

Fieldbus Technology in Industrial Automation

JEAN-PIERRE THOMESSE

Invited Paper

Fieldbus technology in industrial automation is not only relatively complex because of the number of solutions possible, but also, and above all, because of the variety of applications. Ironically, these in turn are responsible for the multitude of solutions available. If the analysis of the basic needs is relatively standard, as they will always involve connecting sensors, actuators, and field controllers with each other, the options in architecture are numerous and can impose the need for certain services. The required performances themselves and the QoS expected fundamentally depend on the applications.

This paper traces this technology from its beginnings, which go back to the first industrial networks in the 1970s. The principal stages of development are recounted, from the initial requirement specifications to the current state of international standardization. The diverse technical solutions are then analyzed and classified. In particular, we study the temporal aspects, the medium access control protocols, and application relationships.

Keywords—Application relationships, architecture, client-server, cooperation models, fieldbus history, medium access control (MAC), protocol classification, publisher-consumer, real time, standardization.

I. INTRODUCTION

For about 20 years now, the word “fieldbus” has been very widely used. Its common meaning is *a network for connecting field devices such as sensors, actuators, field controllers such as PLCs, regulators, drive controllers, etc., and man-machine interfaces*. But this is only an informal definition and needs more in-depth analysis. Fieldbus technology represents a wide domain of problems which are similar, but not exactly identical in nature. Fieldbus technology involves a variety of solutions and techniques which, although frequently seen as closely related, are different from each other. Fieldbus technology is a kind of technical, political, and human adventure, which for more than 20 years has led to a lot of papers in journals, a lot of announce-

ments, a lot of so-called scoops, a lot of conferences and workshops, and a lot of products and standards.

There are many standards. Why?

- The need for a fieldbus technology identified by a number of different end-user companies in a number of different sectors.
- The variety of possible hosts to be connected (the variety of sensors, of actuators, of controllers).

Initially, there was no existing standard so each information technology (IT) provider developed their own solutions in a given sector. These companies realized the strategic importance of the fieldbus in the industrial automation systems. Once their products were developed, they pushed to have them standardized, and hence we have the current vast range of standards, *de facto* and *de jure*.

This paper will try to establish the origins and current status of the fieldbus technology, including technical and scientific analysis, and standardization. Taking into account the number of systems and the number of contenders, the history of the fieldbus is long and the different episodes are numerous. Because of this, a lot of details will not be included, and we shall simply focus on the essential steps in its evolution.

There are two main parts to this paper. The first focuses on the origin of the concept of the fieldbus and on the requirements which led to the beginning of fieldbuses, ending with the current state of standardization. The second part is dedicated to the technical aspects, the services, the protocols, and then the QoS, ending with some communication architecture considerations. The conclusion will focus on perspectives and future possible evolutions.

II. FIELDBUS CONCEPTS: ORIGIN AND REQUIREMENTS

A. How the Fieldbus Concept Began

The fieldbus concept has several origins, but they can be classified into two main groups: the end-user needs and the technological capabilities.

1) Functional End-Users Needs:

a) *Needs for Standardization:* It is appropriate to start with the history of fieldbuses, in order to understand the

Manuscript received April 13, 2004; revised February 14, 2005.

The author is with the Laboratoire LORIA, Institut National Polytechnique de Lorraine, 54516 Vandoeuvre les Nancy, France (e-mail: thomesse@loria.fr).

Digital Object Identifier 10.1109/JPROC.2005.849724

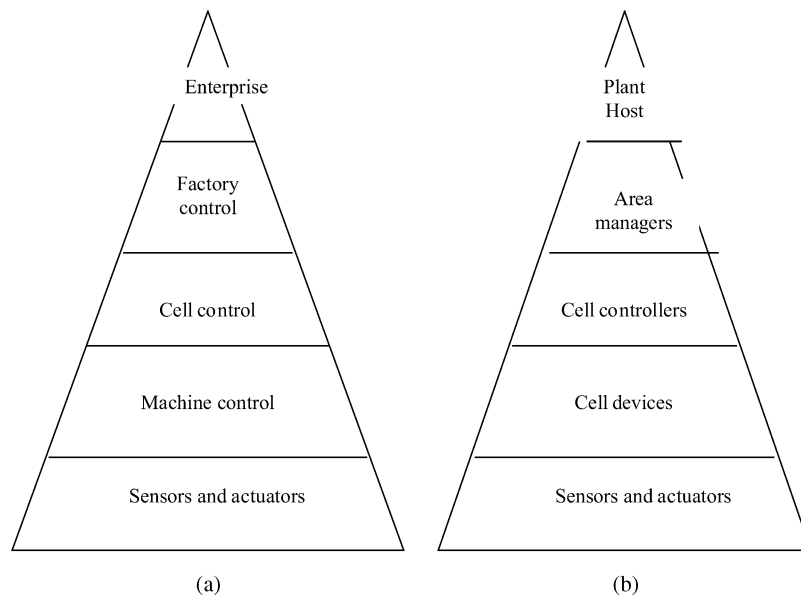


Fig. 1. CIM architecture issued from [13]. (a) Hierarchical architecture. (b) INI-MAP architecture.

differences in approaches. As with most histories, this one has a prehistory. The fieldbus ancestors which stood out the most in industrial automation were Modbus from Modicon (Modicon Bus) [62], [63] and the Westinghouse Distributed Processing Family (WDPF) from Westinghouse [112], [140] because of their seniority, their functionalities, and their worldwide acceptance. Other networks were already in existence, but did not go beyond a few specific applications or domains. For example, the Alliance Research Centre Network (ARCNET) started in 1977 and primarily covered office communication needs [114], before being used in data acquisition [19]. Another network which was highly used in avionics and aerospace applications was the Military Standard 1553 [56], [133]. In the nuclear instrumentation domain, the CAMAC [28] network, created in the 1960s, is considered as the first instrumentation network. Several proprietary networks were in use at the end of the 1970s to connect programmable logic controllers (PLCs) (Allen Bradley Data Highway and Tiway-Texas Instrument Way), [126] as well as in the process industry [136], [158].

b) Manufacturing Automation Protocol (MAP) and Technical and Office Protocol (TOP) Projects: The integration of heterogeneous systems was difficult due to the lack of standards and was expensive on account of necessary gateways, adaptors, and protocol converters. It was at this moment that two U.S. companies started two projects, the aim of which was “the definition of a standard communication profile.” Boeing Company launched the TOP project [26], [36], and General Motors the MAP project [41], [48], [79], [145]. The main objective of the MAP project was the definition of a standard communication profile suited for communication between the design offices and factories and, inside the factory, between workshops and machine tools or robots. General Motors wanted a communication profile for manufacturing applications that could allow all devices to communicate without having to develop specific hardware

and/or software. Boeing’s idea with the TOP project was similar but concerned another issue, namely, communication between business and technical offices.

c) The Concept of Computer Integrated Manufacturing (CIM) and Hierarchical Architecture: CIM refers to automated manufacturing, automated transport of pieces and materials, using computer technologies at all the stages of a product from its design to the manufacture and the quality control. The idea of structuring applications in a hierarchical way by abstracting levels has been used for decades to simplify their design. Fig. 1 shows the five levels model of application architecture defined by the National Bureau of Standards (NBS) in the United States [13], [79]. These models were initially functional, meaning that the main interest lay in the function organization, but not in how it was implemented. They were based on the structure of discrete part manufacturing factories. It was later that they were also used as implementation models (or operational models) (as in Fig. 2).

Each network governs the functions of the layer below and serves as an interface for the layer above. This is how a MAP network in a factory works. All the controllers of cells or of a workshop are connected to the MAP backbone. But each of these controllers is connected to a mini-MAP network which interconnects the machines in the cell. And each machine can use one or several fieldbuses which interconnects the instrumentation to the machine controllers. Notice that the fieldbus and the mini-MAP locations and roles are more or less similar.

In Fig. 2, a TOP network is situated between the *enterprise* and the *factory control* levels, a MAP network is between the latter and the *cell control* level, mini-MAP or sometimes process data highway (Proway) [52], [81] just below that, and then finally the fieldbus network between the *machine control* and the *sensor-actuator* levels, leading to the operational architecture. Mini-MAP [58] or MAP/enhanced per-

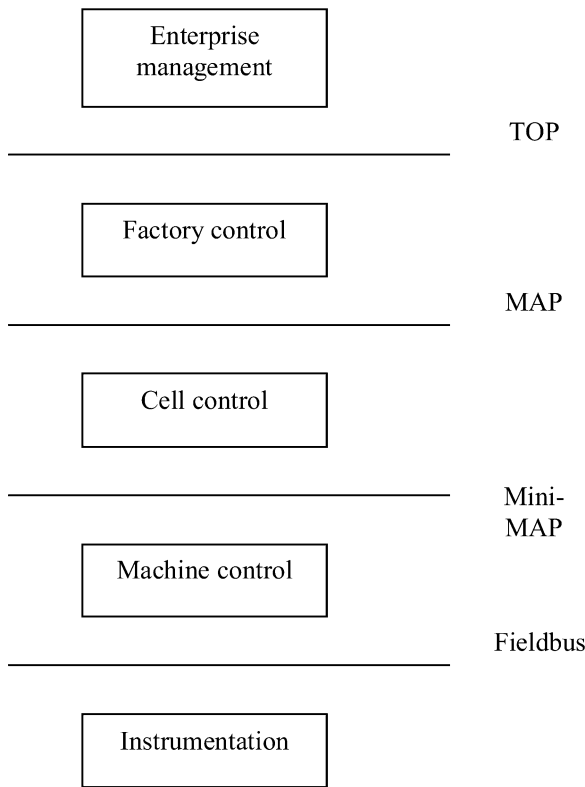


Fig. 2. Operational architecture.

formances architecture (MAP/EPA) [108], [125] was added, based on the factory automation interconnect system (FAIS) specification [115] developed in Japan.

This notion of hierarchical architecture was also developed in process industries, [11], [159] but with a difference in functions. Indeed, the following layers were most often considered: a first layer for reflex automation, a second for the supervision, and a third for optimization.

d) Wiring Simplification Needs: At the lowest level of communication, before the fieldbus era, a lot of standards reigned, for example, the 4–20 mA standard for analog sensors or the 0–24 V for digital inputs, etc. These standards led to a cabling of two wires for each analog point and for each Boolean point (true, false), or each binary digit in a number. The result was the need for a great number of cables in the factories. The design and installation of the wiring were expensive operations, and maintenance or evolution was difficult. This was one of the reasons why end users requested a solution for simplifying these operations: the fieldbus was an answer to this request. This need had already been stated in 1971 [80].

2) Technological Capabilities (Enabling Technologies):

a) OSI-ISO Model: In 1978, work on the communication reference architecture model started in the International Organization for Standardization (ISO), and was to become the Open System Interconnection model (OSI) that we all know today [86], [168]. This model, originally conceived for computer interconnection, brought the right concepts for the understanding of data communication, for the design and standardization of new communication protocols. The 1980s

were to be a very rich period in creativity and innovation in the field of services as well as in protocols.

The OSI model was sometimes poorly understood; we shall come back to this point later in the second part of the paper (in the first section dedicated to the OSI model).

b) LANs and MAC Protocols: In LANs, the stations shared the same transmission support. Logically, without some kind of intervention, all stations would transmit simultaneously. For one station to send at a time, it was necessary to develop medium access control (MAC) protocols. Some of these protocols were called deterministic, i.e., a transmission could occur within a bounded delay. The other protocols, which did not have this property, were called nondeterministic. A deterministic MAC protocol based on the token mechanism [88] was chosen by the MAP project. The TOP project chose a nondeterministic protocol called Ethernet [Carrier Sense Multiple Access-Collision Detection (CSMA-CD)] [87].

Because Ethernet lacked the ability to guarantee latency delay, research for making Ethernet deterministic led to the protocols known as CSMA-Bitwise Arbitration (CSMA-BA), CSMA-Collision Avoidance (CSMA-CA), [16], [91], and CSMA-DCR (CSMA-Deterministic Collision Resolution) [101].

With all these varieties of MAC protocols, LANs exploded. It was attractive to specify one's own protocol, well suited to one's need. The trend was facilitated by the progress made in microelectronics, and design automation.

From another point of view, the LAN technology gave an opportunity to a lot of users to experiment with the distribution of applications. It was a great temptation to experiment with distributing functions on microcomputers, and testing their cooperation through a network. It was the moment in the evolution of industrial applications that the digital control system (DCS) or the direct digital control (DDC) migrated to the distributed control system (DCS) [110], [131], ultimately leading to the systems used today.

c) Microelectronics and Integrated Circuits: The 1970s and the 1980s saw the development of microelectronics, of semicustom and full-custom integrated circuits, the development of microcontrollers, and of digital signal processing (DSP). These were the state-of-the-art technologies that made it possible to design new communication controllers. The first Inter Integrated Circuits (I²C) network was created in 1982 by Philips for the interconnection of ICs in television sets [124]. However, the perspective was not only the integration of protocols into silicon, but also the capability to put "intelligence" inside the smallest device, inside any sensor, or actuator. This digital treatment capability found in each sensor and actuator necessitated new communication means [59]. This was another reason for the development of the fieldbus, and was stated in a report from Prof. Soutif of Grenoble University, Grenoble, France [143] and during a dedicated colloquium in the United Kingdom [65].

3) Conclusion: All the elements were in place for entering into the fieldbus saga. The discussion at that time centered on sensor and actuator networks or instrumentation

networks. The term “fieldbus” had not yet been coined. It would appear only in 1985 at an IEC meeting.

The needs were many and the provider companies recognized great potential in this emerging market. Perhaps the most important reason for fieldbus development was the awareness that it could become the backbone of the future distributed and real-time systems for automation (and then the *bone of contention* for the competition between automation companies). Thus, the specification and the design of numerous fieldbuses began. An initial experiment of a digital fieldbus (1981) was carried out by Brown Boveri Company and Electricité de France with the KSU network at the Thémis [53] solar power plant in the south of France.

In parallel with this innovative design work was the real start of protocol engineering activity, formalization of protocols in terms of automata, Petri nets, etc., and proofs of property, development of languages for specification (ESTELLE, LOTOS, SDL), conformance testing methods, conformance testing procedures and institutions, and arrangements and recognition between national organizations.

