The Multicast Multimedia Conference Recorder

Lambros Lambrinos, Peter Kirstein and Vicky Hardman
{L.Lambrinos/P.Kirstein/V.Hardman @cs.ucl.ac.uk}
Department of Computer Science, University College London
Gower Street, London WC1E 6BT, UK

Abstract

The ability to archive multimedia data is required in most applications of multimedia conferencing. This paper describes the generic client-server architecture of a multicast multimedia server system and the design and implementation of the Multicast Multimedia Conference Recorder (MMCR).

The MMCR is a remotely controlled system capable of recording and playing back multimedia data over the MBone. It has a client-server architecture and the logical independence between components simplifies development and component replication. The MMCR uses indexes to facilitate fast data access, allowing efficient fast-forward, rewind and random access operations. Inter-stream synchronisation between recorded audio-video streams is improved by eliminating the network delay variance.

1. Introduction

Multimedia facilities are available on a variety of platforms for desktop multimedia conferencing; users interact with each other in real-time using audio, video and shared workspace tools. The ability to archive multimedia data from conferences and to introduce stored data into conferences is required in many multimedia conferencing applications. Recording a session, enables anyone to replay it and find out (or review) its content or to introduce parts of the recording in an on-going session in order to illustrate a point.

We consider here systems designed for the Multicast Backbone (MBone)[1], a virtual network layered over the Internet that provides multi-point packet delivery. Multicast-capable multimedia servers offering recording, playback and editing facilities must be an integral part of the emerging multimedia computing infrastructure.

The paper gives a brief introduction to multimedia conferencing and then presents the requirements of multimedia servers. The generic architecture of a multimedia recording/playback system is then described and existing systems are studied. The UCL Multicast Multimedia Conference Recorder (MMCR) satisfies many of the requirements identified. The MMCR is a client-server system designed for remote interactive recording and playback of multicast multimedia conferences. It addresses the characteristics of the various media types and its flexible architecture allows easy addition of new components. The MMCR uses real-time multimedia indexes to facilitate fast data access, allowing efficient fast-forward, rewind and random access operations. Details of the implementation are given and some data quality observations described. Methods to eliminate the network delay variance (imposed on streams during recording) are used in MMCR; these improve playback and stream synchronisation. Finally, future MMCR enhancements are stated.

2. Multimedia Conferencing over the MBone

Low-cost networks and personal computing facilities that can cope with real-time multimedia have encouraged the widespread use of desktop multimedia conferencing. A typical multimedia conferencing session on the MBone consists of real-time audio and video, and 'shared workspace' media (e.g. an interactive drawing tool).

The various applications of multimedia conferencing include distance learning (students at remote sites interact with the teacher[2]), major event broadcasts (e.g. viewing NASA landings) and interactive discussion or consultancy (e.g. research collaboration meetings[3]).

MBone conferencing tools[4-8] are publicly available on UNIX and Windows platforms. Separate tools allow users to use a selection of media, which retains flexibility. 'SDR'[9] is a MBone session announcement tool that can invite users to join sessions.

Data encryption[10] enhances the privacy of MBone multimedia conferencing. Only participants in possession of a valid decryption key can decipher the media streams.

The ability to record live sessions and to introduce pre-recorded ones is important. Recording a session (or parts of it) allows later replay (e.g. if someone missed a seminar). Playing back a clip in a live session may emphasise a point (e.g. parts of operations for medical training, flight simulations in aerospace conferences etc.).

The ReLaTe[2] project pilots remote language teaching. Small group tutorials are run using geographically distributed real language teachers and students. In this scenario, the students and teachers need the ability to review past lessons using a playback tool.

2.1 Multicast data distribution

The MBone provides multi-point packet delivery by using IP multicast[11] that allows a source to send data to multiple recipients without addressing them individually. Multicast-capable routers only forward data to sites that have expressed an interest in the material. Multicast data is sent using the Universal Datagram Protocol (UDP), which provides a best-effort delivery service.

Applications transmitting real-time audio and video use the Real-time Transport Protocol (RTP)[12] on top of UDP. RTP provides information such as encoding type, sender id and data timestamps. RTP is used with an associated control protocol RTCP[12] that provides data delivery monitoring and session participation information in packets generated at fixed intervals. Use of RTP/RTCP does not guarantee quality of service; delivery is not guaranteed and delay may not be bounded. Interactive use of real-time media means that long delays (greater than a few hundred ms) cannot be tolerated.

