Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

TELECONTROL SYSTEM BASED ON TCP/IP

Luis A. Baigorria P., José F. Postigo, Vicente A. Mut, Ricardo O. Carelli, Benjamín R. Kuchen.

Instituto de Automática (INAUT), Facultad de Ingeniería.
Universidad Nacional de San Juan.
Av. San Martín Oeste 1109 – J5400ARL. San Juan – Argentina.
Tel:+54 264 4213303 – Fax: +54 264 4213672
e-mail : {lbaigorria, jpostigo, vmut, rcarelli, bkuchen}@inaut.unsj.edu.ar

Abstract: In this paper, the development and implementation of a client software package for process control is presented. The proposed software is based on a client – server model under an Intranet architecture. The architecture is proposed for a telecontrol system of a real process, including the possibility of integrating I/O devices with data networks based on open protocols such as TCP/IP. This protocol allows the implementation of control systems using a low-cost alternative. Some experiences on a pasteurization laboratory plant are presented to show the proposed architecture performance. *Copyright © 2002 IFAC.*

Keywords : Telecontrol, Telemetry, Communication protocols, Computer-aided control system design, Industry automation.

1. INTRODUCTION

One of the main characteristics of network-based control systems (NCS) is that they have one or more closed control loops via a communication channel. In NCS the research interests are focused on the stability and performance of the control system rather than on the communication network.

Despite the fact that networks play a subordinate role within the control system, they are still important, because the feedback loop must be designed so as to use the communication channel as little as possible, on account that the network has multiplexed signals from other sensors and actuators. In addition, it has other communication possibilities related with other tasks unrelated to control duties, as shown in Figure 1.

The control networks are typically LANs with a set of hierarchical networks above them (Sonderman, 1998). The control loops are closed locally, with some loops remotely closed, such as in teleoperation systems (Castro, *et al.*, 2000) and monitoring systems with sensors distributed in WAN networks (Castro, *et al.*, 1998).

To integrate computer networks into control systems in order to substitute the point-to-point wiring, presents great advantages such as low cost, reduced power requirements, simple installation and maintenance and greater reliability. On account of these reasons, the NCS are becoming a widely preferred option. In recent years, the discussion has been centered on the expansion of Ethernet uses into monitoring and control applications (Laine, 2000). A considerable number of companies have begun to use the Ethernet standard in their industrial devices.

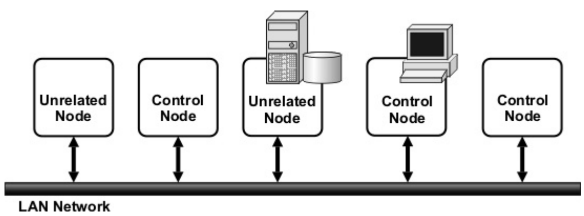

Fig. 1. Network-based Control System (NCS).

2. WORK OBJECTIVES

The main contribution of this paper is in the NCS field where we propose an integration of the available tools and protocols with the Ethernet platform using TCP/IP. With this aim, an architecture of a telecontrol system is developed with the TCP/IP-based communication protocol and the software architecture for a telecontrol application.

The proposed developments are intended as a low-cost NCS implementation that enables to integrate sensors and actuators connected to a single data network for monitoring and control from various sites.

The application objective of this work is to telecontrol an available pasteurization plant using a LAN with standard Ethernet, based on the TCP/IP protocol, without the need of implementing a dedicated data-communication system. The LAN network allows interacting with the application, and is able to go beyond the geographical boundaries of the server by means of the Internet.

2.1 Pasteurization Plant

A lab scale plant is used for the experiences performed. The lab plant consists in a pasteurization process which incorporates a complete range of methods and strategies of process control: from simple feedback loops up to complex cascaded loops, with multiple actuators and sensors, as shown in Figure 2. The plant operates with a high temperature-short time (HTST) industrial process for pasteurization that exposes the product at a fixed high temperature for a short time, usually with bacteriological purposes. This is accomplished through a holding tube that delays the product flow rate. This operation exposes particular control problems such as dead-time. The plant uses a three-stage heat exchanger (recycle - heating - cooling) and a diverting valve to reject the improperly treated product. The plant introduces real control problems, thus allowing to test various control strategies.