B. Development of Fieldbus

In the beginning of the 1980s, several projects started in Europe after the MAP project had began in the United States. In France, the FIP fieldbus project saw light in 1982 under the aegis of the French Ministry for Research and Industry. It is a similar process which led to the Process Field Bus (PROFIBUS [9], [37]) fieldbus project in Germany in 1984, and to the P-Net [40] project in Denmark in 1983. At the same time, in 1983, the Bosch Company developed the specifications of the Controller Area Network (CAN) for cars manufactured in Germany [16], [91], [123]. FIP [151] stands for Factory Instrumentation Protocol and is now known as WorldFIP.

The standardization process began at this time in these different countries and at an international level, with IEC TC 65/SC65C/WG6 [51], simultaneously with the Instrumentation Society of America (ISA) in the United States (in the ISA SP50 (ISA-Standard Practice)).

This beginning shows, with the number of fieldbuses now, that ideas, old or new, were not and are not lacking.

The contenders for the IEC international standard at the early beginning were classified into two subgroups: the first group included solutions based on existing protocols; the second group included only new paper proposals without experiments. Some details on these proposals can be found in several publications [15], [164]–[167]. Two fieldbus types were to be considered, the H1 fieldbus at a low data rate for the connection of some sensors essentially in process control, and the H2 fieldbus at high data rate for manufacturing or for interconnection of several H1 networks.

1) First Group:

a) *ERA Technology*: The U.K. company ERA Technology proposed a fieldbus based on the existing Mil Std 1553B. The proposal extended the current standard for physical performances:

- 1900 m at a data rate of 62.5 kb/s, 750 m at the data rate of 250 kb/s, and 350 m at the data rate of 500 kb/s;
- changes to specifications for spur isolation resistors;
- optional addition of power;
- 32 nodes possible with power and active repeaters.

b) *IEEE P1118*: A U.S. group proposed a fieldbus based on the P1118 project (based on Bitbus [10] from Intel) dedicated to exchanges between microcontrollers for all types of applications. The specifications covered the physical, the data link, and the application layers.

- *Physical layer*: The covered distance was from 2000 to 5000 m with data rates between 50 and 500 kbit/s, with 250-V isolation, optional intrinsic safety, and power with signal. The proposed medium was a twisted pair with possible redundancy.
- *Data link layer*: A master/slave protocol with an optional backup master was required. In case of failure with initial master, the backup master assured availability.
- *Application layer*: Different types of messages and services were specified (broadcast and multicast, datagram, acknowledged datagram, connection oriented) with a response time between 10 and 50 ms, and a minimum of 1 ms, to ensure the physical procedures. The fieldbus had to be optimized for small frames (128 b), with downloading and task control, management tools for device status. The P1118 proposal scope was underlined for distributed intelligent devices in all industries.

c) *Foxboro Proposal*: The Foxboro company presented two complementary solutions for the H1 applications, using enhanced high-level data link control (HDLC) with Manchester encoding and baseband communication and HDLC with nonreturn to zero inverted (NRZI) encoding, and RS 485 for H2 networks. The enhancements of HDLC were related to the error detection mechanism.

The number of undetected errors were to be less than one such error in 40 years [6] at a data rate of 1 Mb/s.

d) *Rosemount Proposal*: Rosemount Inc. presented two solutions, one for the H1 bus using IEEE 802.4 with frequency shift keying (FSK) phase coherent, and one for H2 using IEEE 802.4 with FSK phase continuous. Rosemount started the development of the Hart system in 1985.

2) *Second Group*: Two European proposals, FIP and PROFIBUS, were only paper proposals at this time.

a) *FIP*: The FIP requirements, published in 1984, were developed by a group of end users and labs. The requirements focused on the application needs, periodic updates of data, independence of addresses and locations, coherence and consistency of data. The FIP Club was created in 1986 for the promotion of these specifications.

b) *PROFIBUS*: The PROFIBUS Fieldbus Message Specification (FMS) fieldbus was already a technical solution: a character-based transmission, according to the RS 485 standard, with a token passing method distinguishing master and slave stations. Only the master stations were included in the virtual ring. FMS was a subset of the Manufacturing Message Specification (MMS) [89].

3) Organization Positions:

a) *IEC TC65C/WG6*: The technical committee IEC TC65C/WG6, after having defined Proway (IEC 955) [52] “Inter subsystem communications for industrial process,” was in charge of the fieldbus standardization after a meeting in Montreal, QC, Canada, in May 1985.

b) *ISA SP50*: The American position was given by ISA SP50. ANSI entrusted ISA for the definition of the American fieldbus standard. The position was that it was not necessary to develop a specific American standard, as they had to cooperate directly with the IEC committee for a unique international standard. After a “call for proposals,” the diversity of the protocols proposed made any convergence difficult, if not impossible. The group SP50 of ISA then defined their requirements, which were not very different from those of IEC, and tried to find a common solution among the proposals (Rosemount, PROFIBUS, FIP, ERA, Foxboro).

c) *NEMA*: In the United States, the National Electrical Manufacturers Association (NEMA) created a task force (SC21) which worked with ISA SP50 and IEC to determine a single American and international standard.

4) *Conclusion*: The proposals were very different in terms of requirements and solutions. Many concerned the physical layer, and related aspects such as connectivity, topology, and distances, without any deep investigation into the functional application aspects. All this was something new. For the first time ISA and IEC had to consider:

- on the one hand, existing products;
- on the other, paper proposals, largely based on different views of what a fieldbus should be.

The decision was to write requirements in order to define a fieldbus. The contenders tried to push for the requirement(s) corresponding to their solutions. These requirements are presented in the following section.

C. Fieldbus Requirements

The establishment of fieldbus requirements started the standardization by both the IEC and ISA committees. Before examining the different proposals, it was decided to first express the requirements before choosing or defining a standard solution. ISA SP50 gave a questionnaire to all the members to try to state the real needs of the user. ISA and IEC committees started writing the requirements in 1986.

Without going into details, a very deductive approach was advocated and described in an ISA document entitled “Field Instrument Bus Standard Specification” [82], [83]. But it was difficult to follow the stages described in this document strictly because of the members’s various levels of progress. Some were working on the needs analysis, others on the protocol specifications, and still others on the first implementations. However, it was only in February 1987 that the first version of the final requirements was drafted. For one year, new needs or requirements were proposed at each meeting. But work on the definition of a solution started in Spring 1987 [164], [165]. Therefore, we can see an evolution in the requirements from end users’ needs at

the beginning moving to more and more technical aspects in the later versions.

This next section presents, first, the questionnaire by ISA SP50, then a table summarizing the requirements issued from IEC and ISA. Some requirements from the FIP proposal are included [50] because of their specificity in this arena; an example of operational architecture issued from [166] is also given.

1) *ISA Questionnaire*: ISA published a 15-page “Discussion draft and questionnaire for functional requirements” [82]. This document discussed the requirements for a “low level” industrial fieldbus that connected field devices to higher level monitoring and control systems. Some of the following features were used to distinguish a “low level” field bus from a “high level” bus system such as Proway or MAP. It was structured in four chapters: “Benefits of fieldbuses,” “Describing field devices,” “Information flows,” and “Application environment.”

Seven benefits were identified, and each was to be qualified according to its importance from greatest to least. The benefits were:

- lowering the installation costs;
- ease of adding field devices;
- providing two-way communication with field devices;
- improving the accuracy of information delivered at control room;
- enhancing the maintainability of field devices;
- providing remote access to measurement data through handheld interface;
- more advanced control strategies can be implemented because of improved field data.

The description of the devices consisted essentially of (for each type of sensor):

- the maximum message response time (time between request and delivery of information);
- the message frequency (in average).

The “information flows” part dealt with

- the design philosophies (grouping of devices on a bus based on functional analysis, on geography, etc.);
- the bus control and the exchanges (master/slave; peer-to-peer, etc.);
- the address allocation;
- the fieldbus topology (with distinction of lengths between master and junction box and between junction box and slaves);
- the fieldbus size in number of stations;
- the redundancy possibility.

The application environment analyzed the power requirements, the type of wires, the insulation requirements, and the capability to support flammable atmospheres.

As can be seen, the questions were very end user oriented at this early stage; the environment and management were the two key points of the questionnaire. Technical communication aspects were not dealt with, except on a few points such as the notion of masters and slaves and bidirectional

communications. It was implicit that the fieldbus had to provide two services READ and WRITE. Other services were not considered.

The committees were very optimistic. At the end of 1986, it was expected that the functional guidelines would be available in January 1987 and a standard set in June 1989 [84], [85]. The tables of contents of two future documents (Architecture and Overview; Messaging Service) were published in the working groups on 11 December 1986.

2) *Requirements Summary*: Table 1 summarizes the requirements from IEC, ISA, and WorldFIP.

N.B.: The response time is defined as follows:

- for IEC: time delay between event occurrence and signaling;
- for ISA: the time elapsed between a request and the delivery of information.

3) *Architecture and Functional Aspects*: Fig. 3 explains the position of the fieldbus and the types of equipment to be connected. This presentation of distributed architecture was issued from the IEC functional requirements dated July 1986, which can be found in [164]. This standard provides for serial digital communication to and from field devices; it also provides for attaching more than one addressable field device on one bus. The general requirements were introduced as follows:

The fieldbus will be a serial digital communication standards which can replace present signaling techniques such as 20 mA and 24 VDC, so that more information can flow in both directions between intelligent field devices and the higher level control system over shared communication medium.

Just one fieldbus is needed to allow multipoint attachments for a number of addressable devices. The most common justifications for this design are:

- better quality and quantity of information flow;
- save cable and installation cost;
- ease of adding or deleting field devices in a system;
- fewer connections to devices mounted on moving equipment;
- fewer penetrations through process containment walls;
- save cable and installation weight;
- reduce installation errors;
- reduce terminal and junction boxes.

4) *Conclusion*: This conclusion resumes the official requirements from IEC-ISA and adds some comments related to real-time networking. The requirements stated that two fieldbuses were needed, H1 and H2. Even if they presented some similar functionalities, they differed in speed, distances between stations, number of stations, and services to the user. We can see the needs expressed on the one hand for the low-speed process control applications (H1), and on the other hand for high-speed process applications (H2) (in discrete part manufacturing or in certain process control). The former required a robust physical layer, with a powered bus, with intrinsic safety, and possible reuse of the existing wiring, but in terms of services, it needed READ and WRITE

services, without particular synchronization needs. The latter expressed more requirements in terms of synchronization and distributed control.

Both concluded that fieldbus traffic was either periodic or aperiodic, that it was composed of data and of messages. The data was coming from or going to the final elements in the devices; it was transmitted with status. The messages contained other information.

Regarding the services provided to the user, the requirements cited the services for the exchange of values (READ, WRITE, Information Report, or Notification) and for synchronization. Time stamping was required, but the concepts of consistency or of coherence were not really recognized as necessary. The concept of response time, even if cited and quantified, was not really studied. A rapid classification distinguished process control and manufacturing.

Regarding the mechanisms relevant to MAC and LLC, the question seemed to be eluded; they were to become the major point of discussion and the stumbling block throughout the following years.

The fieldbus was not yet considered as a real-time network but as relative to MAP and Mini-MAP [6]. This was reinforced by the position of PROFIBUS FMS, which appeared at this moment more as a mini-MAP network than a fieldbus.

It was after the appearance of the fieldbus in other applications that the concept of a real-time network would be considered at the international level for standardization. It was after the publication of real-time communication needs by the European MAPs users's group [45], [127] that the work group ISO TC 184, SC5, WG6 TCCA was created to study real-time communication independently of the network's position in the system's architecture.

D. Application Domains

At the beginning of the fieldbus era, only two main domains of application (process control and discrete manufacturing) were considered by the standardization bodies (IEC and national organizations). We have seen that the requirements were quite similar in both cases. And now, as fieldbus technology has penetrated all application domains, it is interesting to observe their similar needs, even *a posteriori*. This section analyzes the different application domains in order to show that the variety of fieldbuses existing today are also a result of these different requirements.

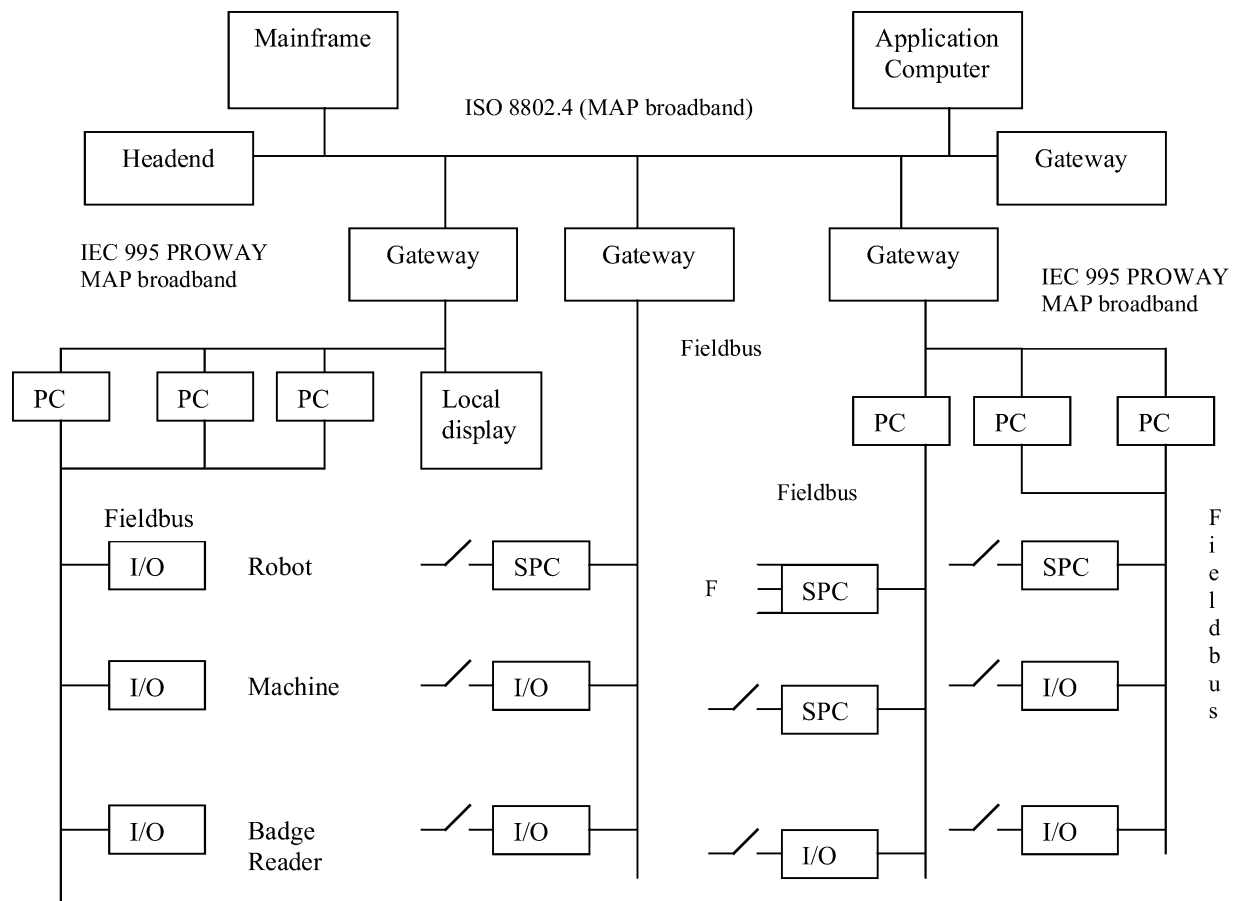
The applications can be seen as a set of criteria for classification of industrial LANs and, more especially, fieldbuses. The criteria are related to:

- the types of traffic, which influence the services and the required QoS (real-time constraints, synchronization needs, etc.);
- the environment characteristics (electromagnetic compatibility (EMC), intrinsic safety, power);
- the dependability constraints (availability, reliability, safety, security, etc.).

