The SICMA teleteaching trial on ADSL and Intranet networks*

F. Cortes[2], V. Darlayiannis[1], M. Herreo[2], G. Kyriakaki[1], R. Lüling[2], Y. Maragoudakis[1], Y. Mavraganis[1], K. Meyer[2], N. Pappas[1]

[1] Laboratory of Distributed Multimedia Information Systems and Applications
Technical University of Crete, Greece
[2] Department of Mathematics and Computer Science
University of Paderborn, Germany

Abstract. The provision of interactive multimedia services, such as video-on-demand, teleshoping and distance learning, to a large number of users, still remains a challenging issue in the multimedia area. Despite of recent technological advances in all levels of the distributed multimedia infrastructure (storage, network, compression, standardization etc..), there is a strong need for feedback from public trials. Trial descriptions and evaluations, by revealing potential system limitations and measuring end user reactions, will provide valuable input towards the large-scale deployment of such services. In this paper, we present the teleteaching trial that was held at the end of 1998, at Limburg University (Belgium), in the context of ACTS SICMA project. We describe in detail the overall architecture and present results/implementation experiences for the parts of the system, putting the main emphasis on the server. We present technical integration issues between DAVIC and Internet server protocols (RTSP, DSM-CC etc..), and discuss the overall trial results.

1 Introduction

The SICMA project (Scaleable Interactive Continuous Media Servers - Design and Application) is part of the program on "Advanced Communication Technologies and Services (ACTS)" of the European Union. The general aim of SICMA is to design a scaleable media server for the delivery of continuous multimedia information, over high speed networks, to a large number of clients. Server's efficiency, interoperable and scaleable architecture are demonstrated with the help of relevant applications, in the context of two public trials. In first SICMA trial, the server was used to support the "Virtual Endeavour" application at Natural History Museum, London, in summer 1997. This year, the server was integrated in a teleteaching application at Limburg University, Belgium, where it provides video on demand services for the students.

The SICMA server complies to the DAVIC standard [3]. The DAVIC Services are built on top of the KYDONIA system [2], a multimedia object-based

* This work was supported by the EU project SICMA, AC071 in the framework of the program "ACTS - Advanced Communication Technologies and Services"

management system, that has been developed by MUSIC/TUC in previous European projects and is expanded within SICMA for providing full multimedia support. The main features of the system are: storage management for multimedia objects, real-time data pumps towards multiple clients over ATM, multimedia object modeling, text and video access methods, browsing and content addressability techniques.

While supporting the DAVIC standard, we decided to expand the SICMA media server to support the IETF standards for delivery of continuous media formats (RTSP/RTP/RTCP). In this way, the server is able to provide services in environments that comply to both standards. Media content is stored once on the server but can be made available using both protocol families.

The final aim of SICMA is to build a parallel multimedia server because parallel systems are able to deal with a very large disk array. Compared to sequential systems, they are able to store a larger number of media streams and to serve the requests of a larger number of users at the same time. While parallel systems are hard to use for certain problems, they seem to be well suited for the delivery of continuous media. This is because users usually access streams stored on the server independently from each other. Thus, a natural parallelism is given and can be exploited most efficiently by a parallel system. In this paper, we present an evaluation of the server system within the teleteaching application that was integrated as the second SICMA trial at the end of 1998. This application demonstrates the performance of the SICMA server for the delivery of continuous media information, within a local Intranet that is build up by ATM and standard Ethernet, as well as by using an ADSL network.

The paper is structured as follows: In next section the teleteaching trial is described in detail. After this, the SICMA media server is presented. Results-experiences from the trial finish the paper.

2 Teleteaching trial overview

The aim of this trial was to demonstrate the performance of the SICMA media server technology by integrating it into an available teleteaching environment. The main requirement was the delivery of MPEG encoded data to client systems, being connected to the server via the Intranet of the University as well as via ADSL lines. In the following subsections we describe the teleteaching system being developed and used for the SICMA project and afterwards the integration of the SICMA media server in this environment.

