

# Internet Protocols for Synchronous Distance Learning

Ch. Bouras  
Computer Technology Institute,  
Greece  
Computer Engineering and  
Informatics Dept., Univ. of  
Patras, Greece  
E-mail: bouras@cti.gr

A. Gkamas  
Computer Technology  
Institute, Greece  
Computer Engineering and  
Informatics Dept., Univ. of  
Patras, Greece  
E-mail: gkamas@cti.gr

Th. Tsiatsos  
Computer Technology Institute,  
Greece  
Computer Engineering and  
Informatics Dept., Univ. of  
Patras, Greece  
E-mail: tsiatsos@cti.gr

## Abstract

*The use of multimedia in telematics and new services has been greatly improved by the introduction of new systems and techniques. These new systems and techniques made the implementation of many telematic applications like tele-teaching, collaborative or tele-medicine applications possible. In this paper, we present design, protocols and implementation issues, which concern an Internet based synchronous distant learning application. We describe in detail the design and the implementation of each application's module and we appose the protocols, which control the communication among the application's components. The described application is a network-based application, which follows the client-server model. The application runs over the Internet and more specifically over the web. The users of the application access its services through web pages.*

## 1. Introduction

In recent years, the computer and communication companies have made great efforts to include multimedia capabilities into workstation and desktop systems. With these achievements, the multimedia teleconferencing system is widely accepted as a major functionality that can support many applications such as collaborative applications or tele-teaching applications.

To form a tele-teaching application, a number of nodes, including multimedia desktop computers must participate over a distributed computing environment. In order to realize a tele-teaching application, someone must implement not only tools for the presentation of the education material, but also a lesson control mechanism. The subject of conference control mechanisms has

engaged researches all over the world ([4], [5], [8]). The lesson control mechanism must not only offer capabilities for the control of the conference like creating a lesson, adding or removing users form a lesson, but must also offer capabilities for the control of the educational procedure, like floor control capability. During a tele-teaching session, one user, the teacher, must have the full control of the lesson's flow. Typical tele-conferencing tools regard all users equal, something that is not desired to a tele-teaching application. Therefore, typical tele-conferencing tools are not suitable for tele-teaching applications. The tools that assist the presentation of the educational material include a shared whiteboard, a shared chat and video and audio communication between the participants.

In this paper, we present the design and the implementation issues of a synchronous tele-teaching application. A synchronous tele-teaching application is an application that offers distance learning with the live presence of the teacher. In this category of tele-teaching, the lesson takes place at a predefined time.

The most prominent features of our application are its fully integrated environment and its platform independence. In addition our application follows is teacher oriented meaning that the teacher has the full control of the educational procedure during the lesson.

## 2. Requirements of tele-teaching applications

Tele-teaching could be regarded as the process of learning with the use of telematics (i.e. the combination of telecommunication, information and multimedia technology and its services).

A tele-teaching environment combines various instructional scenarios such as collaborative learning and education with or without the live presence of the

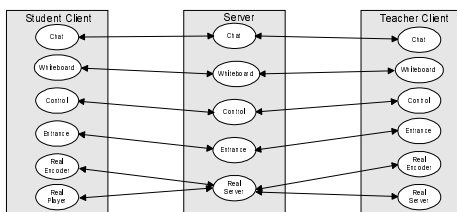
Professor. In this paper we present the design and the implementation of a synchronous tele-teaching application. The students attend the lesson, listen to the teacher and have the ability to interrupt the flow of the lesson in order to submit a question. The teacher has all the necessary authority to control the flow of the lesson (like as in a conventional lesson). For saving network resources, every student can see only one participant every time (the teacher, or a student that submits a question), as a result it is not recommended to use videoconference for the implementation of synchronous tele-teaching.

Someone can find more information about tele-teaching technology and applications on [6].

### 3. General architecture

The architecture of our tele-teaching application is based on the following concepts:

- ◆ An integrated communication environment that offers many functionalities to its users, like interactive communication over the Internet, including audio, video and data exchange
- ◆ A client-server model. We propose a model in which multiple participant clients connect with a server
- ◆ An open platform architecture in order to support different operating platforms, since for the implementation we use Java
- ◆ International accepted standards. We use widely accepted network protocols for the interaction between the clients and the server
- ◆ Access through web pages
- ◆ Object oriented design and implementation
- ◆ Modular design, in order to achieve the goal of expandability and simple implementation



**Figure 1 General architecture**

Figure 1 displays the general architecture of the application. The architecture is based on the client-server model. It uses two kinds of clients: the student client and the teacher client, and consists of the following modules:

- ◆ Server and client entrance modules: These modules are responsible for the acceptance of users request for entrance to a lesson. The client module sends the appropriate information to the server module, which recognizes the attributes of the user and connects the user to the appropriate lesson

- ◆ Server and client control modules: These modules are responsible for the management of the educational procedure. For example these modules are responsible for the floor control during a lesson
- ◆ Server and client whiteboard modules: The server whiteboard module is responsible for the management of the shared whiteboard and the client whiteboard modules are responsible for the representation of the whiteboard contents to the users
- ◆ Server and client chat modules: These modules are responsible for the management of the shared chat capability of the tool.
- ◆ Server and client media modules: These modules are responsible for the transmission of the video and audio information during a lesson.

