MarconiNet – An Architecture for Internet Radio and TV Networks

Ashutosh Dutta * Henning Schulzrinne Yechiam Yemini
Department of Computer Science, Columbia University, New York, NY
{dutta,hgs,yemini}@cs.columbia.edu

Abstract

MarconiNet is an architecture for IP-based radio and TV networks, built on standard Internet protocols including RTP, RTSP, SAP, and SDP. It allows to build virtual radio networks, similar to traditional AM/FM radio and TV networks. It addresses some of the practical problems of building Internet radio networks, including the insertion of local content and advertising.

1 Introduction

So far, the Internet has flourished providing services that do not have direct equivalents in the "real" world, such as email, netnews and the web. Recently, there has been interest in having the Internet serve as next-generation replacement for existing, long-established communications media, in particular the telephone system, radio and TV. In all three cases, the basic functionality of these communications systems has not changed in several decades; addition of features is difficult. Digital audio broadcasting[1] has been proposed, but it simply replaces the transmission infrastructure, without actually offering significant new services beyond an indication of what piece of music is playing. It also does not change the fundamental issue that only a very small number of stations can transmit at any given time.

Earlier attempts at next-generation radio and TV failed largely because they were closed systems adding little beyond the ability to click on ads and save the trip to the local video rental store. In this paper, we are presenting an architecture that tries to highlight the kind of operational issues that need to be addressed if we truly want to create an Internet-based service that combines the best features of both traditional radio and TV and the Internet. Traditional radio has the advantage of offering continuous, themed content without requiring extensive user interaction, making it convenient in situations where the listener is sharing his attention with something else, such as the road in front of him. It also offers far better bandwidth efficiency than true audio or video on demand.

The Internet can offer the advantage of increased total "spectrum", avoiding the problem that only, for example, 45 FM stations can be heard in the New York City metropolitan area [1], with all other voices being limited to being pirate stations. Just like micro-cellular systems increase the capacity of mobile telephone systems, we propose to use micro-cellular radio systems, with transmitters reaching just a few hundred feet in dense urban areas.[2] By using open Internet standards, we can also more easily integrate Internet radio with other Internet mechanisms, for example, email and web. (It would, for example, be simple to "invite" a friend to listen to a radio station using SIP [2].) Given the much better "side information" available, one can also envision devices that automatically record certain types of material for later local replay. (Such systems exist in some European countries for automated recording of TV shows.)

Internet radio is also not limited to serving a single area, but there could easily be "islands" of interest, as multicast limits the distribution to those regions where there are actually listeners.

Below, we adopt radio terminology for conciseness, but the mechanisms are independent of the media carried. We assume the availability of network-level multicast, but RTP translators or UMTP [3] can be used to reach non-multicast-capable networks.

*Presently employed by Telcordia Technologies, Morristown, NJ

[1] see http://www.worlddab.org/

[2] Unfortunately, there do not seem to be variable-power, variable-bandwidth systems which would allow to have fewer, higher-powered wide-area stations in low-population-density areas.

2 Architecture

The architecture (Fig. 1) consists of four components, namely the primary station (PS), local station (LS), possibly an advertisement/media server and the Internet multimedia receiver (IMR) capable of acting as an Internet radio or Internet TV. There are two types of multicast groups in this hierarchical architecture: one distributes programs from primary stations (PSs) to local stations. A local multicast session is created between each local station server and the local receivers for every multicast session that exists between a primary station and the local station.

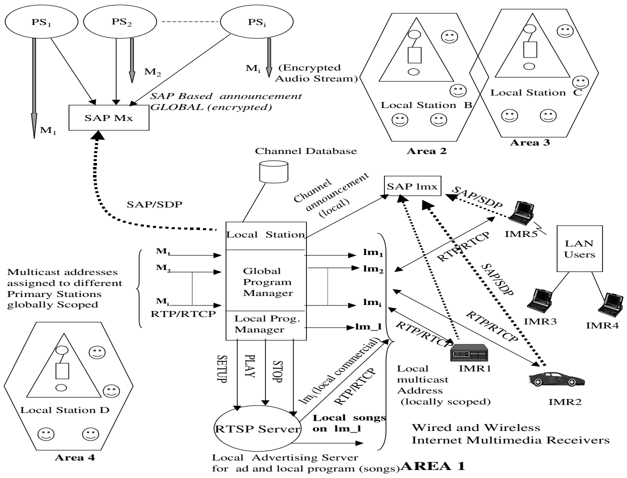

Figure 1: Architecture

The local station combines material from one or more primary stations with local material, including advertisements, and rebroadcasts the stream. Since the local station may combine several sources, it is not simply acting as an RTP mixer. However, it may be useful to still use RTP content sources to indicate the sources of different content, for example, to indicate legal responsibility for a particular segment or copyright restrictions. Local stations may also generate content on a separate local network multicast group, e.g., for feeding locally-produced news segments or relaying part of primary station's program for a specific time during its progrm schedule to the local clients.

Brassil et al. [4] have investigated a different architecture for program insertion where the local content was inserted into the same multicast group as the network programming. For stored content such as advertisement, the techniques described there apply.