2.1 The teleteaching system at Limburg University

The teleteaching system developed at Limburg University Center (LUC) is a generic system for the creation and delivery of multimedia teleteaching content, stored on a server system, to a large number of clients (PC systems). The system is based on standard Internet technologies, e.g. HTML pages, CGI scripts, JAVA applets, standard database integration and related systems. The overall system is based on three types of server systems:

– Web servers (storing conventional web information, pictures, texts, ...)
– Database servers (hosting a relational database allowing to perform search operations on the overall content)
– Media servers (storing media assets, e.g. audio and video streams that are streamed from the server to the client)

The student, as a user of the teleteaching system does not recognize the three systems, but has access to the overall content via a standard web interface that integrates all information sources in a seamless way. The standard way to access the teleteaching application is via a search menu that allows to search for items stored on the web server and on the media server. Using the results of a search operation, the user gets a list of all items that fall into his interest. These can be single lectures, pointers to external resources on the Internet, single items like slides, animations, video streams and many other monomedia items. Using suitable plug-ins these items can be displayed and animated for the user.

To develop a teleteaching application, interested professors get access to a number of tools that allow an easy integration of their content in a full teleteaching application. The tools provide a number of frameworks for the integration and indexing of texts, graphics, video clips and other monomedia components. In this way, an interested professor can easily integrate his content into a teleteaching application without knowing the technical details of the underlying software models and server systems.

2.2 Integration of the SICMA server into the teleteaching application

The second trial phase of the SICMA project, presented in this paper, aims at demonstrating the scalability of the SICMA media server system. In the first SICMA trial [7], only a small number of user clients were used to demonstrate the functionality of the server system. In the second trial phase a larger number of clients are connected to the server system. It was foreseen to allow the integration of at least 70 clients accessing the server in parallel.

The SICMA server system was integrated into a teleteaching application that is already available at Limburg University Center. This teleteaching application is fully integrated into some of the lessons held at the University. The idea of this application is, that some lessons, as well as supporting material for the overall course, are only available in electronic form, so that students have no other choice than using the server to get all material for their exams. The existing system contains a database for searching, a sequential HTTP server that stores conventional material (text, pictures, etc...) and a sequential video server that presents some videos to the students. This sequential video server is proved to be the bottleneck was replaced by the SICMA parallel video server system. The role of the SICMA server is better shown in Figure 1 , that describes the overall chain of the trial. Thus, the SICMA server is fully integrated into a real-world application that is used by students of different faculties at LUC. The current implementation of the teleteaching application is used by students from within the University. For the SICMA trial phase, a number of external student homes are connected to this teleteaching service using ADSL modem technology.

For the SICMA trial phase multimedia material from the Medical Faculty, the Applied Economics Faculty and the Science Faculty consisting of different media types like video, pictures and text are used. The following amount of video information is selected:

- Medical Faculty: about 10 hours of video material
- Applied Economics Faculty: 20 tapes of 0.5 to 1 hour of video material
- Science Faculty: about 3 hours of video material

The groups of students, involved in the second trial are:

- Applied Economics Faculty: More than 100 students were selected to participate in the trial. The course targeted was the language teaching course.
- Medicine Faculty: A group of about 15 to 20 students. The course is a "Capita Selecta" course, in which the professor has "closer control" on the students involved.
- Science Faculty: A group of about 20 students were targeted.
- Some of the students living in the students home were chosen for a residential trial, the other ones have to use computers on the campus.

