

Control and Monitoring System by TCP/IP of a Spectrum Analyzer HP8590L

Guillermo AZUARA; Maria Teresa MENDEZ; Ana María LOPEZ
Universidad de Zaragoza, Escuela Universitaria Politécnica de Teruel, Ciudad Escolar s/n, 44.003 Teruel,
Spain, gazuara@unizar.es, <http://www.unizar.es>

KEYWORDS: *online laboratory, spectrum analyzer*

ABSTRACT: *In this paper, we propose a system that allows multiple users' access to the information that provided by spectrum analyzer, model HP8590L. This goal is carried out by means of a network that operates in real time. This system is at the moment into operation in the electronic laboratory of the Escuela Universitaria Politécnica de Teruel, in Spain, and used in Electronic Instrumentation subjects within a Communication Engineering degree.*

The HP8590L spectrum analyzer, a HP-IB instrument, and the server are connected via GPIB (General Purpose Instrumentation Bus), a parallel bus designed to connect standalone instruments to a computer and optimized for measurements applications. This server, placed on a PC computer, provides the measurements to the rest of users or clients via TCP/IP connections. These connections are implemented with CORBA (Common Object Request Broker Architecture) technology, a standard architecture of distributed objects. Finally, the high level applications associated to the server and the clients have been programmed with Java, creating an interface that resembles the real control panel of the instrument.

The final purpose of this work is that all the students in an electronic instrumentation laboratory (or in different locations) can visualize and measure the signal provided by a single spectrum analyzer. We have taken a special interest in several aspects of the system such as security, speed and transparency of the application. Transparency makes possible that users do not need to know further details about the communication with the instrument that can be placed in the same laboratory or even in a different location.

1 INTRODUCTION

In several subjects of technical degrees, the students learn the structure and performance of electronic instruments that they will surely employ at their future works. The learning of this equipment is generally focused on theoretical aspects since these instruments are expensive and the institutions can afford just a small number of them. In this way, at laboratory, the students merely observe how the teacher makes measurements and cannot experiment directly with the controls and switches nor control the effect of the measurement parameters on the final results. This problem is especially remarkable in small schools as the Escuela Universitaria Politécnica de Teruel is. In our case, the lecturer responsible for the subject "Instrumentation and Electronic Equipment" has at his disposal just one spectrum analyzer, one logic analyzer and one digital universal counter, what is clearly insufficient for a class of 60 students.

We have developed two solutions for this handicap. Adding a data acquisition card to our computers, we have designed virtual instruments with the appropriate software. The second solution is the subject of this paper. We have developed a server-client application that enables several users (clients) to visualize the measurements that a real spectrum analyzer makes in real time (figure 1). The instrument is connected via GPIB to a computer and establishes communication by the IEEE 488.2 protocol. This computer acts as a measurement server that provides, using TCP/IP protocol the signal from the spectrum analyzer to a group of clients. One of those clients (master) can control the performance of the analyzer, establishing the value of different parameters. These commands reach the server that communicates them to the instrument through the GPIB connection. The rest of users (read-only users) just observe the evolution of the measurements made by the master and obtain information from it. The TCP/IP connection is implemented with CORBA (Common Object Request Broker Architecture). The interface of the application has been programmed with Java. This interface is similar to the front panel of the instrument. In this way, students have the feeling of acting directly on the machine and can observe the measured

signal on their PC display. In order to make them possible to control the machine for a while, the application includes an access control that determines if a computer can access the application and who developed the control task.

