

REIHE INFORMATIK

498/98

A Multicast Gateway for Dial-In Lines

Christoph Kuhmünch

Universität Mannheim

Praktische Informatik IV

L15, 16

D-68131 Mannheim

A Multicast Gateway for Dial-In Lines ^{*}

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

Lehrstuhl für Praktische Informatik IV, Universität Mannheim,
L15,16, D-68 131 Mannheim, Germany

July 30, 1998

Abstract

With the increasing number of Mbone sessions the interest of home users to participate is also increasing. Unfortunately the cost for the hardware necessary to participate in multicast sessions over a high speed link are still prohibitively high. In our Interactive Home Learning project we are providing methods to participate in a teleteaching lecture – transmitted on the Mbone – live from a PC at home via a low-bandwidth connection (e.g. ISDN). This paper discusses technical problems and solutions for users who wish to participate in multicast sessions over dial-in lines and presents our approach, the *ISDN Multicast Gateway*, which meets the requirements for an application layer multicast router with a very restrictive broadcasting policy and a dynamic tunnelling mechanism. The ISDN Multicast Gateway allows the transmission of selected multicast sessions over dial-in connections such as ISDN by providing a mechanism to configure a multicast tunnel dynamically. For each selected media stream the desired quality of service parameters can be set interactively, e.g. a certain share of the available bandwidth can be reserved. In order to allow a graceful scaling of video we integrated a simple scaling mechanism for H.261 video streams that controls the temporal resolution of the video.

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

1 Introduction

With the increasing number of Mbone [Dee91, Hui95] sessions the interest of home users to participate is also increasing. Although the cost of computer technology continues to fall rapidly, the hardware and wide area network connections necessary to participate in these lectures at full bandwidth are still prohibitively expensive.

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 [Eck97], Interactive Home Learning is based on the Mbone tools vic, vat and wb [McC95, McC96, Jak96a, Jak96b]. Figure 1 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.

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

^{*}This project is supported by the BMBF (Bundesministerium für Forschung und Technologie)

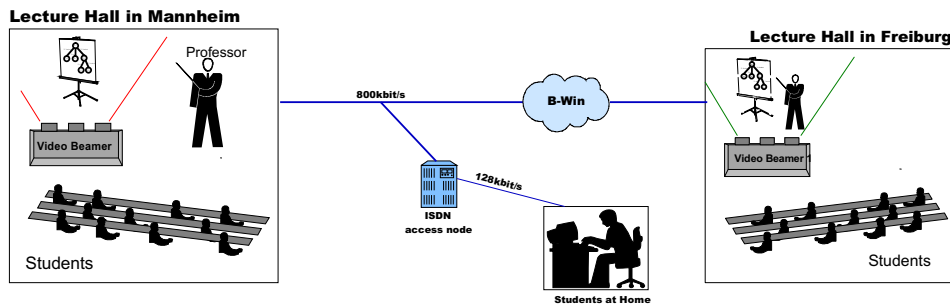


Figure 1: The synchronous Home Learning Scenario

1. Currently, we transmit video at up to 12 frames per second to the remote classrooms, which requires a bandwidth of about 600kbit/s. In order to transmit the video not only to the remote classrooms but also through ISDN to the students's PC at home, the quality of video has to be adapted to ISDN's lower bandwidth (128kbit/s).
2. Since the ISDN lines are point-to-point links in the overall multicast topology tunnels have to be set up. But the students typically don't have permanent IP addresses: the computing centre assigns them temporary addresses for access via PPP. There is no mechanism in the Internet protocol architecture to set up tunnels dynamically of PPP links.

Since unlike Local Area Networks ISDN is not a broadcasting channel direct multicasting is not possible.

In this paper we discuss these technical challenges and propose our approach, the *ISDN Multicast Gateway*. This gateway meets the requirements for an application layer multicast routing demon with a highly restrictive broadcasting policy and a dynamic tunnelling mechanism. It allows tunnels to be set up dynamically and selected multicast sessions to be transmitted 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 have been realised in Wieland Holfelder's MBone VCRoD [Hol97] and the mMOD by Peter Parnes [Par97a].

The remainder paper is structured as follows: The next section summarises related work, and is followed by an analysis of the technical requirements of the scenario. Section 4 describes in detail the technical problems that have to be overcome in order to send high-bandwidth multicast data streams via low-bandwidth unicast connections. Section 5 presents our approach, the Java-based reflector and scaling tool. Conclusions and an outlook in Section 7 finish the paper.

