A&QT-R 2002 (THETA 13)
2002 IEEE-TTTC International Conference on Automation,
Quality and Testing, Robotics
May 23 – 25, 2002, Cluj-Napoca, Romania

NEW COMMUNICATION TECHNOLOGIES FOR DISTRIBUTED CONTROL

Authors: Prof. Dr. Kalman Pusztai, S.L. ing. Gheorghe Sebestyen,
Universitatea Tehnica din Cluj-Napoca, str. G. Baritiu 26-28
Tel: 40-64-143003, Fax: 40-64-194491,
Emails Kalman.Pusztai@cs.utcluj.ro, Gheorghe.Sebestyen@cs.utcluj.ro

ABSTRACT

The present article gives a comprehensive overview on the communication methods and technologies available today as support for distributed control applications' implementation. A number of specific communication requirements and issues are discussed, and some possible solutions are suggested. The article analyzes the possibility of adapting some of today's Internet technologies for control purposes. In conclusion it is shown that control applications need a more reliable and deterministic communication environment, with explicit mechanisms for timeliness, quality of service (QoS) and dependability control.

Keywords: Internet technologies, control applications, real-time communication

1. INTRODUCTION

Complex automation applications, involving a significant number of distributed devices, spread on a wide area need an appropriate communication support. The use of general-purpose communication networks and protocols is limited by the restrictive requirements of timeliness and reliability specific for control applications. As a first approach [1, 2, 3], in the '80-s and '90-s a number of special communication protocols were developed and used. They are known as industrial networks of fieldbuses. But this solution generated a gap between the process-flour applications and the business administration applications. Today new communication solutions and models are required to allow the integration of process control with the distributed information system of an enterprise.

The present paper analyses the specific requirements of the automation field and the possibilities of adapting the new Internet communication technologies to these requirements.

2. SPECIAL COMMUNICATION REQUIREMENTS IN CONTROL APPLICATIONS

The communication environment, as part of a distributed control system, must fulfill the specific requirements of a control application, such as: real-time behavior, reliability, fault-tolerance, predictability and dependability. In some applications there may be some extra-requirements, such as: immunity to high levels of noise, safety in hazardous area (explosion danger), immunity to significant environment changes (e.g. temperature, humidity), dynamic reconfiguration, and others. The fulfillment of these requirements must be guaranteed in the design phase of the project. Special mechanisms should be implemented to control and enforce the fulfillment of some predefined performance parameters.

One important issue is **the reliability** and **fault-tolerance** of the data transfer through the communication network. It is known that most of today's network protocols does not eliminate the possibility of transmission errors. Therefore the communication protocol must contain error detection and correction mechanisms that can mask the accidental transmission errors, guaranteeing that no undetected errors will appear during the lifetime of the system.

Another important issue is **the timeliness** of the data transmission. It is important to notice that the correctness of a real-time system (and most control systems are real-time systems), equally depends on the correctness of data processing and on fulfillment of deadlines. Communication, as part of processing loops, must be made in a predefined time limit (deadline). Messages must be scheduled for delivery based on their time parameters and deadlines. Many industrial protocols incorporate message-scheduling facilities that can guarantee message delivery before their deadline.

In control applications, the system's reaction time is considered an important performance parameter. Usually the required reaction time depends on the time characteristics of the controlled process or on the maximum frequency of the input signals. Therefore message transmissions must have much shorter delivery time limits. These time limits can be achieved only if the messages are short and the medium access control protocol (MAC) assures a short roundtrip of the access token (see token-bus and token ring protocols).

Another issue is **the predictability** of message transmission. For instance Ethernet protocol is a highly efficient transmission protocol, offers a good access time to the communication medium, but it is not predictable. Because of its collision based access method, a message delivery may be indefinitely delayed by collisions. This effect becomes more pregnant in overflow situations. Safety-critical applications do not allow a probabilistic behavior; a slower but predictable protocol is preferred.

3. A BRIEF OVERVIEW ON PROTOCOLS AND COMMUNICATION MODELS USED IN CONTROL SYSTEMS

Computer-based control systems need a communication infrastructure, that connects field automation devices (e.g. sensors, transducers, actuators) with control and supervisory devices (e.g. PLCs, regulators and process computers). Industrial communication networks are specially developed for such purposes. These networks are optimized for the specific data flow of a control system [5], their behavior is deterministic and real-time message delivery requirements can be guaranteed. There are three classes of industrial networks: actuator-sensor networks (used at process-floor level), fielbuses (used for middle complexity systems) and cell networks (used for complex, process optimizations).

These networks assure the low level transmission services, specific for message based communication. But an efficient design process requires more powerful abstract concepts (e.g. sockets, virtual devices), communication models (e.g. client/server, consumer/producer) and tools (e.g. predefined services). By using such high level tools, the automation system designer can concentrate more on supervision and control issues than on communication problems.