Both local and primary station send session announcements, e.g., using SDP [5] and SAP [6] or its hierarchical extensions [7], with the primary station announcements being encrypted.

Since we would not want the local receivers to be able to get the program feed from the primary station directly without the local advertisements during the commercial break, the primary station encrypts its content [8, 9, 10, 11]. Besides, by adopting certain encryption scheme it offers a mechanism so that a local station can choose to relay the programs of a subset of primary stations based on the popularity of the program in a local area. If local stations are not trustworthy, but the network is, it may be necessary to introduce authenticated multicast joins to make redistributing the key less attractive [12, 8]. (The station would then have to retransmit the content after decryption rather than letting a "rogue" listener simply subscribe to the multicast group.)

Almost all radio stations need to know the size of their audience. RTCP [13] provides an easy mechanism for gathering statistics on the size of the audience and their reception quality. Unfortunately, using RTCP as is may not scale or provide the necessary information. For example, the primary station only sees reception reports from its local affiliates (local stations), but has no idea how many people are actually listening. Thus, we propose that RTCP be extended by a user demographics field that simply counts the number of local users. At the local level, there is an additional problem, in that RTCP does not provide accurate audience size indications for larger groups. For example, for a 128 kb/s (MPEG L3) audio stream, the local station can learn about 10 new audience members each second. Thus, even if listeners stay around for an hour, the station will never see more than 36,000 listeners, even with optimization techniques such as reconsideration [14]. Plus, RTCP packets are multicast, raising privacy issues. Multicasting RTCP receiver reports also adds to the state in multicast routers as every listener becomes a data source. It appears that sampling-based techniques [15, 16] are more appropriate here.

With the potentially large number of local stations, we are investigating a scalable mechanism for local-content insertion. The primary station simply transmits the timestamp value of the beginning of the next block, as well as a type code (ad, mandatory local content with no network content available, optional local content, . . .) via the RTCP sender reports. To recover from packet loss, this information should be distributed multiple times in the intervals preceding the local content. (For senders, the expected RTCP

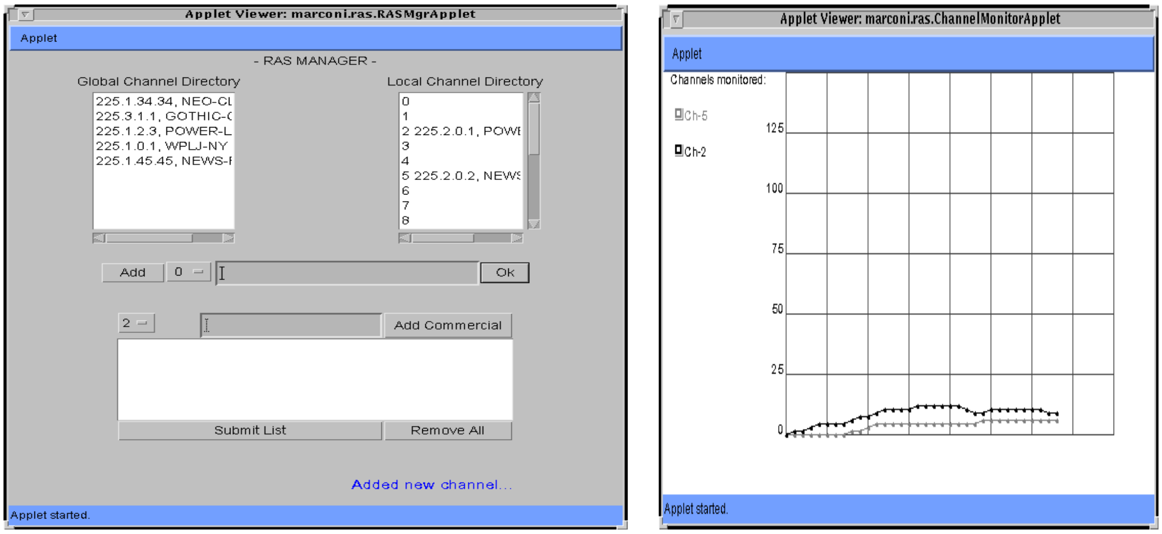

Figure 2: Global Program Manager Prototype

interval is roughly five seconds, so there are ample opportunities for redundancy.)

On receiving the signal for a commercial or local-conent break, the management server at the local station requests the local RTSP [17] server to start playing the local content. During this time, the local station ceases forwarding RTP packets. While RTP timestamps can be aligned with the primary station, the local station has to generate its own sequence numbers.

3 Related Work

Jonas et al. [19] have introduced the concepts of Network Access Points (NAPs) between the clients and the servers and Service Applications (SA) within the network to take care of the scalability and heterogeneity of the receiver sets respectively. Nullsoft Inc. [20] uses distributed server functionality and offers the ability to an end user to be a potential broacaster even over a dialup line using MP3 coding. But it does not use multicast. MCI Worldcom in collaboration with Real Networks is planning to offer a new multicast-based multimedia streaming service called uucast [21] within its Intranet. Presently it limits the usage to its own users, and may not allow any arbitrary customers to act as content providers. The general motivation expressed in this paper is to build an audio/video streaming system over the public Internet which can provide scalability for a large number of audience, offer flexible services to fixed, mobile and wireless users while providing a charging model framework (e.g., local advertisement insertion, local receiver reports) to make it commercially attractive. This design implements scope-based distributed multicast approach, introduces the concept of local stations and uses application layer routing and control for the real-time audio/video streams. Thus it is different than the related systems which exist today.