2 Related Work

Peter Parnes et al. [Par97b] propose a modified multicast routing demon called "*mTunnel*" which allows user-based dynamic tunnelling of multicast packets. 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 to 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 the tunneling mechanism connects only two participants. 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.

An earlier 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 with 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 with CU-SeeMe is that it uses its own conferencing tools and that it does not allow media scaling.

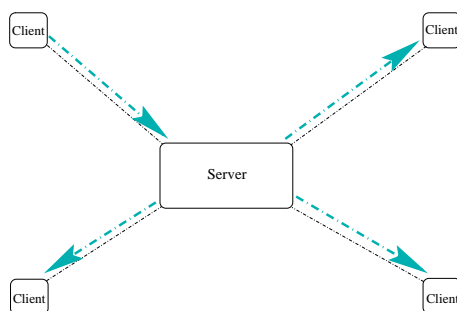


Figure 2: The CU-SeeMe Reflector technique

Another alternative the “*RTP Gateway*” was developed by Amir et al. [Ami95]. 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 multicast or unicast packet streams.

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 fails to exactly match our needs for the following two reasons:

1. The current implementation supports only one output stream so that one gateway has to be started for each ISDN receiver.
2. As the name suggests, the RTP Gateway relies heavily on the RTP protocol. The wb tool, currently the standard shared whiteboard application on the Mbone, does not run over RTP, and therefore is not supported by the RTP Gateway.

We conclude that each of the existing systems solves one specific part of our problem, but there is no integrated solution and just pasting those components together would obviously not work. We claim that multicast routing, scaling and tunneling are closely related, and only an integrated architecture can provide an optimal solution for multicast to heterogeneous end systems.

3 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 95/98 or Linux, participates through

ISDN at a bandwidth of 64 or 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 packets are sent via multicast IP from the local lecture room to the remote lecture rooms. This mixed data stream has to be rescaled to the bandwidth of ISDN (max. 128kbit/s) and transmitted to the student's PCs.

For the transmission of the data streams we are using the Mbone tools. Figure 3 depicts the protocol stack of the most popular Mbone tools *vic* *vat* and *wb*. While *vic* and *vat* are using RTP as transport protocol *wb* relies on a Reliable Multicast Protocol (RMP). These packets are encapsulated in IP multicast datagrams which can be transmitted on any network type, e.g. a LAN or ATM.

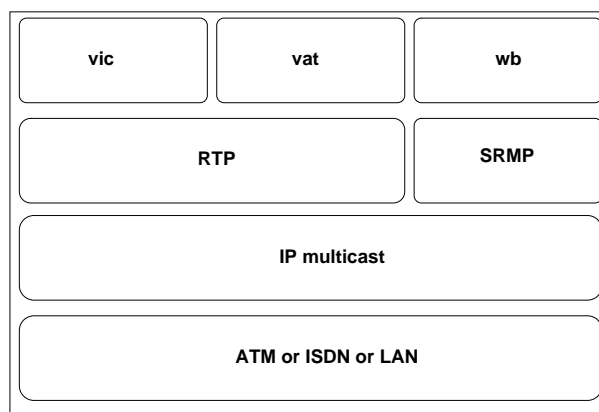


Figure 3: The protocol stack of the most popular Mbone tools *vic*, *vat* and *wb*

4 Technical Challenges

The transmission of Mbone sessions via ISDN is much more complicated than the transmission over LANs or leased lines for the following reasons:

1. The first problem results from the low bandwidth of ISDN. Currently, we transmit audio, video and whiteboard streams at about 400-800 kbit/s between Mannheim, Freiburg and Heidelberg but the bandwidth of ISDN is only 64 kbit/s (128 kbit/s if two B-channels are combined). Therefore we have to implement methods to *scale* the audio and video streams to the lower 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.

For media scaling we can take the importance of the different media streams into account: In teleteaching lessons in computer science the quality of the audio signal is more important than the quality of the video signal since the video signal carries only the "talking head" of the lecturer. Also human perception is much more sensitive to audio errors than to video errors. Therefore, a *priorisation* mechanism is necessary that reserves a minimum share of the available bandwidth for the audio signal.

