

Multimedia Supporting Tools for the CEDERJ Distance Learning Initiative applied to the Computer Systems Course

Edmundo de Souza e Silva,

Federal University of Rio de Janeiro,
edmundo@land.ufrj.br

Rosa Maria Meri Leão,

Federal University of Rio de Janeiro,
rosam@land.ufrj.br

Anna D. Santos

Fluminense Federal University, Brazil
anna@dcc.ic.uff.br

Bernardo C. Machado Netto

Fundação CECIERJ, Brazil
bnetto@land.ufrj.br

Jorge Allyson Azevedo

Fundação CECIERJ, Brazil
allyson@land.ufrj.br

Abstract

The state of Rio de Janeiro launched a major program, the CEDERJ consortium of public Universities, to bring high-education to the interior of the state. We begin by introducing the CEDERJ distance learning initiative in the state of Rio and focus on the Computer Science undergraduate program being offered since March 2005. We discuss the goals of the CS program and the main technological issues involved in achieving such goals. The multimedia supporting tools that we developed, and are currently being employed, will be described and its main features and technological challenges highlighted. We conclude by summarizing the experience gathered so far from the Computer Science CEDERJ program.

Keywords: multimedia tools, distance learning, video on demanda, VoIP, videoconference

1 Introduction

In 1999 the state of Rio de Janeiro launched a major effort to bring higher education to the interior of the state of Rio based on the *open and free university* distance learning concepts that started to flourish in Brazil. A consortium of six public universities was formed in the year 2000 (the CEDERJ consortium) and six undergraduate degree programs started being idealized. One of these was the Computer Systems program which began on March 2005.

The previous decade was notorious by the remarkable booming of the Internet and channel speeds growth from kilo bits to giga bits per second. New applications started emerging as a consequence of the research in streaming media and related issues. It was clear that future networked multimedia information systems would carry a wide variety of applications including digital libraries, video, audio and image services, distance learning and collaboration, networked virtual environments, and entertainment.

It was natural then that the Computer Systems program was idealized to make use of the state-of-the-art tools emerging at that time. This was indeed a difficult challenge, since we need to combine research in the area and prototype development and, at the same time, find ways to deal with the technological limitations of the interior of the State and adapt to our local needs.

A few year before the CEDERJ consortium was launched, the LAND computer network Lab at COPPE/Federal University of Rio de Janeiro lead a research initiative with the Federal University of Minas Gerais and three Universities in the USA (UCLA, UMass and Maryland), funded by a joint grant CNPq/NSF, to investigate many issues related to quality of service that arise in the design of the next generation information systems. Among the goals was the development and evaluation of application-level mechanisms for enhancing quality of service delivered by the network and for disseminating efficiently continuous media (CM) data to clients. This includes mechanisms to cope with packet losses and high delay variability imposed by the network.

As a result of the research above, a rough prototype of a multimedia video server was running in our labs, together with an interactive voice transmission tool that incorporated our research results. The research has continued at LAND-COPPE/UFRJ funded by grants from Faperj and CNPq. In the year 2000 much progress have been made to the point that it was conceivable that we could elaborate a Computer Science course based on the multimedia tools we developed, coupled with the distance learning platform of CEDERJ.

The CEDERJ Computer Systems course was being developed jointly with the CS Departments of both Fluminense Federal University and the Federal University of Rio de Janeiro. Both groups were supportive of elaborating a course tailored to computer science students and making use of state of the art tools in Information Technology. The CEDERJ officials believed in the ideas and gave the green light to implement a course which would make extensive use of multimedia material and application tools. The research groups of both Universities have collaborated since to make this enterprise a reality.

In this paper we focus on the multimedia tools being used for the CEDERJ course on Computer Systems and the issues that had to be overcome to implement our ideas. In section 2 we briefly overview some of the research challenges and describe the main requirements for the Computer Systems

course. Section 3 overviews the RIO multimedia server in use at the CEDERJ sites. Section 4 describes the other multimedia tools that are being deployed. In section 5 we present some of the experience we gathered so far and also what we have learned from a parallel research distance learning initiative in Networking with the University of UMass at Amherst. Finally, in section 6 our conclusions and future challenges are presented.

