Interconnection of Laboratory Equipment via Satellite and Space Links: Investigating the Performance of Software Platforms for the Management of Measurement Instrumentation

Luca Berruti, Franco Davoli, Stefano Vignola, Sandro Zappatore

National Inter-university Consortium for Telecommunications (CNIT) DIST-University
of Genoa Research Unit,– Via Opera Pia 13 – 16145 Genova - Italy
Phone: +390103532990, Fax: +3935322154, e-mail: luca.berruti@cnit.it,
franco.davoli@cnit.it, stefano.vignola@cnit.it, sandro.zappatore@cnit.it

Abstract. In the recent years, systems devoted to access remote laboratories through a networking infrastructure have been actively studied, and a number of specific software platforms, able to manage the instrumentation, have been proposed and implemented. Nevertheless, the problems involved in controlling remote devices on board satellite laboratories or in accessing remote equipment via satellite links are still not sufficiently investigated.

The paper briefly introduces the architecture developed within the Labnet project, especially as concerns its core software component, whose aim is providing unified access to heterogeneous equipment for a multiplicity of users.

Since the use of an earth-to-space link can possible affect the overall performance of systems based on a TCP-IP suite, a number of tests has been carried out, in order to evaluate the effectiveness and robustness of the proposed solution. Furthermore, the results regarding the performance of the Labnet platform are discussed and compared with those achieved by exploiting the facilities offered by a commercial and very popular software package.

1 Introduction

The recent years have seen an increasing use of satellite networks, especially those based on geo-stationary spacecrafts, owing to the offer of low cost terminals, and to new bandwidth availability in Ka band. The commercial exploitation of the Ka satellite band fostered the development of new packet-based multimedia applications, and the porting of a number of functionalities, initially devised for terrestrial networks. Through the last five years, the Italian National Consortium for Telecommunications (CNIT) has undertaken several projects, aimed at deploying a satellite network [1], as well as at providing a platform for remote laboratory management [2, 3] and at performing distance learning activities in higher education [4]. Still on the satellite side, CNIT has been actively involved in SatNEx, a European Network of Excellence (NoE) in satellite communications [5, 6], where research integration, training and dissemination activities are carried out, involving, among other tools, the use of satellite platforms. Recently, the new NoE SatNEx II has been launched, a follow-up of the previous one, which will also include experimental activities over satellite networks.

Although distance learning and training can fruitfully exploit services of a remote measurement platform, the possibility of remotely driving experiments represents an issue of great interest and relevance for many sectors, ranging from medicine and biology to mechanics and telecommunications. For instance, one can imagine a biological analysis test bench on-board a space laboratory, or a set of bio-medical devices mounted on mobile first aid and rescue units, to be used in case of earthquakes and hydro-geological disasters. In these scenarios, tele-measurement platforms can provide researchers on the earth a means to directly investigate phenomena in the absence of gravity, and medical staff a sophisticated infrastructure to carry out diagnoses, as well as environmental biotic and chemical analysis. The former case obviously requires an earth-to-space communication link, while in the latter a satellite link often represents the only available and reliable channel to connect the mobile units to an operative center.

Programmable equipment, remotely controllable devices, and fast/efficient interfaces constitute only the basis of a remote measurement platform. In general, one has to deal with issues concerning the possible heterogeneity of the application environments and of the instruments, the resource sharing, the efficient storage and transmission of data captured and of controls, needed for properly setting all the elements included in a remote laboratory. Hence, the software portion of such a platform plays a very important and critical role, as it must provide data and service abstraction so that end users can effectively receive data collected by the instruments, and send commands and possibly other data to them, thus preserving the realistic aspects of the experiment. In the literature, several architectures have been proposed to remotely control and manage measurement instrumentation [7–12].

As concerns commercial products, the Labview® suite by National Instruments represents one of the most widely employed software packages to remotely pilot instrumentation. In spite of its simplicity of use, no source code is available, and some strategies and communication protocols are partially unknown, limiting the implementation of new functionalities, the software portability and scalability.