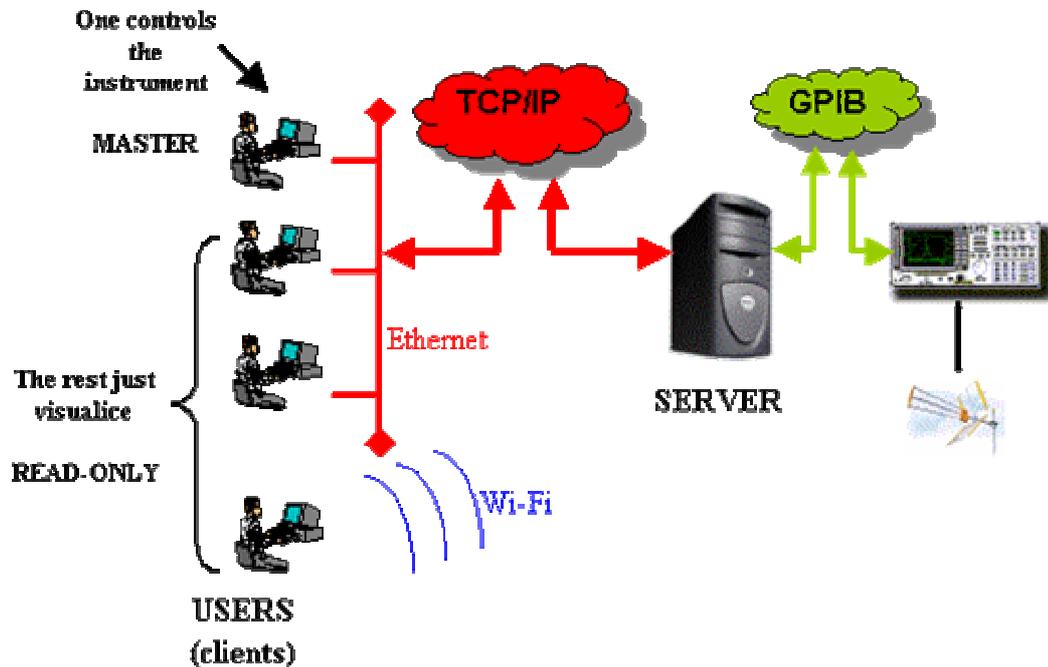


Figure 1- Scheme of the application.

2 ARCHITECTURE AND IMPLEMENTATION

Basic architecture of the system consists of three parts: first, a server (in this case a PC) connects by GPIB (IEEE 488.2) to the spectrum analyzer; this server provides in real time the information that is requested by hosts; it must also manage the simultaneous connections of several hosts (it make sure that only one host is in controller mode). Second, TCP/IP network with different data link layers such as Ethernet (IEEE 802.3) or WirelessLAN (IEEE 802.11). It is also possible a mixture of different technologies but always based on TCP/IP. Finally, one or several hosts simultaneously connected to spectrum analyzer through server; only one of them can control the analyzer while the rest must be in read only mode.

There are three basic parts in the application: the driver, CORBA and user interface.

Driver makes possible the connection between measurement server and spectrum analyzer. In general, drivers are a special kind of software whose function is to relate user demands (for example to click a mouse button) to the device which actually performs these tasks (for example to activate an option in the display). Although there are several control libraries for the analyzer, usually based on SICL (Standard Instrument Control Library) or VISA (Virtual Instrument Software Architecture), none of them allows the use of multiplatform language. Usually these libraries support C code, which depends on the platform. In order to make possible the use of these libraries with different platforms, the source code is implemented by a Java component. This component permits to use SICL or VISA through a Java native interface (JNI), which can invoke C programming language functions. These functions are compiled and packed in a dynamic library. This library can be called by a Java class in runtime.

User interface is developed in Java. The Java programming language is a high-level language that can be characterized by the following buzzwords: simple, architecture neutral, object oriented, portable, distributed, high performance, interpreted, multithreaded, robust, dynamic and secure [1-2]. With this component, users can control the spectrum analyzer as if they were operating with the real device, because the user interface provides the same controls as the front panel of analyzer.

CORBA (Common Object Request Broker Architecture) is an OMG's (Object Management Group) open, vendor-independent architecture and infrastructure that computer applications use to work together over networks. The separation of interface from implementation, enabled by OMG IDL, is the essence of CORBA. Using the standard protocol IIOP, a CORBA-based program can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network. It can handle a large number of clients, at high hit rates, with high reliability. CORBA applications consist of objects, that is, individual units of running software that combine functionality and data, and that frequently represent something in the real world. Typically, there are many instances of an object of a single type. For each object type, an interface in OMG IDL must be defined. The interface is the syntax part of the contract that the server object offers to the clients that invoke it. Any client that wants to invoke an operation on the object must use this IDL interface to specify the operation it wants to perform, and to marshal the arguments that it sends. The IDL interface definition is independent on the programming language, but is compatible with popular programming languages via OMG standards: OMG has standardized mappings from IDL to C, C++, Java, COBOL, Smalltalk, Ada, Lisp, Python, and IDLscript. The interface to each object is defined very strictly. In contrast, the implementation of an object - its running code, and its data - is hidden from the rest of the system (encapsulated). In the present project, CORBA is used for managing connections between server and hosts. The structure is provided by IDL libraries and a compiler. In this way it is possible the use of services and high functionalities of ORB (Object Request Broker).