We shall not analyze the environment and dependability characteristics because they are too dependent upon the application itself, and dependent even on the location of the ap-

Table 1
Principal Requirements

	IEC	ISA	WorldFIP
Application	APDU format For process <ul style="list-style-type: none"> • Measurement • Alarm setting • Status • Final element current status For Installation <ul style="list-style-type: none"> • Tag number • Additional information for maintenance • Manufacturer's data 	APDU formats Different types of data <ul style="list-style-type: none"> • From binary (16 bits) to floating points (32 bits) and large blocks optional Exchange of data with status Unsolicited messages Configuration data Status of devices Control of access with password	Functional requirements <ul style="list-style-type: none"> • Two types of application exchanges : identified data and messages • Full logical connectivity • Any function on any processor can access all information
Application architecture			Distributed or centralized
Coherences			Synchronous or asynchronous processes Time coherence of data Time coherence of actions Space consistency of copies
Insulation	Galvanic insulation at 500V	Galvanic insulation at 250V	
Power	Powered and non powered fieldbuses	Power with signal optional downstream from the junction box	Wire-powered stations
Flammable atmosphere	Special requirements for connection in hazardous environments	Optional Intrinsic safety	Intrinsic safety
Medium	Industrial grade twisted pair Reuse of existing shielded twisted pair Co-axial or twin axial cabling	Twisted pair and reuse of existing shielded twisted pair	Twisted shielded pair, coaxial or fiber optic Reuse of existing wiring
Topology	<ul style="list-style-type: none"> • Single wiring line with spur wiring from the trunk cable to each station • Single wire link with looping to each station terminal 	A stem with a fan at the end (bird-foot at each junction box)	
Stations management	Addition and removal of sensors and controllers while operating		
Configuration	Configuration operations to be defined in the standard	Configuration operations to be defined in the standard	
Redundancy		Media	Media, connections, components
Performances			
Number of stations	30 stations	5 to 32 for H1 32 for H2	
Length and data rate	350m – 150 messages per second 350 m – 10 000 messages per second 40 m – 5 000 messages per second	Two distances : 750 m and 1850 m H1: 10 messages per device and second	Frequencies required by applications
Response time	5 ms in manufacturing automation 20 ms in process control	H1: 100ms H2: 1 ms	
Integrity of Data		One undetected error every 20 years at the specified data rate	
Addressing	Of a station Of a final element in a station	Maximum 256 addresses, for stations or final elements	Maximum number of final transmission objects : 2^{16}
Protocols	Exchange between stations or between final elements With or without retransmission Access arbitration could be transferred through the bus.	READ/WRITE of application messages to field devices	Single master or multiple masters Broadcast Multicast With or without retransmission



PC = Programmable Computer or Process Controller
 SPC = Slave Process Controller (1 or 2 loops)
 Slave Programmable Controller (5 to 10 steps)
 I/O = Single or small group of Analogue, Digital In or Out
 F = Final element such as Flow Meter, Valve or Relay

Fig. 3. System with MAP/PROWAY and fieldbus (issued second draft—fieldbus standard for use in industrial control systems; from [166]).

plication. Also, the presence of perturbations is not the property of a given domain; for example, a flammable atmosphere may be encountered in several contexts.

The question of real time is common to all domains as well. It is not expressed in the same “units,” but the constraints are potentially the same. This question will be addressed in the next part.

Now, as traffic is always periodic or aperiodic, we shall briefly analyze its characteristics for the following types of applications: discrete manufacturing applications, process control industry including energy production, building automation, control of utilities networks, transportation systems, and embedded systems.

1) *Discrete Manufacturing Applications:* A discrete manufacturing application is characterized by the fact that, between two operations, a product is in a stable state, i.e., it is not damaged if stocked between these operations. This criterion allows for decomposing the application into subapplications relatively independent from a time point of view, with each subapplication being attached to an operation or to a machine.

In such applications, it is then natural to distinguish the communications within a machine from those between machines. In the former, the stations are the sensors, the actuators, the axis controllers, the regulators, and other PLCs. Traffic is essentially periodic and is relevant to the fieldbus. There are needs for broadcasting, for distribution of control algorithms, and for synchronization between application processes.

In the latter, the exchanges are more asynchronous [production orders, report of activity, downloading of programs (or of “domains” in MMS terminology)]. The synchronization between the subapplications is more relevant to production management and productivity criteria than to real-time constraints and process dependability or of product quality criteria. Indeed, such a synchronization failure does not affect the process nor the product. Even if this intermachine traffic may be supported by some fieldbuses, it is more the role of cell networks or factory networks.

The environment depends essentially on the factory domain and may lead to the use of special media adapted to the EMC.

The reliability criterion applies to the fieldbus inside a machine, for the quality of the products and for human (operators) safety. Other criteria, such as availability, are not usually critical but are important for productivity.

2) *Process Control Industries*: The continuous processes are characterized by the fact that the products are continuously produced through a sequence of operations (assumed by different machines) with no stable state between two successive operations. Iron–steel industry processes, many chemical and biological processes, the paper mill industry, and energy production are considered in this category.

Inside a given machine, the traffic is very similar to that described in the previous section. The fieldbus must assume real-time traffic between sensors, controllers, and actuators. But considering the need for synchronization between successive machines, some real-time traffic introduced by the distributed control between the concerned controllers is also supported by the fieldbus.

The characteristics of these processes differ essentially in their time constants, e.g., the speed control in a steel mill and the temperature control in a blast furnace. The lack of a stable state leads to very strict time constraints for the synchronization of operations, e.g., the controller coordination of the sequential elements of a rolling mill. The real-time QoS depends on the criticality of the application.

In terms of environment, continuous processes, especially in the chemical industry, are the domain of intrinsic safety for devices powered by the network. It is also the domain for protection against electromagnetic perturbations (industries with electrical motors). Redundancy is often desired, sometimes necessary, for dependability and safety (people, environment, production tools).

3) *Building Automation*: This type of application concerns the surveillance of houses or buildings, access control, heating and air conditioning, and management of utilities and electric domestic appliances. The applications are more relevant to data acquisition and supervision than to control. The control functions are often very simple.

The range of sensors is very broad. A lot of sensors are ON/OFF (open/close, enable/disable). Others measure the usual physical input variables (temperatures, levels, speeds, etc.). Finally, some are camera based and need image analysis for remote monitoring. The real-time constraints are not numerous. Only some cases, such as burglar alarms or control of elevators and access, are constrained.

As far as the wiring is concerned, it usually represents a significant part of the cost. Therefore, wireless or power line communications are being used more and more in this kind of application. The environmental conditions are not really too demanding, but the great number of devices and controllers lead to very complex systems. All possible topologies must be available for adapting the architecture of the system to any type of building or group of buildings. The dependability is not specified as in industry but it is also very important:

- reliability—an elevator or a heating system cannot fail on Christmas evening;

- availability of communication resources in case of an emergency in a remote health care monitoring;
- safety and security for protection against vandalism or unauthorized people.

4) *Control of Utilities Networks*: These applications consist of the remote monitoring and the control of very large networks for the distribution of water, gas, hot steam, or electricity. They are no longer located in a small area such as a factory or a building, and the networks are no longer really LANs, and yet these applications really are of the same nature as the previous. The functions are the remote monitoring and control of stations (pumping, stocking, transformer stations), pipes, and lines. Operators are only in central control rooms for exploitation and maintenance organization.

Traffic consists of status variables and events as well as the transfer of information between the intermediate stations of fluid/current transport.

The synchronization of data acquisition is often important for establishing the order of events. The data rate depends on the complexity of the system considered.

The networks that convey the data for monitoring and control have the same dependability roles as a fieldbus in a factory. The only difference is in the distances covered; the medium and the physical layer protocol must be adapted to the distance. Power line protocols are used in electrical networks. Optic networks are used in transformer stations. Radio waves are often used to connect very remote stations. It is also now a preferred domain for Internet use.

5) *Transportation Systems*: A transport system is an infrastructure for the transport of people and/or freight. The applications in these systems cover the management of a railway network, the remote control of urban traffic, the monitoring of highways, etc.

Traffic is composed of status variables, events, and device command and control. The topology of the fieldbus depends on the geography of the system considered. The safety constraints are often very important. Dependability is also very crucial, especially availability, even during maintenance or updating operations.

It is possible to include in this category of applications the control and management of telecommunications networks (telephone networks, mobile networks, etc.)

6) *Embedded Systems*: These systems are now in many products, from cars to buses to trains, but also in major electrical domestic appliances (refrigerators, etc.). In vehicles, the application consists of various functions:

- control of the motor(s), of the braking system, of the stability, of the gearbox;
- assistance to the driver, or to automatically pilot the vehicle (as in some trains);
- other functions are related to the energy consumption, such as optimization;
- management of lights, glass-cleaner;
- management of passenger access in trains, ticketing;
- maintenance.

The distances are short; the environment possibly very demanding (in cars, for example); safety is a major constraint.

These applications present time constraints depending on the functions considered. The motor is controlled every 10 ms; the response time of a braking request must be as short as possible.

The term “embedded system” is also used for different equipment such as refrigerators, coffee machines, and washing machines. Each time a piece of equipment is built with “intelligence” and communication capabilities, it can be considered as an embedded system. We also speak of “ambient intelligence” [1] with the expectation of a fully communicating environment and many autonomous intelligent devices in the near future. New problems will occur such as connectivity, safety, confidentiality, integrity, etc.

7) *Synthesis*: This brief study of the application domains shows that the basic fieldbus specifications in terms of functions and of services are very similar in each of the applications.

The exchange of data (values, status, and events) is the main function of the fieldbuses for automatic control, but also for maintenance and management. We shall see that if the requirement is relatively simple, the solutions are numerous (in terms of protocols). Other functions are required but as options or “nice” functions. Synchronization is one of these functions. It is, nevertheless, necessary for the management of distributed systems. The fact that this function was not considered 20 years ago shows that a lot of people did not think that the fieldbus would change the application architecture and design. They thought it would only be a simplification of the wiring. Consequently, the only cited time constraints were the response time and the frequency of the exchanges, which allow for a very simple calculation of the load on a fieldbus and then its proportioning.

The maximum values given for each fieldbus (maximum length, maximum number of stations, maximum data rate, maximum frequency of data update, etc.) are limitations for each application design. Because of these limitations it is sometimes necessary to use several fieldbuses (and other networks) on which the architecture of the application will then depend.

This notion of architecture is not as simple as is usually understood. The word “architecture” is sometimes used for topology; it may be used with the same sense as in the title of the OSI model, and in this case, it then represents an organization of services and protocols. Here, the word “architecture” represents the organization of the automatic control application implemented around a fieldbus and other industrial networks. Architecture is typically defined by diagrams as seen in Fig. 3. The question of architecture is inherent in the requirements for setting up a fieldbus. This was not the question before.

Fig. 3 shows an operational architecture, because it indicates the devices and the fieldbuses actually in operation. But this kind of diagram leaves much to be desired. Indeed, nothing is said about the functions implemented in each station represented by a box, and nothing is said about the cooperation between the functions nor the exchanges supported by the fieldbuses and other networks. Before designing such an architecture, it is necessary to carry out a

functional analysis which must specify the application, the functions and subfunctions, their interactions, and their communications. The result is a functional architecture which ideally may relegate the components and the networks to second place, focusing on the functions. A functional architecture being specified, the designer has to choose the networks, the components, the devices, and the distribution of the functions in the devices. This is the real design stage of the solution, taking into account the constraints, defining which station is connected to which fieldbus, and distributing the functions in the devices.

It is only after this stage that the choice of the architecture takes place, along with the choice of the fieldbus. Now it is the choice of the fieldbus which has an impact on, and sometimes imposes, the choice of the architecture, depending on the services provided by the communications system. It is clear that the existence of certain services determines the distribution facilities.

These architectures will be applied to all domains of application, with a hierarchy from the first-level fieldbus, up to the highest level. Let us take some examples.

In a train, the usual architecture has two levels; each wagon has its own fieldbus, and another fieldbus interconnects them all and has a gateway to the external world.

In a building, an architecture with three levels can be considered; here, each apartment has its own fieldbus; they, in turn, are interconnected by a floor fieldbus, themselves interconnected by a building fieldbus. Other fieldbuses may be associated with the control of the elevators.

In the control of a pipe (be it for gas or water, or some other medium), we could have a fieldbus along the pipe, structured in “segments” of the maximum length for the chosen fieldbus. Each fieldbus of a segment connects all the devices of the segment, and a special site serves as concentrator. The concentrators may be interconnected by another upper level fieldbus or network.

E. Current Standardization

1) *Introduction*: Fieldbus standardization is a subject which has led to a great number of publications for the past twenty years. Regularly, during the 1990s journals published the progress of the standardization process (control and instrumentation, control engineering, measurement and control). Words such as “war,” “battle,” “winner,” “peace” appear in the titles [25], [47], [122]. Optimistic opinions (“a standard will be obtained at the end of the 1980s”) expressed in 1987 all the way up to 1996 [54] decrease after this period. Skepticism and interrogations start to appear [64], [117], [119], [135], [163].