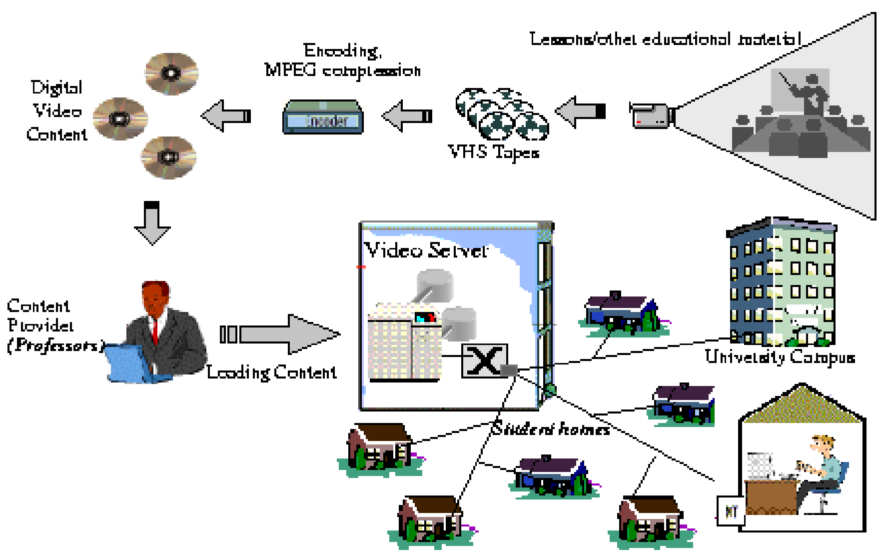

Fig. 1. VoD teleteaching architecture

The SICMA media server is based on the Parsytec CC parallel computer system. This general-purpose distributed memory parallel computer architecture is configured to be used as a media server with a number of disks and external communication devices (ATM cards, Ethernet cards) as presented in Figure 2.

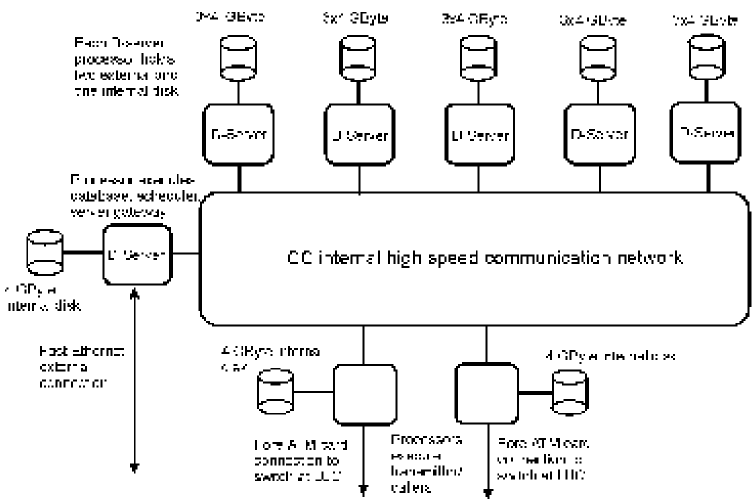

Fig. 2. Server Component architecture

The architecture allows to store about 70 GByte of media content and deliver this via two ATM (155 Mbit/sec) cards to the external communication network. The server can be accessed (for controlling the delivery of media streams) via a Fast Ethernet interface. Figure 3 presents the integration of the SICMA media server based on the CC system in the overall structure of the network that was used for the second trial phase of SICMA. The SICMA server is connected to a Ethernet switch at LUC that leads to the class rooms as well as to a SDH networks that connects to the ADSL central office switches.

3 The SICMA media server

3.1 General

From June to September of 1997, the SICMA server was used in the context of the first SICMA trial, at the Natural History Museum of London [7]. The server was delivering, over an ATM network, multimedia material (videos, images) to two PC clients that were located in a museums exhibition room. The main goal concerning the server, was to test the applicability of the DAVIC protocols (DSM-CC, CORBA IIOP, MPEG2 TS over AAL5 etc..) in an open public