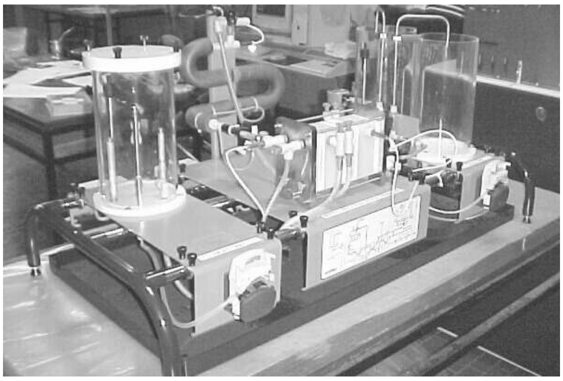

Fig. 2. Pasteurization process plant.

3. A LOW-COST OPTION

The communication channel is a LAN with standard Ethernet, based on the TCP/IP communication protocol. With the open architecture of PC networks based on Ethernet it is possible to configure almost any conceivable system. By applying these networks to new users of industrial equipment it is hoped to have the same open architecture and the ability to combine equipment from different suppliers that share the environment.

TCP/IP is the most widely used protocol in Ethernet systems. Using a protocol that is common with the information systems of companies allows the data be easily and readily shared from the manufacturing areas into the R&D laboratories. It enables as well to connect all levels of a corporate company, from the manufacturing plant to the administration and management floors, thus becoming a significant low-cost option.

3.1 Ethernet Features

Ethernet has some features that renders it more advantages than other specialized industrial control networks. The wiring-type makes it easy to install and to expand. Low cost many-port devices called hubs are easily connected to the network backbone, thus becoming an easy task to add devices to the network. The maintenance of an Ethernet is much cheaper and easier than any other network type. Ethernet is available in 10Mbps or 100 Mbps versions, providing a wide bandwidth which is translated into a faster response for real time applications (Winkler, 1999).

These features make of Ethernet a valid option that competes with the remaining control networks which were developed to render determinism in communication times, but paying a very high cost for this deterministic capability.

3.2 The TCP/IP Protocol

The TCP/IP protocol is widely supported as a communication protocol because it can be applied to almost any physical transmission media (Lutzs, 1998). TCP/IP protocol allows to co-exist different software protocols and different hardware TCP/IP components, providing –besides- an addressing system that enables to identify each node univocally over the remaining network nodes.

3.3 *Architecture of the Telecontrol System*

The architecture proposed for the TCP-based telecontrol system is shown in Figure 3.

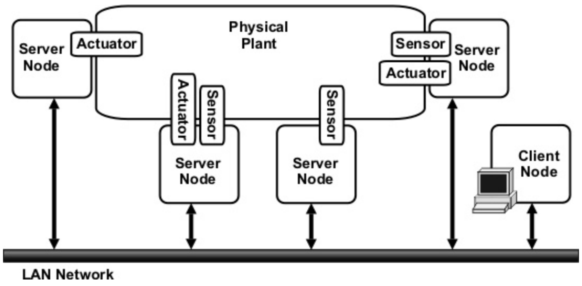

Fig. 3. Telecontrol System Architecture.

The *telecontrol system* is composed with several client-server structures. In these structures, the clients are implemented by the software developed at the control PC, and the servers are at the nodes connected to the network that manage the I/O points (Comer, D. E. and Stevens, L. D., 1994). There is a server at each network node. The nodes that manage the I/O points can be configured remotely, and they perform the updating of the values measured by the sensors. Besides, they are in charge of performing the control actions received from the clients at the output points.