Several updates on the situation have been regularly published; some are listed here in chronological order: [32], [33], [35], [46], [60], [104], [128], [138], [153], [154].

The international standardization concerned essentially the IEC, but also the ISO (TC 184 SC5 WG6), which was in charge of the Time Critical Communication Architecture (TCCA) specification. In Europe, in the middle of the 1990s, the CENELEC decided to define European norms while waiting for an international solution from IEC. Indeed, at the

beginning of the 1990s, different lobbying groups appeared only to disappear: the Open Fieldbus Consortium (OFC), the International Fieldbus Group (IFG), the Interoperable System Project (ISP), etc. There was no compromise and no possible consensus between any two opposing blocks. Even though a group of experts wrote a complete specification for the data link layer [70], [71], including the different concepts and mechanisms, the opposition continued. Because this specification was refused by the minimum minority in 1998, and caused a great problem at the highest IEC level, the Committee of Action of the IEC issued some kind of ultimatum for the working group. The result today is the current content of the IEC 61 158 with eight completely heterogeneous and incompatible fieldbus families.

2) *IEC Standardization*: After a lot of episodes and developments, the IEC 61 158 standard, including a large set of services and protocols, is defined as follows. It is structured by layer, according to the OSI model architecture reduced as mini-MAP, or MAP-EPA, to the physical layer, the data link layer including the MAC and the application layer (cf. Section III-A).

a) *IEC 61 158 Standard*: The main standard is IEC 61 158. The first standard (IEC 61 158-2), published in 1993 [67], defined the physical layer.

The other parts are:

- 61 158-3: data link layer service specification;
- 61 158-4: data link layer protocol specification
- 61 158-5: application layer service specification
- 61 158-6: application layer protocol specification.

These specifications are a collection of different national standards or specifications.

The data link parts (IEC 61 158-3 and 61 158-4) cover eight types listed below:

- Type 1 is the TR1158, the compromise standard proposal refused by a minority of members in 1999, which led to a publication of indignation by Patricio Leviti [104].
- Type 2 is the ControlNet specification.
- Type 3 is the PROFIBUS specification.
- Type 4 is the P-Net specification.
- Type 5 is the FOUNDATION Fieldbus specification.
- Type 6 is the SwiftNet specification.
- Type 7 is the WorldFIP specification.
- Type 8 is the INTERBUS specification.

The application layer specifications (IEC 61 158-5 and 61 158-6) covered ten different types. The first eight are associated with the data link layer. The two others, Type 9 and Type 10, define the FOUNDATION Fieldbus H1 network and PROFINet, respectively.

Two other parts were planned, 61 158-7 for network management and 61 158-8 for conformance testing procedures. They were canceled, because of the existence of proprietary tools for configuration and network management, and for conformance tests of the different types. The maintenance of this standard is now entrusted to SC65C/MT1 (MT stands for Maintenance Team).

b) *IEC 61 784*: A project for a new standard has also been started for the definition of the Communication Profile Families (CPFs) inside the IEC 61 784 standard. Its objective is to clarify the situation created by the number of variants and options in the IEC 61 158 standard. While it defines services and protocols by layer, according to the OSI model reference architecture, the 61 784 standard proposes a specification for a complete stack of protocols based on the previous options. These communication stacks are called profiles.

This standard is composed of two parts; the first, the IEC 61 784-1 standard, is composed of 18 profiles; and the second part, 61 784-2 on Real-Time Ethernet, in progress (work started mid-2003), is composed of nine proposals, all based on Ethernet.

This new project has been entrusted to the new working group IEC SC65C WG11. It is structured as follows.

Structure of IEC 61 784-1:

- The current CPFs are defined in the first part.
- CPF 1 FOUNDATION fieldbus CPF 1/1 H1, CPF 1/2 High-Speed Ethernet (HSE).
- CPF 2 ControlNet CPF 2/1 ControlNet, CPF 2/2 EtherNet/IP.
- CPF 3 PROFIBUS, CPF 3/1 PROFIBUS-DP, CPF 3/2 PROFIBUS-PA, CPF 3/3 PROFINet.
- CPF 4 P-Net, CPF 4/1 P-Net RS 485, CPF 4/2 P-Net RS 232.
- CPF 5 WorldFIP, CPF 5/1 WorldFIP, CPF 5/2 WorldFIP Device WFIP.
- CPF 6 INTERBUS, CPF 6/1 INTERBUS, CPF 6/2 INTERBUS TCP/IP, CPF 6/3 subset.
- CPF 7 SwiftNet, CPF 7/1 SwiftNet transport (without application layer), CPF 7/2 Full stack.

Structure of 61 784-2, under current specification:

- CPF 2 ControlNet;
- CPF 3 PROFIBUS, PROFINet;
- CPF 6 INTERBUS;
- CPF 10 VNET/IP (Virtual Network Protocol);
- CPF 11 TCnet;
- CPF 12 EtherCAT (Ethernet for control automation technology);
- CPF 13 EPL (Ethernet PowerLink)
- CPF 14 EPA (Ethernet for plant automation);
- CPF 15 MODBUS-RTPS (real-time publish-subscribe).

3) *ISO Standardization*: In 1990, a new work item for the ISO TC184 SC5 WG6 Time-Critical Communication Architecture (TCCA) group was started, following the analysis of the MAP experiments to define real-time communication requirements and recommendations [127]. The European MAP user group published a list of requirements for real-time communication. At the same time, the fieldbus appeared as a real-time network [130]. The study of a communication architecture was published as a technical report [57], [90]. Following this work, the network management of time-critical communication systems (TCCS) was also studied [93].

4) *CENELEC Standardization*: Four European standards have been published and updated several times in order to provide international standards where the IEC lacked.

a) *EN 50170 [21]*: The EN 50170 was published in 1996 with three national standards: P-Net from Denmark, PROFIBUS-FMS from Germany, and WorldFIP from France. FOUNDATION Fieldbus was added to EN 50170 as an addendum in 2000, jointly with ControlNet [27] and PROFIBUS-PA [39].

b) *EN 50254 [22], [23]*: The EN 50254 was also published to include fieldbuses with higher performances for the transmission of short frames: INTERBUS, PROFIBUS DP, and Device WorldFIP [3], [22].

c) *EN 50325*: The EN 50325 standard covers the profiles derived from the CAN protocol (and of the ISO 11 898 standard), as DeviceNet, SDS, and CANopen, which are also parts of the IEC 62 026.

d) *EN 50295*: The EN 50295 standard is a standard defining the actuator and sensor interface (AS-i) protocol [7].

F. Conclusion

This first part presented the history of fieldbus and its requirements. They were written between 1984 and 1987, after which choosing a standard was possible. But it was not to be so simple, particularly with the development of other standards by other committees, especially in the ISO, and with the start of other fieldbuses for car automation, building automation, trains, etc. Obviously, the entire story was not to be played out in a single scene.

The concept of architecture must not be forgotten because the ultimate desire of the end user is really not a fieldbus, but an operational architecture which meets his needs in terms of dependability, performance, and cost.

III. FIELDBUS TECHNICAL ASPECTS: SERVICES AND PROTOCOLS

This second part is dedicated to the technical analysis of fieldbuses.

What services are provided by a fieldbus? According to what protocols? According to what communication stack? Looking at the requirements, we can see that some are already structured in terms of OSI layers: more generally, some of them address the services required by the end users (i.e., the application layer in OSI terms), others are related to the physical transmission and coding, and still others express properties or performances.

For 20 years, all papers and contenders (except some such as LonWorks [105]), agreed with the fact that a fieldbus is designed according to a reduced OSI model. But what is the reality? The first section will analyze this model and the particularities of fieldbuses regarding the OSI concepts.

The OSI model gives the structure for analyzing the different technical aspects, from the topology and the cabling to the application services provided to the users. We shall follow this structure for presenting the different choices, the different solutions to fulfill the requirements.

A normal approach would lead to choosing the services from the requirements, and then the protocols from the chosen services and required QoS. But as was already said, the diversity of the applications, the approaches, and the competitors did not allow for simple deductive and objective reasoning.

The requirements being determined, the choices for services were not too broad, but the same cannot be said for the protocols—especially at the MAC layer.

Before going into detail, we shall present the architecture recognized for the fieldbus and analyze the necessary OSI concepts in the first section. Traffic will be analyzed in the second section, before studying the main relationships (or cooperation models) at the application level in the third section. The fourth section is dedicated to the study of MAC protocols, and the fifth to the communication architectures.

A. OSI Model and Fieldbus

The first version of the OSI model was published [86], [168] when the work on fieldbuses started. Most fieldbuses were presented according to a three-layer architecture. It was not new; other architectures had also been considered, for example, the mini-MAP, Factory Automation Interconnection System (FAIS), Collapsed Architectures, and Enhanced Performances Architecture. This reduced architecture came from the MAP Task Force, which claimed that a real-time network must have only the physical, the MAC and the logical link control (LLC), and the application layers. This concept was introduced to reduce the delays observed in the first implementations of the MAP seven layer profile. It was a mistake which led to concluding that to improve performance (to boost communication), it was necessary to reduce the OSI model to a more simplified one.

Let us analyze this point and the particularities of fieldbuses regarding the general-purpose mechanisms and concepts defined by the OSI model.

1) *Fieldbus Architecture Model*: It is common to say that a fieldbus has three layers:

- the physical layer;
- the data link layer, including implicitly the MAC layer;
- the application layer.

What happened with the other layers of the original OSI model? What about the network, transport, session, and presentation layers, as well as another layer that was added, the eighth, or user, layer (to be discussed later)?

a) *Physical Layer and Topologies*: The physical layer is always necessary. All the topologies in fieldbuses are found here; bus, star, ring, tree, and other topologies supporting store and forward transmissions.

b) *Data Link Layer*: The data link layer is also necessary, but we shall see that the problem of transmission errors is not treated the same as in OSI networks. This paper considers that the MAC layer is included in the data link layer, because it is thus in all fieldbus standards. The MAC is obviously necessary and all existing protocols can be used.

c) *Network Layer*: The network layer is not a part of the usual fieldbus architecture model. It was introduced in the

OSI model to integrate the routing function in the topologies allowing for several paths. The network layer is not necessary if only a single path is possible between stations. In most cases, even if the general application architecture is complex, bridges may interconnect the different fieldbuses, and no network protocol is needed.

d) *Transport Layer*: The transport layer was introduced in the OSI model for providing end-to-end control of the exchanges between two end stations, without considering the underlying mechanisms (routing, data link protocol, physical wiring, etc.).

To do this, the emitting transport layer cuts the messages into small packets which are transmitted separately from one point to another until they reach the receiving transport layer. They are then reassembled to reconstitute the initial message and delivered to the application. There is a mechanism to control the proper reception and possible retransmission.

These protocol mechanisms are carried on the messages and are similar to those in the data link layer which are applied to a frame [data link protocol data unit (DL-PDU)].

Regarding the lack of a transport layer in a fieldbus, the end-to-end control is then done at the data link layer. And it may only apply to a frame and not to another PDU. The application protocol data unit (A-PDU) must then be shorter than the longest DL-PDU. Even if most of the fieldbus A-PDUs are short, operations such as program downloading and uploading are then made more complicated, and even impossible. Therefore, functions such as fragmentation and reassembling are sometimes included in the application layer implementation.

e) *Session Layer*: Regarding the session layer, it was introduced in the OSI model to facilitate managing the exchange of very large messages. It does not have a role in most fieldbuses even if some synchronization functions could be considered as relevant to an OSI session concept.

f) *Presentation Layer*: Regarding the presentation layer, its role is not only necessary, but also fundamental, in order to provide a common language of exchange between stations with different internal and local syntaxes. It is often included in the application layer. For a reciprocal understanding of the exchanged information, a comprehensible coding of the A-PDUs from both parties is necessary. With OSI, the ISO standards Abstract Syntax Notation Onez (ASN1) and Basic Encoding Rules (BER) are used but are not efficient for fieldbus. Therefore, other kinds of coding are used [129], often associated with the name of the data exchanged.

g) *Application Layer*: The application layer is obviously necessary. It may be defined according to different models. This layer normally includes the “user layer” presented below, because, as stated in the OSI standard, “the Application layer has no upper interface.”

h) *“User Layer”*: The application layer defines elementary types of objects such as integers or chains of characters. But applications manipulate several types of objects such as speed, temperature, pressure, etc. The need to define these types of objects, in addition to those that exist in application protocols, was felt early on in the first networks. This

is how the “companion standards” were defined. These standards proposed specific objects for each application domain, like robotics, numeric commands, process controls, etc. With fieldbus, these functions are integrated in what is called the “user layer.” We find here, obviously, definitions for types and objects, but also for standardized functions that are called “function blocks” [72], [75]–[77], which correspond to particular treatments of the objects, such as conversion between units, filtering, linearization, etc.

It is called “user layer” in order to express the idea that it is the way by which the user “sees” the fieldbus and communication [106]. This is directly issued from the necessity for the end user to “ignore” the communication techniques. This approach, which came from the MMS works [89], was already recognized by Pimentel [129], and is now the base for defining the Electronic Device Description Language (EDDL) [77] coming from the DDL defined in the Highway Addressable Remote Transducer (HART) Fieldbus [109].

“Profile” is also used to describe the concept of possible options in the protocols of the stack and in the companion standards. For example, we see the “pressure sensor profile,” different “actuator profiles,” etc. This word “profile” has, then, two meanings: one for designating the choices of protocols and protocol options in a real OSI stack implementation, and the other for the integration of the dedicated functions of given devices.

i) *Conclusion*: In conclusion, to say that a fieldbus is always based on a reduced model is a gross misunderstanding. Let us recall that the OSI model is a conceptual, not an implementation, model. A fieldbus usually presents all the functionalities provided by the seven layers of the OSI model. But, in terms of implementations, other choices are possible. For example, the transport functionalities may be implemented within the application and presentation layers.

Furthermore, considering that some protocols could be implemented in different stacks, it was necessary to define some kind of “interface layer” (sometimes called a “glue”) to satisfy the implementation constraints. These “glue layers” may also implement some intermediate layer functionalities. The lower layer interface (LLI) in the PROFIBUS fieldbus and message control services (MCS) in the WorldFIP fieldbus are such examples.