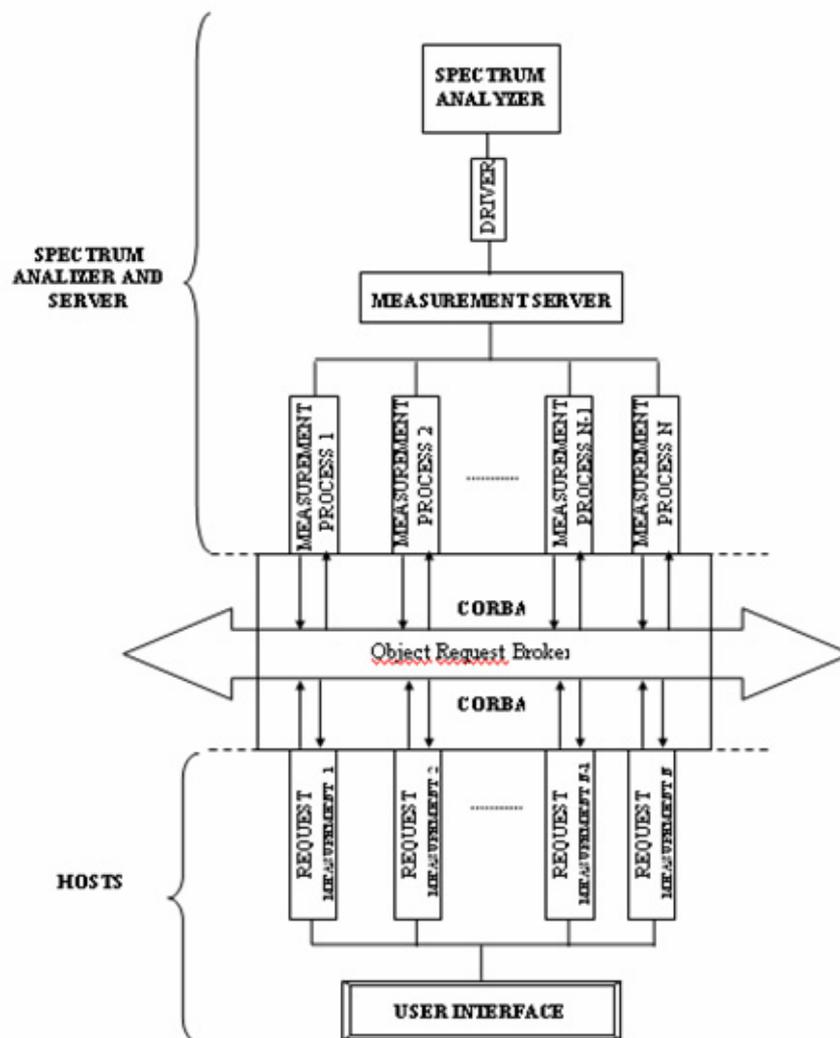


Figure 2- Basic programming structure

The application includes an **access control**. It controls if one user is authorized to accede to the system or not. This module implements a dialog between host and server. It works by sending an access request to a servant object in the server. It can get four replies: user no registered (access denied), incorrect password (access denied), time not assigned (access denied) and user authorized (access allowed).

We observe (figure 3) how application works.