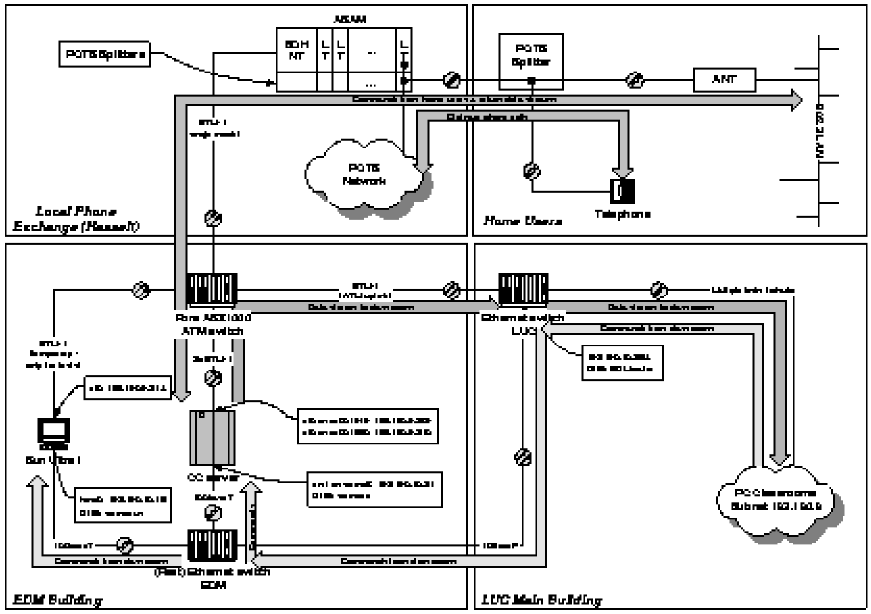

Fig. 3. Trial Network Architecture

trial, as far as there were not any previous DAVIC implementations and public experiments.

The second SICMA trial put requirements on the server system that had to do more with performance than interoperability aspects. The server had to support a large number of concurrent clients (at least 70), manage a big amount of video content (70 GBytes) and prove its ability to scale up and down. Moreover, we decided to develop, integrate and test in this trial, the server-related video streaming protocols that were starting to appear from the Internet community (RTSP, RTP etc..). The server was successfully integrated into the teleteaching application environment and provided a video-on-demand service for the students of the University. In this chapter, we describe the core architectural components of the server, together with results and experiences on critical points, gained from system development and testing in the teleteaching trial.

3.2 System Architecture

The server software architecture is comprised of several modules which enable the provision of interactive multimedia services, in a way compliant to state-of-the art standards (DAVIC, RTSP). Moreover, the design follows a scaleable approach without any performance bottlenecks. In this way, server throughput can be increased by the addition of extra resources (processors, network cards etc..). Figure 4 describes a general view of the core system components.

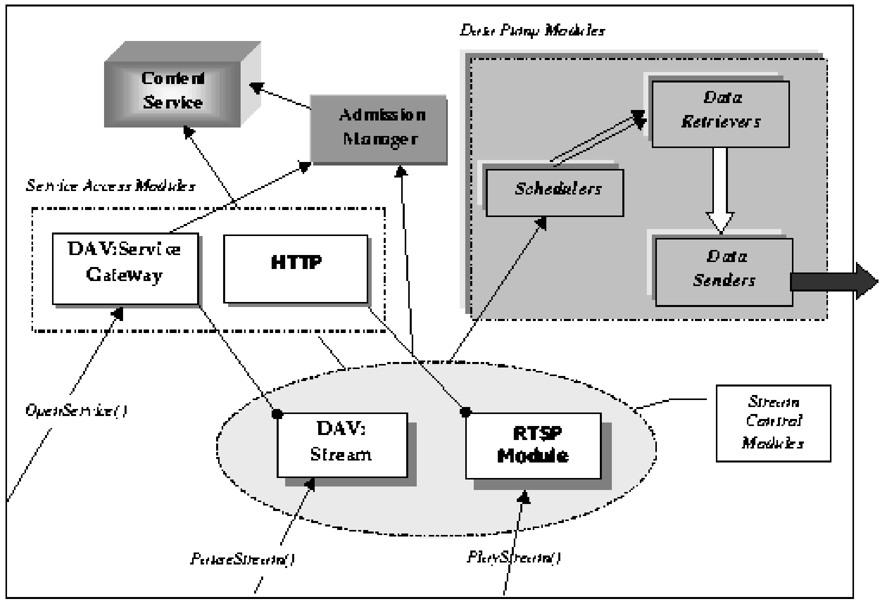

Fig. 4. Server Architecture

The server architecture, shown in the above picture, enables the provision of full video-on-demand functionality: insertion/deletion of video titles, playing/browsing on video streams. The server components can be divided in the following groups:

- The Service Access group that provides functionality for opening stream services and is comprised of the DAVIC Service Gateway element for the DAVIC clients and the HTTP server for the Internet clients.
- The Stream Control modules, that are responsible for controlling the advance of media streams (Stream Element for DAVIC and RTSP for Internet).
- The Content Service that implements the physical data model of the server, maintains descriptors, indexes etc.. It is used for as a storage manager for inserting/deleting video content.
- The Admission Control component that maintains system load and "decides" for the acceptance of new client requests.
- The Data Pump modules (Schedulers, Data Retrievers/Senders) that retrieve multimedia data from disks and send it over the network.