Although a wide variety of architectures were proposed, few appear to be oriented to remote measurements for a distance learning experimental environment [13–15] and, for the most part, no support (e.g., multicast) is provided for an efficient and simultaneous dissemination of the measured data to a potentially large group of users. Furthermore, the application scenarios commonly involve the use of local area networks or terrestrial communication channels. The aspects concerning the exploitation of satellite links for accessing remote laboratories and measurement equipment are not sufficiently well focused and considered. The present paper addresses these problems and provides some experimental investigation.

The paper is organized as follows. Section 2 describes an ad-hoc designed software platform, which was originally developed within the LABNET project [3]. In the third Section, a number of tests, aimed at evaluating the performance of the system, are shown and the related results are discussed. The goal is to highlight the effects of a satellite link on the performance. Furthermore, a comparison between the behaviour of the proposed platform and a commercial one is presented, in order to prove the effectiveness of the solution devised. Finally, in the last Section, the conclusions are drawn.

2 LABNET Server Architecture

An efficient tele-laboratory system requires a Supervising Central Unit (SCU), whose task is to control and monitor the instrumentation involved in any experiment. The SCU decouples the user(s) (and the user software tools) from any issue related to the commands/controls and the communication protocols specific of each equipment. In other words, the SCU plays the role of an "interface" between the *inner* laboratory space (i.e., "the domain" of the real instruments and of their specific rules and protocols) and the *outer* laboratory space (i.e., the user space, the "domain" of abstracted instruments and standardized protocols).

Within the LABNET project, an ad-hoc SCU, called LabNet-Server (LNS), was proposed and its development started. Through the last four years, the Labnet Tele-Laboratory Architecture has been continuously evolving, and a number of versions of the LNS were released, in order to better meet the new requirements of the laboratory system. The current LNS is able to address several crucial concerns, such as: i) the intrinsic heterogeneity of the application environments and of the instruments, ii) the software portability and scalability, iii) the level of flexibility, and iv) the capability of efficiently exploiting IP-based satellite channels, as well as of multicasting the data gathered from the instrumentation for an efficient use of the transmission resources. All these aspects, although quite relevant, are not sufficiently well focused, and often neglected, in some products available on the market. Figure 1 illustrates the LNS architecture and shows its main components.

Owing to the LNS, the facilities of the remote laboratory can be exploited by means of any common Internet browser, which communicates with the LNS. The latter maintains a real-time database, which grants access to data gathered from the

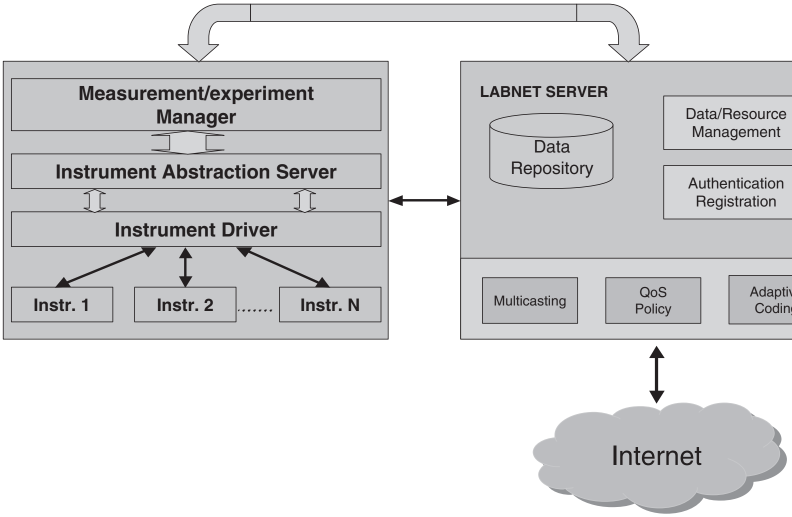

Fig. 1. Overall software architecture of the LABNET server.

instrumentation. In order to access the laboratory, a user must contact the Labnet Web-site and choose an experiment. In turn, the Web server uploads the client with a proper archive of Java applets and the LNS starts an experiment session, setting-up any kind of actions required for the correct execution of all the measurements.