For general-purpose (non-control) applications (e.g. distributed databases, banking, interpersonal communication, etc.) a number of communication models and programming tools were developed and successfully used, such as: the client/server model, the remote process calls (RPC), the object-based transparent access model (CORBA, COM and DCOM), web-based access (HTTP, CGI, XML, etc.), mobile agents, and others. The question is if these new models and tools can be used also for control applications. The answer to this question is neither simple nor unique.

Communication models used in control applications must offer better support for periodic data transfer (e.g. for data acquisition and control generation), for priority specification and for delay measurement and control. Typically used communication models are:

- the consumer/producer model
- the master/slave model
- the time scheduled communication – a time slice is allocated for every network node
- the virtual manufacturing device model

In the producer/consumer model "producer" nodes transmit periodical information concerning the process variables managed by them and one or more "consumer" nodes may receive the information; a given network node may be in the same time a producer for some variables and a consumer of others. This model assures a periodic data flow, location transparency and a certain level of fault tolerance.

In the master/slave model, master nodes have the ability to initiate data transfers, sending or requesting data to or from slave nodes; the complexity of the communication protocol is concentrated in the master nodes, requiring small resources for slave nodes. This model is recommended in industrial networks, where a wide range of automation devices is involved, with different processing capabilities.

The time-scheduled model is preferred in time critical applications, where a certain periodic data flow must be assured. An off-line transmission schedule is defined, that can guarantee the fulfillment of message transfer deadlines.

The virtual device model offers a class of virtual objects adapted to the needs of a control application (e.g. input/output signals, event objects, program and context objects) and a standard data/service access interface. This model solves the compatibility, interoperability and interchangeability issues between different types of automation devices.

These communication models are easy to implement on a local network, but there are much harder to use on an intra- or inter-net environment. In the latest networks there is a pour control on the quality of services and the communication load patterns are unpredictable. This is caused by the fact that the process control data must be mixed with an unknown data flow.

But on the other hand, there is a significant demand for Internet-based distributed applications. End-users want to visualize and control their production facilities through the Internet. Therefore new models are needed, models that are common for the Internet environment.

3. INTERNET TECHNOLOGIES FOR PROCESS CONTROL

In the followings a number of Internet technologies are analyzed as possible solutions for process control communication.

The first step in using the Internet technologies in process control is to implement the TCP/IP protocol stack on automation devices. A device is recognized and it can communicate on the Internet only if it has an IP address (real or virtual) and if it understands the TCP/IP protocol. But the TCP/IP protocol stack is rather complex and requires a multi-tasking or at least a multi-threading operating system (UNIX, Linux, Windows). Many simple automation devices have limited resources (memory, processor speed) and don't have an operating system. The question is if the TCP/IP protocol stack can be implemented in these conditions. A number of attempts were made and some interesting solutions are available.

A first approach was to build a simplified version of the TCP/IP stack on an MS-DOS kernel. The proposed application forces limitations on the IP packet length (max. 1024 bytes), on the number of buffered packets and it simulates a multi-tasking environment with a periodical pooling of the higher level services (Telnet, Ping and FTP). The servers accept only one client at a time.

A more interesting solution is proposed in [9], where the idea of embedded Web server is promoted. The author builds the whole protocol stack (including the Web server) from scratch, without the use of any operating system or kernel functions. The application, written in "C", can be ported on X86 or PIC17/18 embedded platforms. Special programming techniques were needed to generate a minimal code that fits in the very limited memory space of a micro-controller (e.g. PIC17/18). With just a few components (a PIC18 microcontroller, an Ethernet controller and some adapter circuits) and a firmware written in the internal memory of the microcontroller a Web server is built. The Web server can show a limited number of predefined HTML pages (1-3 pages) that contains the state of some input signals. This prototype architecture may be used as the intelligent part of an automation device. In this way, a process device can be seen from a remote place using a common Web browser.

The use of HTTP protocol for process control purposes is still a controversial solution. The main advantage of this approach is that no specialized tools are needed for remote access, any web browser may be used for this purpose. Also, one of the most common ways of getting information from the Internet is through Web pages. But the HTTP protocol does not offer the security and reliability level required in a control application; it has no mechanisms to control the transmission time, and transmission errors or failures are solved in a too primitive way. Therefore this protocol, in its basic form, is used mainly for visualization purposes. Probably some extensions should be added to the HTTP protocol to solve these problems. These extensions may be embedded in the text of some scripts.

Another drawback of the HTTP protocol is that information is transferred in a single direction: from the server to the client. But in control applications a bi-directional data transfer is needed. The user should have the possibility to change the state or value of some process variables and also to activate or stop some remote tasks. Commands should be transmitted through the web page from the client part. This problem can be solved in two ways: by using the servlet technology, or by using the CGI (Common Gateway Interface).

The servlet technology allows to fill-in some forms at the client-end, and to process the data contained in the form, at the server-end. This mechanism can be used to transfer commands, to change the value of some output variables, or to activate some functions in the server. The data is transferred only at an explicit request (when the Submit button is pushed). An applet may be used to automatically send and request periodical data.