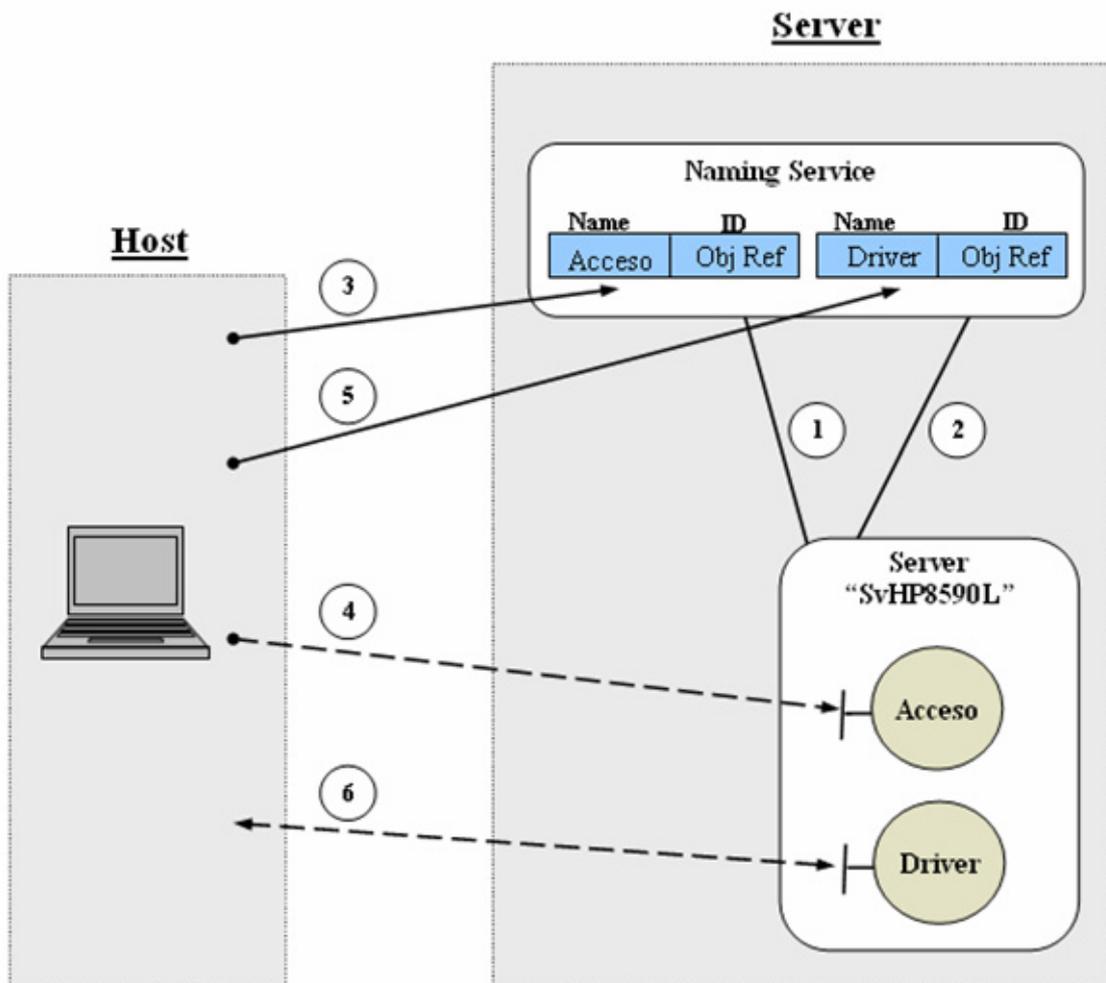


Figure 3 – General diagram of application performance

First (1), the application "SvHP8590L" publishes the reference of object "Acceso" in the naming service. Then (2), the application "SvHP8590L" publishes the reference of object "Driver" in this naming service. These two steps are necessary for providing to hosts, the services associated to these serving objects ("Acceso" and "Driver"). Next (3), host must obtain from naming service the reference to object "Acceso", and then can invoke the control access service (4) for checking if it is authorized to uses the system. Finally, the host (if it is authorized) obtains the reference to object "Driver" (5) and can start interoperating with the spectrum analyzer (6).

3 PERFORMANCE AND RESULTS

In order to make use of the application in the laboratory, it must be loaded in all the computers that are going to form the LAN. One of them, provided with the necessary GPIB hardware, is connected to the spectrum analyzer. This computer will be the server, and the server program "SvHP8590L" must be executed. Any computer that wish to access to the information from the analyser, must be connected via TCP/IP to the server and execute the client program.