Many multicast tools (such as shared-workspace) require reliable data delivery. A reliable multicast protocol (on top of UDP) guarantees that the data will eventually reach all participants. An example is Scalable Reliable Multicast (SRM)[13] where recipients detecting loss send out a general request for repair and the nearest up-stream source retransmits the requested data.

2.2 Multimedia Conferencing Standards

Apart from the data delivery protocols, other standards may influence the design and operation of a recording/playback system.

The Session Description Protocol (SDP)[14] is an IETF proposed standard for describing conferencing sessions. SDP information includes session duration, encoding formats, and the multicast addresses used for the session’s media streams. The Session Announcement Protocol (SAP)[15] is a standard for announcing sessions and the Session Initiation Protocol (SIP)[16] is a standard for inviting particular entities to join a session. SAP and SIP use SDP to provide the session's details.

2.3 Transmission versus Storage Formats

The RTP/RTCP data transmission format allows senders and receivers to discover missing packets. If the real-time data incorporates Forward Error Correction (FEC)[17] (e.g. a low quality version of an earlier audio packet[18]), receivers attempt to repair the stream. For reliable multicast traffic, retransmission requests and any re-transmitted data can be ignored if the data has already been received by the recorder.

Thus, by using the properties of the tools it is possible to reconstruct a reasonable replica of the data originally transmitted at the time, but not later. Subsequent requests for transmission of the stored data may be over a network with different error characteristics or speeds so that different transmission formats may be optimal.

Most of the MBone-based recorders just store the data packets; this makes storage and replay easier but may lead to sub-optimal performance. Most current storage formats are vendor-specific. There has been little activity on standardising these (except ASF[19], QuickTime[20]).

3. Multimedia recorder/player requirements

This section identifies the requirements that a system for recording/playing back MBone sessions must fulfil.

3.1 Recorder requirements

- ***Recording from the MBone*** the data transmitted during a session. The various data transmission protocols (e.g. RTP, SRM) used must be handled appropriately.
- ***Error Recovery*** techniques (i.e. FEC) used in the transmission should be utilised by the record mechanism prior to storage of the data.
- ***Data Formatting*** capabilities are required as it may be best to store the data in a format different from the transmission format. The storage format should include timing and stream information to allow faithful play-out of the stored data.
- ***Access to a large data repository*** (the users do not need to know its location) is essential due to the amount of data contained in the media streams (especially video).
- ***Encrypted data storage*** may be desirable (not withstanding the access control mentioned below). The recorder will normally be made a trusted member of an encrypted session to utilise any data repair operations prior to storage and to allow the use of encryption algorithms more suitable for storage.

3.2 Player requirements

- ***Playing back to the MBone or to a single user*** the streams (all or a selection) of an archived session.
- ***Random access*** facilities (including ‘fast-forward’ and ‘rewind’) are required during playback.
- ***Encrypted data playback*** to maintain confidentiality is required. Additional checks will be needed before decryption keys are provided.
- ***Data formatting*** capabilities are needed in the player

to enable it to reformat the steams according to the re-transmission medium and use the appropriate packaging like RTP, SRM, etc. Additionally, the ability to play back media streams other than those of pre-recorded MBone sessions (e.g. digitised MPEG streams) can be useful.

3.3 Other requirements for a multimedia server

- ***Authentication*** mechanisms using Access Control Lists and strong authentication should be used to prevent unauthorised access to the server's facilities and data.
- ***Remote Interactive*** set-up and control of recordings and playbacks is essential. A locally controlled system is too inflexible.
- ***Editing*** facilities are desirable to enable users to create their own material from pre-archived sessions.
- ***Browsing*** facilities for retrieving/searching details of existing recordings (e.g. subject, media types) are needed.
- ***Annotation tracks*** can be a very useful adjunct to aid navigation in streams during playback and editing.

4. Multimedia Recording & Playback systems

Some existing multimedia server systems have the MBone as their application area whereas others aim to work over the Internet or high speed Intranets.

4.1 Generic Systems

All systems have a similar architecture (Fig. 1).

