REIHE INFORMATIK

4[illegible]/98

Teleteaching over Low-Bandwidth Network Channels

Christoph Kuhmünch
Universität Mannheim
Praktische Informatik IV
L15, 16
D-68131 Mannheim

Teleteaching over Low-Bandwidth Network Channels *

Christoph Kuhmünch
cjk@pi4.informatik.uni-mannheim.de

Lehrstuhl Praktische Informatik IV[†] University of Mannheim,
L15,16, D-68 131 Mannheim, Germany

April 6, 1998

Abstract

Teleteaching has become an important application of the Internet and the MBone. Unfortunately the costs for the hardware, necessary to participate in remote lectures, are still prohibitively high and the degree of distributiveness of the implemented scenarios is very low. In the Interactive Home Learning project we plan to provide methods to participate in a Teleteaching lecture live from a PC at home via a low-bandwidth connection (e.g. ISDN). This paper summarises technical aspects of this learning scenario and presents our approach, the fully *Java-based Reflection and Scaling Tool jrst*, which meets the requirements of an application layer multicast routing demon with a highly restrictive broadcasting policy and a dynamic tunnelling mechanism.

Keywords: *computer-based distance learning, ISDN, MBone, media scaling*

1 Introduction

Teleteaching has become an important application of the Internet and the MBone. Several projects all over the world have investigated technical and pedagogical aspects of this new paradigm of teaching. Since 1996 our institute has transmited complete lectures and seminars [Eck97, Eff97, Gey97] on a regular basis to other universities by making use of the MBone [Dee91, Hui95] and its tools. Vic, vat and wb [McC95, Jac96a, Jac96b, McC96] are used to transmit audio and video of the lecturer to the remote side. Moreover, we make use of animations and simulations developed in Java. In this context we developed the *Java Remote Control Tool*[Kuh98], which allows us to control and synchronise animations running on several distributed machines. Although the costs of computer technology continue falling rapidly, the hardware and Wide Area Network connections necessary to participate in these lectures are still prohibitively expensive. Because of these high costs the degree of distributiveness of the scenarios implemented so far is very low. Currently we transmit lectures to classrooms at not more than two or three other universities.

In the Interactive Home Learning project we plan to provide methods to participate in a Teleteaching lecture via low-cost low-bandwidth ISDN connections. Similar to our other Teleteaching scenarios, Interactive Home Learning is based on the MBone tools. Figure 1

*This project is supported by the BMBF (Bundesministerium für Forschung und Technologie)
†URL `http://www.informatik.uni-mannheim.de/informatik/pi4/`

depicts the scenario. Lectures will be transmitted via high-bandwidth ATM connections to the remote lecture rooms and simultaneously to the student PC at home via a low-bandwidth ISDN connection.