When these steps have been followed, the first window that appears on the computer screen is the **access control** (figure 4). The users are requested to write an identifier of user and the associated password. It is indispensable in case the system should be used for e-learning, because this module warrants that the user, who has reserved a session, will be able to open session as “master” or controller. When assigned period of time expired, the user is ejected and other user can enter as “master”. The teacher assigns schedules, and can modify them in real-time.

Once the client enters the application the interface observed is shown on figure 5. There are four different zones. First, the **display**, similar to the display of the real instrument, where the signal at the input channel of the analyzer is drawn. As in the real instrument, this display also shows information about some of the measurement parameters as, for example, attenuation, span or sweep time. The signal on the screen is updated almost in an instantaneous way when any change on the analyzer parameters is done.



Figure 4 - Access control window

The **option menu** holds several sets of parameters. The parameters included in each one of these sets are shown on the **option parameters** panel when this set is selected. These groups of parameters are **frequencies**, that comprises the parameters that determine the visualization range; **amplitudes**, that includes attenuation, linear or logarithmic scale and detection mode; **filters**, where we can find the parameters related to the detection filters and the sweep time; and **setups**, where it is possible to save and open particular measure configurations. All of these parameters can be modified on the **option parameter** panel in a simple way, introducing numerical values on text boxes and selecting units with traditional menus. The last zone is the panel of **measurements on the display**, that includes several kinds of markers that make possible to obtain exact values of frequency and/or amplitude on the display. Also, it is possible to compare the trace at the moment with other saved signals as the maximum or minimum values obtained.

As it was mentioned before, there are two different kinds of clients. Both types observe the same interface on the computer screen. The master, the one who connects first to the server and is authorised by the access control, is able to act directly on the analyzer controls. The values of the parameters selected to this user are transmitted to the server that translates this information into GPIB commands that are sent to the instrument. The rest, observe on the screen the evolution of the signal when the master modifies the parameters, and can make measurement on the display with the elements on the **measurement on the display** panel, but cannot act on the elements of the rest of menus and panels. Of course, the control user can also use the options of the **measurement on the display** panel.

