Multimedia Virtual Classroom [1]

José Luis González Sánchez, Alfonso Gazo Cervero, Antonio Plaza Miguel,
Alberto Gómez Mancha, Marisol Sánchez Alonso

jlgs@unex.es, agazo@policc.unex.es, aplaza@policc.unex.es,
agomez@unex.es, marisol@unex.es

Computer Science Department
Universidad de Extremadura, Escuela Politécnica de Cáceres
Avda. de la Universidad s/n, 10071 Cáceres, España

Abstract

Virtual Classroom Project proposes new techniques to improve teaching quality at Technical School in Cáceres, offering remote services to all university community. All School computing resources and Internet services are available through remote connections. The service successful achieved justifies the development of audio and videoconference systems to extend Virtual Classroom. This multimedia add-on would allow geographically sparse receivers to attend classes and conferences.

Our system can be used in two different ways: students could inspect recorded lectures and teaching material at home; furthermore, real-time audio and videoconferences are broadcasted to our students. The difficulties founded in our system implementation are widely discussed along this paper. This service permits us to study the information flow and to propose new communication protocols with QoS in IP and ATM networks.

Keywords: *Linux*, teleworking, freenet, remote connection, bandwidth, communication protocols, *CTN (Conmuted Telephonic Network)*, *ATM*, multimedia information, *multicast*, *QoS.*

Related subjects: Computer networks, multimedia, distributed systems.

---

[1] This research has been supported by the Education and Youth Council of Junta de Extremadura (Spain) under expedient number MCD97D024.

1. Introduction

Virtual Classroom [1] has been the first attempt to provide university community teleworking and to alleviate the problem of collapsed classrooms in peak hours and empty ones in non school hours (evening hours, weekends and holidays). Afterwards we have seen the necessity to access multimedia resources, so we have developed system extensions giving rise to *Multimedia Virtual Classroom.* This service is running and operative at the moment and is presented as a distributed system intended to support remote connections to multi-user operating systems (*Unix*, *OpenVMS*, *Linux*, *Netware*) that make advanced telematic services accessible to their users.

This document firstly presents the system topology and design decisions in its elaboration. Next offered multimedia services are presented, followed by the technical considerations that these multimedia services imply and the *multicast* extensions that we have added to the initial *Virtual Classroom* project. After that, we present the statistics of the service use during the first months, and the current works in progress using information flows that *Multimedia Virtual Classroom* service generates. We finish the paper with our conclusions.

2. System Topology and Design Decisions

In this section the *Multimedia Virtual Classroom* service topology is presented, with all the services that are provided.

In *Figure 1* the four main blocks are identified. *Block A* corresponds to a room with 30 *Linux*/*Windows 95* systems with an outside connection through a *SPARC* station that acts as *router*. *Block B* is the one that performs the *remote access service* from *Multimedia Virtual Classroom*. *Block C* is connected to the *Ethernet* buses that are located in the campus where the *Multimedia Virtual Classroom* resides. There are lots of PCs connected to this *Ethernet* buses; those computers are usually situated on teaching staff offices. Finally, *Block D* represents a PC network under Windows 3.11 which is mainly used to make practical assignments that do not have high demands of hardware. This network has a *NetWare* server that acts as file server and *IPX router*. Moreover, there is a TCP/IP *bridge* that joins this network with one of the campus *Ethernet* buses.

Many decisions behind the *Multimedia Virtual Classroom* service design were taking mainly depending on two factors: firstly, we intend to give a *distributed* character to our system, trying a fault in one system not affects the global *Multimedia Virtual Classroom* system operation. In the second place, a major factor that limits the complete project, cost, and a high exploitation of existing infrastructure. We tried to reuse all existing computing resources, as well as to reduce new equipment costs.