In the control PC, there are clients available to carry out the communication with the server nodes. The PC software performs the control system by receiving the system response in real time, through the network and from the nodes. It also computes the control signals and transmits the control actions to the server. The server applies the control action at the corresponding point according to its configuration.

Advantages of the Proposed Architecture

Considering the proposed architecture, it is possible to implement servers with more economical devices and with lower power rates, but with lower computational power as well. The servers tasks are limited to apply the control actions received from the clients, to keep updated the values of the variables measured by the sensors and to communicate the data to the client. This architecture enables the connection of many PCs operating coordinately via the network, thus permitting to increase the computational power for implementing complex control laws, independently of the server's limitations. Moreover, it provides greater flexibility in the design process of the controllers due to the freedom to implement the software by means of any controller type at the client.

3.4 Communication Architecture

The TCP/IP communication architecture was developed for communicating between the clients and servers, where the IP datagrams are exchanged among the nodes through the network layer. With the TCP/IP protocol, several clients with different IP numbers can

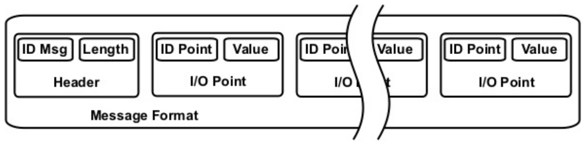

Fig. 4. Typical message format. This datagram has a variable length defined in the *length* field, indicating the number of points included in the datagram package.

be concurrently connected to the same server, and require data from it simultaneously.

Datagrams format

Various message types were defined in order to be interpreted by the clients and servers. A typical datagram transmitted between the client and the server is shown in Figure 4.

3.5 Telecontrol Software

With the tools available for object-oriented programming, such as the Unified Modeling Language (UML) and design programs such as Rational Rose®, it is possible to develop in a simple and orderly manner a powerful application composed of several subsystems developing specific tasks.

UML allows the designer to develop several different types of visual diagrams that represent various aspects of the telecontrol system. *Use-Case diagrams* and *Class diagrams* shows interaction between classes, which represent the telecontrol system functionality. *Sequence diagrams* are used to show the flow of functionality through the communication process to perform the control task. *The State Transitions diagrams* are proper to represent the various states in which the object that perform the communication can exist. The *Deployment diagrams* show the physical layout of the network for the telecontrol system, and where the various components will reside.

Using UML, the designer can easily map the diagrams to C++ or Java code, then he/she can ensure that the requirements were actually met by the code, and the code can easily be traced back to the requirements.

The developed software is made up of several subsystems interacting with each other in order to perform the telecontrol tasks. It includes, basically, the subsystems for *configuration, control, communication and visualization*, as shown in Figure 5.

3.6 Configuration Subsystem

When the user is carrying out the configuration of the telecontrol system, the system updates the information

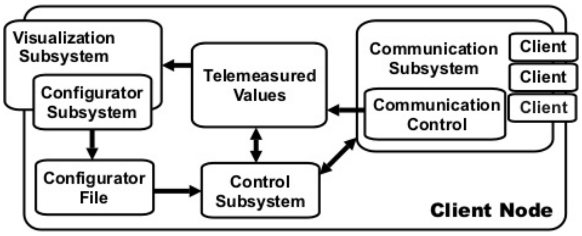

Fig. 5. Telecontrol application subsystems.

of the I/O points of the nodes through polling of each node, using a specific message. Each node returns in a data package the information of the I/O point on which it acts. This information is stored in a file that will be used by other software subsystems.

The user begins the process by constructing or modifying the control structure that is stored in a file in order to be used in the control subsystem. The construction of the architecture begins by choosing the controller type to be used in the control loop. Then, the "input points" are chosen from a list, which will be used in the control algorithm. The same is done with the "output points", on which the control actions are to be performed. The loop's configuration ends by specifying the parameters of the controller according to the specific controller type, as shown in Figure 6. The configuration process ends when the user "adds" the controller to the file.