2. In order to transmit multicast media streams via ISDN, we have to use the multicast tunnelling mechanism.

The tunneling mechanism was integrated into IP multicast in order to cross wide area network lines and routers within the Internet without multicast capabilities as is

depicted in Figure 4. Tunnels are currently the only possibility to introduce multicast routers to each other in a WAN. In order to accomplish this a multicast router (Router A) encapsulates multicast packets in unicast packets with the destination address of the next multicast router (Router B). When such an encapsulated packet arrives at the destination multicast router it is unpacked and routed like a normal multicast datagram. Like most dial-in network connections ISDN access nodes are not capable

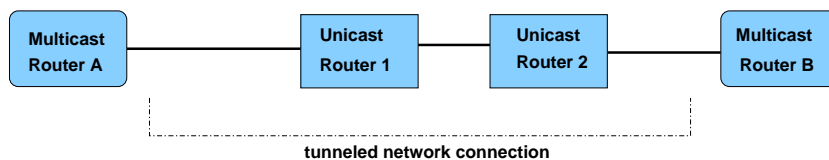


Figure 4: Principle of the tunneling mechanism

of multicasting; thus we have to use the tunneling mechanism. In our In fact, we have to set up a tunnel to each student's PC. If a student disconnects the tunnel will break down and has to be set up again by the network administrator when the student reconnects. 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. Windows 95/98 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 unwanted multicast packets, which to make matters worse, will regularly be re-broadcasted 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.

Table 1 summarises the major problems of the scenario.

5 ISDN Multicast Gateway

Since none of the tools described in section 2 solves our problem, we developed our own solution, tailored to the needs of our synchronous Interactive Home Learning scenario. Our "*ISDN Multicast Gateway*" 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 with computers.

The gateway basically follows the architecture of a reflector [Dor95] that receives packets from a multicast group and forwards them via unicast to ISDN users. Vice versa, packets from ISDN users are forwarded to the multicast group and to the other ISDN users.

No.	Problem
1	<p data-bbox="395 786 667 824">Dynamic Tunneling</p> <ul data-bbox="437 842 1238 1081" style="list-style-type: none"> <li data-bbox="437 842 1238 902">• Tunnelling is a static routing mechanism but the students connect and disconnect dynamically through the universities' PPP/ISDN ports. <li data-bbox="437 913 1238 1010">• Multicast routing demons are not available for all platforms, and their administration is difficult. In the current protocol implementation a multicast demon is required to set up a tunnel. <li data-bbox="437 1021 1238 1081">• Tree pruning has to be implemented more strictly than in other environments.
2	<p data-bbox="395 1140 900 1178">Priorities, Flow Control, and Scaling</p> <ul data-bbox="437 1196 1238 1473" style="list-style-type: none"> <li data-bbox="437 1196 1238 1256">• In contrast to the ATM connection between lecture rooms, the bandwidth of ISDN is limited to 128 kbit/s (2 B-channels). <li data-bbox="437 1267 1238 1305">• The media streams have to be scaled to the bandwidth of ISDN. <li data-bbox="437 1317 1238 1406">• Some media streams are less important than others; e.g. in teleteaching in computer science a good audio quality is more important than good video quality. <li data-bbox="437 1417 1238 1473">• The transmission of large postscript files by the wb causes high peak bit rates.

Table 1: Major problems with the Home Learning scenario

5.1 Functionality

5.1.1 Dynamic Tunneling

As already explained in section 4, ISDN is not capable of multicasting within the network. Therefore the gateway adopts the functionality of a multicast router, whose tunnels can be configured dynamically by the user. With the help of a graphical user interface, ISDN users can notify the gateway server, and after the login and password have been verified the server sends a list of session announcements [Han97a] to the client, where they are displayed in a list box. The user may now choose several sessions to be joined. Each data stream from a session can be assigned a priority according to which a certain share of the total bandwidth of the ISDN line will be reserved for the data stream.

At the client end decoder tools such as *vic*, *vat* and *wb* are started automatically with the necessary parameters. Once established the tunnel works bidirectionally: the user may actively participate in the session. Media streams are transmitted unicast from the ISDN user to the server and the server, which forwards them to the multicast group and to other ISDN users.