But the OSI model is not just a layered architecture, it is also the definition of several concepts, services and protocols, addressing, Service Access Point, multiplexing, grouping, point-to-point or not, broadcasting, flow control, acknowledgment, etc. Considering these concepts, fieldbuses differ from general-purpose networks. The differences are studied in the next section.

2) *Basic OSI Mechanisms and Fieldbuses*: This section analyzes some OSI concepts highlighted by communication needs.

a) *Point-to-Point, Multipoint, Broadcasting*: All fieldbuses provide point-to-point communication, and some provide broadcasting capabilities. When provided, physical broadcasting is obtained thanks to fieldbus topology through the diffusion of signals. But the data link layer protocols are

all point-to-point and do not take into account the fact that a given frame may have several simultaneous receivers. The problem of reliable broadcasting [24] has never been dealt with in existing fieldbuses. A single protocol (WorldFIP) addresses this problem at the application layer and proposes a mechanism for verifying space consistency. Space consistency is a property, which defines copies of data. It is verified when the copies of data on different stations are equal. This mechanism provides a kind of global acknowledgment (acknowledgment of a set of frames in a single A-PDU) [137].

b) *Connection or Connectionless Protocols:* The connection mechanism was introduced to dynamically manage the resources necessary for communication between two entities. In the case of the fieldbus, as a lot of operations are statically defined, it may be considered that the connections are permanently established at the configuration or commissioning stage. In fieldbuses, operations with or without connections should be possible. But it is important to define multipoint connections for multipoint communications.

c) *Buffer Versus Queues:* This item analyzes how the PDUs to be sent and received are stored in the communication stacks at the sender and receiver sites. Usually, the PDUs to be transmitted are stored in queues at the sender site; they are also stored in queues at the receiver site. They are generally managed according to the first in, first out strategy (FIFO) but another schedule may be obtained through priorities or deadlines. The idea in these classical communication systems is that all the PDUs must be processed.

In fieldbus-based applications, due to periodic traffic, we can drop old PDUs in favor of the most recent. The strategy of storing all the PDUs in queues is not suited to this behavior. Therefore, this data is not stored in queues but in buffers, which always contain the last value produced or received. The reader can find a very good analysis of these mechanisms in [103].

d) *Control of Errors or Status Versus Event:* The detection of errors or the control of the exchanges is either done by the sender, or by the receiver.

When does the sender control the exchanges? It must control the exchanges when they are randomly initiated, or when the message has the semantics of an event. The sender decides the transmission; the receiver is not informed and will only be so at the reception of the message. The sender controls the transmission by waiting for an acknowledgment of the receiver.

When does the receiver control the exchanges? It controls the exchanges when they are regularly initiated; when the message has the semantics of a status, independent of the time-triggered or event-triggered paradigm (see latter subsections for description). The receiver waits for a periodic reception in time-triggered systems, or waits for the response to a request if the exchange has been so initiated. It is the receiver, who is in charge of transmission control.

In fieldbuses, both of these situations are encountered and so, fieldbuses normally have to provide both of communication mechanisms.

e) *Acknowledgment or Not:* Acknowledgments were introduced in protocols so that a receiver informs the trans-

mitter of a message whether or not it has been well received or not. In fieldbus applications, aperiodic exchanges must be correctly received, then acknowledged and possibly repeated. On the other hand, periodic exchanges do not need to be acknowledged. So if there is an error in periodic traffic, the receiver can ignore it and wait for correct data to follow. But it is not sufficient that a message be received without error, it must be received at the right moment. The temporal aspect is important. The management of errors, the recovery strategy, must be placed under the control of the user, i.e., the application processes [45].

f) *Flow Control:* In general purpose networks, flow control is necessary for preventing congestion, for satisfying previous requests, for keeping one's engagements; flow control starts with an admission strategy and test. Flow control is important when traffic changes very quickly.

In the case of fieldbuses, the flow control may be seen as a function of the configuration stage, it is essentially a feasibility study, a test of schedulability. At runtime, flow control is useful for random traffic management.

B. Traffic Classification and Characteristics

1) *Typical Exchanges:* The requirements have specified the types of traffic and their main functional characteristics. They are mainly constituted of input and output variables, what we call state variables, and events. Input-output variables, internal variables and states (as in state-transition models or state control) are considered as *status*. Changes in the status are considered as events.

But traffic may also include some files for downloading device domains, and service requests and responses, especially for the management of application processes and stations.

For some of the transfers, the temporal characteristics are frequency, jitter, lifetime, response time, simultaneity, and the temporal and space coherences or consistencies.

Frequency indicates the rate at which the data is updated, and *jitter* is a variation in the periods; *lifetime* indicates the duration that the data values are significant, and *response time* is the delay between a demand and the result. *Simultaneity* indicates that several operations or events occur at the same time, i.e., in a predefined time interval or time window. When several operations occur in a given time window, they are called *time coherent*.

For other transfers, no such constraints exist, but their required QoS is more related to the absence of errors, to the delivery order and/or to the recovery mechanisms. In other words, the required QoS depends on the traffic considered. Safe and secure transmission is required for file transfers, and respect of time constraints is required in the case of exchanges of status.

Following these requirements, traffic may be considered as composed of two types of information exchanges: *identified data* and *usual messages*, as in all ISO communication systems (Fig. 4). Identified data is all the data known by the control system such as the input issued from the sensors, the commands to the actuators, and so on. They are essentially real time and periodic data. Identified data has

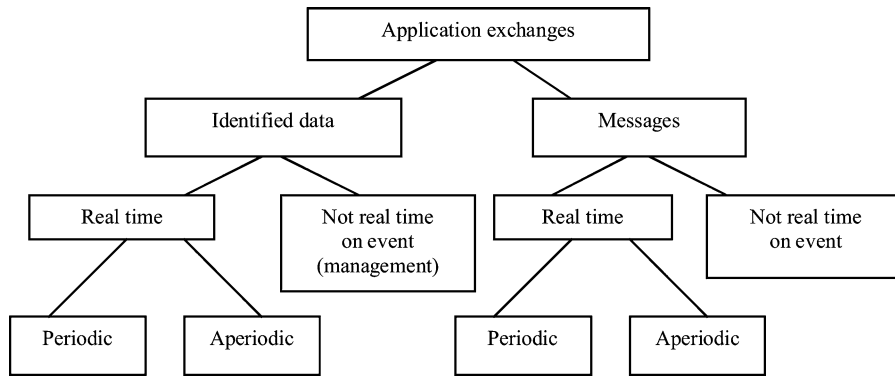


Fig. 4. Traffic classification.

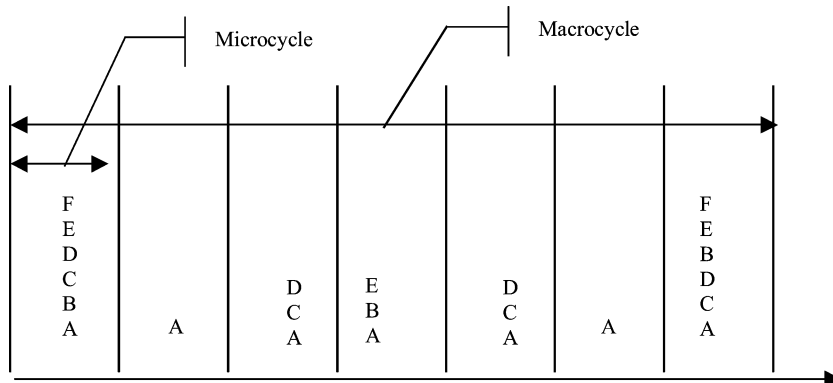


Fig. 5. Example of periodic traffic.

only one producer, but one or more consumers. Rather than producer–consumer, we may say publisher–subscriber (see Section III-C).

These so-called messages are issued from any application process which needs to send something to another one.

Considering the different fieldbuses, this classification is not always so clear. Some provide only exchanges of identified data; some provide only exchanges of messages. This distinction is useful for two different reasons.

- First, because considering the identified data, only the successive values are of interest, and they can often be immediately accessible through the name of the object without having to treat a message in the different layers.
- Second, the values of this data can be stored in buffers and not in queues as with messages. This point will be examined later.

2) *Typical Traffic of Identified Data:* A fieldbus has to transmit essentially the values of data between sensors and controllers, between controllers and actuators, and between controllers themselves. These exchanges, called “identified data traffic,” are known once the application is specified. They may be managed in the client–server as well as in the producer–consumer model or their extensions [29], [44], [148], [162].

a) *Periodic Traffic:* Periodic traffic is induced by the sampled systems theory, which is the basis for automatic control and detection of events. Most identified data is the input

or output of control algorithms. They must often be transmitted periodically. This traffic is deduced from the periodic polling of input in normal centralized systems. The periods of exchanges may be different for each kind of data. A jitter may or may not be accepted. It is clear that the protocols will play a major role in the respect for periodicity without jitter. These systems are based on state traffic and are sometimes called “time-triggered systems.”

Fig. 5 shows a general example of periodic traffic. It shows the updating of A at each elementary period, of C and D every two elementary periods, B and E every three elementary periods, and F every six elementary periods. The macrocycle is the period equal to the *lowest common multiple (LCM)* of the periods. And the microcycle is a time interval equal to the *highest common denominator (HCD)*.

b) *Aperiodic Traffic:* All data may be transmitted cyclically, as is the case in some fieldbuses. But the obtained global traffic may be too great for its nominal data rate. In this case, aperiodic exchanges of some data is more advantageous. Indeed, some state values do not change at a predefined period and may be transmitted only on a change basis.

The *random* or “*on demand*” traffic takes place in the free time slots left by the periodic traffic.

The schedulability of traffic may be analyzed at a configuration stage and online [4], [116], [134], [161].

c) *Time-Triggered or Event-Triggered Systems:* Distinguishing periodic and aperiodic traffic is relevant from two points of view of the application. Kopetz

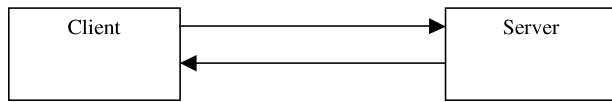


Fig. 6. Client-server model.

compared these approaches [94]. The main comparison criterion is the capability to meet the application time constraints. Most fieldbuses favor a kind of time-triggered system. But some of them combine both approaches, events being managed by a periodic server.

3) *Messages*: We call messages all the exchanges that are not relevant to the previous exchanges of identified data. Messages are exchanged during configuration and maintenance stages. They are used for downloading and uploading. In fieldbuses which do not consider the traffic of identified data, everything is considered a message. When only this traffic is considered, it is then necessary to distinguish the real-time and non-real-time messages.

4) *Conclusion*: All this traffic may be managed in very different ways, with priorities to one type or another, with more or less predictability, and so on. These mechanisms are relevant to the MAC layer and will be studied later (see Section III-D). The service and protocol characteristics will be summarized according to the following points: peer-to-peer versus multipeer or multicast, confirmed (or not) services, acknowledgment or not, connection or connectionless protocols, and flow control.

Time constraints are statically or dynamically specified and managed [17]. The specification of the time constraints may be determined at the connection opening, at the configuration stage, or dynamically at the service request. They are then managed differently by static or dynamic scheduling [14], [134]. Jointly scheduling tasks and messages is still an open problem [18].

C. Cooperation Models, QoS

Cooperation models represent how two or more application entities cooperate to obtain a given objective. Two distinct transaction families can be distinguished: the client-server family and the publisher-subscriber, also known under the name producer-consumer.

1) Basic Models:

a) *Client-Server*: In the client-server model, two entities cooperate (Fig. 6). The server is an entity, which provides a service, i.e., which executes an action on the account of a requester, which is called a *client*. This model is more useful for transmitting state data than event data. Event data detected by a server is only transmitted if the client requests the transfer through a READ service.

Normal client-server: The client-server interactions are broken down into four steps, request, indication, response and confirmation (cf. Fig. 7). An *indication* is an event by the server that indicates the reception of a *request*. And the *confirmation* is the counterpart regarding a *response* by the client. This model is used by all the application protocols, which are more or less derived from MMS. The client-server

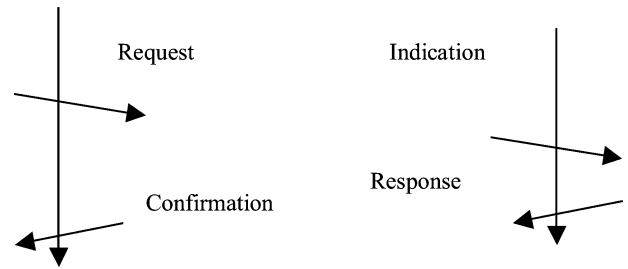


Fig. 7. Usual client-server interactions.

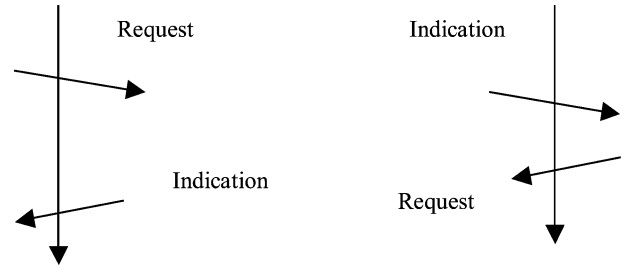


Fig. 8. Unusual client-server interactions.

model is used in INTERBUS, PROFIBUS-FMS and DP, in AS-i, in P-Net, and in WorldFIP.

The semantics of the response may vary from one service to another. For example, the answer may be significant to the request acceptance; it may be significant to the service execution beginning or to the result of the service execution.

In the case of a READ service, the value of the read objects is carried by the response. The request contains the name of the object, and depending on the local addressing mechanism, the means to access the object. The response contains the value or the reason for failure, and when provided, the timeliness attributes. The object may be, *a priori*, a simple variable or a complex structure.

All the application layers which provide services according to the client-server model are more or less built on the MMS model. They propose the management of objects such as tasks (create, kill, start, resume, and stop), variables (read, write), domains (downloading and uploading), and so on. Only one subset of services is generally provided.

In terms of timeliness, the duration of such an operation can be subdivided into three terms: request transfer, action execution, and response transfer. The duration may vary according to the latency time of transfers depending on MAC and on latency time on the server depending on its current load.