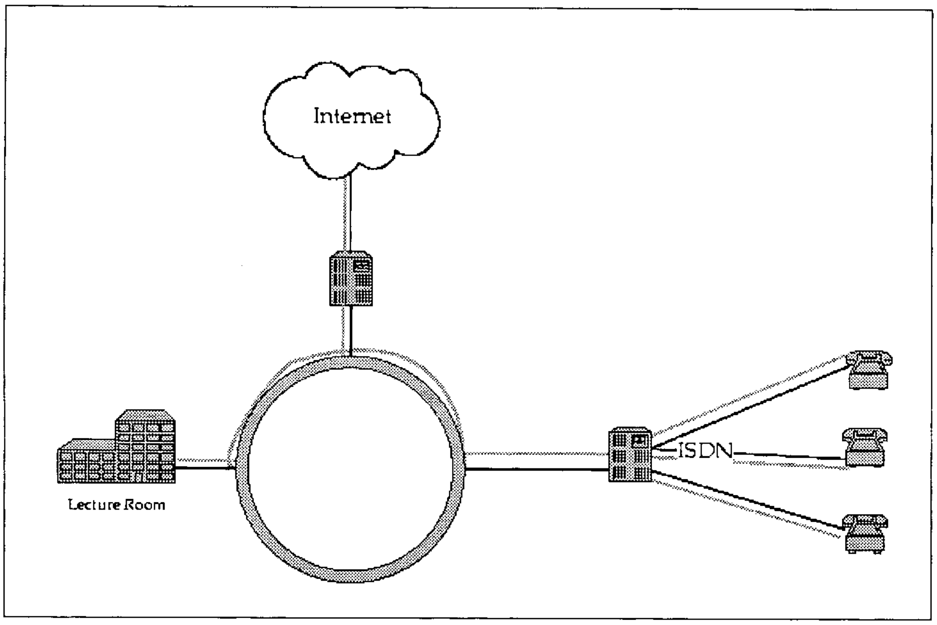

Figure 1: The synchronous Home Learning Scenario

The simultaneous transmission of media streams through high-bandwidth Internet connections and low-bandwidth ISDN connections creates two challenging technical problems:

1. Currently, we transmit video at up to 12 frames per second to the remote classrooms, which requires a bandwidth need of almost 600kbit/s. In order to transmit the video not only to the remote classrooms but also through ISDN to the student's PC at home the quality of video has to be adapted to the lower bandwidth of ISDN (128kbit/s).
2. Since ISDN is not a broadcasting channel, unlike Local Area Networks, direct Multicasting is not possible.

In this paper we discuss these technical challenges and propose our approach, a fully *Java-based Reflector and Scaling Tool (jrst)*. Jrst meets the requirements of an application layer multicast routing demon with a highly restrictive broadcasting policy and a dynamic tunnelling mechanism. That tool allows to dynamically set up tunnels and transmit selected multicast sessions through channels without multicast capabilities. The system fulfils not only the requirements of the Interactive Home Learning scenario but can also be integrated into other services e.g. Media on Demand servers and databases as they are realised in Wieland Holfelders MBone VCRoD [Hol97] and the mMOD by Peter Parnes [Par97a].

The rest of the paper is structured as follows: The next section analyses the technical requirements of the scenario. Section 3 describes in detail the technical problems which have to be overcome in order to send high-bandwidth multicast data streams via low-bandwidth unicast connections, followed by a description of related work in section 4. Section 5 presents our approach, the Java-based reflector and scaling tool. Conclusions and an outlook in section 6 finish the paper.

2 Requirements of the Scenario

For the IHL scenario we have to consider two groups of participants: The first group is connected by high-bandwidth Internet connections, uses Unix workstations and is connected to the MBone. The second group uses PCs with Windows 95 or Linux, participates through ISDN, which provides a bandwidth of 128 kbit/s, and the Point-to-Point-Protocol (PPP) [McG92] is used to transfer IP datagrams through the ISDN channels. As shown in Figure 1, audio, video, and whiteboard datagrams are transmitted via multicast IP from the local lecture room to the remote lecture rooms. This data stream has to be rescaled to the bandwidth of ISDN (128kbit/s) and then transmitted to the student's PC's.

In order to transmit audio, video, and whiteboard streams through high-bandwidth links and ISDN connections, we have to solve several problems, which are described in the following section.

3 Technical Challenges

While the transmission of MBone media streams via Ethernet causes no problems the transmission via ISDN is much more complicated for the following reasons:

1. ISDN is not a broadcast medium like Local Area Networks. Hence, in order to transmit multicast media streams via ISDN, we have to use the multicast tunnelling mechanism. In fact, we have to set up a tunnel to each student's PC currently an impossibility, because tunnelling is a static routing mechanism. If a student disconnects the tunnel will break down and has to be set up again by the network administrator when the student reconnects. Moreover, dial in connections usually use the Point to Point Protocol (PPP)[McG92] with dynamic IP addresses Thus the IP address of a user is not known in advance. Furthermore, the tunnelling mechanism only works between two multicast routers, which causes difficulties if the student's PC is based on Windows 95. Windows 95 supports IP multicasting but so far there are no multicast routing demons available. Even if we could find ways to set up tunnels dynamically, multicast routing algorithms like DVMRP (Distance Vector Multicast Routing Protocol) [Dee89, Dee91] would soon cause the ISDN line to collapse when data from other sessions like the NASA Space Shuttle Mission are broadcasted into the tunnel. Of course, these streams will be pruned after a timeout but until then, the ISDN connection is flooded by unwished multicast datagrams. Additionally, the data-streams will regularly be broadcasted again during the grafting and pruning process. To summarise, the multicasting problem can only be handled by a tunnelling mechanism with much more restrictive pruning algorithms.