5.1.2 Priorities and Flow Control

Priorities and flow control are realised by means of Priority Queues (PQ). A PQ implements a FiFo queue with a priority parameter p that stands for the bandwidth reserved by a certain ISDN receiver for a certain media stream and is measured in bits per second. A PQ is assigned to each pair of ISDN receiver and media stream to be transmitted and incoming packets from the multicast group are enqueued in the appropriate PQ of each receiver. After a datagram has been dequeued, a deadline for this PQ is calculated, thus at any given time the deadline for the first packet of all PQs is known. In the next cycle a datagram is dequeued from the PQ with the earliest deadline (EDF= earliest deadline first scheduling). Empty PQs are skipped. The deadline d (a timestamp in the future) is calculated in the following way: Let B be the total bandwidth of the connection to the client (e.g. 128 kbit/s) and p the priority value reserved for a media stream (e.g. 64 kbit/s for audio). Furthermore let $|pkt|$ be the size of the datagram, that currently has to be transmitted, and timestamp the current value of the “system-clock”.

$$d = timestamp + \frac{|pkt|}{p} \quad (1)$$

Furthermore, for each ISDN receiver an interval timer is instantiated. Scheduled by this timer the packets are dequeued from the PQs of this receiver and sent. The calculation of the wake-up interval t of the timer is similar to the calculation of the deadline of a PQ:

$$t = \frac{|pkt|}{B} \quad (2)$$

Since empty priority queues are skipped the unused bandwidth that has been reserved for a media type is distributed among the other media types.

Figure 5 depicts an example scenario where a receiver has joined a session with the typical three media types audio, video and whiteboard.

The algorithm described so far assigns priorities above the minima in proportion to the minimum values, i.e. a 32kbit/s stream will get higher priority than a 16 kbit/s stream

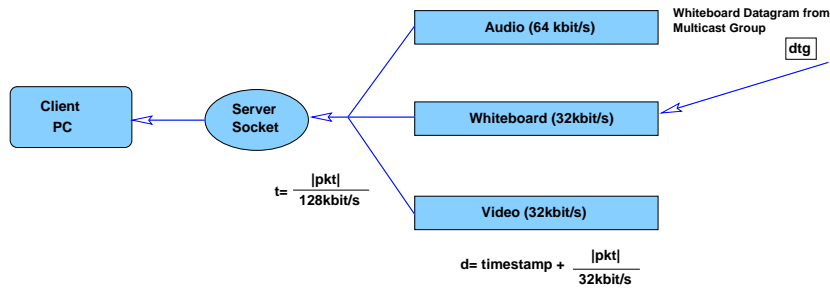


Figure 5: Priorisaties, flow control and filtering through priority queues

when all minimum requirements are fulfilled. We are currently developing models for more sophisticated priority algorithms.

Consider an example scenario where an ISDN receiver has joined a session with an audio, a video, and a whiteboard stream through a 128 kbit/s connection. For audio the user has reserved $p_{audio} = 64 \text{ kbit/s}$ while whiteboard and audio both receive $p_{wb} = p_{video} = 32 \text{ kbit/s}$. Incoming packets are inserted into one of the three queues according to its media type. The packets are removed in a process scheduled by an interval timer. In each cycle of the process a packet is removed from the PQ with the earliest deadline; i.e. when a packet has been removed from the video queue, its deadline is equal to $d = \text{timestamp} + (|pkt| / 32 \text{ kbit/s})$ and the timer interval is set to $t = |pkt| / 128 \text{ kbit/s}$. In other words, the effect will be that video and whiteboard will get 25% of the bandwidth each; audio will get 50%.

5.1.3 Scaling

When a new packet from a media stream arrives it is inserted into the corresponding PQ of each receiver. Since the dequeuing rate is typically lower than the rate of arrival rate, scaling of media streams becomes necessary in order to prevent the overflow of the priority queues. For each media type a scaling (i.e. packet discarding) algorithm has to be devised. For the first prototype of our gateway we have concentrated on video scaling. The quality of video is defined by its three dimensions color resolution, spatial resolution and temporal resolution. The color resolution is defined by the number of bits used to represent the color of a single pixel, the spatial dimension is defined by the width and height of the video, and the temporal resolution is represented by the number of frames per second. Several approaches have been developed in order to scale video in any of these three dimensions [Kuh98b, Bur83, Ami96, McC96, Mer97, Hof96].