Unusual client-server: An “unusual client-server” model is, in fact, composed of two sequences of unconfirmed services. This model (Fig. 8) is also called *client-server*, even if the client is in charge of the association of the indication to the previous request. The same services as in the previous section may be defined, but with only the Request and Indication primitives. The response to a READ indication is a WRITE request. The response to a WRITE indication is also a WRITE request. This model is used, for example, in the BatiBus network [8].

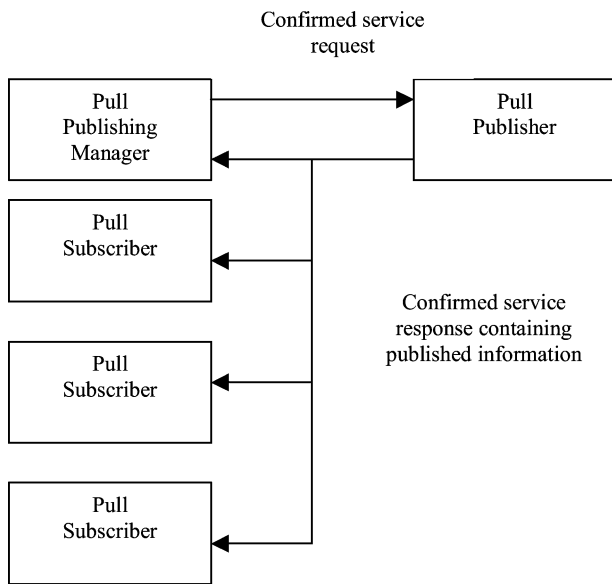


Fig. 9. Pull publisher-subscriber model.

b) *Publisher-Subscriber*: The publisher-subscriber interactions, involve a single publisher application process (AP), and a group of one or more subscriber APs. This type of interaction has been defined to support variations of two models of interaction, the “pull” model and the “push” model.

- **Pull model**: In the “pull” model, the publisher receives a request to publish from a remote publishing *manager*, and broadcasts (or multicasts) its response across the network. The publishing manager is responsible only for initiating publishing by sending a request to the publisher. Subscribers wishing to receive the published data listen for responses transmitted by the publisher. In this fashion, data is “pulled” from the publisher by requests from the publishing manager. A confirmed service is used to support this type of interaction.

Two characteristics of this type of interaction differentiate it from the other types of interaction.

First, a typical confirmed request/response exchange is performed between the publishing manager and the publisher. However, the underlying conveyance mechanism returns the response not only to the publishing manager, but also to all subscribers wishing to receive the published information. This may be accomplished by having a protocol mechanism in an underlying layer, which transmits the response to a group address, rather than to the individual address of the publishing manager. Therefore, the response sent by the publisher contains the published data and is multicast to the publishing manager and to all subscribers.

The second difference occurs in the behavior of the subscribers. Pull model subscribers, referred to as pull subscribers, are capable of accepting published data in confirmed service responses without having issued the corresponding request. Fig. 9 illustrates these concepts.

- **Push model**: In the “push” model, two services may be used, one confirmed (1 and 2) and one unconfirmed (3). A confirmed service is used by the subscriber to request a binding to the publisher. The response to this request is returned to the subscriber, following the client-server model of interaction.

The unconfirmed service (3) in the “push” model is used by the publisher to distribute the information to subscribers. In this case, the publisher is responsible for invoking the correct unconfirmed service at the appropriate time and for supplying the appropriate information. In this fashion, it is configured to “push” its data onto the network. Subscribers for the push model receive the published unconfirmed services distributed by publishers. Fig. 10 illustrates the concept of the push model. In Fig. 10, the sequence “request-confirmation” (noted 1 and 2) represents a subscribing phase. The publishing operation (3) is triggered by the push publisher itself, each time necessary.

According to the pull model, it is possible to define a READ service initiated by the pull publishing manager, similar to the READ service of the client-server model. The difference is that, in the pull model, all the subscribers receive the READ confirmation under the form of an indication, because there is no prior request.

According to the push model, it is more a service resembling an information report initiated by the push publisher which may be considered as a server.

The push publisher-subscriber model is well suited for transmitting event data. It may be used for services as “event notification” request and indication or “information report” request and indication defined in MMS.

The publisher-subscriber models are used for exchanges (READ and WRITE services) between buffers. The following fieldbuses use this model: WorldFIP, CAN, LonWorks, EIBus, ControlNet, SwiftNet, and FOUNDATION Fieldbus.

In terms of timeliness, the duration of the exchanges done by these models depends only on the MAC protocol, coming from the latency of the sending operation by the push publisher.

c) *Manager-Agent*: The manager-agent model is similar to the client-server one. It is the model used by the Simple Network Management Protocol (SNMP), in conjunction with a management information base (MIB) based on a tree structure. This protocol provides a push publisher-subscriber service, the so-called TRAP request and indication.

2) Other Models:

a) *Client-Server Multiconfirmations*: This service model defines several responses (and then confirmations) for a single request (Fig. 11). It is of interest in the case of long service execution. The semantics of the responses and of the confirmations may be the following: the first response indicates that the request is possible and taken into account by the server. The second response indicates that the service starts its execution. The last one delivers the results of the service.

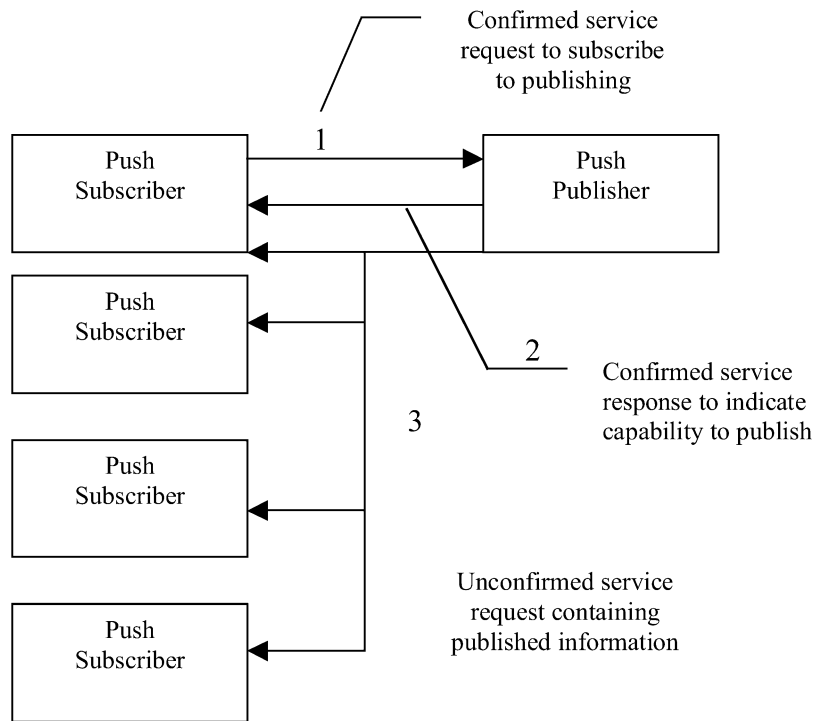


Fig. 10. Push publisher-subscriber model.

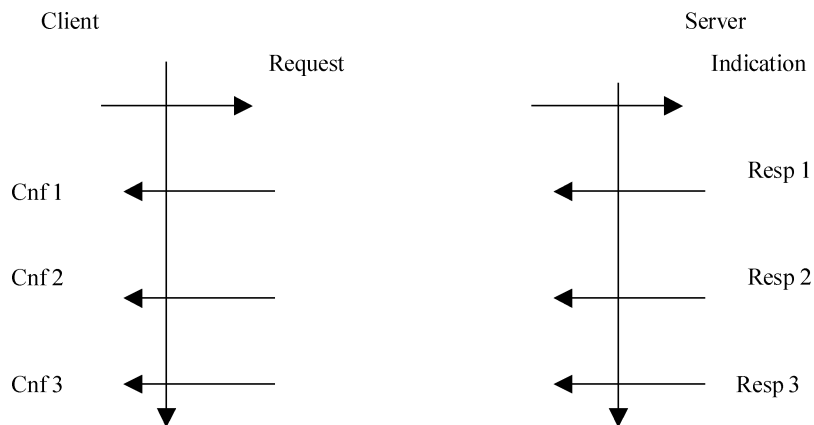


Fig. 11. Multiconfirmation client-server.

This model of cooperation is well suited for long duration services, when a server is overloaded, and when the execution of a service may take a long time. It allows the client to know the status of its request, and it is possible for the client to establish time constraints (delays or deadlines) for each of the responses. Cancellation of a service request may then occur when the constraints are not met. Such a model may be used for any service. This model is not implemented in standardized protocols.

b) Client Multiserver: The client multiserver model (Fig. 12) is a particular case of the client-server model. A given request which cannot be processed by a single server can have several servers that may answer it. In this case, there is a function to break down the request into subrequests adapted to the capabilities of the different partial servers.

The client does not know all the partial servers. The decomposition of the initial request into several is not known to

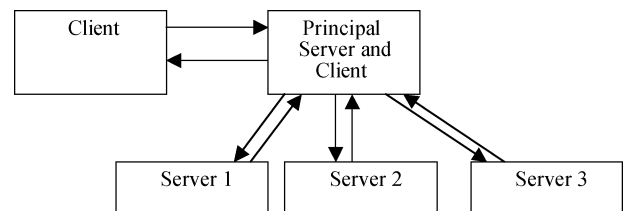


Fig. 12. Client multiserver.

the client. Some synchronization between the partial server actions can be requested and verified by the principal server.

The problem is that, upon the definition of the result in the case of all partial servers not being able to provide a correct response, the global response to the client then becomes partial. If the response can only be complete or negative, the problem does not exist.

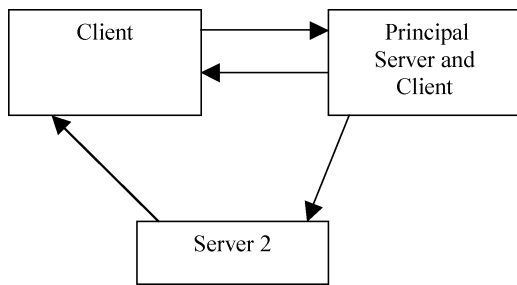


Fig. 13. Third-part model.

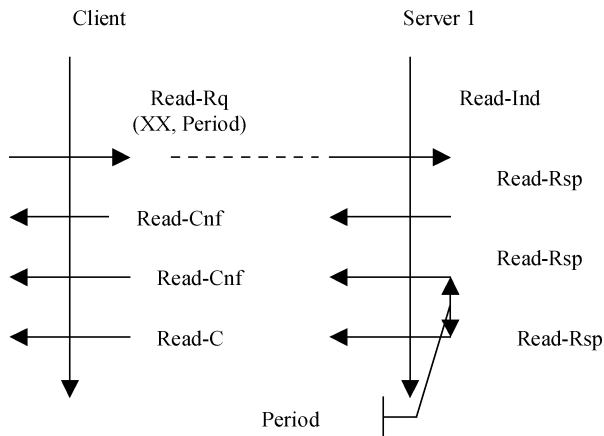


Fig. 14. Multiresponses.

If a response is partial, the client must know the composition, in order to correctly identify the lacking parts of the response. This model has been studied in different works [29]. It can be implemented above MMS protocols, but is not recognized in standards.

c) *Third-Part Model*: This model is a particular case of the previous one. A client requests a service from a server which is unable to provide the service but which knows the appropriate server able to do it (Fig. 13). Several scenarios may be considered and possible failures must be detected and corrected.

d) *Multipublisher–Multisubscriber*: This model is of interest for synchronizing the activities of publishers. If, for example, several pieces of data must be produced at the same time (i.e., in a given time window), it is easy to synchronize the producers and then to apply one of the two publisher–subscriber models to provide information to the subscribers.

The time constraints, which may be specified, are the global response time (less than the period, obviously) between request and confirmation, or the server response time between the indication and the response.

This model of cooperation is essential when properties such as the time coherence of data production, of data publishing, or of data consumption are required.

This model is used in WorldFIP networks and in all the IEC 61 158 Type 1 compatible networks.

e) *CS Mono Request Multiresponse*: This model provides the following timed behavior. It is similar to the push publisher–subscriber model. The request “Read-Rq” may be compared to the request to become a subscriber.

This model (Fig. 14) is of interest in fieldbuses for the managing of periodic exchanges, with a single request.

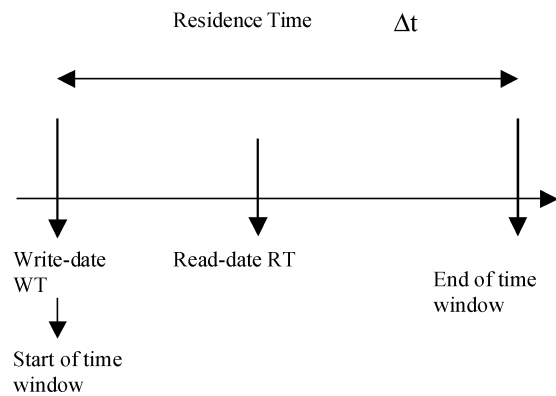


Fig. 15. Residence attribute.

This model may be extended with the means to define a starting event, for example, a date or a condition and, similarly, an ending event, for example, duration or a condition or a date.

This model can be compared to a periodic client–server consisting of periodically requesting the service provided by the server. The difference is on the temporal QoS. A periodic client–server is periodically triggered on the client site, but with transmission and server delays, the period may not be met at the server site. In the multiresponse model, the period is managed at the server site and can then be more strictly respected.

f) *Conclusion*: The different cooperation models represent the rules, which are furnished by the application and presentation OSI layers. They make up the application service element (ASE), which is called “*industrial messaging*” or “*fieldbus application layer*” (FAL) [74]. A lot of messaging services and protocols have been defined in different domains, which may be used in fieldbus applications: MMS [89], SNMP, MPS [2], [69], [149], [150], [152], and IEC 870-5, [66]. All the application layers in existing fieldbuses provide a subset of these application service elements.

3) *Timeliness*: The word “timeliness” means all the temporal aspects of operations, of data, and more generally of dynamic system components. Timeliness is expressed through dates or time stamps, through durations and through Boolean attributes, which determine if a temporal property has been met or not.