2 Issues

2.1 The overall course requirements

With respect to the CEDERJ Computer Systems (CS) course, several requirements were imperative. First, we need to consider the quality of education. The main asset of the Computer Science Departments at the public Universities involved in the CS program is the high quality of their faculty. So, one driving force towards bringing high quality education to the students in the interior of Rio was to make available to the students the expertise of our faculty, in the form of their lectures, as close as possible to their on going classes delivered at the Universities. On the other hand, the CEDERJ students would have to be able to attend classes at any time according to their availability. These requirements pointed towards preparing lectures to be recorded and streamed to the students as needed. They would then have the opportunity to "attend" the same classes available to our local students, at any time.

Another issue to be considered is *class interactivity*. In a local environment students ask questions and are called to participate in class. If what is available is only a tape of each class, the student would become *passive* spectators, not actors in class. How could we mimic the *interactive* nature of a local lecture was another challenge.

A third issue related to the above was the CS laboratory. The main laboratory for computer science students is the computer itself. Therefore, we envisioned lectures that would make extensive use of computers to encourage, throughout all course, a good match between theory and practice.

These three initial issues pointed to the use of multimedia servers to stream the lectures to students in the CEDERJ sites. Each lecture would contain a pre-recorded video and associated slides synchronized with the video. The lectures would contain a list of topics so that each student could jump to any class topic at any time, independently of the others.

In addition, in order to bring interactivity to the lectures, the multimedia server and associated clients would have to be designed to support interactivity in the "classroom". The slides would be the interface to implement interactivity, that is, they would be programmable to execute tasks launched from a mouse click in the slide area. In other words, the lecturer would have the possibility to ask the student to interact with the slides, for instance to call a browser, to see an animation related to the presentation, to execute computer commands in search of the correct answer. In summary, our solution was to permit that computer commands could be executed from the slides. As an example, the lecturer has the possibility to ask the listener to press buttons in the slide canvas to start, say, a simulator and then guide the student throughout all steps necessary to execute a simulation. The lecturer could also ask questions to the student who would have to click buttons to find the correct answer.

Consequently, a streamed lecture could be as interactive as the lecturer wanted.

Similar to all CEDERJ courses, the CS courses rely on a hierarchy of tutors. One type are the local tutors that support the students locally at each site and, as such, play a very important role in the distance learning process. There are also the remote tutors, who are graduate students located at each university, in charge of replying to individual questions of CEDERJ undergraduate students.

The CS course had an additional challenge: Mostly sure we would not be able to find CS support tutors in the interior of the State, and would have to rely on the remote students. The obvious solution was then to make extensive use of video conferencing, and provide real time video/voice and a whiteboard to the tutors. In addition we would need tools that could adapt to drastically varying network conditions at each CEDERJ site. That is, CEDERJ sites were likely to be connected to high congested low line speeds (few mega bits per second). Therefore, the video conference tools would have to cope with possibly high jitter and packet loss rates and so would have to implement application level algorithms to deliver good quality despite of the varying network conditions.

2.2 The system requirements

As described above, the course requirements demanded for the use of a multimedia video server with several unique attributes and the use of video conference tools for tutoring with application level QoS adaptation algorithms. The special requirements were briefly introduced above. In what follows we outline a few issues at the system level.

The main characteristics of multimedia applications leading to difficulties in end-to-end systems design are the very large bandwidth and storage needs, incorporating vastly different performance and reliability requirements, often coupled with real-time constraints. Thus, one challenging task in designing multimedia end-to-end systems is to satisfy the real-time requirement of continuously transmitting a multimedia object, from the storage subsystem through an unreliable network with unpredictable QoS capabilities. In [6] many of these issues are discussed and solutions highlighted.

Our multimedia server would have to include algorithms to solve the issues in [6] to stream good quality video at each CEDERJ site, to handle a variety of objects (video, slides, photos, text, etc.) and cope with highly interactive users. In addition, it was our goal to use a regular PC as the basic hardware and to run the application software over a free open source operating system (Linux).