The Java applets carry out the actual communication between the LNS and the user stations. Data exchanged can be roughly divided into two groups: the first one consists of commands toward the server, and then to the instrumentation on the field; the second one is related to data gathered by the LNS from the instruments and addressed to the clients. In order to assure a good level of interaction, the LNS can adopt the most suitable data coding, QoS strategy and, if needed, it can enable multicast transmission to save bandwidth. In this manner, the LNS hides the languages/environments specific to the laboratory test bench from the final user. It is just a task of the LNS to communicate with the proper experiment manager that actually operates the data exchange, by exploiting the services offered by the Instrument Abstraction Server (IAS). The IAS eventually communicates with and controls the physical devices through a set of drivers.

Finally, a set of ancillary modules completes the LNS in order to i) grant access only to registered users, who possess the proper access rights; ii) schedule the resources among competing users.

3 Performance Evaluation

A significant number of tests has been performed on the LNS with a number of clients, connected via a real satellite link. The tests were aimed at i) evaluating the efficiency of the LNS in terms of the possible delay and jitter of data packets observed at the receiver end, and ii) comparing the effectiveness of the proposed software platform with the "Data Socket Server (DSS)", a component of the Labview® package. The DSS, similarly to the LNS, stores and publishes data gathered from instrumentation for any possible use of the client stations. The choice of the DSS is motivated by the fact that Labview is a well-known package, very popular in the fields of tele-measurements and instrumentation remote control.

The experimental set-up, involving the satellite link, is depicted in Fig. 2.

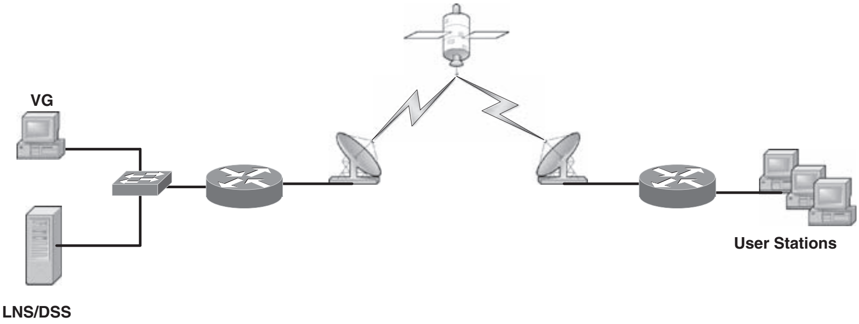

Fig. 2. The experimental set-up.

The "variable generator" (VG) plays the role of an experiment manager: It (quasi-) synchronously produces a set of data packets, conveying a group of 60 variables (52 of them scalars of 4 bytes each, 8 one-dimensional arrays of 1024 bytes each). When the LNS is in use, the VG periodically generates a single packet, containing all the variables. Upon receiving the packet, the LNS decodes it, updates the values of the referenced variables in its repository, and, finally, for each variable, it generates as many data packets as the number of client stations connected (we have chosen not to adopt the multicast option in the LNS for fairness of comparison). Therefore, if v is the number of variables, and the client stations are c, the LNS periodically sends v*c data packets on the satellite link. In the set-up centred on the DSS, the VG periodically updates 60 variables (of the same type and size as in the previous case) in the repository maintained by the DSS. The latter, in turn, notifies the clients the new values of the 60 variables. As the internal working mechanisms of the DSS are undocumented, the number and the role of packets involved in the overall process is not well known.

Since the total net payload (consisting of variables generated at the VG) is the same in both cases, the possible differences in performance can be attributed to the different protocols, data storing, retrieving, and forwarding strategies adopted by the LNS and the DSS. For all our tests, we employed a real satellite link (DVB-RCS like) in Ka band, exploiting the Skyplex processor onboard Hot Bird 6. Although the gross bandwidth amounts to 2 Mbit/s, the net capacity measured at the IP layer never exceeds 1.2 Mbit/s, owing to various overheads. The modems/routers at the earth stations are produced by Viasat, which seems to adopt a FIFO queue at the IP layer, and the number of user stations participating in all the tests is three.