Once the package has been made with the information from the control loops, it is used by the control subsystem.

3.7 Communication Subsystem

The techniques used by telemetry systems to update the remote variables are applicable to the NCS as well.

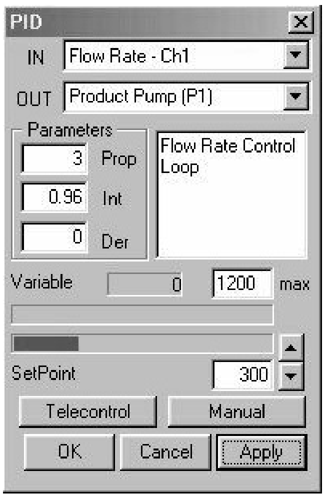

Fig. 6. PID Controller configuration window.

The choice of data gathering technique used by the client can also limit the ability to use particular communication techniques (Aubin, P., 1997). The available techniques are:

- *Polling*: the *client* asks in sequence the data from each *server* node.
- *Polled Report by Exception Techniques*: the *client* asks only for the changes on the variables from *server* nodes.
- *Unsolicited Report by Exception Techniques*: the *server* node begins the transmission without having been first solicited by the *client*. A background poll performed by the *client*, verify that data are updated and checks the "health" of remote devices.
- *Quiescent Techniques*: the *server* always begins the communication, the *client* (application) never asks information from any remote site. The most bandwidth efficient technique is the quiescent operation.

Among the above techniques, the most appropriate one for a telecontrol implementation resulted the *Quiescent Technique*, in which each server sends a message to the client in order to carry out the control action to be applied on a specific controller. As it was mentioned before, the method consists of sending the variables sensed in the server node, to the client at a sampling period rate specified by the controller. Once the datagram is received, the client sends out the variable with the communication subsystem, and it "tells" the control module to perform the computation of the control action. Then the control action is transmitted to the node that has the variable on which it is intended to act upon. In this case, each of the *server* nodes has information about both the sampling period and its clients connected. Moreover, the server node sends the information periodically, according to the sampling period.

3.8 Control Subsystem

The technique used for computing the control action is based on sending periodically the data from the nodes. Two developed subsystems are combined to perform this task: the control subsystem and the communication subsystem.

The communication module is designed to receive data packages from the nodes servers, to interpret the messages and to obtain the variables contained in the message. Once the variable has been identified, such an information is given to the control subsystem that executes the controller corresponding to the received variable from the communication subsystem. The control subsystem is in charge of executing the controller and points out the control action along with the identifier of the output point existing at some node server. The communication module determines the node such a variable belongs to, and sends the message through the network along with the control action to be applied to the corresponding actuator.

3.9 Server node architecture

The proposed *server node architecture* is simple, as shown in Figure 7. Its operation includes the possibility of accepting several clients that *subscribe* for the variables contained at the node, using a specific message. The *server* node has a *timer* configurable from the client, in order to send the sensed variables that had been subscribed before. It also receives and performs the control actions at the corresponding output variables.

The *server* node electronic implementation includes a bus to connect the signal conditioning circuits for the sensors and actuators. The server node reads and writes through the bus the measured values from the signal conditioning circuits connected to it, keeping its complexity very simple. It is also expected to connect, to the bus, 4-20 mA loops drivers to communicate with the sensors and actuators.

The *server* node also implements a second *timer* to detect communication failures, or the increasing of the communication delay. When the *server* node receives control actions it resets the *timer* value. If the *timer* finishes, the *server* node sets the output variables with secure operation values, and waits for the client to send new control actions or to reset the communication. The secure operation values and the period of time are configured for every output variable from the *client* node, using a specific message.

Several experiments were carried out and, among them, an application was run simulating the behaviour of a node, with I/O points, complying with the protocols defined in the application. It implemented a server using the sockets of the Winsock APIs of Miscrosoft Windows® (Jones, A. and Ohlund, J. 1999). The main program incorporated the possibility of implementing a client- server connection in the same PC, to implement a local controller, for experimental purposes.