We chose the RIO multimedia server prototype we developed in our labs as a result of the joint research projects mentioned in section 1. We also used the *Freemeeting* tool for interactive real time video conferencing coupled with the TGWB distributed whiteboard, all developed in our labs, as a result of cooperative projects. All the tools are freely available.

The tools are under continuing development to include new algorithms and techniques to improve performance and to provide solution according to the course demands. We incorporate theoretical results from our group to lab prototypes. The technology transfer is done continuously.

Examples of novel developments include: admission control and streaming algorithms to cope with jitter [5]; bandwidth sharing algorithms tailored for interactive customers [14,3,16,7]; packet recovery algorithms to cope with losses [8,11]; statistical models for users accessing video servers target to distance learning [18].

In the next section we describe the tools under use for the CEDERJ CS course.

3 The RIO server

The RIO multimedia server is a universal storage system that employs the random allocation technique [17] and uses block replication to achieve high performance and dependability. RIO supports different types of media such as video, audio, text, images. In addition it implements techniques for delivering real time data with stringent time requirements.

The random I/O technique was conceived at UCLA [13] and a basic prototype was initially developed at UCLA to support virtual worlds. Through the joint collaborative project mentioned in section 1 the server was improved and used as a testbed for laboratory experiments with the participant institutions. From 1999 on, the development has continued at Land/COPPE-UFRJ. The server was almost entirely redesigned and two new clients developed: one for visualization allowing synchronization with slides, tailored for distance learning and; another tailored for managing the RIO objects. Furthermore, new algorithms were incorporated to improve performance at the network level, and to efficiently share bandwidth among high interactive clients.

Figure 1(a) shows the RIO basic architecture. Clients request data to the server which in turns dispatches the requests to the storage servers. These access data and deliver the blocks directly to the clients. The server and storage servers can reside in one machine or be completely distributed in different machines. Currently, each CEDERJ site has a complete server/storage-server unit in a single PC.

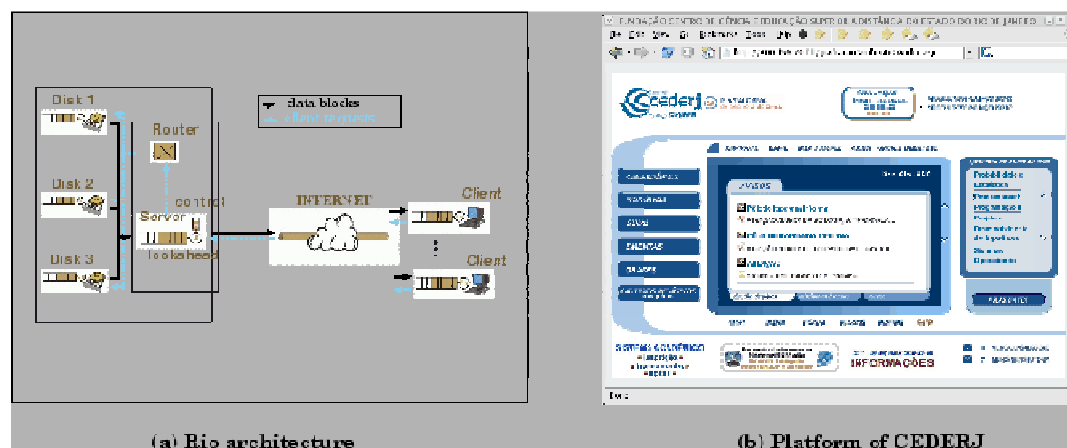


Figure 1: RIO multimedia server architecture and the CEDERJ platform.

The random allocation technique is very efficient to our needs. It does not distinguish among any of the different types of media being stored and the random nature of accessing files facilitates many of the control functions needed, besides providing high performance which results in more clients being

served. A single RIO server in a single regular PC can currently handle around 80 active clients, which is twice the capacity of the laboratories at the current CEDERJ sites.