In the set-up centred on the LNS, the operating system was Linux (kernel v. 2.6.11); in the one involving the DSS, Windows XP Pro sp2 was used. All the PCs employed were Fujitsu-Siemens Scenic P300 VKM266, and the switch was a CISCO Catalyst 2900xl.

Specifically, for each experimental set-up described above, the VG quasi-synchronously produces, every D seconds, a set of 60 variables, as already specified. Hence, the total net payload amounts to 8400 bytes. The "variable generation time" D (i.e., the time interval between the generation of two consecutive sets of variables) was set in different experiments at 1000, 500, 350, and 300 ms, respectively. Although the dispatching and publishing mechanisms of the DSS are not well known, the performances of the LNS and DSS can be jointly evaluated by considering the jitter of the "variable" transit time, viz the time a variable waits to be notified to a client station since its generation at the VG.

Let us consider a generic variable within the set periodically produced by the VG. Since the latter is implemented as a task in the user space, the generation time of the variable is itself affected by a certain level of jitter, owing to the intrinsic, non-real-time nature of the operating system. Therefore, the generic instant t_k when the k-th variable is produced can be written as

$$t_k = kD + \varepsilon_{tx} \tag{1}$$

where D is the (theoretic) "variable" time, and ε_{tx} is a random variable expressing the uncertainty about t_k. The algorithm used by the VG assures that t_k has zero mean.

The user station receives the k-th variable at the instant T_k given by

$$\mathrm{T_k} = \mathrm{t_k} + o + \varepsilon = \mathrm{k}D + \varepsilon_{\mathrm{tx}} + o + \varepsilon \tag{2}$$

where o is a fixed time offset due to the physical transmission (about 560 ms at our latitudes), and ε is a random variable, expressing the uncertainty related to the time spent in i) the queuing process at the VG, ii) the LNS / DSS to perform their tasks (packet decoding, data archiving, dispatching, . . .), iii) the modem/router queues, iv) the switch processor on-board the satellite, and v) the de-queuing/de-assembling process at the receiver end. It should be remarked that, in the case of the LNS, the first and last mentioned processes are quite tiny (the LNS protocol is based on UDP); thus, ε is essentially due to all the remaining contributions. On the contrary, when the DSS is in use, it is impossible to determine the different time contributions of ε. In both cases, ε represents the total uncertainty due to all the active processes involved in advertising that a variable has changed.

The notification instant of the (k+1)-th variable, $\mathrm{T_{k+1}}$, is given by

$$\mathrm{T_{k+1}} = (\mathrm{k} + 1)D + \varepsilon'_{\mathrm{tx}} + o + \varepsilon' \tag{3}$$

Then, the time interval between the arrivals of two consecutive variables is

$$\Delta\mathrm{T} = D + \alpha_{\mathrm{tx}} + \beta \tag{4}$$

where $\alpha_{\mathrm{tx}} = \varepsilon'_{\mathrm{tx}} - \varepsilon_{\mathrm{tx}}$ is a zero mean random variable, whose variance σ_α^2 is twice that of $\varepsilon_{\mathrm{tx}}$, and $\beta = \varepsilon' - \varepsilon$.

As α_{tx} and β are independent and α_{tx} has zero mean, then

$$\mathrm{E}[\beta^2] = \mathrm{E}[(\Delta\mathrm{T} - \mathrm{D})^2] - \mathrm{E}[\alpha_{tx}^2] = \mathrm{E}[(\Delta\mathrm{T} - \mathrm{D})^2] - \sigma_\alpha^2 \tag{5}$$

Since

$$\mathrm{E}[\beta^2] = \mathrm{E}[(\varepsilon' - \varepsilon)^2] = \mathrm{E}[(\varepsilon')^2] + \mathrm{E}[(\varepsilon)^2] - 2\mathrm{E}[\varepsilon'\,\varepsilon] \tag{6}$$

under the assumption that ε' and ε are independent,

$$\begin{aligned}\mathrm{E}[\beta^2] &= \mathrm{E}[(\varepsilon')^2] + \mathrm{E}[(\varepsilon)^2] - 2\mathrm{E}[\varepsilon']\,\mathrm{E}[\varepsilon] \\ &= 2\left\{\mathrm{E}[(\varepsilon)^2] - \mathrm{E}[\varepsilon]^2\right\} = 2\sigma_\varepsilon^2\end{aligned} \tag{7}$$

where σ_ε^2 is the variance of ε.