The modules that affect servers interoperability with the external world (clients, networks) are the Service Access modules, the Stream Control modules as well as the Data Senders. These components implement protocols identified in DAVIC and Internet standards and inter-operate with other systems in a predefined way (certain interfaces and protocols stacks). The scalability of the server is achieved mainly from the Data Pump modules that can be scaled up if more processors and networks cards are added to the system.

3.3 Service and Resource Access Modules

After establishing a connection and initiating a session with the server, client applications issue a command for opening a specific service/resource that resides in the server domain. The server finds the resource, opens it and sends the necessary information back to the client for interacting with the resource.

For this role, DAVIC has identified the Service Gateway Service Element that organizes the service domain of the server and enables external clients to discover and select available services [3]. More details about the Service Gateway Element can be found in [7]. One of the requirements of this trial was to support different client platforms, client systems with different capabilities. For this reason, a profile describing client's hardware and software characteristics (e.g protocols stacks, modem bandwidth, presence of MPEG-2 demultiplexers, network interface), is passed to the server, during the opening of a service. The structure of the client profile is described in the DSM-CC standard [5]. The server uses this information to identify if the client is able to use a specific service. The Service Gateway element is unique for the server system, and thus must have a very good performance for handling all client requests. The implementation is based on threads.

For the integration in Internet environments and the use of RTSP/RTP protocols, the Service Gateway module is implemented in form of a HTTP server that offers access to web pages that host the services offered by the media server. Using this open approach, that allows to access media streams stored on the server via URLs (the RTSP standard takes up the URL format of web servers using an own protocol, e.g. rtsp://.... to access media streams) it is possible to integrate non-realtime media data, e.g. standard web information and realtime data in a seamless way.

3.4 Stream Control Modules

The SICMA server provides functionality for manipulating continuous media streams. This is done through the interfaces of DAVIC Stream Service Element and RTSP module that enable VCR-like operations on video streams (stop, pause, fast-forward etc..). The interface of Stream Service Element is defined by the DSM-CC standard [5]. It employs a state machine to keep information about the current status of the stream. It also simulates an internal clock for predicting the position on the timeline of the media stream. The implementation has to be lightweight thread-based with not much CPU overhead- because for each client there exists a separate Stream Service Object. Another important point is the ability of Stream Service objects to discover abnormal client termination, by including a mechanism to check the validity of the command channel connection. In this way a Stream object does not wait for the next request, but when it encounters a broken connection, releases all the reserved resources (memory, threads, bandwidth).

Using the RTSP protocol for manipulating the delivery of continuous media streams from the server to a client, it is possible to control this delivery in different ways. As an example it is possible to give an exact timing when the delivery of a media stream should start. Thus, media delivery can be scheduled according to a play-list. This feature can be used in broadcast delivery of media streams, e.g. for Business TV applications or teleteaching. Another interesting feature is to start the delivery of parts of the streams identified by timing indexes. This allows collecting interesting topics from a larger media stream within one web page where each link points to different parts of the media stream. In this way, interesting indexing mechanisms for media streams can be implemented.

3.5 Content Service

The content service element for the SICMA media server gives information for the internal organization of the media storage and delivery to the other modules of the server. The content service element provides information about the data placement to identify the storage devices where parts of the media stream can be found. In this way, it also provides information for a random access to the media stream that is necessary to implement functionality as described above for controlling the delivery of the media streams. The content service element is also responsible to perform insertion and deletion operation for media streams. Thus, it provides a suitable user interface that allows for the administrator to manage the overall media server content. For the SICMA media server, it is possible to manage the content via a web interface that allows performing a remote content managing.

The implementation of VCR functionality, e.g. Fast-Forward and Fast-Reverse is also managed by the content service element. We implement these functions by storing separate media streams for fast forwarding in both directions. This is the most general way of implementing this function. It also provides best performance results as the visual quality is better using this approach like others delivering only some parts of the stored media stream (I-frames). We have also implemented an index for MPEG-1 System streams for random positioning within the stream.