4 Conclusion and Open Issues

MarconiNet is still work in progress. This paper discussed the high- level design of the architecture and its functionality. Many of the features like advertisement insertion, security, giving the ability for anybody to be a broadcaster has been prototyped in the current MarconiNet testbed. Current work includes adding a charging model for commercials and pay-per-listen programs, inserting a global station segment while playing the local program, and a scalable directory structure (tuner) for the Internet multimedia receiver. Above features have been prototyped using Java and C on the Solaris platform and have also been experimented in IEEE 802.11 based wireless environment. Figure 2 shows a snapshot of the part of Global Program manager prototype where it illustrates how the channels are selected, how the list of commercials is created for each primary station, and how a channel monitor keeps track of the listeners for any particular channel. We have plans for extending this testbed to a hybrid network covering many autonomous systems where there is no multicast support between different networks and thus include UDP server functionality for converting multicast stream to unicast stream and vice versa on a real-time basis. Future work might include building hardware dedicated to Internet radio or TV. Radio hardware can be relatively simple, since only multicast IP, IGMP, UDP, RTP, SAP and SDP are required, but not DNS or TCP.

One of the problems with the ability to listen to any channel anywhere is that local wireless spectrum may be exhausted. There may be the need to develop, for example, popularity-based spectrum management, so that the wireless bandwidth is allocated to achieve the maximum total utility summed across the local user population. However MarconiNet's distributed server approach would augment the spectrum management technique described above.

For radio and TV, channel surfing is common. Thus, deployment of these solutions in bandwidth-constrained networks is only feasible where the IGMP leave latencies are low and IGMP snooping or similar layer-2 mechanisms have been deployed. (For IGMPv2 [18], the default leave latency is about one second.) However, with large user populations, the IGMP join and leave messages themselves may become a burden.

It would be desirable to implement payment schemes that allow other models beyond the three current ones: public financing, advertising and on-air solicitation for donations. We intend to explore hybrid payment models, where the radio program is made available in two versions: one, with advertising or fund-raising support, without payment, and as a paid service, but without ads or fund raising.

Authors would like to acknowledge the contribution from the project students Yulia Averbukh, Kritatee Bulsook, Maksim Khorovskiy, Hong-Ryul Kim, Jay Kim, and Aleksandr Vosokoboynik who have implemented parts of MarconiNet.

References

[1] H. Schulzrinne, "Re-engineering the telephone system," in *Proc. of IEEE Singapore International Conference on Networks (SICON)*, (Singapore), Apr. 1997.

[2] 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.

[3] R. Finlayson, "The UDP multicast tunneling protocol," Internet Draft, Internet Engineering Task Force, Feb. 1998. Work in progress.

[4] J. Brassil, S. Garg, and H. Schulzrinne, "Program insertion in real-time IP multicasts," *ACM Computer Communication Review*, vol. 29, pp. 49–68, Apr. 1999.

[5] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.

[6] M. Handley, "SAP: Session announcement protocol," Internet Draft, Internet Engineering Task Force, Nov. 1996. Work in progress.

[7] R. Finlayson, "The multicast attribute framing protocol," Internet Draft, Internet Engineering Task Force, Jan. 1998. Work in progress.

[8] A. Ballardie, "Scalable multicast key distribution," Request for Comments (Experimental) 1949, Internet Engineering Task Force, May 1996.

[9] H. Harney and C. Muckenhirn, "Group key management protocol (GKMP) architecture," Request for Comments (Experimental) 2094, Internet Engineering Task Force, July 1997.

[10] B. Cain, N. Doraswamy, and T. Hardjono, "A framework for group key management for multicast security," Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[11] S. Mittra, "Iolus: A framework for scalable secure multicasting," *ACM Computer Communication Review*, vol. 27, pp. 277–288, Oct. 1997. ACM SIGCOMM'97, Sept. 1997.

[12] N. Yamanouchi, O. Takahashi, and N. Ishikawa, "IGMP extension for authentication of IP multicast senders and receivers," Internet Draft, Internet Engineering Task Force, Aug. 1998. Work in progress.

[13] 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.

[14] 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.

[15] J.-C. Bolot, T. Turletti, and I. Wakeman, “Scalable feedback control for multicast video distribution in the internet,” in *SIGCOMM Symposium on Communications Architectures and Protocols*, (London, England), pp. 58–67, ACM, Aug. 1994.

[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.

[19] Karl Jonas, Mathias Kretschmer, Jens Modeker, “Get a KISS - Communication Infrastructure for Streaming Services in a Heterogeneous Environment,” *ACM Multimedia 1998*, (Bristol, UK), Sep. 1998.

[20] www.nullsoft.inc

[21] www.uu.net