The clients implement algorithms to cope with high delay jitter. In addition bandwidth sharing algorithms are implemented to make efficient use of the bandwidth to the clients sharing a network channel [12,14,3]. We have also implemented routines that collect useful statistics. These include measures of the behavior of clients and performance measured of the system. The first are extremely useful for developing models of the students accessing a distance learning class, such as the distribution of time in play mode, distribution of the slides accessed, the probability of accessing a giving slide in a class, etc. With these we will be able not only to create models to improve the performance of the RIO server [18] but also to have a better understanding of the behavior of students while accessing the class material.

The main interface of the platform of CEDERJ is illustrated on the Figure 1(b). From this interface the user is able to choose one of the CS courses. When the user clicks on the link of the selected course, a screen with all its classes is displayed. (One example of this screen is presented on the Figure 2(a).) After selecting one of the classes, the video interface of the RIO server pops up. Figure 2(b) shows an example of this interface. The figure displays the video, the slides and the index that allows the user to access a specific topic.

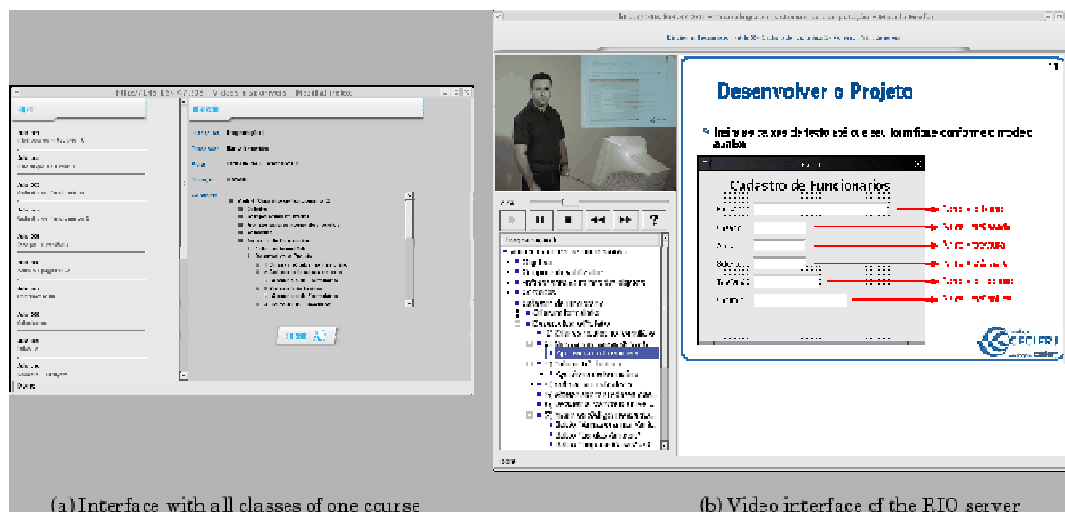


Figure 2: RIO multimedia server interfaces.

The client-object manager can execute most functions of a file manager in graphic or text mode. These functions include copying, moving, deleting, renaming objects and directories, and the editing of text objects [15]. Several servers can be managed in graphic mode. In addition to the client-object manager, another module is being developed to provide, *in real time*, deletion and insertion of stored objects, without halting the operation of the server [4].

To facilitate the access to the class material stored in the RIO server, a boot DVDROM was created. These are intended to students: who cannot visit the closest CEDERJ site in a regular basis and; have no broadband access to the Internet but; have a PC with DVD reader. The DVD contains the basic Linux OS with the RIO client/server software and a few (usually three) classes stored. In addition, all necessary software needed by the stored classes are included in

the DVDROM. Any student with a computer has access to the site environment without being necessary to resort to complex software installations. Booting the PC with the DVDROM is what is required.

4 Freemeeting

Freemeeting is a cooperative environment developed by our group that integrates interactive video/voice tools (VivaVoz and ComitVideo) and a distributed whiteboard and chat (TGWB). In fact, Freemeeting is an interface that manages all those individual communication tools which were independently developed by us since 1997.