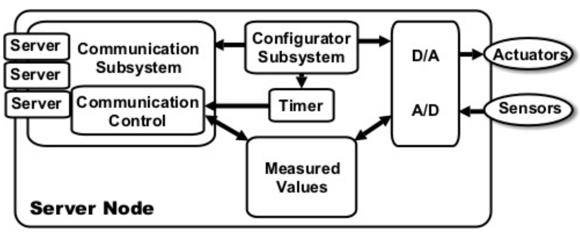

Fig. 7. Server node architecture.

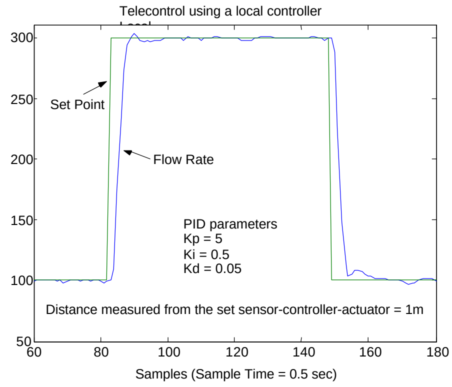

Fig. 8. Step response of the flow-rate control loop locally controlled.

3.10 Experimental results

Some experimental results were obtained from controlling the described pasteurization plant using local and remote controllers connected within an Intranet existing at the INAUT´s building. The first results allowed to obtain conclusions about the effects of the delay on the communication on the control loop and, thus, to test the proposed application in order to perform future developments.

Figure 8 shows the response to the step input signal of the flow-rate control loop applying a local controller. Figure 9 depicts the step response of the system remotely controlled. By comparing the corresponding responses, it can be concluded that both controllers behave acceptably, because the delay in data transmission is small (<1 ms) as compared to the sampling period of the control loop (0.5 sec). As shown in Figure 9, when the network speed is high and the traffic is minimal, the effect of inserting data in the control network originates a small time delay that varies randomly over the Ethernet network. This delay is acceptable in those control loops applied to slow-dynamic plants (such as the one used in this paper), which do not require large sampling frequencies.

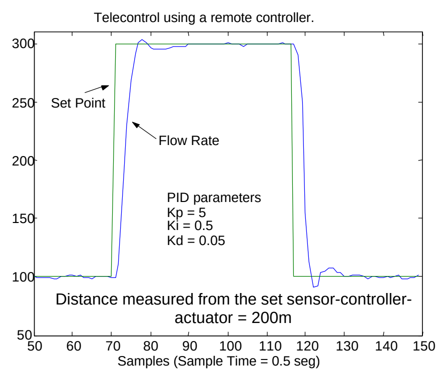

Fig. 9. Step response of the flow-rate control loop remotely controlled.

However, in the case of network congestion, or larger time delays due to the distance involved in the control loop, weaknesses arise on the above approximation. Many of the delay sources do not pose any difficulty on most applications. However, the delay induced on the network that takes place when exchanging data between the devices connected to the shared common medium, can be constant or variable in time, thus degrading the performance of the control system designed, even up to the point of de-stabilising the system (Imer and Basar, 2001).

Some of the known compensation approaches are based on the model of the system, as the one proposed in García, *et al.* (2000), or the schema based on wave transformation variables (Benedeti *et al.,* 2001), that guarantee the stability with fixed time delay and variable time delay. Nevertheless, one of the most widely used controller for inherent time delayed systems is the Smith predictor-based controller, that has the effect of remove the delay within the control loop, and realize effective closed-loop control of the un-delayed process using a conventional controller (Wood, 1998). In this case, we applied the Smith predictor-based controller in the experiments to deal with the communication delay, with acceptable results.