2. The second problem results from the low bandwidth of ISDN. Currently, we transmit data streams at about 400-800kbit/s between Mannheim and Heidelberg but the bandwidth of ISDN is only 64kBit/s (128KBit/s if two B-channels are combined). Therefore we have to implement methods to scale the audio and video streams to its lesser bandwidth of ISDN. Furthermore, flow-control mechanisms and buffers have to be implemented in order to smooth peak bit rates caused, for example, by whiteboard operations.

Table 1 summarises the major problems of the scenario.

No.	Problem
1	• direct transmission of Multicast datagrams over ISDN is impossible. • tunnelling is a static routing mechanism but the students connect and disconnect dynamically through the universities ISDN ports. • multicast routing demons are not available for all platforms and their administration is difficult. • the pruning has to be implemented more strictly than in other environments.
2	• in contrast to the Internet connection between two lecture rooms, the bandwidth of ISDN is limited to 128KBit/s (2 B-channels). • the media-streams have to be scaled to the bandwidth of ISDN. • The transmission of large postscript-files via ISDN causes high peak bit rates.

Table 1: Major problems of the scenario

4 Related Work

Peter Parnes et al.[Par97b] propose a modified multicast routing demon *"mTunnel"* which allows user-based dynamic tunnelling of multicast datagrams. The tunnel is set up by a special routing demon at either end of the tunnel. Beside the tunnelling mechanism the system also provides user influenced QoS: A data stream can be piped through a translator that can scale the data stream or change the encoding method. The last function which shall be mentioned here is the mTunnel's ability to prioritise data streams. This is especially useful in teleteaching scenarios because good audio quality in a lecture is much more important than high video quality.

However, the mTunnel does not exactly match the requirements of IHL because only two participants are connected through the tunnelling mechanism. This means that each student would have to set up his own tunnel in order to participate in the lecture. Our scenario needs a mechanism that connects several participants.

Another approach is *CU-SeeMe*, developed at Cornell University [Dor95]. This system uses a special server called a "Reflector". Figure 2 explains the reflector technique: Clients communicate to each other via a reflector server. Each client sends its media streams to the server which sends a copy to all other clients connected to the server. Clients can either use Unicast addresses or multicast addresses. The drawback of CU-SeeMe is that they use their own conferencing tools.

The *"RTP Gateway"* was developed by Steve McCanne [McC96] at UC-Berkeley, CA. It allows to rescale the audio and video stream of a MBone session. The Gateway consists of two tools, the Audio-Gateway (agw) and the Video-Gateway (vgw). The vgw receives a video stream (sent by the video tool vic), decodes the pictures in the stream, recodes them with a chosen encoding algorithm and in a chosen quality, and sends them to the output stream. The input and output streams can be either a multicast or unicast address.

The RTP Gateway solves the scaling problem mentioned in table 1. Furthermore it can be used to translate a multicast group address into a unicast address. Unfortunately it does not exactly match our needs for the following two reasons:

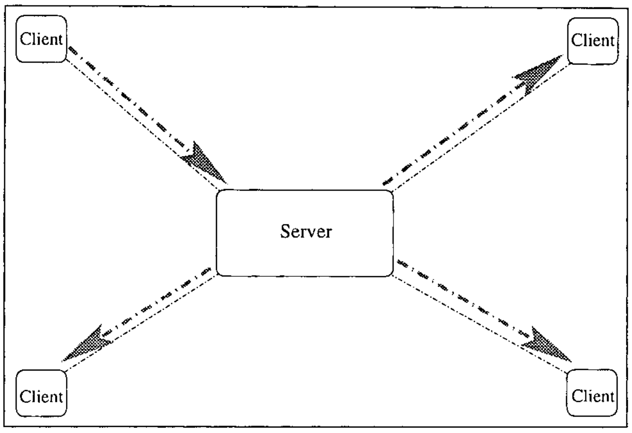

Figure 2: The CU-SeeMe Reflector technique

1. First of all, the current version supports only one output stream, which means that several gateways have to be started in order to support multiple ISDN connections.
2. As the name suggests, the RTP Gateway relies heavily on the RTP protocol. The wb-tool, which is currently the standard shared whiteboard application, does not support RTP, and therefore it is not supported by the RTP Gateway.