3.6 Admission Manager

The admission control module is responsible for handling user requests for accessing the media server and starting the delivery of a media stream. Because of the inherent realtime properties of continuous media as video and audio and because of the large requirements in terms of processor and disk performance it is necessary to perform an accurate bookkeeping of resources that are already used by some users for the delivery of media streams. The admission manager needs two kind of information. The first is a detailed performance model of the media server, e.g. bandwidth of the storage devices, external communication interfaces, processors. For a distributed memory parallel computer system that is used in the SICMA project another important factor is the bandwidth of the internal communication network connecting storage devices and external communication interfaces. The second information needed to decide about the acceptance of a user request is the performance that is needed to deliver a specific media stream in terms of disk bandwidth, communication bandwidth and other measures.

Whereas the information about the media server is a function of the concrete machine and maintained by the admission manager, the information about the resources that have to be allocated for the delivery of a stream are hosted by the Content Service.

On request of a user to deliver an identified media stream, this stream is delivered if the available resources are large enough to satisfy the resources necessary for the delivery of this stream. In this way, the admission manager decides about the acceptance of a user request on the basis of the available and requested resources.

3.7 Scheduler

The scheduler is one of the central components of the media server as it is responsible for the actual delivery of media information from the server to the clients. In SICMA we use a scheduling scheme based on earliest deadline first scheduling. Thus, the scheduler maintains a timeline identifying the time for the delivery of a data packet from the server to the client for each of the stream. Whenever this time has expired, the scheduler triggers the data retrievers to send the requested media packet to the data senders which deliver this to the external communication device.

3.8 Data Retrievers / Data Senders

The increased demands of this trial, in terms of high data bandwidth and large number of concurrent clients, lead to redesigning the Delivery System (DS) of the SICMA server. The architecture of the new Delivery System follows the design criteria of the proxy at server architecture that is recommended in the literature for large scale parallel video- on- demand servers [6], [8]. According to this architecture, the DS runs in a multiprocessor environment with a number of delivery nodes that communicate and receive appropriate data blocks, via high speed (server internal) network, from a set of storage nodes/processors (Data Retrievers or DAMS) and route them to the customers. The term proxy refers to the functionality of the delivery nodes to resequence and merge the data units, that receive from multiple storage nodes, into a coherent video stream before it will be delivered to a client. Based on these special buffer management techniques the DS is capable to serve/transmit video streams that have been stripped on the storage nodes of the server. Furthermore, the scalable design of the system enables its expansion to as many as available nodes equipped with network transmission cards. In this way (parallel/ distributed scalable architecture) a grate number of parallel video streams is efficiently supported by the server.

The DS, as part of a DAVIC compliant server like the SICMA-server, followed the DAVIC 1.0 specification [3] and adopted the S1 data flow. DAVIC 1.0 defines the S1 flow as a uni-directional flow from the Server to the clients carrying encoded MPEG-1, MPEG-2 video/audio content using MPEG-2 transport protocol (MPEG-2 TS over ATM/AAL5). The actual implementation of the S1 followed by the DS, considers a generic (abstract) transmission module able to support not only MPEG-2 Transport Stream over ATM/AAL5, but a variety of communication protocols. So far, apart from the mapping of MPEG-2 TS over ATM/AAL5, also UDP/IP and TCP/IP over ATM as well as RTP protocol libraries have been implemented and embodied in the transmission module. In the trial, RTP and UDP/IP protocols were employed for the delivery of video streams from the server to the clients during the trial phase. The software architecture of the DS, that is also referred as Video Pump Subsystem (VPS), is pictured in the following figure and consists of three main submodules named Load Scheduler Module (LSM), Video Pump Generator (VPG) and Video Pump Objects (VPOs).