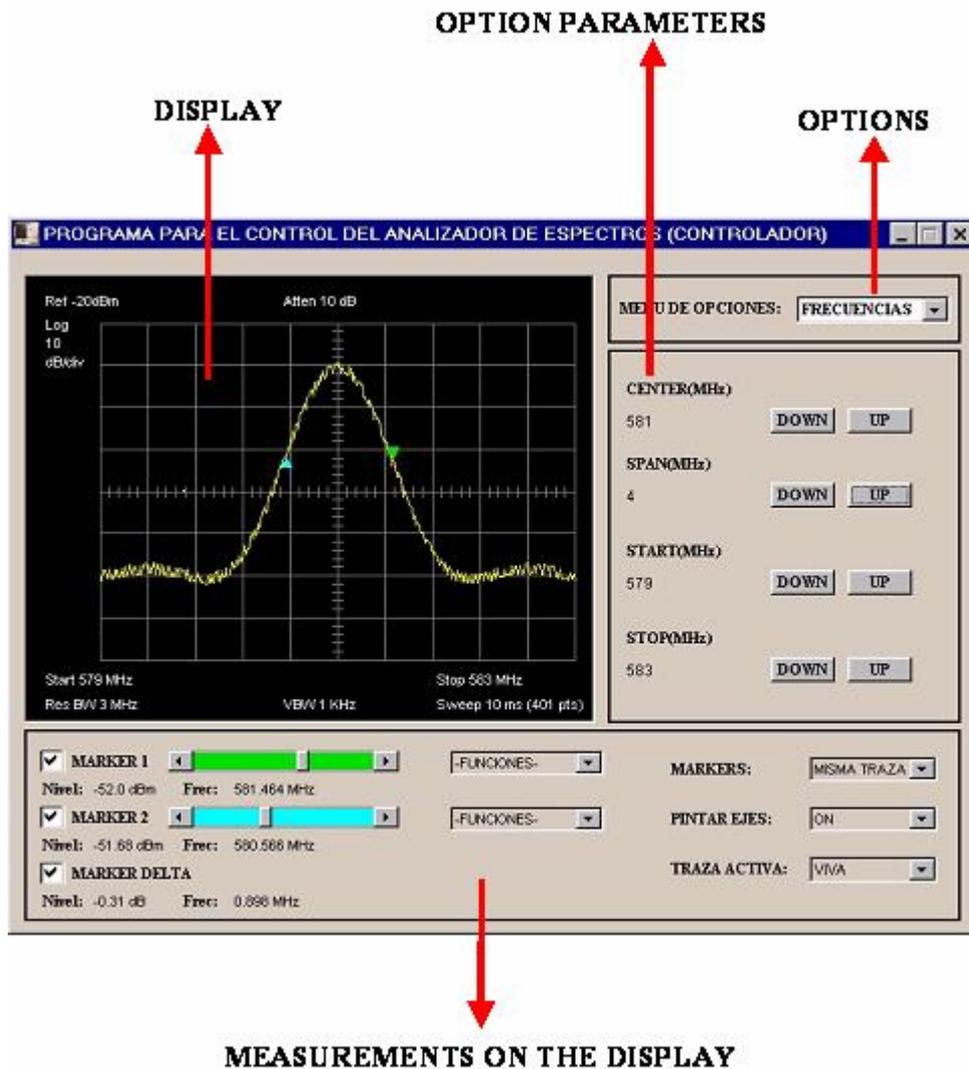


Figure 5 – Application interface. The options parameters corresponds to the set “frequency”

The system performance has been tested in a LAN, supported by Ethernet technology, formed by 21 computers. All of them received the information of the server in a correct way, visualizing the changes done on the analyzer almost at real time. At the same time two computers were connected to the server with wireless technology. The number of users connected corresponds to the total number present at the laboratory, we are sure that more PC’s could form a part of the system. The use of a multiplatform programming language as Java makes possible operation in different operative systems, with successful results in Windows 9X, Windows 2000 and Windows XP.

This system has been already employed as part of the laboratory activities related to the spectrum analyzer learning. Several important objectives are achieved. Previously, all the students should observe the analyzer signal on the small instrument display and had no possibility of making measurements on the signal. Now, they can observe clearly the signal on their computer screen and can use the markers to obtain information from it. The application is totally transparent to them, so they visualize or control the instrument without knowing how the communications between the different elements take place. At the same time, the answer of the system is fast enough and there is not any kind of annoying delays. The access control prevents unwanted users to access the instrument and at the same time allows that all the students control the parameters of the analyzer, establishing a timetable. Another important advantage is the possibility of adding measurement functions that the original instrument does not provide. These new functions are included in the high-level application software. In our case, our system permits to compare different signals that the software has previously saved. New type of markers have been added too. In a future a database could be even associated to the application.

This system can permit in a future time that the subject of electronic instrumentation can form part of a distant teaching programme since the laboratory activities can be done via TCP/IP.

4 CONCLUSIONS

The application developed permits to make laboratory experiences in a large group of students, with just one instrument. Though it is not possible that all the users change the measure parameters at the same time, they can observe clearly and at real time the evolution of the measured signal and the influence of the instrument parameters on it. Besides, they can obtain information from this signal thanks to a set of markers.

In this way, a better use of the resources is made, what is important at small institutions with insufficient funds. At the same time, this system permits to work with instruments that can be located far away from students that need a computer available that can establish a TCP/IP connection. Developing similar systems (programming the specific drivers) for other electronic instruments, an experimental programme for the distant learning of this kind of equipment would be possible.

REFERENCES

- ORFALI, R., HARKEY, D. *Client/Server programming with java and corba*. New York [etc.]: John Wiley, cop, 1998, 1022 p. ISBN 047124578X.
- About the Java Technology* [online]. Sun Microsystems. [cited 2004-03-14]. Available from web: <URL: <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>>
- CORBA® BASICS* [online]. Object Management Group. [cited 2004-03-14]. Available from web: <URL: <http://www.omg.org/gettingstarted/corbafaq.htm>>