Combining Eq. (5) with Eq. (7) gives

$$\sigma_\varepsilon^2 = \tfrac{1}{2}\left\{\mathrm{E}[(\Delta\mathrm{T} - \mathrm{D})^2] - \sigma_\alpha^2\right\} \tag{8}$$

Hence, in a laboratory platform managed by the LNS or DSS, the RMS of the time needed to advertise a user station that a variable at the VG has changed can be estimated by

$$\mathrm{RMS}(\varepsilon) = \sigma_\varepsilon = \sqrt{\tfrac{1}{2}\left(\mathrm{E}[(\Delta\mathrm{T} - \mathrm{D})^2] - \sigma_\alpha^2\right)} \tag{9}$$

In practice, by evaluating the variance of the "variable" time computed over all the user stations, viz $\mathrm{E}[(\Delta\mathrm{T} - \mathrm{D})^2]$, and by computing the variance of the "variable" time at the VG, viz σ_α^2, it is possible to estimate the standard deviation of ε, that is the root mean square of the time the variable needs, "passing-through" the LNS or the DSS, to be notified, via a satellite link, to a user station. Finally, the overall variance of the time a general variable needs to reach a user station since its generation at the VG is computed by averaging the RMS (ε) over all the variables involved in the experiment.

Besides the experimental set-up previously mentioned (see Fig. 2), other two quite similar set-ups have been exploited, in order to carry out altogether three groups of tests, for both the LNS and the DSS. The first group is centred on the set-up sketched in Fig. 2 and includes the actual satellite link. In the second group of tests, the client stations are connected to the LNS (or DSS) via a terrestrial link, whose bandwidth (1.2 Mbit/s) amounts to the average bandwidth available on the satellite link. In this case, the experimental set-up includes two CISCO routers, back-to-back connected by means of two synchronous serial ports. In order to operate under conditions as similar as possible to those of the previous case, the queues of the router interfaces were set to FIFO. Eventually, in the last group of tests, the client stations are directly connected to the LNS (or DSS) by means a high speed (100 Mbit/s) LAN: no routers and satellite links are employed. In this manner, as the latencies owing to the communication channels can be neglected, we simply evaluate the jitter of the transit time inserted by the LNS (or DSS). In other words, ε used in equation (2), here represents a random variable, expressing the uncertainty related to the time spent by the LNS (or DSS) to perform their tasks.

The results achieved by the tests are summarized in Tables 1 and 2. The former shows data related to the LNS; the latter reports data obtained with the DSS. The columns labelled RMS refer to the Root Mean Square of the delay jitter (i.e., the difference between the expected and the actual variable transit time, viz the time a variable needs to reach a client since its arrival to the LNS). For every test, the values have been calculated by averaging the RMS of the delay jitter over all the variables involved in the experiment and, for each variable, over 10 repeated sets of 1000 transmissions.

Table 1. Estimated packet loss and RMS of data packet transit time vs packet time (i.e., the time interval between two successive packet departures at the VG) for a LAN, a terrestrial and satellite link, respectively, when the LNS is used to access the laboratory.

	LAN		Terrestrial		Satellite	
Variable Time [ms]	Loss [%]	RMS [µs]	Loss [%]	RMS [µs]	Loss [%]	RMS [µs]
1000	0	70 ± 2	0	139 ± 71	0	16615 ± 70
500	0	72 ±3	0	170 ± 78	0	15980 ±240
350	0	75 ± 4	0	258 ± 90	0	11030 ± 530
300	0	71 ± 5	1.6	14141 ± 78	2.3	9120 ± 212

Table 2. Estimated packet loss and RMS of data packet transit time vs packet time (i.e., the time interval between two successive packet departures at the VG) for a LAN, a terrestrial and satellite link, respectively, when the DSS is used to access the laboratory.