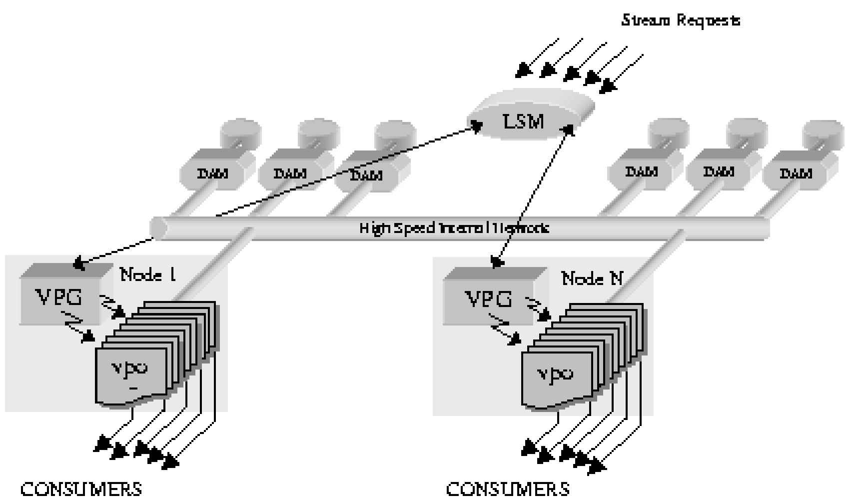

Fig. 5. Delivery System Architecture

The LSM receives requests from the Stream Control modules, to serve the delivery of new video streams and, based on load balancing scheduling techniques, distributes the requests to the less weighted delivery node. The selection of the appropriate delivery node is based on statistics that show the number of video streams served in each node, as well as on their encoding rates. The VPG is located on each delivery node and its responsibility is to fire new VPOs when it is triggered by the LSM to do so. It is also responsible for the estimation of the delivery load in the node that is located as well as to inform the LSM about the status of this load.

The Video Pump Objects constitute the heart of the delivery system since they encapsu- late primary functions like intelligent buffer management, rate control, packetization and delivery operations. For every requested video, the VPG fires one VPO to serve the delivery of the corresponding stream to the consumer. Each VPO consists of 2 threads, the Receiver and the Transmitter which are sharing a common buffer. The Receiver fills the buffers with the stripped data units of the video stream that receives from the Data Retriever (DAM) modules. It is also responsible for merging the stripped data units in order to reconstruct the video blocks (proxy functionality). The Transmitter at the same time reads the full parts of the buffer and transmits the data to the client according to the encoding bit rate of the stream. The synchronization between the Receiver and Transmitter threads of each VPO is coordinated via inter-process communication (semaphores).

4 The Client Application

The client system used in the teleteaching trial at Limburg University is a standard PC (with a minimum of 166 Mhz, Pentium processor). This PC runs one of the standard web browsers (Microsoft Internet explorer or Netscape communicator) and accesses the overall server system via a web interface. The access to standard web information is done us- ing suitable plug-ins that are available for these media types. The audio/video material integrated into the teleteaching application was encoded in MPEG-1 quality using a bitrate of 1.5 Mbit/sec. To decode this media element the Active Movie system available on Windows 95/98/NT is used. To allow an online streaming from the media server to the user client a client software has been developed that establishes a communication lines between client and server.

For the use of the DAVIC compliant server a Corba interface was used here. This is implemented by setting up a permanent process on the client that communicates with the browser on the one hand and the server on the other hand allowing a flexible and elegant Corba interface in this way. To use the RTSP interface, the communication primitives were directly integrated into the plug-in. Both plug-ins allow an easy navigation and VCR like functionality of the media stream.

5 Trial Results and Experiences

The second SICMA trial focussed on the delivery of a complete teleteaching application to a number of about 40 clients installed within the University of Limburg, as well as to about 40 clients installed at student homes and connected to the University network via ADSL. The trial was set up in the last semester of 1998 at the University and supported the lectures in different faculties over this semester. Concerning the innovation of this trial, only very few experiences had been made before in connecting student apartments with ADSL lines to the University backbone. A similar trial is currently held at the University of Muenster, Germany where lecture information is delivered to student homes via ADSL. During this trial phase, a large number of students accessed the server system using the DAVIC compliant interface as well as the RTSP/RTP interface. It was observed that after the trial integration and initial tests the SICMA media server was stable and supported the envisioned application well. System architecture fulfilled interoperability, scalability and performance requirements. The server was able to support approximately 70 concurrent clients, retrieving MPEG-1 video streams at 1.5 Mbits/sec. This throughput limitation was mainly due to CPU overhead on Data Sender processors. Response time between subsequent requests was tolerable. Scalability was achieved by increasing the number of Data Retrievers and Data Senders. One scheduler was able to support 70 concurrent clients, so it was not necessary to fire more schedulers for the trial needs. As it concerns server functionality, the full range of video on demand operations was supported: insert/delete videos, play, stop, pause, fast-forward, fast backward. An important issue is the ability to install video content, at the same time when the server is serving consumer clients. This feature was not supported, so the system was not accessible from students, during content insertion. We are working on this, and especially to a more advanced feature, the ability to store video content in real time, while being able to deliver other videos at the same time.