For simplicity and due to performance we decided to use a simple scaling mechanism that reduces the temporal resolution [Kuh98a] of intra-encoded H.261 video streams by relying on the RTP header information [Sch96b, Sch96a] as it is produced, e.g. by the video conferencing tool vic: If the length of a PQ reaches a certain upper limit all packets, carrying data from the latest video frame are removed from the queue, and packets of this video frame all later arriving will also be discarded. Information about the frame number and encoding type is taken from the RTP header. This method gracefully reduces the bandwidth of the video signal. Other more complex scaling algorithms will be integrated into future versions of the gateway. Furthermore we plan to integrate transcoding and multiplexing strategies for audio streams.

5.2 Overall Architecture

The ISDN Multicast Gateway system consists of a server and a client component. The client component, programmed in Java, mainly provides a graphical user interface for the communication with the server component. The server component carries out the scaling and reflecting of the data streams: it is implemented in C++ to ensure good performance. In principle it works in the following way: Users can connect to the server through the client application via a dedicated port number. Once a user has logged in the client will receive a list of sessions available for transmission. The user will then choose a session to join. A request for the selected session will be sent to the server, which then will transmit and rescale the media stream according to the bandwidth available.

The following list summarises the features of the ISDN Multicast Gateway:

- distributed client-server oriented architecture,
- integrated solution with an-easy-to-handle graphical user interface,
- client is implemented in Java and therefore platform-independent,
- support of the most popular MBone Tools (audio, video and whiteboard).

5.2.1 Technical Outline of the Server Component

The server consists of the components indicated in Figure 6:

- Client Communication Manager,
- Session Announcement Protocol Listener (SAP-Listener),
- Reflector Service, and
- Media Filtering Service.

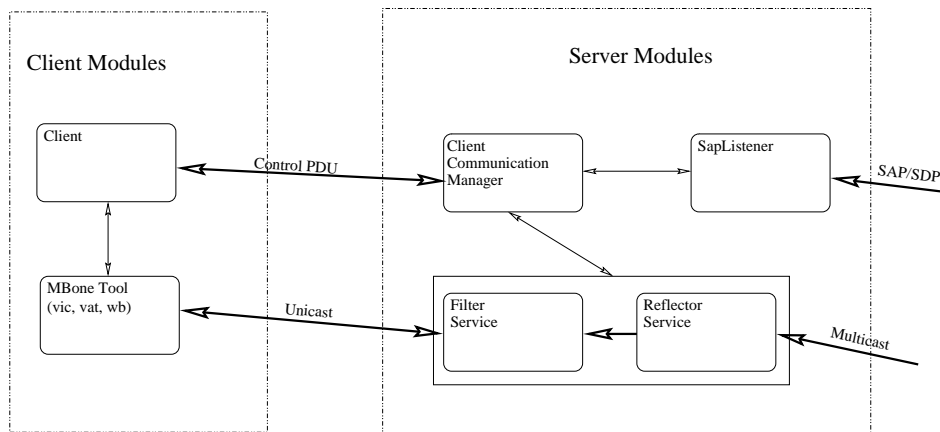


Figure 6: The ISDN Multicast Gateway.

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. Clients communicate with the server by transmitting control protocol data units (PDUs) to the remote site. A PDU holds a command token along with parameters. The token and the parameters are transmitted in a single datagram as character strings separated by zero-bytes. Table 2 summarises the most important PDUs we have defined so far.

PDU	Parameters	Description
ConReq	login, password, bandwidth	Connect Request PDU is sent by a client in order to connect to the server.
ConConf	ctrl-id	Connect Confirm PDU is sent by the server if a connect request of a client is accepted. The server sends a unique identifier (ctrl-id) to the client.
SesInf	SDP-description	A SesInf PDU is sent by the server if a new session description is received by the SAP-Listener component. The PDU contains the new session description.
SesReq	ctrl-id, sdp-id, media-id, bandwidth	A Session Request PDU is sent by the client in order to request the transmission of a media-stream of a certain session.
SesAck	port	A Session Acknowledgement PDU is sent by the server. The PDU holds the information on the port to which the data is sent.

Table 2: Control PDUs of the ISDN Multicast Gateway

The Session Announcement Protocol Listener receives Mbone session descriptions, described according to the session description protocol SDP [Han97b] and announced in conformance with 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 is the central component and the most time-critical part. When a client requests a multicast session, the client communication manager adds the requesting client to the list of receivers of the corresponding Reflector Service. Multiple clients can participate in a reflected session. 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. Its task is to broadcast incoming multicast packets to all of its customers.