The first tool developed was VivaVoz [10] to transmit voice over IP. It implements an efficient packet loss recovery algorithm [9]. The tool evolved and new algorithms to improve the quality of the received voice were added. Based on statistical models, the tool predicts future error rates and, from that, adapts to improve the voice quality [8]. This is particular relevant to environments where error rates vary from a few percent to say 25-30%. Large error rates are not uncommon in our Universities environment. The problem is even worse in the interior of the State. Some commercial tools fail in such lossy environments.

Different speech codecs are available such as GSM and SPEEX (these are free open codecs developed elsewhere.) Combined with our packet recovery algorithms, the tool can transmit data at speeds from 2.4 to 128 Kbps, according to the desired quality and channel conditions. The voice tool has also a modulo that allows full duplex conference calls (N to N communication).

VivaVoz has embedded many options for collecting statistics. These are extremely useful for research and for analyzing the quality of a conversation and for study the performance of the implemented algorithms.

Commit video can capture, transmit and receive video at different rates and formats. It uses the freely available ffmpeg program for formatting and MPlayer to display the video. A full range of data rates are available to the user and N to N communication is possible.

Both the video and voice tools were largely used since 2002 in a distance learning effort between the LAND/COPPE-UFRJ group and the University of Massachusetts, Amherst networking group. Several joint courses and seminars were developed. This helped to shape the tool and to improve its robustness.

The TGWB (Tangram whiteboard) is a distributed whiteboard that can be shared by all users. That is, any user can alter simultaneously any area in the canvas. The whiteboard is implemented on the top of TGIF (Tangram Graphic Interface facility) [2] which is an open 2D vector-oriented drawing tool. TGWB implements several routines to guarantee data consistency. A reliable multicast library [1] is included to allow N to N communication. Since most networks do not implement IP multicast, a modulo (mcastproxy) was developed to receive data from multicast groups (where multicast is available, such as local networks) and re-transmit data via unicast channels to a list of given addresses [1].

Figure 3(a) shows a session with Freemeeting. Similar to all the tools we develop in the lab, it is an open source tool and it is currently in use in the tutorial sessions for the CS courses, and integrated with the CEDERJ platform.

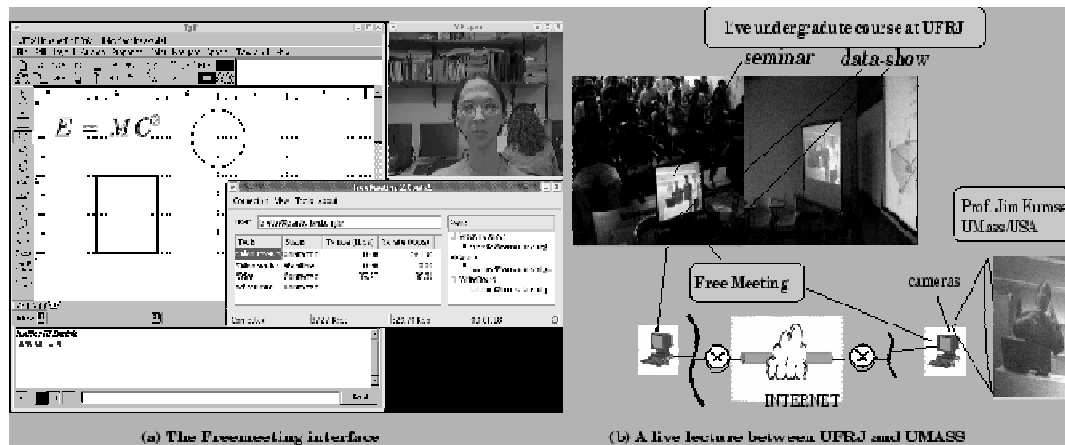


Figure 3: The Freemeeting interface and a lecture between UFRJ and UMass.

As mentioned before, Freemeeting is extremely important to allow local graduate students to fill the tutor gap due to the lack of human resources in computer science in the interior of the state. The tool is adapted to our local conditions and can evolve to respond to the needs of our course. We expect that the professors involved in the course soon will be able to use the tool for seminar sessions transmitted to the interior improving the interaction between the participant universities and the satellite sites.