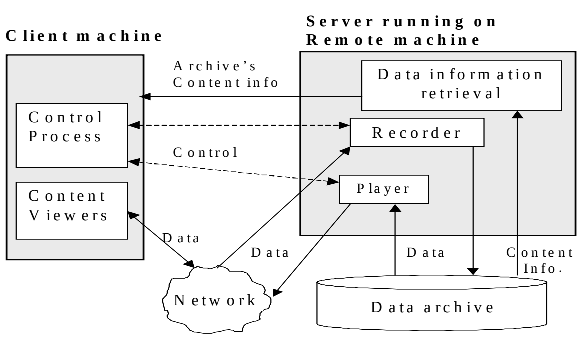

Figure 1: Generic Multimedia Server architecture

Systems have differing characteristics with respect to the generic architecture shown in Fig. 1. Many Video-on-Demand systems (e.g. Real-Video[21], Net-Show[22]), concentrate only on the Player section and play through an Intranet, transmitting multiple unicast streams (without multicast). They usually have proprietary data formats in the data archive. Some have WWW based clients with specific plug-ins for the receivers. Many such systems are designed for high performance, multiple sessions.

In the rest of this paper we only consider systems that work with multicast and use the MBone conferencing tools for their recording/playback operations.

4.2 Standards for Recording/Playback systems

There are considerable variations on the client-server dialogue. While many of the current systems have *ad hoc* control of the Server from the client, some are starting to use the Real-Time Streaming Protocol (RTSP)[23] to specify this control. RTSP is a recent IETF standard with the general aim of controlling all classes of recorders and players. There is insufficient experience on whether enough common features are used so that clients from one system can control servers from another.

Many server systems allow clients to browse the archives using a WWW interface. Again, the way that information on stored sessions is shown on the client varies in the different implementations.

Some standards used in multimedia conferencing (Section 2.2) can be used during recording/playback of sessions. The information in a Session Description is required for setting-up the recorder and it should be stored with the recorded streams. There are implementations of SAP (e.g. 'sdr') which allow for a Record option; this could start a specific recorder when the session is started. Additionally, SIP can be used to invite a player (not yet done but we expect this to happen shortly) to an on-going session to arrange for the introduction of recorded clips.

The recorders may operate in several ways. The simplest is to store the media packets as they arrive, usually together with time-stamps. As discussed in Section 2.3, it is often better to store the media in a more suitable storage format. This requires that some parts of the original media tool are also embodied in the recorder. This allows the recorder to undertake the stream repair so that the information stored on the server has the same quality as that in co-located real-time conferencing tools.

The players must operate in an analogous way to the recorder. If the native media packets are stored, then the player can play these back using the stored timestamps to pace the output. If a different storage format has been used for the media data the player must also incorporate format converters and parts of the media tools.

4.3 Existing Multicast Multimedia Servers

It is important to review briefly some systems that now exist. Some of the system components identified in Fig. 1 may be missing in specific implementations.

The MBone VCR On Demand (MvoD)[24] system offers interactive remote recording/playback of MBone sessions. The client (VCR-like interface) can initiate and control the facilities of the remote server. The system has its own protocols for client-server communication and an interface to SAP for receiving session announcements. MVoD only supports RTP-compliant media.

The Interactive Multimedia Jukebox (IMJ)[25] uses a scheduling mechanism to distribute RTP audio and video over three multicast channels. Playback of a session is requested via a WWW-based user interface; the scheduler then assigns a slot on a channel. The IMJ lacks in end-user interactivity; recorded material is only produced off-line and there is no stream control during playback.

A commercial system is Cisco's IP/TV[26]. It has a client-server architecture and the facilities to record live MBone sessions. It can also multicast stored AVI video streams. IP/TV plays out sessions based on a schedule.

The MASH project[27] at UC Berkeley incorporates a record/playback system that uses RTSP for client/server communication and the WWW for content advertisement. Special attention is given to annotations for identifying important parts of a session. Recent work has focused on reliable client-server communication[28].

5. The MMCR

The Multicast Multimedia Conference Recorder (MMCR) is a system specifically designed for recording and playing back multimedia conferences over the MBone. It provides a vehicle for the minimal facilities required and can be extended to provide much more. Its architecture (Fig. 2) consists of the graphical client and the server (incorporates the playback, recording and browsing mechanisms); logical component independence simplifies development and component replication.