5 A Java-based Reflection and Scaling Tool

Since none of the tools described in the previous section solves our problem, we developed our own solution, tailored to the needs of our synchronous Interactive Home Learning scenario. Our *"Java-based Reflector and Scaling Tool (jrst)"* enhances and combines the features of the approaches mentioned above. Our intention is to create an integrated service that can even be handled by first year students with little experience in computer science.

The system consists of the following two applications:

- Reflector and Scaling server
- Reflecting and Scaling client

The core of the tool is a Reflector Server, which works basically as described in figure 3. Users can connect to the server through a client application via a dedicated port number. After the login process the client will receive a list of sessions available for transmission. The user then will choose a session to join. A request for the selected session will be transmitted from the client to the server, which then will transmit and rescale the media stream according to the bandwidth available to the client.

The following list summarises the features of jrst:

- Distributed client-server oriented architecture.
- Integrated solution with an-easy-to-handle graphical user interface.
- Implemented in Java and therefore platform-independent.
- Support of all MBone Tools (audio, video and whiteboard).
- Client and server communicate via the open standard Real Time Streaming Protocol (RTSP)[Sch98], which makes it possible to integrate its scaling and reflector services into other applications, e.g. Video-on-Demand Servers.

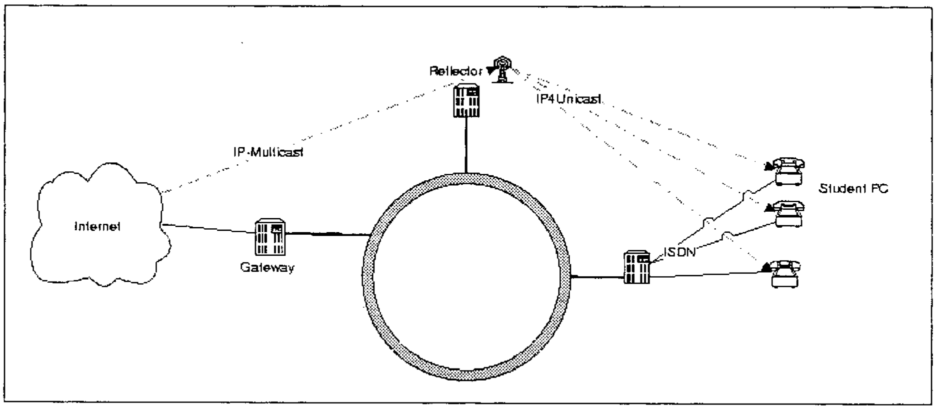

Figure 3: The Reflector-technique

5.1 Technical Outline of the Server Component

The server is the main part of the system and consists of the components indicated in Figure 4:

- Client Communication Manager
- Session Announcement Protocol Listener
- Reflector Services
- Media Filtering Services

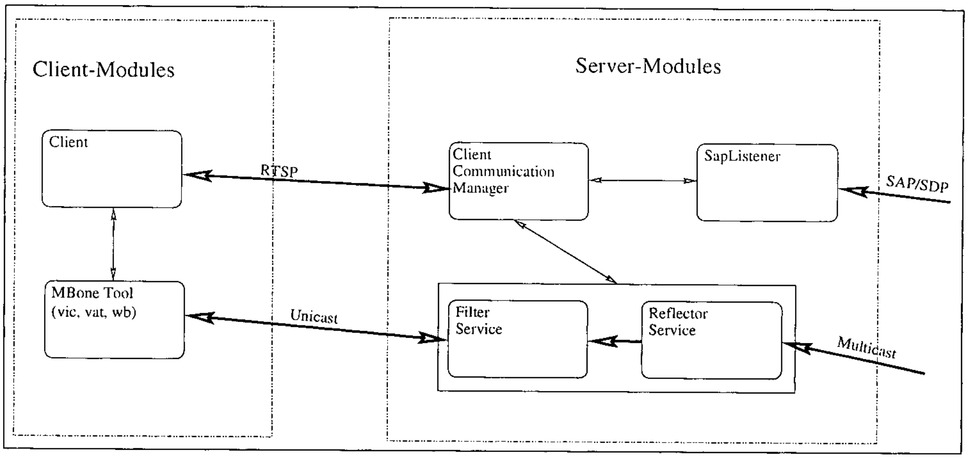

Figure 4: The Java Reflector and Scaling Tool

The Client Communication Manager is the central management component of the server. It provides the service interface for the clients and starts and stops reflection and scaling services.