	LAN		Terrestrial		Satellite	
Variable Time [ms]	Loss [%]	RMS [µs]	Loss [%]	RMS [µs]	Loss [%]	RMS [µs]
1000	0	16750	0.2	103000	60	–
500	0	14500	25	189000	82	–
350	1.3	24500	60	–	96	–

The Tables report the results organized according to the group of tests the data belong to. Specifically, the group named “Satellite” refers to the tests involving the actual satellite link; the group named “Terrestrial” refers to the second group of tests above described, which permit to investigate the behaviour of the LNS (or DSS) when a terrestrial link (1,2 Mbit/s) is in use; finally, the group named “LAN” is related to tests performed on a high speed LAN. Some comments regarding the Tables are in order. As regards Table 1, the number of repetitions assures that the range specified for each value is characterized by a confidence level of 99%. The RMS values reported in Table 2 have a tolerance of 20% with a confidence level of 95%. Furthermore, whenever the variable losses exceeded 30%, we have preferred to omit the corresponding RMS for two reasons: i) owing to the significant variable losses, there are too few data in order to compute a stable and reliable RMS value; ii) such high values of loss are often associated to instabilities in the processes controlling the DSS: indeed, the DSS crashed during several tests.

The results highlight that the LNS performance is almost the same in the case of LAN and a terrestrial link, while a satellite link yields higher RMS values. Although the RMS values in this latter case are significantly higher than those measured in the other cases, the RMS values never exceed 3% of the variable time. Moreover, no loss is present for variable times of 1000, 500, 350 ms. The losses at 300 ms, both in the case of terrestrial and satellite link, are due to the queue length, inadequate to completely allocate room for the data bursts associated to the transmission of packets from the LNS.

On the contrary, the performance of the DSS, especially as concerns the packet loss, dramatically decreases when a satellite link is in use. Comparing the columns of Table 2 related to the terrestrial and satellite links, highlights how the propagation delay, inherent to the satellite link, strongly affects the overall performance of a tele-measurement platform centred on the DSS. Furthermore, the DSS appears unable to manage bursts of variables, whose inter-arrival times are less than 350 ms.

There are a number of reasons for the different behaviour of the LNS and the DSS, and it is not simple to motivate them. The DSS uses TCP as a transport protocol, whose performance may be negatively affected by the presence of a large bandwidth-delay product, whereas the LNS relies on UDP. Moreover, the mechanism of bandwidth allocation, which controls all the satellite modems/routers, seems to further reduce the efficiency of TCP. It is quite difficult to motivate the behaviour of the DSS in the absence of information regarding its internal structure. Therefore, the DSS appears more suitable to manage asynchronous controls and data within networks characterized by high bit-rates.

4 Conclusions

The paper has presented a possible extension of the tele-laboratory system designed within the Labnet and related projects. Specifically, the attention has been addressed to evaluate how an earth-to-space link can affect the overall efficiency of the supervising central unit, the software component that plays an important role in the entire system. To this aim, a number of tests have been carried out, whose results prove the effectiveness of the proposed solution. Furthermore, a comparison with a very popular commercial software package has highlighted that the devised platform appears more suitable to be exploited in all those contexts that include communication channels characterized by high delay-bandwidth products. Further work is in order to investigate whether some performance improvements might be achieved, by implementing the software within a multi-thread environment.

Acknowledgements

This work was partially supported by the Italian Ministry of Education, University and Research (MIUR) within the PRIN-CRIMSON project, and by the European Commission under the SatNEx Network of Excellence.