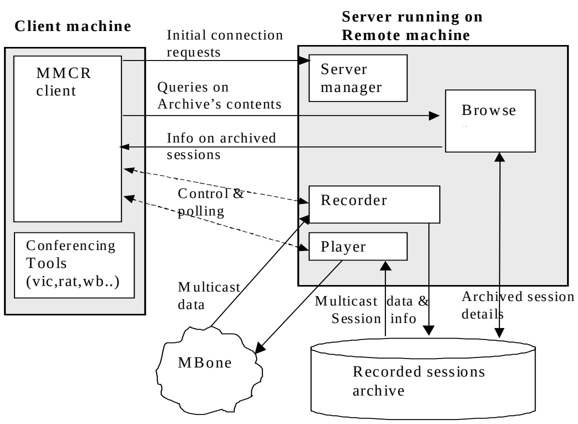

Figure 2: The MMCR architecture

All server components can access the data archive to store/retrieve recordings and their details. Much research has considered ways of providing efficient storage/access mechanisms for Video On Demand systems [29,30], that require high bandwidth delivery. However, with the current MBone bandwidth limitations the simple disk model used here is adequate when considering the simultaneous replay of a small number of sessions.

5.1 The Server and its components

The server acts as the single point of contact for recording, browsing and playback. The **server manager** controls the whole service; it has a separate interface for each task (can add more interfaces e.g. for editing). The client requests a service and the server manager starts the appropriate mechanism. Once the mechanism has started, the client communicates directly with that mechanism. Each mechanism has its own text-based control protocol.

To record the media streams the **recorder** need not be an active part of the conference; it simply 'listens' to the specified multicast groups and collects the data.

The files of a recorded session are organised in a 3-level hierarchy. At the top is the conference header file that contains general session details (title and description as given by the user or included in the Session Description) along with the recorded media details (name and type of each medium). Then are the per-medium header files that log each independently stored source that sent data of this medium (participant's name and location of source's files). At the end are each source's files which consist of a header file (holds the location of the other source's files) and the data, index and annotations files.

Storing data on a per-source basis allows utilisation of all kinds of networks as users with bandwidth limitations can play a subset of the available streams (e.g. all the audio and a subset of the bandwidth consuming video streams).

An important feature of the system is the **real-time multimedia index.** For each source recorded there is an index created – the 'primary index' (Fig. 3) that contains an entry per data packet received. Each index entry consists of the time the packet was received at the recorder and the offset into the data file that the packet begins. This structure is used for efficient data access as its size is a lot smaller compared to the data size.

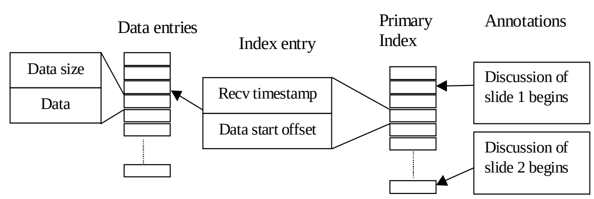

Figure 3: The Primary index structure

The addition of annotations allows users to add meta-data to a recording to describing particular sections. Each annotation points to the relevant index entry. Annotations can take the form of a book contents list and provide rapid access to stream sections that are of interest to a user.

The **browsing mechanism** provides a listing of the sessions in the server's archive. A title keyword search facility is also available to help identify titles of interest.

Further details about a particular session can also be obtained to assist a user in deciding which session to play back. They include the session's title, description and the media that constitute the session. Additional information on each medium includes the data type (e.g. RTP) and the names of the participants (where available) to help users identify and select only the required data streams.

The **player** schedules real-time packet transmission based on the timestamp in the index entry. RTP-compliant media provide additional information that can be used for a smoother playback. Other packets (e.g. shared workspace data) are scheduled based on their received timestamp (i.e. with the inter-packet gap they arrived at the recorder).

The different media characteristics affect the fast-forward and rewind operations. Audio and video are continuous media; moving to a random point in the stream simply involves skipping intermediate parts and restarting at the new position. The RTP data packets must be modified to maintain the continuity in timestamps and sequence numbers. For non-continuous media (such as shared workspace), fast-forward should involve the transmission of intermediate parts so that the data set is complete at least from a checkpoint (e.g. start of a page).

5.2 The implementation of MMCR

MMCR is currently under development as part of the MECCANO project[3]. The client software and the server (runs on Unix workstations) are written in Java and the playback, recording mechanisms are written in Objective C. The client is totally independent of the server and runs on Unix workstations and PCs.

Currently, MMCR has its own text-based control protocol for client-server communication. This will be replaced with an implementation of RTSP. The Session Description Protocol is used for recording initiation and scheduling (this is achieved via a 'record' option in 'SDR').