The CGI technology is a more flexible way of transmitting information through web pages. It allows dynamic web page generation, based on clients' requests and local variables' state. When a location (URL) is accessed a program is executed. The program parses the request message and generates a web page in a HTTL format. The page shows the state of the process variables. The program can modify the output variables' values.

For complex distributed applications, a multiple service-based approach is recommended. According to this model, a number of servers fulfill some services useful for process control. The following services are considered necessary:

- global time service – responsible for time synchronization and ordering
- event management service – responsible for event detection, registration and notification
- configuration management – responsible for detecting and maintaining a specified system configuration
- replication and fault tolerance service – assures the required reliability level
- periodic data acquisition and control generation service – delivers the input and output data for the control loops

These services represent the middle tire in a three-tire architecture. They offer a transparent access of the application tire to the physical and logical resources of the system.

4. SOME EXPERIMENTAL RESULTS

Most of the communication models and technologies presented in the previous chapters were implemented and tested by the authors in different control applications.

In the project "ASI-Master Interface Implementation" a number of issues concerning the implementation of a low-level industrial network protocol were analyzed and solved. The most critical part of the project was the fulfillment of tight time requirements. Special programming techniques and execution models were used for this purpose [8]. The communication functions are partitioned on three hierarchical levels, which are executed in parallel.

Another interesting experiment was the developed of a special communication protocol for a medical supervisory system [10]. The protocol allows the acquisition of medical data (EKG signals, blood pressure, oxygen concentration, body temperature, etc.) from a set of patient monitoring devices. A special attention was given to alarm detection and timely transmission. A time division multiple access protocol was used for a better control of message timeliness.

For complex distributed control systems a service-based framework was developed [6]. The following services were implemented: event management service, time service and transparent access service. This work is still in progress, a number of new services being in the design phase. The result of this research work will be a set of middle-ware services useful for a rapid prototyping and development of complex process control applications.

The authors experimented some dedicated control applications in which communication plays an important part. A track surveillance system was developed based on the Global Positioning System (GPS) and on the Mobile telephony communication system. A special protocol was used to transmit the geographical position and trace of a set of trucks. Another project (still in progress) tries to solve the remote control of a video camera through an Internet connection. The camera can be rotated around two directions and it can transmit video information from an isolated place. The communication protocol must assure a constant video-data stream on a network with variable load.

These experiments showed that communication issues are an important part of a modern control application and new communication protocols and models are needed to solve the specific requirements of this field.

5. CONCLUSIONS

Distributed control applications require more deterministic, reliable and timely communication means and tools. For safety or time critical applications a special class or industrial networks was developed. For less critical applications a set of general-purpose protocols and technologies can be successfully adapted. Of special interest are the new Internet technologies, which open new perspectives in the design of complex distributed control systems.

This article presents some issues and possible solutions to these problems. The ideas presented in this article are based on a number of research projects and experimental results and obtained by the authors in this particular field.

REFERENCES

1. Cardeira C. , Simonot-Lion F, Bayard M. [1995], “ Intelligent Field Devices and Field Buses: Impact on Applications Design Methodology”, *Studies in Informatics and Control* Vol 4 No. 3, pp 255-262
2. Fredriksson L.B. [1994], “Controller Area Networks and the Protocol CAN for Machine Control Systems”, *Mechatronics* , Vol. 4 No.2 pp 159-192,
3. Koubias S, Papadopoulos G., [1995], “Modern Fieldbus Communication Architectures for Real-Time Industrial Applications”, *Computers in Industry journal (ELSEVIER)* Vol. 26 pp. 243-252
4. Pasadas R. , Almeida L. Fonseca J. [1997], “ A Proposal to Improve Flexibility in Real-Time Fieldbus Networks “ IFAC SICICA ‘97 , *3rd IFAC Symposium on Intelligent Components and Instruments for Control Application* , Annesy, Franta
5. Sebestyen G. [1999], “ Industrial Communication Networks – Modeling and Simulation”, Technical Report, UTCN, pp. 1-55
6. Ghe. Sebestyen, G. Buzas “Distributed Services in Control Systems” *OSPMA-FieldComms99 Conference, Open Solutions for Process and Manufacturing Automation*, Telford, UK, 1999
7. Ghe. Sebestyen, G. Buzas “Internet Technologies for Remote Process Control and Supervision” *OSPMA-FieldComms99 Conference, Open Solutions for Process and Manufacturing Automation*, London, UK, 2000
8. Kopecz H. s.a. “Distributed Fault-Tolerant Real-Time System: The Mars Approach”, *Technical Report on the MARS project*, 1989
9. P. Smith, "Embedded Web server design", Application notes, Microchip's web page (www.microchip.com)
10. Ghe. Sebestyen, A. Peculea, L. Toderean, [1998] A Real-Time Medical Supervisory System, *Proc. A&Q'98 International Conference on Automation and Quality Control*, Cluj-Napoca, p A154-160