The Media Filtering Service receives incoming data stream packets and either enqueues them to a PQ or discards them as described above. In the current version only video streams are scaled down and we use the very simple temporal scaling mechanism described in Section 5.1.3.

5.2.2 Technical Outline of the Client Component

The client component is available both as a stand-alone Java application and as an applet. When a client connects to a server it receives the announcements of those Mbone sessions¹ currently available. The user may then choose the sessions to be received. The client sends the request back to the server process which then transmits the requested data stream to the client. At the client site the Mbone tools are started automatically.

Figure 7 illustrates the graphical user interface of the client.

¹announced on sdr

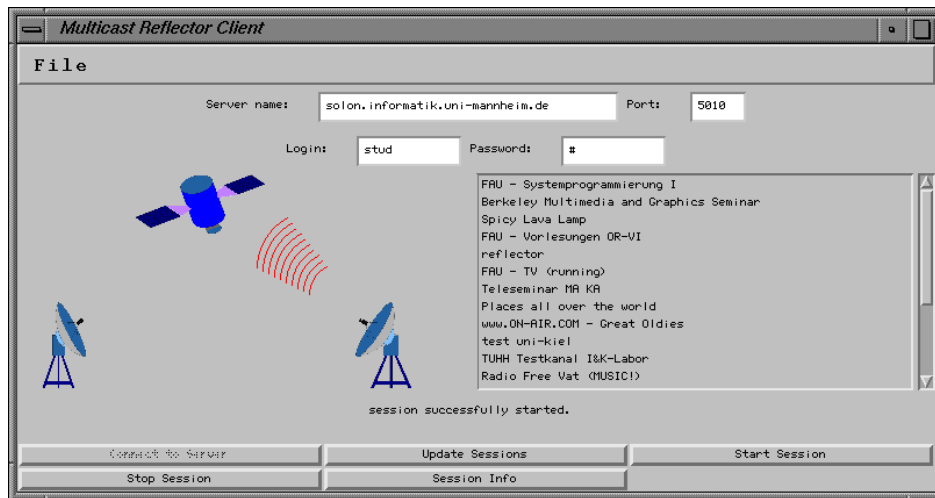


Figure 7: The graphical user interface of the client

6 Experiences

During the last summer term we extensively tested the ISDN Multicast Gateway in our Interactive Home Learning scenario. In this scenario 12 students received our teleteaching lecture in computer science through the ISDN access node of our computing center. Simultaneously the lecture was transmitted to lecture halls of other universities. The multicasted video stream had a bandwidth of about 600kbit/s with up to 12 frames per second. In order to transmit the video to the students at home we had to set the priority factor of the video PQs to 64kbit/s which resulted in a low quality video signal with only a few frames per second. The probant students judged the quality of the video as sufficient. The voice of the lecturer was transmitted in GSM compressed format which resulted in a data stream with about 15kbit/s. The students found the quality of the audio acceptable for a lesson in computer science although they mentioned that they found it hard to concentrate for more than 60 minutes and a better audio quality would be appreciated. The whiteboard data stream could only be received by 6 students who used Linux on their PC at home. We had no problems to transmit the whiteboard data stream with a priority value of 32 kbit/s although it sometimes took quite some time to transmit large postscript files.

Conclusively our evaluation proved that even low bandwidth connections such as ISDN allow the participation in a multicast session with media streams. In future we plan to make use of new dial-in connections like ADSL which provide a higher bandwidth for users at home.

7 Conclusion and Outlook

In this paper we described Interactive Home Learning, a teleteaching scenario focusing on low-cost, low-bandwidth Internet connections via ISDN. We discussed technical challenges in this scenario and described earlier work on this topic. We then presented our approach, the *ISDN Multicast Gateway*. It implements an application layer multicast routing system with built in media scaling. The client component of the ISDN Multicast Gateway is implemented completely in Java using JDK1.1 and thus works on every platform. The server component is written in C++ and was ported successfully to several different platforms (SUN Solaris 2.6., Linux, and Irix). The system fulfills not only the requirements

of the Interactive Home Learning scenario but can also be used for other services, such as Media-on-Demand servers and multimedia databases.