The system supports recording and playback of any type of media sent over the MBone (e.g. RTP audio/video, shared workspace etc). When recording RTP data, the RTCP messages are also stored. For RTP-compliant audio and video streams some data correction is performed. Encrypted sessions can also be recorded and played-back (for playback, receivers need the original session's decryption keys) but the server can not extract any information from the encrypted data.

Partial recording of a conference is possible both as to the streams (e.g. only record the audio) and the duration (e.g. record for 1 hour). The facility to 'pause' and 'resume' recording is also provided.

The browsing mechanism is implemented as described in Section 5.1. The required session and the selected media sources are passed from the browser to the player to access the corresponding data streams and send them to the specified multicast groups. Users must start the appropriate conferencing tools to receive the data. The playback functionality provided consists of play, pause, fast forward, rewind and random access operations.

6. Practical experience and data repair

The 'best effort' model for data delivery relies on the current network conditions. As a result, the quality of the real-time data may be degraded. During playback, the player aims to reproduce the original session (as produced by the original sources) as accurately as possible but this relies on the quality of the data received by the recorder.

We used the MMCR implementation to monitor the effect of the network on the data arriving at the recorder. Our observations showed a need to eliminate the network delay (jitter) imposed on packets to improve playback and stream synchronisation. Algorithms to partially eliminate this effect were devised. We can extract loss and timing information from RTP-compliant media and the work described below only applies to these media types.

The graph below shows the difference in jitter (in milliseconds) between the first packet of an audio talkspurt and the rest of the talkspurt's packets. The data source was in the US and the recorder in the UK. The graph represents one minute of audio. Clearly, the jitter can vary significantly (especially when the data source is far from the recorder) and in some cases it has a small negative value (these packets were not delayed as much as the first).

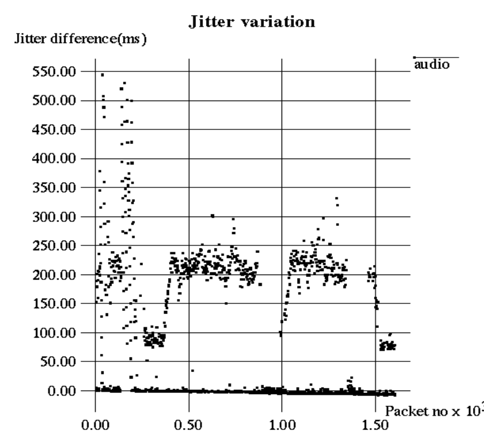

Figure 4: Packet jitter variation

An exact replay of the original conference (excluding any lost data) is possible if the jitter is completely removed. The real-time representation contained in the RTCP packets combined with the RTP data timestamps can be used to retrieve the actual time the packets were sent. But, there is no guarantee that the clocks of the machines from the original session were either synchronised with each other or with the server. Unsynchronised clocks will result in out of phase streams; thus, we do not attempt to calculate the original sender transmission time. As a partial solution, we calculate the minimum jitter (as seen by the recorder) and use it to eliminate the jitter variation.

This method is based on a combination of the RTP timestamp in the data packets and the received timestamp in the index (Fig. 3). It is performed on the data streams immediately after recording. Because of the differing characteristics of audio and video, different (but similar) algorithms are used. The table below shows an example of the algorithm's results for audio.

Packet number	1	2	3	4	5
Packet RTP timestamp	2385920	2386240	2386560	2386880	2387200
Packet receive timestamp	238ms	265ms	301ms	330ms	372ms
RTP difference from first *	-------	40ms	80ms	120ms	160ms
REC difference from first	-------	27ms	63ms	92ms	134ms
New start REC timestamp**	210ms	-------	-------	-------	-------
New REC timestamp ***	210ms	250ms	290ms	330ms	370ms

* The difference in milliseconds between the current packet and the first, derived by using the RTP timestamps and the audio frequency
** Smallest of (Packet's received timestamp – RTP difference from first)
*** Result of (new start REC timestamp + RTP difference from first)

Using the RTP timestamps we eliminate the variation in the jitter imposed by the network during recording. Additionally the original inter-packet gap is restored. That results in an acceptable level of inter-packet synchronisation as the data is transmitted with the same order and inter-packet gap it left the original sender. Packets are sent to the network smoothly avoiding bursts that may cause further losses and will arrive to the receivers with only the jitter variation imposed during playback. Thus, lip-synchronisation[31] becomes easier for receivers who expect packets to arrive within a certain time period.