The Session Announcement Protocol Listener is capable to receive session descriptions, described according to the session description protocol SDP [Han97b] and announced according to the session announcement protocol SAP [Han97a]. Received session descriptions are stored in a local cache and transmitted to the client upon demand.

The Reflector Service accomplishes the bulk of the work and is the most time-critical part. When a client requests a multicast session, the Client Communication manager adds him to the list of receivers – called customers – of the corresponding Reflector Service. The Reflector Service is created when a client requests a session for the first time and destroyed when the last of its customers leaves the session. The task of the reflection service is to broadcast incoming multicast datagrams to all of its customers and vice versa.

The Media Filtering Service controls external filter mechanisms. In the current prototype, we rely on the rtp gateway described in Section 4. Furthermore, the filtering service provides flow-control mechanisms and buffers for selected streams in order to smooth peak bit rates, for example caused by whiteboard operations (loading of a new page).

The current prototype is implemented completely in Java except for external filters, although we are considering porting time-critical parts of the server to C/C++, in order to improve the performance.

5.2 Technical Outline of the Client Component

The jrst client – also implemented in Java – is available as a stand-alone application as well as as an applet. When a client connects to a server, it receives the announcements of those sessions currently available. The user then can choose the sessions to be received. The client sends the request back to the server process, which then transmits the requested session to the client. At the client site, the required MBone tools are started automatically.

Figure 5 illustrates the graphical user interface of the client.

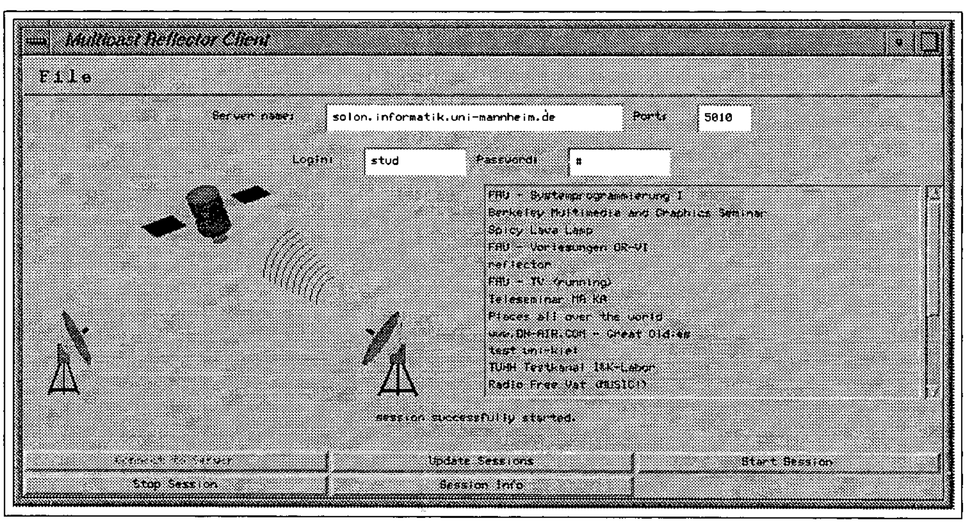

Figure 5: The graphical user interface of the client

6 Conclusion and Outlook

In this paper, we described synchronous Interactive Home Learning, a Teleteaching scenario which focuses on low-cost, low-bandwidth Internet connections such as ISDN. We discussed technical challenges in this scenario and described several existing solutions. We then presented our approach, the *Java-based Reflector and Scaling Tool (jrst)*, which implements an application layer multicast routing system with a highly restrictive broadcasting policy. Jrst is completely implemented in Java using JDK1.1 and, thus works on every platform, that supports Java version 1.1. The system was tested on SUN Solaris 2.5., Linux, Windows 95 and Windows NT. The system fulfils not only the requirement of the Interactive Home Learning scenario but can also be integrated into other services, e.g. Media-on-Demand servers and databases. During the forthcoming term, we will evaluate our prototype implementation to obtain performance results. Due to performance problems with the programming language Java we are considering porting time critical parts to C/C++ in order to optimise performance.