Future work will comprise more complex scaling algorithms for video, which will allow a more graceful scaling. Also we are investigating the use of RTSP [Sch98] as our control protocol, which would allow a more facile integration of the ISDN Multicast Gateway into other tools.

8 Acknowledgment

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

References

- [Ami95] Elan Amir, Steven McCanne, and Hui Zhang. An Application-level Video Gateway. In *ACM Multimedia, San Francisco*, pages 255 – 265, November 1995.
- [Ami96] Elan Amir, Steven McCanne, and Martin Vetterli. A Layered DCT Coder for Internet Video. In *Proc. of IEEE International Conference on Image Processing ICIP '96, Lousanne Switzerland*, pages 13 – 16. IEEE, September 1996.
- [Bur83] Peter Burt and Edward Adelson. The Laplacian Pyramid as A Compact Image Code. *IEEE Transactions on Communications*, 1983.
- [Dee89] Stephen Deering, C. Partridge, and D. Waitzmann. Distance Vector Multicast Routing Protocol. Internet Request For Comments, IETF, RFC-1075, August 1989.
- [Dee91] Stephen Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, 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 in Telecommunications and Multimedia – A Compound Organizational, Pedagogical and Technical Approach. In *Proc. ED-MEDIA/ED-TELECOM'97, Calgary, Canada*. AACE Association for the Advancement of Computing in Education, 1997. [on CD-ROM only].
- [Han97a] Mark Handley. SAP: Session Announcement Protocol. Internet draft, IETF, Multiparty Multimedia Session Control Working Group, draft-ietf-mmusic-sap-00, expiring 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, expiring date September 26th, 1997 1997.
- [Hof96] Don Hoffmann and Michael Speer. Hierarchical video distribution over Internet-style networks. In *Proc. of the IEEE Conference on Image Processing*, pages 5 – 8, Lousanne, Switzerland, September 1996.

- [Hol97] Wieland Holfelder. Interactive Remote Recording and Playback of Multicast Videoconferences. In *Proc. 4th 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, 1st edition, 1995.
- [Jak96a] Van Jakobson and Steven McCanne. Video conferencing Tool (vic). Lawrence Berkeley Laboratory, <ftp://ftp.ee.lbl.gov/conferencing/vic>, 1996.
- [Jak96b] Van Jakobson and Steven McCanne. Visual Audio Tool (vat). Lawrence Berkeley Laboratory, <ftp://ftp.ee.lbl.gov/conferencing/vat>, 1996.
- [Kuh98a] Christoph Kuhmünch, Thomas Fuhrmann, and Gunther Schöppe. Java Teachware - The Java Remote Control Tool and its Applications. In *Proc. ED-MEDIA/ED-TELECOM'98*. AACE Association for the Advancement of Computing in Education, 1998. [on CD-ROM only].
- [Kuh98b] Christoph Kuhmünch and Gerald Kühne. Efficient Video Transport over Lossy Networks. Technical Report 7-98, University of Mannheim, April 1998. <http://www.informatik.uni-mannheim.de/cjk/publications>.
- [McC95] Steven McCanne and Van Jacobson. vic: A flexible Framework for Packet Video. In *MultiMedia '95 (San Francisco)*, New York, November 1995. ACM, ACM Press.
- [McC96] Steven McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, Ca, USA, 1996.
- [McG92] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). Internet Request For Comments, IETF, RFC-1332, May 1992.
- [Mer97] Michael Merz, Konrad Froitzheim, Peter Schulthess, and Heiner Wolf. Iterative Transmission of Media Streams. In *Proceedings of the conference on Multimedia '97*, pages 283–290. ACM, 1997.
- [Par97a] Peter Parnes. mMOD: The multicast Media-on-Demand system. In *NOSS-DAF'97*, 1997.
- [Par97b] Peter Parnes. mTunnel: a multicast tunneling system with a user based Quality-of-Service model. In *Proc. 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97)*, 4. IDMS, September 1997.
- [Sch96a] Henning Schulzrinne. RTP Profile for Audio and Video Conferences with minimal control. Internet Request For Comments, IETF, RFC-1890, January 1996.
- [Sch96b] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications. Internet Request For Comments, IETF, RFC-1889, January 1996.
- [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, expiring date August 2nd, 1998. 1998.