4. CONCLUSIONS

The development of a telecontrol system of a lab pasteurization plant based on an Ethernet network as a communication channel has been presented in this paper. The feasibility of using the TCP/IP protocol for remote process control has been tested.

Regarding that a real process was used instead of a PC-simulated process, it was shown that the client-server model is appropriate for the telecontrol of a process as a low-cost alternative.

Finally, it could be mentioned that this methodology has the advantage of an easy and ready implementation of other control schemes, such as parameter tuning of the on-line controller at the client, while the server is transmitting the system's response (Gillet and Gorrochategui, 1998).

Future works include the design and implementation of several controllers for NCS, that compensate the induced variable delay in the network over an Internet link for control purposes. These controllers may differ significantly from the design of traditional centralised control systems.

5. REFERENCES

Aubin, P. (1997). Telemetry Systems, Techniques and Standards. *Instrumentation and Control Exhibition Symposium, June 1997.*

Benedetti, M., Franchini, M., Fiorini, P., Postigo, J. F., García, C. E., Carelli, R. (2001). Compensation Methods for Time-Delay Teleoperation. *Proc. of the IX Reunion de Trabajo en Procesamiento de la Información y Control,* Vol. 2, pp. 423-428. Santa Fe, Argentina, September 2001.

Castro, A. C., Postigo, J. F., Manzano, J., Kuchen, B., (2000). Sistema de Teleoperación de Robots con realimentación de fuerzas basado en redes TCP/IP. Proc. of the *XVII Congreso Argentino de Control Automático,* pp 201-206. Bs. As, Argentina, September, 11-13.

Castro, A. C., Zavalla, E. M., Martín, M. L. (1998). Modelo Cliente-Servidor para la administración visual en edificios. *XVI Congreso Argentino de Control Automático*, pp. 543-548, Bs. As., Argentina. August, 17-21.

Comer, D. E. (1995). *Internetworking With TCP/IP Vol I: Principles, and Architecture. Third Edition.* Department of Computer Sciences Purdue University. Ed. Prentice Hall International Inc.

Comer, D. E. and Stevens L. D., (1994). *Internetworking With TCP/IP: Client-Server Programming And Applications.* AT&T TLI Version. Department of Computer Sciences Purdue University. Ed. Prentice Hall International Inc.

García, C. E., Carelli, R., Postigo, J. F., Morales, B., (2000). Time delay compensation control structure for a robotic teleoperation system. *Proc. of the SICICA ´2000 - 4th IFAC International Symposium on Intelligent Components and Instruments for Control Applications*, Vol. I, pp.321-326. Bs. As., Argentina. Sept. 13-15.

Gillet, D., Gorrochategui, E., (1998). Remote Experiment with LabView. *National Instrumentation Newsletter*, Vol. 10, Nº 3.

Imer, O.Ç., Basar, T. (2001). Control of Congestion in High-Speed Networks. *Semiplenary Lecture in the ECC ´2001. Seminario de Vilar, Porto, Portugal. Sept. 4-7, 2001.*

Jones, A. and Ohlund, J. (1999). *Network Programming for Microsoft Windows,* pp175 - 226. Mircosoft Corporation (1999). Redmond Washington, United States of America.

Laine, T. (2000). Impact of Internet technologies on industrial fieldbuses. *Proc. of the 4th IFAC International Symposium SICICA ´2000, Vol. I, pp. 57 – 62.*

Lutz, T. A., (1998). Using TCP/IP as an instrument interface. Sensors, July, 1998, pp 43 – 46.

Sonderman, D., (1998). Getting Control of the Process. Sensors, October 1998, pp 12 – 18.

Winkler, R., (1999). Clearing Up the confusion on Ethernet. Sensors, January 1999, pp 28 - 35.

Wood, H. C., Olson, M. R. (1998). Time-Delayed remote control systems using Smith controllers. *Proceedings of the IASTED Internacional Conference Intelligent Systems and Control*, pp. *36-39. Halifax, Nova Scotia, Canada.*