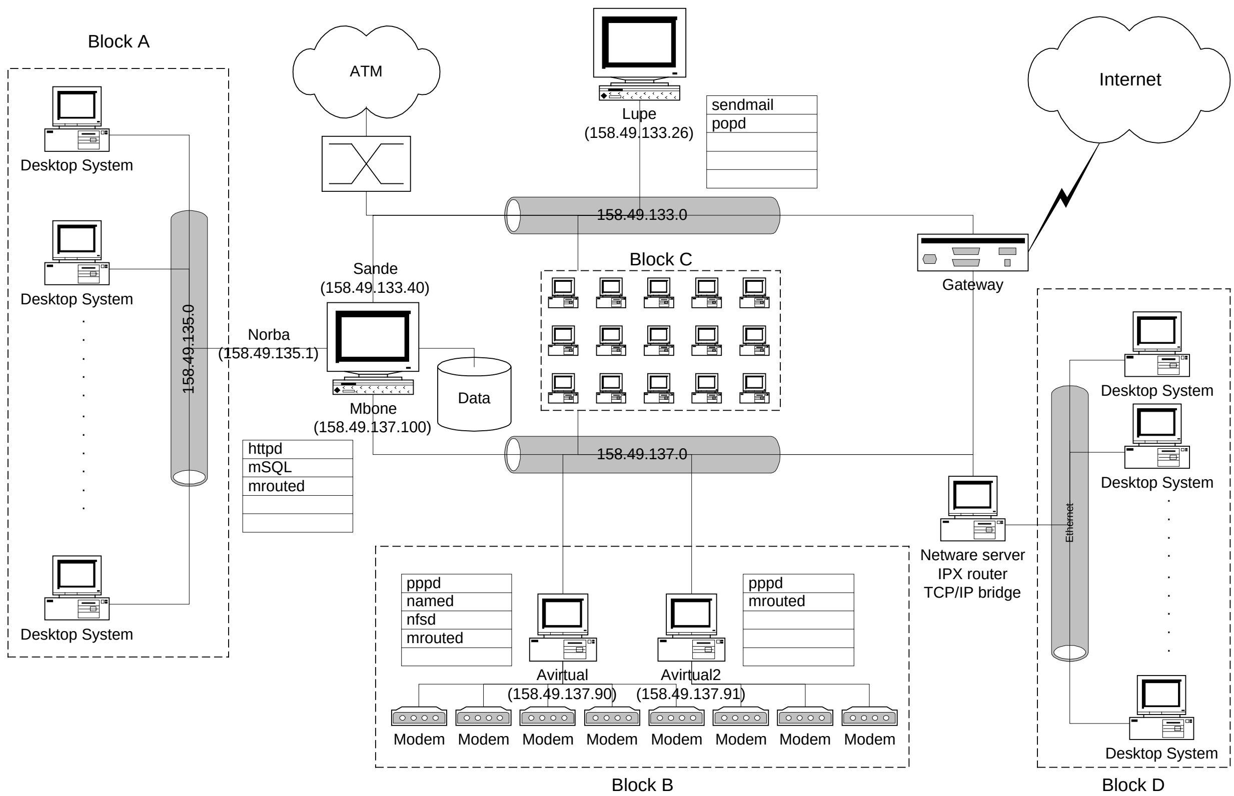

Figure 1. ***Multimedia Virtual Classroom*** **System Topology**

Next we describe the basic services that the *Multimedia Virtual Classroom* offers.

2.1. Electronic mail

The electronic mail service weaknesses recommended isolating the electronic mail service in a dedicated machine. Besides, if the electronic mail server breaks down, this fault would have a little impact if the machine were devoted to this service.

The electronic mail service was installed in a *SPARC* station under *SunOS 4.1.3*, and its first task was to substitute immediately *sendmail* with the last available version. A *POP daemon* [2] was compiled and installed, which, nowadays, do the users that enter the systems via modem use the most. It is also planned to install an *IMAP server* [3], because this protocol allows users to store all their mail at the server, even with a folder organisation in their own *mailbox*. In this way, mail can be consulted from any machine, without transferring it, as the *news* service works.

2.2. Web service

When we planned the necessity of an information system for the *Multimedia Virtual Classroom* service, we immediately think on a web-based system. We estimated a high number of accesses to this system (at the moment there are more than 20000 accesses per month), that implies a workload that should be managed by a machine with a high computation capacity. So we thought in a *SPARC System 10* that used to work only as a gateway between different subnetworks of the University. This way, we also get a direct access to the information system from any of the subnetworks, reducing network traffic.