Trial tests shown that a critical point, concerning server stability, is the proper release of resources (e.g socket connections, buffers), in case when clients terminate abnormally, without closing their sessions. The server has to discover such situations and all modules related to the provision of the service should be informed to release reserved resources. The current integration between DAVIC and Internet protocols in the server can be con- sidered as an initial step towards building a multi-protocol server. We are currently investigating a server architecture that will allow for dynamic switching between different protocol families, in way transparent to the client application.

Since the PC clients of the trial were not equipped nor with MPEG-2 TS demultiplexers, neither with ATM network cards (due to cost reasons), it wasnt possible to investigate the performance of the server under the situation of having to transmit small size packets (376 bytes), as it is specified by DAVIC (mapping of 2 MPEG-2 TS packets to one AAL-5 packet or 8 ATM cells [4]). Recently, FORE released an ATM API for Windows, so we have the chance, together with an MPEG-2 TS demultiplexer card from Optibase to investigate performance issues for the S1 data flow of DAVIC (MPEG-2 TS/AAL5). Another major topic was the absence of an appropriate protocol/system to reserve network resources. This could be done if there existed a DAVIC compliant ATM based- delivery system or the RSVP protocol for IP environments was supported in the trial network infrastructure.

Reactions of students and teachers at Limburg University indicate that an increased support of teaching with audio/visual elements is of large interest. This is because the videos, giving detailed description of certain topics that could not be covered by conventional material, eased the learning and understanding of some topics considerably. But as this depends always on the available material, more attention has to be put on the selection and indexing of material. In current phase, we run a detailed evaluation of the trial. Users are interviewed how the technical system as well as the integration of content supported their learning process.

6 Summary - Future Work

We presented a teleteaching trial, held in Limburg University (Belgium), in the context of ACTS SICMA project. The video on demand needs of the teleteaching application were supported by the SICMA system, which was responsible for managing and delivering MPEG video streams over the local university Intranet, composed of ATM and standard Ethernet, as well as over ADSL lines to student homes. The server supports DAVIC and IETF protocols and follows a scaleable architecture. Results and implementation experiences from the trial were presented. Future work on the server includes full DAVIC and IETF compatibility, real time storage functionality and deployment of a distributed server hierarchy. The main conclusion is that there are not any major technical problems for the provision of interactive distributed multimedia services, in large scale.

References

1. S. Christodoulakis and Peter Triantafillou: Research and Development Issues for Large-Scale Multimedia Information Systems. *ACM Computing Surveys - Special Issue on Multimedia Information Systems*, December 1995.
2. S. Christodoulakis, N. Pappas, G. Kyriakaki, Y. Maragoudakis, Y. Mavraganis, ganis, C. Tsinaraki: The KYDONIA Multimedia Information Server, *Proceedings of Second European Conference on Multimedia Services Applications and Techniques*, ECMAST 97, Milan
3. Digital Audio-Visual Council, Service Provider System Architecture and Interfaces, DAVIC 1.0 Specification Part 03, 1995.
4. Digital Audio-Visual Council, High and Mid Layer Protocols, DAVIC 1.0 Specification Part 07, 1995.
5. Digital Storage Media Command and Control International Standard, ISO/IEC JTC/SC29/WG11 N1300 p1 MPEG 96/. (Tampere).
6. Jack Y.B Lee: Parallel Video Servers: a tutorial, *IEEE Multimedia, April, June 1998, pp 20-28.*
7. C. Brandt, G. Kyriakaki, W. Lamotte, R. Lüling, Y. Maragoudakis, Y. Mavraganis, K. Meyer, N. Pappas: The SICMA Multimedia Server and Virtual Museum Application, *Proceedings of Third European Conference on Multimedia Services Applications and Techniques, ECMAST 98, Berlin.*
8. A.L. Narasimha Reddy, R. Haskin: Video Servers, *in The Communications Handbook*, CRC Press, Oct. 1996.