7. Conclusion and further work

There is currently a lot of interest in the field and some commercial systems are emerging. Nevertheless, we consider MMCR to be a suitable prototype for doing research in the field of Multicast Multimedia Servers. In the near future it will support RTSP allowing people to use their own RTSP clients. The facility to 'invite' the server to playback a recorded session using SIP will also be provided. Other issues that will be examined in further work include:

- Distributed recording that will minimise losses and jitter. This will improve the data quality in recordings.
- Making the server a trusted member of an encrypted session to enable decryption of the streams allowing extraction of useful information (e.g. RTP timestamps).
- Storage in an appropriate format after stream repair.
- Provide editing facilities by allowing copying of the required index segments and joining them to form another index. All index copies will point to the single data copy.

This paper has presented the requirements and design/implementation of a multicast multimedia server. The MMCR is not a complete system, but its flexible architecture makes it suitable for research purposes.

Acknowledgements

The work on MMCR is funded by DARPA and the European Commission (MECCANO project - Telematics Applications Programme, Research Sector).

References

[1] H.Eriksson – "MBone: The multicast backbone" – Communications ACM, vol. 37, Aug. 1994
[2] See http://www-mice.cs.ucl.ac.uk/relate.html
[3] See: http://www-mice.cs.ucl.ac.uk/meccano
[4] S.McCanne, V.Jacobson -"vic: A Flexible Framework for Packet Video" - ACM Multimedia, November 1995
[5] See: http://www.cs.ucl.ac.uk/mice/rat
[6] V.Jacobson, S.McCanne – "vat – X11 based audio teleconferencing tool" - vat manual page - Lawrence Berkeley Labs, University of California
[7] V.Jacobson, S.McCanne-"Using the LBL network whiteboard" - Lawrence Berkeley Labs, University of California
[8] M.Handley, J.Crowcroft – "Network Text Editor (NTE): A scalable shared text editor for the MBone" - ACM SIGCOMM 1997
[9] See: http://north.east.isi.edu/sdr/
[10] ANSI X3.92-1981. American National Standards Data Encryption algorithm. American National Standards Institute. Approved Dec 1990
[11] S.Deering – "Multicast Routing in a Datagram network" – PhD thesis – Stanford University
[12] H.Schulzrinne, S.Casner, R.Frederick, V.Jacobson-"RTP: A Transport Protocol for Real-time applications"-RFC 1889
[13] S.Floyd, V.Jacobson, S.McCanne – "A Reliable Multicast Framework for light-weight Sessions and Application Level Framing"–IEEE/ACM transactions on networking – Nov. 96
[14] M.Handley, V.Jacobson – "SDP: Session description protocol" – RFC 2327
[15] M.Handley – "SAP: Session Announcement Protocol" – Internet draft
[16] M.Handley, H.Schulzrinne, E.Schooler - "SIP: The Session Initiation Protocol" – Internet draft
[17] J.Rosenberg, H.Schulzrinne - "An RTP payload format for Generic Forward Error Correction" - Internet Draft
[18] Perkins et al - "RTP Payload for Redundant audio" - RFC 2198
[19] Advanced Streaming Format (ASF) - Microsoft– See http://www.microsoft.com/asf/
[20] See: http://www.apple.com/quicktime/
[21] See: http://www.real.com
[22] See: http://www.microsoft.com/ntserver/netshow/
[23] H.Schulzrinne, A.Rao, R.Lanphier – "Real Time Streaming Protocol (RTSP)" – RFC 2326
[24] W.Holfelder - "Interactive Remote Recording and Playback of Multicast Videoconferences" – IDMS'97
[25] K.Almeroth, M.Ammar – "The Interactive Multimedia Jukebox (IMJ)" – Seventh International WWW Conference, Brisbane, Australia, Apr 1998
[26] See: http://www.cisco.com/iptv
[27] See: http://www-mash.cs.berkeley.edu/mash/
[28] A.Schuett et al - "A Soft State Protocol for Accessing Multimedia Archives" - NOSSDAV, July 98
[29] M.Buddhikot – "Design of a large scale multimedia server" – Proc. INET 94
[30] K.Nishikawa – "High performance VOD server aims" – Globecom 95
[31] I.Kouvelas, V.Hardman – "Lip synchronisation over the Internet: Analysis and implementation" – Globecom 96