It was installed an *HTTP server* and a database server to gather references to stored information [4] in the information system.

In the event of a fault in this system, in the same way as the information system will be affected, routing among the subnetworks will too, both *unicast* and *multicast* mode [5]. Failure probability should be minimum, not just because of its participation in the *Multimedia Virtual Classroom* service but because it is an important part of the University communication system.

2.3. Remote Access Service

When we designed the remote access service, our first idea was to have a system composed by a group of machines, so a failure in one of them would not imply the global failure of the remote access service.

Because the system will have a number of *CTN* lines, we have to explain the way these lines work. When there are a group of telephone lines associated with the same number in a system, the main number is set up in the so-called *jumping device*. When a call is received via the main number, this device will try to connect with the first line. If it is occupied, it will pass to the second one, trying all the free lines, as in *Figure 2*.

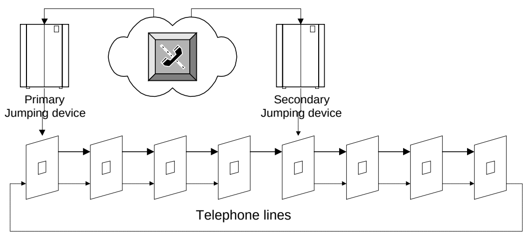

Figure 2. Jumping device

Let 's suppose that the machine where the modem installed in the first line is connected fails. If this machine fails, the first modem will not answer the call, and as the line is not occupied, the jumping device will not act. So we decided to install two *jumping devices*. The first one, the main one, begins with the first modem. The secondary number begins with N/2-th modem, being this modem controlled by a different machine from the first modem. If a user tries to connect using the main number and the line is not picked up, he could try with the secondary number.

Once we have the *CTN* lines, the remote connection protocol that is going to be used has to be decided. When we talk about Internet, there are two basic remote connection protocols:

- *SLIP* [6]
- *PPP* [7]

We decide on implantation of a remote access service based on *PPP*, because this protocol allows *IP* address negotiation in a dynamic way, and a wider range of compression protocols that are negotiated between the two ends.

An undeniable advantage of the newest operating systems is the incorporation of *PPP* as a part of their basic distributions, but, sometimes, manufacturers slightly modify the protocol to adapt it to their operating system features [8]. So we decided to adopt an operating system that provide facilities for future changes and protocol extensions.

Furthermore, as long as the remote access service will also supply *multicasting-based services* [5], a *UN*X* system seemed the best choice. We took these two factors and the implantation cost into account, and finally we decided to use *Linux*. Two personal computers with serial multiport cards were used.

A series of automatic connection scripts to the more used operating systems were written to make easy the access in *PPP* mode. In this way, a user that wants to connect via *PPP* will not have to select the suitable options from menu.

Present *CTN* technology permits to use a bandwidth up to 33600bps. However, to optimise the available bandwidth through the *CTN* lines, 33600bps modems with V.42 compression support where installed. In this way we have a hardware level compression that will improve transmission rate, via *PPP* or through a terminal emulation.

Although we had a hardware level compression scheme, we also implemented some software level compression schemes, supported by the *PPP* protocol, so they can't be applied to the terminal emulation sessions. These schemes are:

- Van Jacobson compression [9]
- BSD compression [10]
- Deflate compression [11]

Van Jacobson compression is only applied to the *TCP/IP* headings, and not globally to the packages contents sent through the *PPP* line. In this way it is possible to use it with an information compression scheme, *V.42*, *BSD* or *Deflate*. *BSD* compression as long as *V.42* compression is based on *LZW* techniques. Difference between *V.42* and *BSD* is that *V.42* uses a code size of 10, 11, or 12 bits, whereas *BSD* can use a size up to 15 bits, with a better compression ratio. But d*eflate* compression uses the same compression algorithm that *gzip*, so it has better compression ratios that previous schemes.

Tests done since service start point to the software compression activation if the operating systems allow it, disabling the modem hardware compression that usually is slowest and offers lowest compression ratios.