References

[1] F. Davoli, G. Nicolai, L. S. Ronga, S. Vignola, S. Zappatore, A. Zinicola, "A Ka/Ku Band Integrated Satellite Network Platform for Experimental Measurements and Services: the CNIT Experience", *Proc. 11th Ka and Broadband Commun. Conf.*, Rome, Italy, Sept. 2005, pp. 715–723.

[2] F. Davoli, S. Vignola, S. Zappatore, "A Multimedia Laboratory Environment for Generalized Remote Measurement and Control", *Proc. 2002 Tyrrhenian Internat. Workshop on Digital Communications (IWDC 2002)*, Capri, Italy, Sept. 2002, pp. 413–418.

[3] F. Davoli, G. Spanò, S. Vignola, S. Zappatore, "LABNET - Towards Remote Laboratories with Unified Access", *IEEE Trans. Instr. Meas.*, Oct. 2006 (to appear).

[4] G. Mazzini, A. Ravaioli, C. Fontana, P. Toppan, O. Andrisano, L. S. Ronga, S. Vignola, "The Teledoc2 Project: a Heterogeneous Infrastructure for International E-Learning", in F. Davoli, S. Palazzo, S. Zappatore, Eds., *Distributed Cooperative Laboratories - Networking, Instrumentation, and Measurements*, Springer, New York, NY, 2006, pp. 475–496.

[5] M. Werner, A. Donner, R. Sheriff, F. Hu, R. Rumeau, H. Brandt, G. Maral, M. Bousquet, B. Evans, G. Corazza, "SatNEx – The European Satellite Communications Network of Excellence", *Proc. 59th IEEE Vehicular Technology Conf. (VTC 2004-Spring)*, Milan, Italy, May 2004, vol. 5, p. 2842.

[6] B. G. Evans, "SatNEx – A European Network of Excellence in Satellite Communication", *Internat. J. Satell. Commun.Network.*, vol. 23, no. 5, p. 263, Sept./Oct. 2005.

[7] F. He, F. Wang, W. Li, X. Han, J. Liu, "Object Request Brokers for Distributed Measurement", *IEEE Comp. Appl. in Power*, vol. 14, no. 1, pp. 50–54, Jan. 2001.

[8] D. Grimaldi, L. Nigro, F. Pupo, "Java-based Distributed Measurement Systems", *IEEE Trans. Instr. Meas.*, vol. 47, no. 1, pp. 100–103, Feb. 1998.

[9] L. Nigro, F. Pupo, "A Modular Approach to Real Time Programming Using Actors and Java", *Proc. 22nd IFAC/IFIP Workshop on Real Time Programming (WRTP-97)*, Lyon, France, Sept. 1997, pp. 83–88.

[10] M. Bertocco, F. Ferraris, C. Offelli, M. Parvis, "A Client-Server Architecture for Distributed Measurement Systems", *IEEE Trans. Instr. Meas.*, vol. 47, no. 5, pp. 1143–1148, Oct. 1998.

[11] J. W. Overstreet, A. Tzes, "An Internet-Based Real-Time Control Engineering Laboratory", *IEEE Contr. Syst. Mag.*, vol.19, no. 5, pp. 19–34, Oct. 1999.

[12] S. You, T. Wang, R. Eagleson, C. Meng, Q. Zhan, "A Low Cost Internet-Based Telerobotic System for Access to Remote Laboratories", *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 265–279, July 2001.

[13] A. Ferrero, S. Salicone, C. Bonora, M. Parmigiani, "ReMLab: A Java-Based Remote, Didactic Measurement Laboratory", *IEEE Trans. Instr. Meas.*, vol. 52, no. 3, pp. 710–715, June 2003.

[14] X. F. Yuan, J. G. Teng, "Interactive Web-Based Package for Computer-Aided Learning of Structural Behavior", *Comp. Appl. in Engineering Education*, vol. 10, no. 2, pp. 79–87, Oct. 2002.

[15] T. Murphy, V. G. Gomes, J. A. Romagnoli, "Facilitating Process Control Teaching and Learning in a Virtual Laboratory Environment", *Comp. Appl. in Engineering Education*, vol. 10, no. 3, pp. 121–136, Jan. 2003.