For the issue of the state transition machine on the server modules, the simple way is keeping up only one state transition machine for each module per lesson. In regular situations the single state transition machine of a module would be sufficient to control the operation of the multiple client, if the multiple requests were serialized. However, if both users attempt to change to different states simultaneously, uncertain anomaly would be caused. To overcome the above problem, we use two types of state machines for each module: one is the state machine for the lesson and the other is a state machine for each user connection (we call this machine handler). By distinguishing the state machine of each user connection from the lesson machine; we avoid various problems that the access to shared resources by many users at the same time may cause.

### 4. System modules description

In the next paragraphs we describe each module in detail. The functionality of the server is exposed to clients via simple string-based messaging protocols. We choose string-based protocols, instead of other solutions (for example binary encoding), because string-based protocols have the advantage that they are easily manipulated by programming languages, can be transferred via application-level text-based transports and are easily logged and debugged.

#### 4.1. Entrance modules

The entrance modules are responsible for the acceptance of the users to a lesson. The entrance procedure works as follows: First the server checks if the user is a teacher or a student of the lesson. If the user is the teacher of the lesson (this means that the lesson did not start yet) the server creates a new session object that represents the new lesson. In the case that the user is a

student then if the lesson has already started (that means the teacher of the lesson has passed the entrance procedure and the system has created the corresponding session object) the system creates a new user object that corresponds to the new student of the session, if not the system sends to the user a message to inform him that the lesson did not start yet.

## 4.2. Lesson Control modules

Since the lesson control modules are accomplished by exchanging various information between the nodes in distributed environments, the designated protocol for the lesson control is considered as a mandatory part of the tele-teaching application.

The lesson control modules in our application are responsible for two main tasks: the transmission of the educational material and the management of the floor control capability.

When a user connects to the server, the server gives to his client information about the educational material of the lesson. The educational material consists of slides in the format of .gif or .jpeg images. When the user's client connects to the server, submits a request "Get LessonImages" the server replies with information about the location of the lesson's images (with the reply "LessonImages <image1> <image2>...").

Floor control refers to the coordination mechanism that governs access to the "floor", that is a shared resource. The most common resources that need floor control are the audio channel, where only simultaneous use makes sense, and access to pointers or to a shared applications. In our application the shared resource is the control of the application, which means that the use of the video-audio channel and the control of the whiteboard and chat modules.

When a student wants to be granted the floor, he must click to the appropriate button in his interface and must give an explanation why he wants the floor. Then the student client sends an appropriate packet ("Permission asked from X") to the server and the server forwards the message to the teacher client. The teacher client receives the student's request and adds it to the "floor list". When the teacher decides to give the floor to a particular student, he selects the student from the "floor list" and clicks on the appropriate button. Then the teacher's client sends to the server the appropriate packet ("Permission given to X") and when the server receives the packet it, sends the student which has been granted the floor the packet "Permission granted" and to the other students the packet "No more permission". When a student's client receives a packet "Permission granted", it activates the access to the shard recourses. When a student's client

receives a packet "No more permission" it informs the student that an other student has taken the floor.

At any time the teacher has the authority to remove the floor from any student. When the teacher removes the floor, the teacher's client sends a packet "Permission removed" to the server. The server forwards the packet to the student which has the floor and sends the others users the packet "More permissions" to inform them that the other student lost the floor. Teacher has the capability to activate and deactivate the capability of the students to request the floor with the use of the packets "Interruption" and "No interruption".

For fairness reasons, when a packet with a student's request for granting the floor passes through the server, the server adds a timestamp to that packet. With this approach the teacher can see in the "floor list" the time which each student made the request to grant the floor.

The following table summarizes the messages of the control protocol.