Continuing with the optimisation of the remote access service, new techniques to minimise *DNS* resolution latency [12][13] were considered. A *DNS* server is implanted in one of the machines that take part in the access remote service. This machine is set up to the *DNS* server is a secondary server of the zone the University systems belong, as long as a cache server [14]. Cache service stores *DNS* requests made to the server. When a *DNS* request is done to the server and its name do not pertains to the University area, server consults its internal tables to check if it has this entrance. If it has the entrance, it checks if it has got out of date, and if it is still active, it returns the request. In that way, it has not been necessary to send any package through the *Ethernet* network to solve a request. We determine if an entrance has got out of date checking *SOA* record that every *DNS* area has.

In their *resolv.conf* files, rest of the machines that take part of the remote access service will have the *IP* address of the *Multimedia Virtual Classroom* machine where the *DNS* server is running. Furthermore, they will have the IPs from the rest of *DNS* servers from the University. In that way, the disablement of the names server will avoid if the *DNS* server in the *Multimedia Virtual Classroom* context fails.

DNS server has been installed in only one machine instead of in all offering remote access service machines by the following reasons: i) Since all access remote machines are directly connected to a *Ethernet* 10BaseT network the request response time and network traffic impact are low enough; ii) Server configuration in cache mode allows us to reuse previous request resolution. Nevertheless, if there were different *DNS* daemons, the same request over different daemons could be sent, which implies adittional resolution time.

3. Multimedia Services offered by *Multimedia Virtual Classroom*

From the underlying infrastructure consisting of a remote access system and an information server through the web, the development of a system to manage teaching activities is justified, but our attempt will be to offer a proper multimedia service to the University community.

Some of the tools provided to teachers are:

- Real-time transmission of theoretical and practical lessons (audio and video) to those students, which are physically away from the classroom.
- Additional teaching support activities are provided, such as the shared desktop service. It can be used both by remote users and/or in the classroom computers in which practical lessons are imparted.
- Record of sessions for future review.
- Advertise sessions and reference to stored material.

Before implanting these services as a part of *Multimedia Virtual Classroom*, a typical situation is described: if it is desired to transmit a lesson with real-time audio and video, i.e. a camera and a microphone for the teacher to represent the sender and the students, which are attending to a practical lesson through their computers, to represent the receivers, as it is showed in Figure 3.

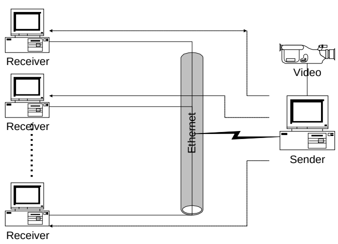

Figure 3. Real-Time Multimedia Transmission Diagram

Then, the number of connections established by the sender must be the number of students. The sender must be both collecting audio and video information and encoding it and sending independently that information to every one. Since all receivers are connected to the same *Ethernet* bus, and taking into account that the *Ethernet* buses provided by our University are based on 10Base-T technology, the total bandwidth required by 20 students will be superior to our available bandwidth. So, if we tried to offer the service using this scheme, the network would collapse.

One solution could be the limitation of the transmission bandwidth or the maximum number of receivers. Nevertheless, the next section proposes an approximation to eliminate the mentioned limitations.

4. Technical Remarks

The restrictions mentioned above are decisive, so the scheme is changed for transmission and reception we change the scheme for transmission and reception. The previous scheme was based on an *unicast* addressing, and now different kinds of addressing in IPv4 are presented.

4.1. Addressing

IPv4 allows us three kinds of addressing:

- *Unicast*: A package transmission to a specific machine. It can be seen like simplex mode transmission, where a node sends information (identifying the address of the receiving node) and another node receives it. Since most of the high-level network protocols (*ISO Transport*, *TCP* or *UDP*) only provide *unicast* transport service, which only allows sending packages from a node to another simultaneously, this is the most usual kind of addressing in the Net. If a point-to-multi-point communication with *unicast* transport service is desired, the solution is replicated mode, which generates and sends a copy of every package to each of the receiving nodes.
- *Broadcast*: Transmission of a datagram to every single node of a sub-network. A node sends information and the information is received by every node in its sub-network. It is the most commonly used mode of transmission in LAN´s such as *Ethernet*.
- *Multicast*: The previously mentioned replicated *unicast* mode is not an efficient solution. The best way to transmit information from a sender to various receivers is to use a *multicast* transport to commit the operation with a unique call to the transport service. The *multicast* mode is logically situated between *unicast* and *broadcast*; it allows *multicast* packages to be sent to different nodes which have decided to be members of a *multicast* group, even if they are situated in different sparse sub-networks. *Multicasting* is the capability of sending information to different nodes through a unique atomic operation, i.e., to every node simultaneously and not one by one, what will require a different operation to every receiving node. *Multicast* minimises the network traffic by transmitting every package only to a *unicast* group address. Each package will be re-sent through the distribution tree to the receivers. The datagrams are only sent to those machines waiting for them. Besides, the relative *multicast* efficiency in relationship with *multicast* and *broadcast* is incremented as the number of receivers and their dispersion increases. Finally, another key point in favour of *multicasting* is the fact that some networks such as *Ethernet* support both *unicast* and *broadcast* in the physical layer. A meaningful exploitation of the local network capabilities is achieved by implementing *multicast* servers.

4.2. Multicast Groups

Multicast addressing adjusts precisely to the goals we try to achieve: each *multicast* group has its equivalence in a class containing at least a sender and a receiver. Just as in a university campus different lessons can be imparted simultaneously, there can exist different *unicast* groups at the same time.

A client can join or leave a *multicast* group as well as a class. This operations are committed using the *IGMP* protocol [15][16], used by *multicast* routers (*mrouters*) to determine the branches of the distribution tree to which they have to re-send the *multicast* packages. If there is no client attached to any *multicast* group in a branch of the tree, *multicast* packages must not be sent through that branch. If there are clients attached to a *multicast* group, *mrouter* will only re-sent the *multicast* packages correspondent to the group the clients of this branch are attached to; the *multicast* packages from other *multicast* groups will not be re-sent through the branch because no receiver is waiting for them.

4.3. MBone

In the previous points we have described the situation of a network in which every router is a *mrouter*, but this ideal situation does not correspond with reality. To commit *multicast* transmissions between two networks joined by a *unicast* router, the upcoming solution is to transform the router in a *mrouter*, but this is not always possible. The problem gets worse if we want to transmit *multicast* information between two geographically distant networks, with lots of *unicast* routers.

The solution to the problem is the creation of tunnels between *mrouters* on *unicast* networks. This is the basis of *Mbone*, which can be defined as a *multicast* network integrated within Internet that uses tunnels to connect the *mrouters* not directly visible to *multicast*. This way we can achieve a *multicast* transmission scheme extended through different continents.

5. *Multicast* Extensions for *Multimedia Virtual Classroom* Service

As it has been described before, the original goal was to implant a system based in *multicast* communication will significantly reduce the negative effect produced by the network traffic increase due to the transmission of the multimedia resources. By reducing the network traffic:

- The impact on the communications of the campus backbone will be reduced.
- Multimedia transmissions quality within the context of the *Multimedia Virtual Classroom* service (*QoS* [17]) will be highly improved.

5.1. Multicast Routing

As it is showed in figure 1, most of the users of *Multimedia Virtual Classroom* service will use either one of the three *Ethernet* buses or a *PPP* link through the remote access service. Our principal interest is to define an efficient *multicast* routing scheme among the three *Ethernet* networks without defining any tunnel.

Once this subject is solved, we can appreciate how *multicast* packages will not go through *PPP* lines, as *IGMP* requests will not be able to reach the *Ethernet* bus because of the absence of a *multicast* routing scheme in the machines that offer the remote access service. We solve this problem by executing a *mrouted* command in every machine. Nevertheless we find a new problem: *mrouted* can only define *multicast* routes (referred as virtual interfaces in *mrouted* documentation) on those network interfaces active at the time *mrouted* was executed. The *PPP* interfaces are only active when the connection is already established. So, if we try to execute *mrouted* by including it in the start-up of the system, it will not perform correctly as it only finds one network interface (the interface to the *Ethernet* bus), and decides there are not enough interfaces (distribution branches) to perform the routing, aborting its own execution.