Timeliness has been studied (and is still being studied) for a long time in different communities, with the first work on real-time languages and on formal methods for time and dynamic system modeling. A lot of papers have been published on these subjects. Historically speaking, the following are of interest because they introduced the concepts now available with the truly “real-time” fieldbuses. For the topics related to real-time systems and languages the reader may consult different papers: [12] for the PEARL language, Gertler in [55] proposed the first synthesis on real-time languages; Deschizeaux [34] and Kronental [97] proposed elements for the standardization of real-time operating systems; Thomesse, in [147], introduced timing considerations and mechanisms in real-time distributed applications; Le Lann [100] and Lamport [98] were among the first to formally introduce the problems of time in distributed systems. For time concepts and

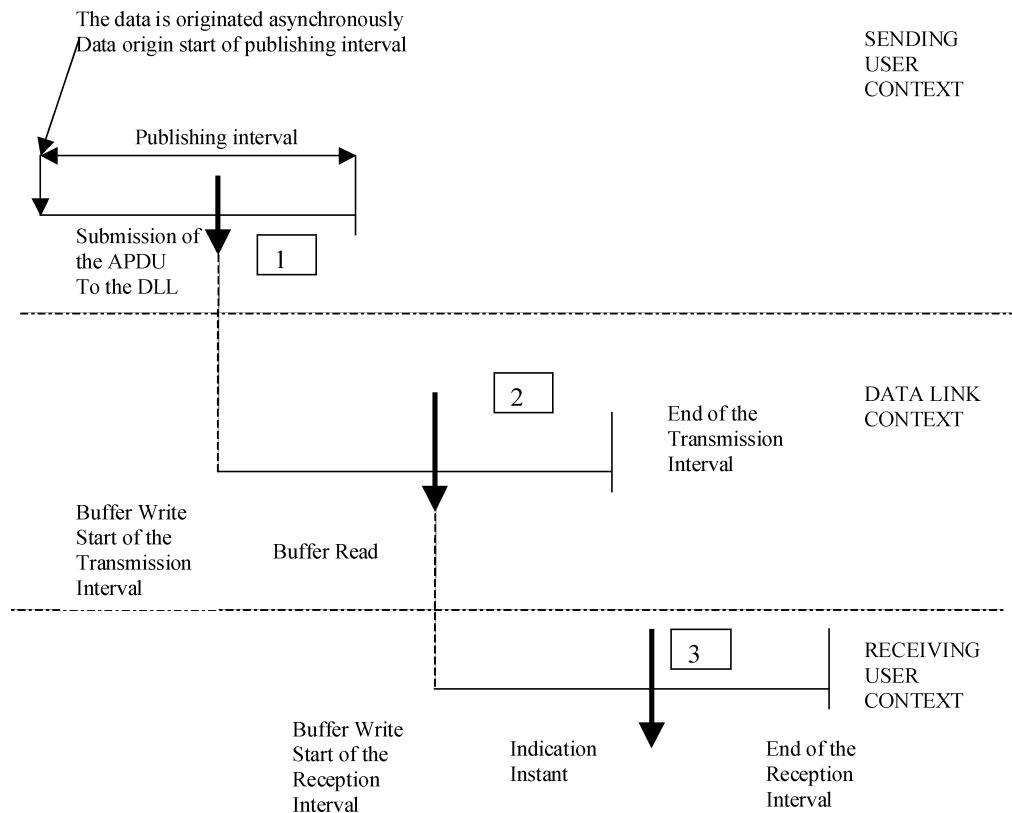


Fig. 16. Application layer residence mechanism (issued from [74]).

modeling, the reader can consult [30], which explains the different types of time constraints; [113] for an overview of time concepts in real-time applications; [5] and [121] for a formal presentation of the introduction of time in logic and state-transitions systems. Reference [160] proposes extensions to UML for time consideration and modeling.

The objective of this section is to give the particularities of timeliness in fieldbus services and protocol. We shall not present the generalities such as time stamping or clock synchronization. This section will focus on the definition of timeliness attributes in order to verify if time constraints have been met or not.

The first such attributes were proposed in the WorldFIP application layer. And they are now redefined in some profiles of the IEC 61 158 standard. The idea was to verify if a given operation had occurred in the right time interval, i.e., in a given time window. Here we shall only give an example in Fig. 15, issued from the WorldFIP standard and from the IEC 61 158-3 data link layer standard [73].

The “residence” timeliness is an assessment based on the length of time that a datum has been resident in a buffer, which is the time interval between:

- 1) the moment when the buffer is written; and
- 2) the moment when the buffer is read.

Given ΔT the length of the residence time interval, the data link residence is defined as follows: $0 \leq (RT - WT) < \Delta T$, where RT and WT are the READ and WRITE instants.

This type of timeliness was called *asynchronous* in previous French and European standards.

Applying this principle to the communication between a publisher and a subscriber, it can be seen that three operations in three intervals can be controlled. A global analysis of these attributes may be found in [107].

The publishing operation is controlled at the publisher site by an attribute which is transmitted along with the value to the subscribers. They, in turn, can then know if the publisher has met its own constraint or not. For example, such an attribute can indicate if the period of a periodic publishing operation has been respected. Or such an attribute can indicate if the publishing occurred before a given deadline after a request, Action 1 in Fig. 16.

A similar control can be placed at the receiving site. For example, a given value must arrive periodically or before a given deadline after a request. An attribute may be computed at the receiving instant (Action 3 in Fig. 16), in order to be transmitted along with the value to the application entity afterwards. Such an attribute can also be defined at the Data Link layer to control if the sending instant occurs in a given time window (see Action 2 in Fig. 16).

A subscriber then receives, not only a value of data, but also attributes which indicate if the successive operations have occurred in the right time window, or on time. These attributes represent the QoS from a time point of view. The subscriber may then decide what to do according to the quality of the data. The reader may find the basis of the QoS in [92].

D. Data Link and MAC

1) *MAC Classification:* The usual MAC protocols are based on one of three following classes: controlled access, TDMA, or contention (see Fig. 17). If a control is used, it

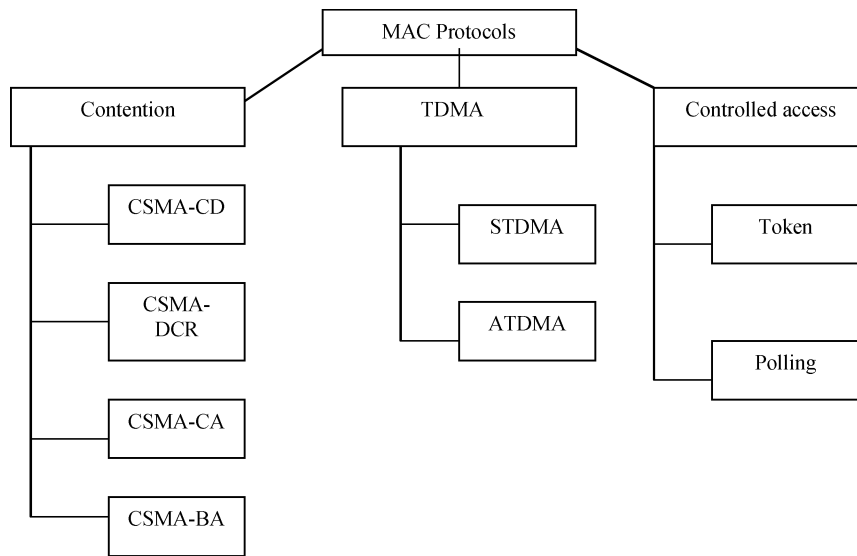


Fig. 17. MAC protocol classification.

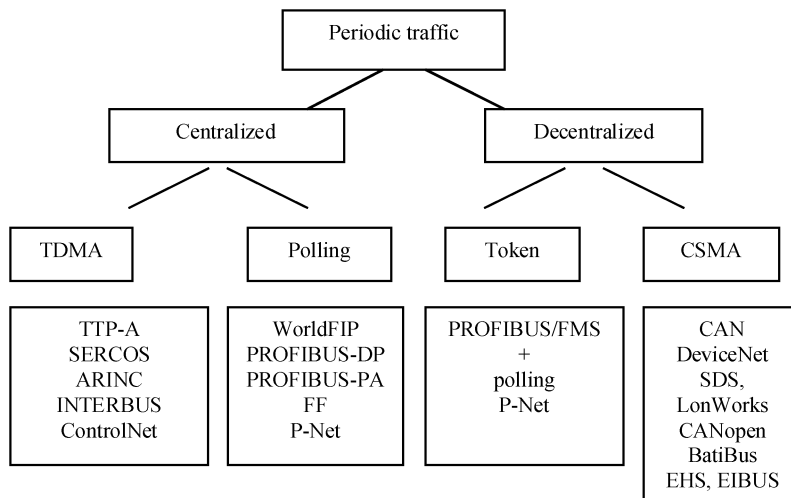


Fig. 18. Periodic traffic in fieldbuses.

can be centralized or decentralized. In the case of TDMA, the classification is not so easy, the access is always decentralized, because the decision to send is taken individually by each station, but the clock synchronization function itself may or may not be centralized. In applying such a classification to fieldbus MAC protocols, it is necessary to distinguish the management of periodic and of random traffic, as shown in Fig. 4 (see also [102]).

Regarding how periodic traffic is managed, it is either centralized, or decentralized. Figs. 18 and 19 show two classifications of the fieldbuses according to their MAC protocol, regarding the periodic and the aperiodic traffic. In the case of decentralized management, each station must decide, at its allotted time to send, which traffic it should prioritize. In the case of centralized management, a periodic server deals with this problem. A station can ask for an additional right to send when it is periodically polled (special frame on demand) or the server systematically and periodically allots a time slot for aperiodic traffic (time slot in each frame).

The contention protocols cover all CSMA variants. The controlled access protocols are used the most in large field-

buses, with thousands of stations. All protocols use MAC addresses, which are either a station address or a logical address (source address), which is more efficient for cyclic traffic. Addressing by the name of the identifier is used by WorldFIP, CAN, BatiBus, EIBus, and FOUNDATION Fieldbus. Otherwise, a classical addressing mode is used.

2) *TDMA Class*: This class represents the protocols which give the right to send on the medium according to a rule, such as time division multiple access (TDMA).

a) *General Principle*: TDMA is based on dividing the access time of the medium into slots, which are allotted to the stations according to a given strategy. The slots may or may not be equal in duration. Each station may send a frame of a given length at a defined moment. In synchronous TDMA, the access is periodically allotted, as indicated in Fig. 20. In asynchronous TDMA (ATDMA, Fig. 21), the slots are allotted to the stations according to their needs. This means that a station without generated traffic does not use its slots, such as the Sites 2 and 4 in Fig. 21. While in synchronous TDMA the address of the sender is implicitly given by the relative position of the slot, in asynchronous TDMA, each

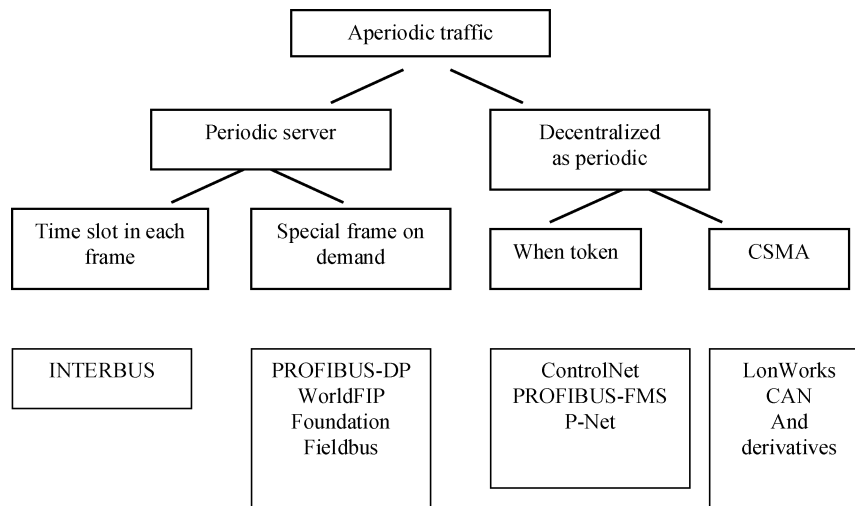


Fig. 19. Aperiodic traffic.

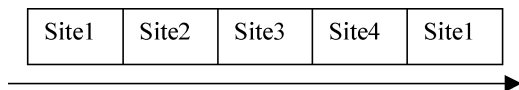


Fig. 20. Synchronous TDMA.

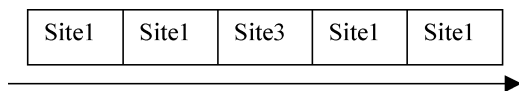


Fig. 21. Asynchronous TDMA.

slot must contain its address or its identification. In STDMA, the nominal data rate of the network is equal to the sum of the stations' loads; in ATDMA, the total load of the stations may be greater than the nominal data rate of the network.

b) Variants: The variants concern the following points.

The content of a slot: the content of a slot may be the value of the data, or a frame issued from a station containing (possibly) the values of different data. In the former, a same station may then have several rights in a same round, possibly to send more often than others.

The length of slots: all the slots are of the same length (as in digital phone systems, because all the traffic is the same), or of different lengths in order to take into account the needs of each station.

The clock synchronization: the clock synchronization is the basis for defining the starting instant of transmission for each node. This synchronization may be done in a centralized way as in TTP-A (Time-Triggered Protocol) [96] or by a distributed algorithm as in TTP-C. It is also important to note Time-Triggered CAN (TT-CAN) and Flexible Time-Triggered Protocol (FTT-CAN), which, being based on CAN, introduce a time-triggered mechanism.

c) Examples: The TDMA principle is used for periodic traffic by TTP [94], [95], ARINC protocol family, SERCOS [68], and ControlNet [27]. INTERBUS on a ring topology is similar to TDMA; a single frame is divided into

as many fields as the number of stations. Each station has the right to send in its own field.

d) QoS: The temporal QoS is generally good, the frequencies are met, and no jitter occurs when the clocks are well synchronized. It is supposed that each station respects its sending time. The periodic transmission of the time-constrained data may be guaranteed under certain hypotheses [142]. The clock synchronization is not a topic of this paper, but the reader may consult [78] for an example of clock synchronization algorithm.

3) Polling Class:

a) General Principle: The polling class represents the protocols that allow the right to send by sending an explicit message (the poll message) to the station, enabling it to send. The poll message is always sent by a special station, called *Master*, *Arbitrator*, or *Manager*, etc.

b) Variants: The variants are related to the addressing method and to aperiodic traffic management. Some are static, others dynamic [134].