5 Experience

In the above sections we mention that we have developed the tools over many years supported by cooperative projects. In particular UCLA and UMass/Amherst were major partners abroad. In Brazil, UFMG has been our partner for continuous media streaming research and UFF has been involved in the research activities related to CEDERJ. In this section we comment on some of the experience we gathered over the years.

Since 2002 we have been developing with UMass distance learning activities¹. We elaborated several joint courses and enrolled in those senior level undergraduate and graduate students from UFRJ. Some of the courses were regular UMass courses which were broadcast live to our laboratory from the USA, and completely interactive. Students at UFRJ actively participated in the class that used our voice and video tools. Brazilian students did all exams and homework assignments as the USA colleagues and were evaluated as US students. In Brazil, we encouraged discussions among local students on the class material. We also stored the lectures in an earlier version of the RIO server for consulting. All students obtained excellent evaluations. Furthermore, one of the UMass professor gives each semester a lecture as part of a regular undergraduate Computer Network class at UFRJ. Figure 3(b) shows one of these lectures, using our tools.

Indeed, this environment is very similar to what we envision for the CEDERJ CS course, except that the CEDERJ students do not attend live classes (but could attend live seminars). However, the remote interaction with the tutors, the access to recorded classes, and the evaluation process is similar.

Other classes UMass/UFRJ were organized in a series of seminars which were presented both by USA and Brazilian students. Even international

student panels were organized. These all worked very well and shows that the approach we are implementing for CEDERJ can work, including the supporting tools. For instance, CEDERJ students at each site could interact with other students in other sites and attend live lectures delivered by UFF and UFRJ faculty.

The CEDERJ CS course started in March 2005. Since then the students have been using the RIO server and the boot DVDs. The feedback we have had so far indicate that the students enjoy the classes and the environment prepared. The video conference environment has run experimentally during the second semester of 2005. We had evidence that for one class with no site tutor, as soon as the video conferencing started with a remote tutor, the grades improved. As the tools become widely used, we expect to make the quality of education better and bring the CEDERJ students as close to our universities as possible. With the video conferencing tools we hope to increase the participation of students at the CEDERJ sites which is essential for the learning process.

6 Conclusions

In this paper we briefly described some of the major technological issues involved in preparing the CS CEDERJ undergraduate courses. Most of the problems we have encountered so far are identical to those found in our regular *presential courses*, such as: insufficient time to study (most of the students have a full time job), low attendance in office hours and poor understanding of the basic concepts due to low quality education at the high school level. Additional problems include the lack of trained personal to maintain local labs. However, as the years go by, we expect that CEDERJ will drastically impact positively the quality of education in all levels. As connectivity improves we will be able to have more people trained in the interior of the State and so create a much needed local critical mass in computer systems.

The solutions we provide for the CS course is the result of research done at the Universities involved. In fact, this is a good example of technology transfer from the research lab to a full fledge initiative, impacting education. It is also an excellent example of a successful long term cooperation among universities in Brazil (UFRJ, UFF, UFMG) and abroad (UCLA and UMass).

The tools are in constant development. We plan to continue the research and development as the demand for new applications and features increase. This is a major advantage of using open source code developed at the university. Examples of future developments are the improvements of RIO for full distributed large scale operation (GIGA RNP-Finep project) and application of this technology for continuing education in medicine using wireless channels.

The classes developed for the CEDERJ Computer Systems course are available for free access at the participant Universities. Local students will benefit from this since they will be able to complement the material taught in their local classes. We hope that CEDERJ is a major seed to the Brazilian *OpenCourseware* initiatives that is bringing open free education to anyone and, as a consequence, help to close the so called digital gap.