The only solution to the problem is to execute *mrouted* only when there is a *PPP* connection established. Besides, *mrouted* does not support the automatic addition of network interfaces, so a refresh signal must be sent to the daemon every time a *PPP* connection establishes or breaks. Scripts that will be invoked by the daemon dealing with *PPP* connections implement this: *pppd*.

5.2. Multimedia Services

Once *multicast* capabilities are available, applications oriented to send and receive audio (*RealAudio*, *RAT*, *VAT*) and video (*RealVideo*, *VIC*, *IVC*) in *multicast* mode can be used. These tools are available in the system of information of the service. We remark that the transmissions in deferred time are committed in *unicast* mode. To transmit a deferred time transmission in *multicast* mode, all receivers should send reception request simultaneously.

6. System Use Statistics

Figure 4 shows the increase in the exploitation of the system since its implantation. It is important to emphasise that this stage began in January 1998. Since then, there has been an appreciable increment both in the number of remote connections and the elapsed time of the system. At the moment, until the system is satisfactorily tested, we have established a limited number of users.

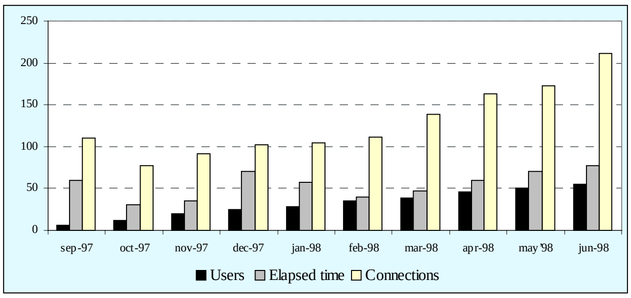

Figure 4. Use statistics

7. Future Work

Our future work will include the next research fields: i) when a local *ATM* network will be installed in our University, implement the scheme showed in figure 1 in order to provide *Virtual Classroom* with *QoS* (throughput, delay, jitter), required for the multimedia transmissions. ii) Design a new protocol oriented to control *IP* traffic, which must be able to conjugate *QoS* with reliability. This protocol will guarantee the reception of multimedia information by different users of a *multicast* group simultaneously.

8. Conclusions

We have presented the *Multimedia Virtual Classroom* service as a low cost and distributed system. Our main aims are: i) improve the use of internal network resources and ii) offer advanced multimedia services in our University context, which are perfectly extendable to other environments. Some of the difficulties presented by our service have been solved using audio conference, videoconference and real-time shared desktops (*Mbone*) and also with the recording of sessions to be received in deferred and emitted through *unicast*.

9. References

[1] A. Plaza, C. Ramos, A. Gazo, J.L. González. "*Servicio Aula Virtual de la Escuela Politécnica de Cáceres*". IV Jornadas de informática de Las Palmas de Gran Canaria

[2] *Post Office Protocol*. RFC 1460

[3] *Interactive Mail Access Protocol.* RFC 1203

[4] *http://webepcc.unex.es:* Páginas de información del servicio *Aula Virtual Multimedia*

[5] J.L. González, J. Caballero, M. Sánchez. *"MBone y las aportaciones del multicasting a los servicios multimedia en Internet"*, Novática nº 124, Dic 1996. Pp.38-39.

[6] Nonstandard for transmission of IP datagrams over serial lines: SLIP. RFC 1055

[7] *The Point-to-Point Protocol (PPP)*. RFC 1661

[8] *PPP Vendor Extensions*. RFC 2153

[9] *Compressing TCP/IP headers for low-speed serial links*. RFC 1144

[10] *PPP BSD Compression Protocol*. RFC 1977

[11] *PPPDeflate Protocol*. RFC 1979

[12] *Domain names - implementation and specification*. RFC 1035

[13] *Clarifications to the DNS Specification*. RFC 2181

[14] *Negative Caching of DNS Queries (DNS NCACHE)*. RFC 2308

[15] *Host extensions for IP Multicasting*. RFC 1112

[16] *Internet Group Management Protocol. Version 2*. RFC 2236

[17] Steinmetz, R., and Wolf, L.C.,"Quality of Service: Where are We?," *IWQOS'97*, pp.211-221, (1997).