7 Acknowledgment

This project is supported by the BMBF (Bundesministerium für Forschung und Technologie/ Federal Minitstry of Research and Technology).

References

[Dee89] S. Deering, C.Partridge, and D. Waitzmann. Distance Vector Multicast Routing Protocol. Internet Request For Comments, IETF, RFC-1075, August 1989.

[Dee91] S. Deering. *Multicast Routing in a Datagram Internetwork.* PhD thesis, Stanford Univerity, California, USA, 1991.

[Dor95] T. Dorcey. CU-SeeMe Desktop Video Conferencing Software. *Connexions,* March 1995.

[Eck97] Andreas Eckert, Werner Geyer, and Wolfgang Effelsberg. A Distance Learning System for Higher Education on Telecommunications and Multmedia - A Compound Organisational, Pedagogical and Technical Approach. In *ED-MEDIA/ED-TELECOM'97, Calgary, Canada.* AACE Association for the Advancement of Computing in Education, 1997.

[Eff97] W. Effelsberg. Das Projekt TeleTeaching der Universitäten Mannheim und Heidelberg. In teleteaching, distance education, mbone [Eck97].

[Gey97] W. Geyer, A. Eckert, and W. Effelsberg. Project TeleTeaching Mannheim - Heidelberg. In teleteaching, remote lecture room, distance learning [Eck97]. in German.

[Han97a] Mark Handley. SAP: Session Announcement Protocol. Internet draft, IETF, Multiparty Multimedia Session Control Working Group, draft-ietf-mmusic-sap-00, expirery date May 25th, 1997 1997.

[Han97b] Mark Handley and Van Jacobson. SDP: Session Description Protocol. Internet draft, IETF, Multiparty Multimedia Session Control Working Group, Internet draft draft-ietf-mmusic-sdp-03, expirery date September 26th, 1997 1997.

[Hol97] Wieland Holfelder. Interactive Remote Recording and Playback of Multicast Videoconferences. In 4^{th} *International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97),* 4. IDMS, September 1997.

[Hui95] Christian Huitema. *Routing in the Internet.* Prentice-Hall PTR, Englewood Cliffs, NJ 07632, 1^{st} edition, 1995.

[Jac96a] Van Jackobson and Steven McCanne. Video Conferencing Tool (vic). Lawrence Berkeley Laboratory, ftp://ftp.ee.lbl.gov/conferencing/vic, 1996.

[Jac96b] Van Jackobson and Steven McCanne. Visual Audio Tool (vat). Lawrence Berkeley Laboratory, ftp://ftp.ee.lbl.gov/conferencing/vat, 1996.

[Kuh98] Christoph Kuhmünch, Thomas Fuhrmann, and Gunther Schöppe. Java Teachware - The Java Remote Control Tool and Its Applications. In *ED-MEDIA/ED-TELECOM'98*. AACE Association for the Advancement of Computing in Education, 1998. accepted paper.

[McC96] Steven McCanne. The RTP Gateway (rtpgw). Lawrence Berkeley Laboratory, ftp://daedalus.cs.berkeley.edu/pub/vgw/alpha-test/, November 1996. s. auch hui95.

[McC95] Steven McCanne and Van Jacobson. Vic: A Flexible Framework for Packet Video. In *Multimedia*, San Francisco, November 95. ACM, ACM Press. s. auch hui95.

[McG92] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). Internet Request for Comments, IETF, RFC-1332, May 1992.

[Par97a] Peter Parnes. mMOD: The multicast Media-on-Demand system. In mbone, mvod, vcr on demand [Hol97].

[Par97b] Peter Parnes. mTunnel: A Multicast Tunneling System with a User-based Quality-of-Service Model. In 4^{th} *International Workshop on Interactive Distributed Multimedia Systems and Telecomminication Services (IDMS '97)*, 4. IDMS, September 1997.

[Sch98] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). Internet draft, IETF, Multiparty Multimedia Session Control Working Group, draft-ietf-mmusic-rtsp-09, February, expirery date August 2nd, 1998 1998.