Bibliography

- 1 Jorge Allyson Azevedo. Transmissão multicast confiável: Aplicação na ferramenta tangram whiteboard e experimentos na internet. Master's thesis, UFRJ/COPPE-PESC, Dezembro 2002.
- 2 W. C. Cheng. <http://bourbon.usc.edu/tgif/>. TANGRAM Graphic Interface Facility.
- 3 C. K. da S. Rodrigues and Rosa M. M. Leão. Novas técnicas de compartilhamento de banda para servidores de vídeo sob demanda com interatividade. In *XXIII Simpósio Brasileiro de Redes de Computadores (SBRC2005)*, Brasil, Maio 2005.
- 4 Sandoval da Silva Gonçalves. Reorganização e de Objetos multimídia em Tempo Real quando da Remoção e Inserção de Unidades de Armazenamento do Servidor RIO. Tese de mestrado, IC/UFF, September, 2006.
- 5 Adriane de Quevedo Cardozo. Mecanismos para garantir qualidade de serviço de aplicações de vídeo sob demanda. Master's thesis, UFRJ/COPPE-PESC, 2002.
- 6 E. de Souza e Silva, R. M. M. Leão, B. Ribeiro-Neto, and S. Campos. Performance issues of multimedia applications. *Lecture Notes in Computer Science*, 2459:374-404, July 2002.
- 7 Edmundo de Souza e Silva, Rosa M.M. Leão, and Morganna C. Diniz. The required capacity distribution for the patching bandwidth sharing technique. In *IEEE Internat. Conference on Communications (ICC 2004)*, pages 2179 - 2183, 2004.
- 8 Flávio P. Duarte, Edmundo de Souza e Silva, and Don Towsley. An adaptive fec algorithm using hidden markov chains. *Performance Evaluation Review*, 31(2):11-13, 2003.
- 9 D.R. Figueiredo and E. de Souza e Silva. Efficient Mechanisms for Recovering Voice Packets in the Internet. In *Proceedings of IEEE/Globecom'99, Global Internet: Application and Technology Symposium*, pages 1830-1837, December 1999.
- 10 D.R. Figueiredo, F.P. Duarte, and E. de Souza e Silva. Implementação de um aplicativo de voz com análise e caracterização de seu tráfego. In *SBC - XXIV SEMISH, XVI CTIC*, pages 531-540, 1997.
- 11 Fernando Silveira Filho and Edmundo de Souza e Silva. Um método de previsão de perdas de pacotes com aplicações à transmissão de mídia contínua. In *XXIV Simpósio Brasileiro de Redes de Computadores (SBRC2006)*, 2006.
- 12 M. L. Gorza. Uma técnica de compartilhamento de recursos para transmissão de vídeo com alta interatividade e experimentos. Tese de Mestrado, UFF, DCC, 2003.
- 13 R.R. Muntz, J.R. Santos, and S. Berson. A parallel disk storage system for real-time multimedia applications. *International Journal of Intelligent Systems, Special Issue on Multimedia Computing Systems*, 13(12):1137-1174, December 1998.
- 14 B. C. M. Netto. Patching Interativo: um novo método de compartilhamento de recursos para transmissão de vídeo com alta interatividade. Tese de mestrado, UFRJ, COPPE/PESC, February 2004.
- 15 Bernardo Calil Machado Netto, Jorge Allyson Azevedo, Edmundo A. de Souza e Silva, and Rosa M. M. Leão. Servidor Multimídia RIO em Ensino

- a Distância, 2005. Anais do  Fórum Internacional de Software Livre, páginas 91-95.
- 16 Carlo Rodrigues and Rosa M. M. Leão. Técnicas para Sistemas de Vídeo sob Demanda Escaláveis. In *XXIV Simpósio Brasileiro de Redes de Computadores*, 2006.
 - 17 J.R. Santos. *RIO: A Universal Multimedia Storage System Based on Random Data Allocation and Block Replication*. PhD thesis, Univ. California, Los Angeles, 1998.
 - 18 D. Tomimura, R. M. M. Leão, E. de Souza e Silva, and F. Silveira Filho. Caracterização do Comportamento de Usuários Acessando um Vídeo de Ensino à Distância. In *SBC2006 - V Wperformance*, 2006.

¹ Much earlier, around 1996, we run an experimental class with USC.