**Table 1: Messages of the control protocol**

Request	Reply	Parameters	Description
Set EndofLesson	Broadcast EndofLesson	Set the keepRunning parameter false	With this packet the teacher end the lesson
Set NoInterruption	Broadcast NoInterruption	Set the parameter Interruption false	Users do not have the capability to interrupt the teacher
Set Interruption	Broadcast NoInterruption	Set the parameter Interruption true	Users have the capability to interrupt the teacher
Set PermissionGiven <user>	Broadcast NoMorePermission	Set the parameter MorePermission false	One user grant the floor
Set PermissionRemoved <user>	Broadcast MorePermission	Set the parameter MorePermission true	The user lost the floor
Set PermissionGranted	No reply	Set the parameter PermissionGranted true	This message inform a user that has the floor
Set PermissionRemoved	No reply	Set the parameter PermissionGranted true	This message inform a user that lost the floor
SetPermissionAsked <user>	No reply	Add to the floor list the <user>	This message inform the teacher that a users wants the floor
Get LessonImages	LessonImages <image1> <image2> ...		The client gets information about the location of the educational material

## 4.3. Whiteboard modules

The whiteboard modules are responsible for the operation of the shared whiteboard. The server whiteboard module acts like a reflector, which reflects the packets that it receives, to all the users. The teacher whiteboard module and the student whiteboard module are the same modules. In the teacher module the access to the control of the whiteboard is always active and in the student module the access to the control of the whiteboard is active only when the teacher has granted to the student the floor.

The shared whiteboard offers to its users many functionalities like displaying slides, write text, make straight lines, rectangles or draw a line in different colors.

When the user, that has the floor, draws something in the shared whiteboard, the user's module sends a packet, which described the user action, to the server module and the server broadcasts the packet to all users participating in the lesson. The packets that the whiteboard modules exchange have information about the type of the action (for example change of color or draw of line) and the characteristic of the action (for example the co-ordinates of the line).

In addition the whiteboard offers an undo capability. In order to implement the undo capability, modules store each whiteboard action into a stack. When the users that has the floor selects the "undo" button, then sends to the server an "undo" packet and the server broadcasts the packet. When a client's module receives an "undo" packet it removes the last action from the stack. In the same manner we implement the redo capability.

The following table summarizing the messages of the whiteboard protocol:

**Table 2: Messages of the whiteboard protocol**

Request	Reply	Parameters	Description
Draw StraightLine <characteristic1> <characteristic2> ...	Broadcast Draw StraightLine <characteristic1> <characteristic2> ...		Draw a straight line to the whiteboard
Draw Text <text> <characteristic1> <characteristic2> ...	Broadcast Draw Text <text> <characteristic1> <characteristic2> ...		Draw a text to the whiteboard
Draw Rectangle <characteristic1> <characteristic2> ...	Broadcast Draw Rectangle <characteristic1> <characteristic2> ...		Draw a Rectangle to the whiteboard
Draw Line <characteristic1> <characteristic2> ...	Broadcast Draw Line <characteristic1> <characteristic2> ...		Draw a line to the whiteboard
Set Color <value>	Broadcast Set Color <value>	Set the parameter Color to <value>	Define the color
Set Undo	Broadcast Set Undo		Go one movement back
Set Redo	Broadcast Set Redo		Go one movement forward
Display Image <value>	Broadcast Display Image <value>	Set parameter CurrentImage to <value>	Display to the whiteboard the image <value>

#### 4.4. Chat modules

The chat modules are responsible for the operation of the on-line chat capability. The server chat module acts like a reflector, which reflects the packets that it receives, to all the users. The teacher has the capability to control, which of the users has the ability to send messages through the shared chat. Usually in tele-teaching application a shared chat is used by the students when they want to submit a question to the Teacher. The packets that the chat modules exchange between them, have the following format: "AddToChat <User> <Message>".

#### 4.5. Multimedia modules

The multimedia modules of our application consist of the commercial products of Real Server, Real Player and Real Encoder. The server module (Real Server) is a server, which is based on the streaming technology and offers advanced streaming capabilities.

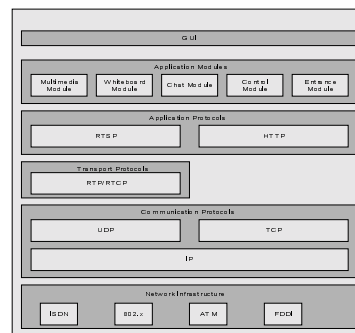
In tele-teaching applications, it is not need for viewing the video of all the participants at the same time during a lesson. During tele-training applications, all the participants attend the speaker, which for the most time is the teacher. For this reason we transmit only one video window to the participants during the lesson, the video of the speaker.

The multimedia modules co-operate as follows: The participant that is speaking uses the Real Encoder application, with the option of live encoding, and sends his video to the Real Server. The Real Server transmits the video, that it receives, and the users receive the video with the use of Real Player.

The use of Real Server Platform for the implementation of the multimedia modules has the advantage that we can create easy asynchronous versions of our lesson with the recording capability that the Real Server Platform offers.

#### 4.6. Communication between the Server and the Clients

The Figure 2 displays the protocol stack of our application.



**Figure 2 Protocol stack**

The data is transmitted using TCP or UDP connections depending to the data characteristics. For example voice or video data are transmitted with the use of UDP connections because for this type of data, the transmission speed is more important than the reliability. For data types that reliability is critical (like control messages) the application uses TCP connections. We use the Real Server platform, for the transmission of the video and the audio, which utilizes protocols like RTSP (Real Time Streaming Protocol which offers enhanced streaming characteristics) and RTP/RTCP (Real time Transport Protocol / Real time Transport Control Protocol, which offers advanced

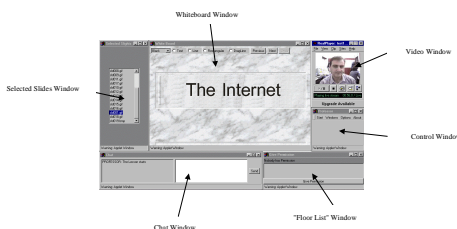
transport capabilities for real time data). In addition the Real Server platform offers the capability of using multicasting for the transmission of the audio and video. The above characteristic is important, especially during lessons with a big number of participants because the use of multicasting reduces the network load.

The communication between the server and the client modules (except the multimedia module) is based on TCP datagrams that are transferred through TCP sockets. The server binds four ports to the server hosting machine, one for each of the following server modules: chat, whiteboard, control and entrance. All the packets that we have described in previous paragraphs are transferred through those sockets.

## 5. Implementation issues

A very significant characteristic of the application is the selection of Java as the programming language with which it was developed. During the design phase, the demand for full portability of the application was considered crucial. That is, the application should be able to be executed independently of the workstation of the user. At the same time it should have the ability to use the Internet infrastructure for reaching every potential user. In order to achieve what was mentioned above and to build a very easy-to-use tool as well, Java was selected. With the use of Java our application can run over any platform, as long as the platform supports the Java Virtual Machine. The

Figure 3 displays the teacher's user interface.



**Figure 3 User interface**

We implement the server as a Java servlet and the users' clients as a collection of Java applets. For simplicity reasons and in order to achieve the goal of easy extension we implement each module of the system as different applets.

As we have already stated, the system design and implementation is based on object oriented concepts. The basic objects of the system are the user object, that represents a user of the system (either student or teacher) and the session object that represents a lesson.

For the implementation of the user interface we use the Abstract Windowing Toolkit (AWT) of Java. Because Java is a multi-platform solution, the AWT provides a

common interface to the native GUI components on a wide variety of platforms. This abstraction makes the AWT highly portable.

## 6. Conclusion – Future work

In this paper we have described the design and the implementation of a synchronous distance learning application. In order to realize this application, we have designed protocols for the interaction between users' clients and the server, and we have implemented a prototype application with the use of Java.

Our next step is the expansion of the application with new functionalities, like the application sharing functionality. We plan to implement an application-sharing module for our application based on the T.120 standard of ITU-T. With this module our application will offer capabilities for distributed collaboration among its users.

## 7. References

- [1] Ch. Bouras, A. Gkamas, V. Kapoulas, P. Lampsas, Th. Tsiatsos, A platform for the implementation of the services of an educational network, 15th IFIP World Computer Congress TELETEACHING '98 in Vienna and Budapest 31 August - 4 September 1998.
- [2] Ch. Bouras, A. Gkamas, Th. Tsiatsos, A Web-based Distributed Environment to Support Teleteaching: Design and Implementation Issues, 3rd International Workshop on Query Processing and Multimedia Issues in Distributed Systems (QPMIDS), September 1st-2nd 1999 (in Conjunction with 10th International Conference DEXA99), Florence, Italie.
- [3] Ch. Bouras, A. Gkamas, Th. Tsiatsos, Distributed Learning Environment Using Advanced Services Over the Internet, Third IASTED International Conference, Internet and Multimedia Systems and Applications, October 18-21, 1999 - Nassau, Grand Bahamas.
- [4] T. Hodes, M. Newman, S. Mccanne, R. H. Katz, And J. Landay, Shared Remote Control of a Video Conferencing Application: Motivation, Design, and Implementation, Proceedings of SPIE Multimedia Computing and Networking, San Jose, CA, January 1999, pp.17-28.
- [5] H Kim, H. Lee, B. Shin, S. Hwang, Egccm: A Conference Control Module for T.120 based Multimedia Teleconference, Proceedings of International Conference on Telecommunications, Porto Carras, Greece, 21-25 June 1998, Volume IV, Pages 135-139.
- [6] D. Minoli, Distance Learning Technology and Applications, Artech House, Boston – London, 1996.
- [7] RealNetworks : <http://www.real.com>.
- [8] Schubert, D. Sisalem, A floor Control Application for Light-Weight Multicast Conferences, Proceedings of International Conference on Telecommunications, Porto Carras, Greece, 21-25 June 1998, Volume IV, Pages 130-134.