Addressing methods: there are two main subclasses: the first designates each station by its address, the second by the identification of the data to be sent. The former indicates the station explicitly and, in the latter, it is implicitly designated, as in the producer-consumer cooperation models.

Aperiodic traffic: different techniques are used to manage aperiodic traffic. For example, WorldFIP uses a dynamic scheduling of requests for aperiodic traffic taking place in the free time slots of the periodic traffic; INTERBUS uses, in each cycle, a 2-B field in the periodic frame to transmit information on demand. ControlNet uses a round robin algorithm for managing aperiodic traffic.

c) Examples: Centralized MAC fieldbus representatives are P-Net, WorldFIP, AS-i [7], PROFIBUS-DP, and PROFIBUS-PA. A station is in charge of the distribution of the access control. INTERBUS [38] may also be considered as a polling protocol, because each station periodically receives the right to send from a central master. It may also be considered as a kind of TDMA on a ring topology, analogous to the Cambridge ring. It could also be modified in a multimaster protocol [20].

d) *QoS*: A polling MAC can guarantee the periods without jitter if some mechanisms (anti-jabber) are developed to avoid overly long frames from being transmitted. The polling technique favors periodic traffic and time-triggered systems.

The time coherence constraints are easier to manage if a multicast is allowed and a consensus mechanism used in order to ensure the distributed copies are identical. WorldFIP is typically such a fieldbus [137].

One may raise the objection that a centralized system is not robust. Some fieldbuses allow a redundancy of the bus controller, or of the bus control function, which may be implemented on several stations (PLC, regulators, sensors, and so on).

To introduce dynamic behavior in statically defined systems, different operating modes may be defined as according to the Fohler proposal [49].

The bus arbitrator of WorldFIP may be duplicated; a token-like mechanism allows a bus arbitrator to give the bus control to another arbitrator, as with the master stations in PROFIBUS.

The draft proposal IEC 1158-3 included services coming from WorldFIP and PROFIBUS standards, allowing centralized access control as well as decentralized.

4) *Token Class*:

a) *General Principle*: This class represents the protocols which provide a control access similar to the polling class, which can be used with a bus or a ring topology, but is decentralized.

b) *Variants*: The variants are related to the role of the stations in the fieldbus, be they masters or slaves, to the form of the token, and to the passing method. The role of the stations: master and slave stations may be distinguished from each other, such as in PROFIBUS. Master stations constitute a virtual ring over a bus topology. They poll slave stations when they hold the token.

The form of the token: it may be an explicit message but it may also be implicit; for example, when a round robin scheduling is used (ControlNet for the aperiodic traffic management), the token is automatically and implicitly passed between stations with successive addresses.

c) *Examples*: The first was PROFIBUS-FMS, which defined a token passing mechanism between the master stations, and a polling mechanism between a master station and the slaves. P-Net provides a similar mechanism but with an implicit token, as ControlNet for the aperiodic traffic management.

d) *QoS*: The temporal QoS guarantees that bounded transmissions (with bounded jitter) are respected due to dependability hypotheses. The respect for periods is less strict than with TDMA or polling because of token management. If the token holding time of each station is strictly constant, and if no errors occur, the periodicity is respected. Jitters may appear in the case where the previous hypothesis is false [31]. Two successive polling operations of a same slave by two masters may lead to temporary inconsistencies between the state information. From this point of view, PROFIBUS FMS was more a Mini-MAP-like profile than a fieldbus.

5) *Link Active Scheduler (LAS)*:

a) *General Principle*: The general principle consists of giving the responsibility of traffic scheduling to a specific station (the LAS). But it has the capacity to delegate responsibility to another station with token passing or with an order to distribute data for a given duration. It is based on a mixed mechanism of PROFIBUS token passing with the WorldFIP bus arbitrator. It comes from the IEC TC65C WG6-Data Link Layer Working Group Committee (IEC 61158-3 Type1), which tried to find a common solution for the much wanted international standard. No variant is known at this moment.

b) *Example*: The FOUNDATION Fieldbus has implemented this mechanism.

c) *QoS*: The temporal QoS is similar to the one obtained with the polling technique.

6) *Contention or CSMA Class*:

a) *General Principle*: The CSMA class represents all the protocols, which are based on any variant of the Ethernet principle. The principle is to wait for the channel to be free to send a frame. Collisions may occur, and the variants propose different recovery mechanisms.

b) *Variants*: The variants are CSMA-CD, CSMA-CA, CSMA DCR, and predictive p-persistent CSMA as in LonWorks.

The most known variant is CSMA-Collision Detection (CD), which is not very common in fieldbuses except when the maximum load is relatively low in relation to the nominal data rate. An example is the Poste de Contrôle-Commande Numérique (PCCN) network for electrical transformers.

CSMA-CA is used in building automation networks such as BatiBus [8], EIBus [43], EHS [42], in car networks such as CAN [91]. It is often a CSMA with forcing capabilities, often called CSMA-CA for “collision avoidance”; it means that even in case of a collision, a single frame may be transmitted, the one with the most priority.

Other CSMA variants (CSMA-DCR) have been defined to guarantee an upper limit to transmit all collided frames [101]. These protocols are based on a partitioning of the stations’ ability to transmit. The protocol is robust in the sense that if a station fails, the others are not concerned at the MAC level. The problem is that a more urgent frame must wait until the end of the transmission of all the previous collided frames.

Another variant is found in LonTalk protocol [105]. It is based on predictive p-persistent CSMA. This method consists of estimating the backlog to adjust the medium access delay according to the network’s current load.

- *Examples*: The examples of CSMA variants are CAN, SDS [118], [141], DeviceNet [120], LonWorks, BatiBus, and EIBus.

CAN is used in DeviceNet and in SDS as a “sub-fieldbus” in a machine.

- *QoS*: The temporal QoS may be predictable in the case of CAN based networks. This mechanism may guarantee the periodicity under some hypotheses [156], [157]. The data rate is limited by the length of the medium, typically 1 Mb/s for a length of 40 m.

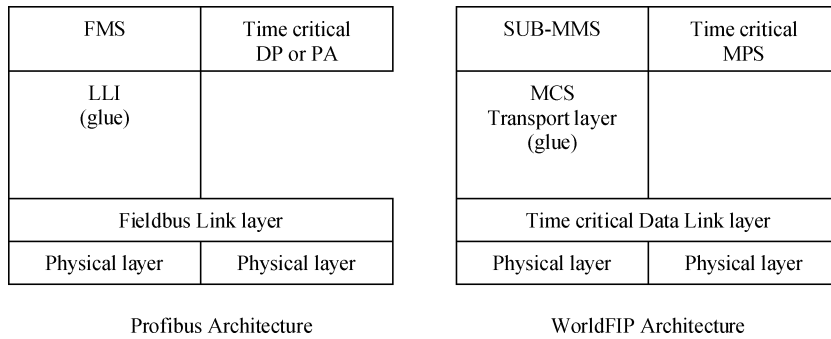


Fig. 22. PROFIBUS and WorldFIP architectures.

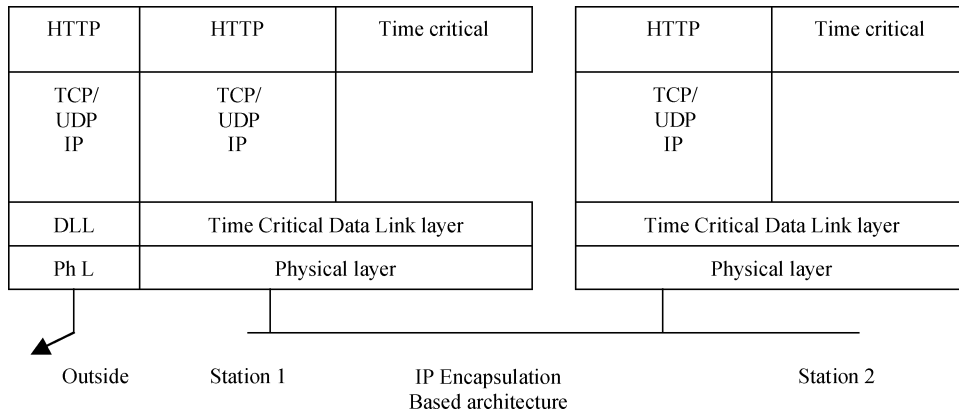


Fig. 23. Encapsulation.

The industrial Ethernet solutions and switched Ethernet are not studied here; other papers in this issue are dedicated to these solutions.

7) *LLC*: The LLC is not distinguishable from the MAC in fieldbuses. However, the LLC services may be identified in the data link layer specifications. The usual LLC services are known under the names LLC type 1, 2, or 3. The major fieldbuses provide LLC type 3-like services, without connection in the OSI sense, with immediate acknowledgment for the real-time traffic. LLC type 1 is also used. Transmission without acknowledgment may be of interest for periodic traffic. The failure or error detection is then made by the receiver(s), when it is made by the sender with LLC type 2 services and protocols. The latter are used for messaging traffic, in order to provide the right transmission safety. Even if from a service point of view, fieldbuses are very close to IEEE 802.2 specifications, from the protocol point of view, they are all different and incompatible.

E. Communication Architectures

After having analyzed the cooperation models at the application layer, and the MAC, let us now further investigate the communication architectures.

1) *Two Stack Architectures*: The communication architectures have been veritably and explicitly developed according to, and thanks to, the OSI reference model. The first modifications (or extensions) of this model were introduced with the IEEE 802 model and with the MAP-EPA reduced model [99]. Both of these extensions have their

own reasons for introduction, but arguments for real-time needs were put forward to promote MAP-EPA. It is a known fact that speed alone is not enough to meet the real-time constraints [144] and that, when a minimum amount of resources are available, the scheduling of the tasks and messages within resource allocation is the only solution. That is the main reason for the definition of most of the fieldbus “real-time” MAC protocols including, more or less explicitly, a scheduling of the messages. In parallel, for configuration or maintenance operations (downloading), normal (not real-time) protocols are necessary. In brief, the internal communications can only be ensured with a two-stack architecture (see Fig. 22). Another reason for a two-stack architecture is that the fieldbus needs to communicate with the outside world.

The use of Web technologies and the emergence of Ethernet at the field level contributed to the study of architectures. Even if some papers in this issue deal with these questions, let us introduce the problem before going back to the future.

2) *Introduction of Web Technology*: Considering a fieldbus with dedicated time-critical data link layer (TC-DLL) protocol, a solution for compatibility with Web technology is the tunneling of IP datagrams inside TC-DLL PDU (Fig. 23). Station 1 is a gateway to the outside world. The HTTP frames which are fragmented into IP datagrams can be transported between fieldbus stations after being encapsulated into TC-DLL PDUs. Szymanski in [146] analyzes the solutions for introducing Web technology into process control.

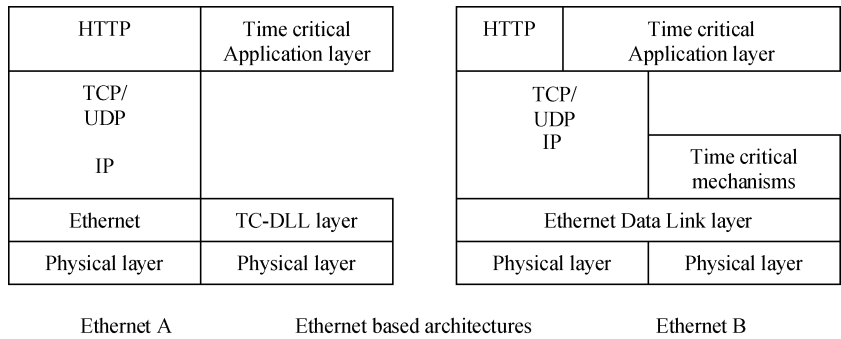


Fig. 24. Ethernet-based architectures.

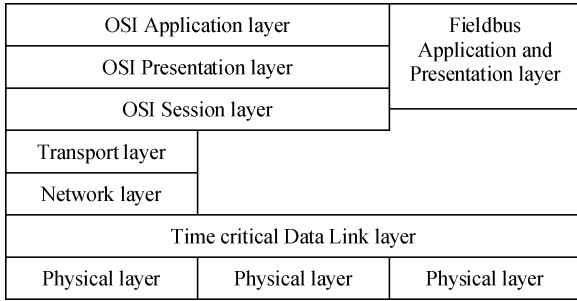


Fig. 25. A time-critical communications architecture (from [127]).

3) *Introduction of Ethernet as Fieldbus MAC Protocol*: A trend was started some years ago to use Ethernet as the TC-DLL [61] in automation [111]. For that, some mechanisms were added to Ethernet to obtain two channels, one for real-time traffic, one for the rest. This is the principle used by all the data link layers, which support different types of traffic.

Since the end of the 1990s, Ethernet has been proposed as a standard for the MAC. And upon Ethernet, naturally, the promoters thought to use the TCP/IP stack and the Internet application layer protocols. This was the reason for the standard project “Real-Time” Ethernet (RTE) of the TC65 SC65C WG11, standard IEC 61784 part 2. The drawbacks of Internet are, moreover, the nonpredictability and the connector technology for the industrial environment. The latter problem was resolved, but as for the former, some mechanisms need to be added to make Ethernet predictable, if this expression can be used. Both of the solutions shown in Fig. 24 can be used. It is also important to notice that the same application layer can be used over the different stacks as has already been done in the CIP solutions family [139].

4) *Toward a Common Stack?*: In a 1991 paper, a three stack architecture was proposed by Phinney [127], as indicated in Fig. 25. The idea of this architecture was to provide a common data link layer with different qualities of service for typical fieldbus traffic, as analyzed above in this paper as well as for file transfers with all the necessary security and dependability. Such a data link layer provides real-time features, associated with connection mechanisms, acknowledgment, bridging capabilities, etc. It was obviously possible to adapt this layer to any kind of physical layer, and to build upon different stacks starting from a full OSI stack,

for covering general purpose networking needs, going all the way down to a reduced stack for very specific real-time fieldbus-based applications. A full stack is used for normal communication and can be implemented with TCP/UDP and IP protocols. A medium stack can be used when neither routing nor fragmentation/reassembling are necessary. The stack on the right is the normal time-critical stack.

The TC-DLL should be the proposal for IEC 61 158 [70], [71].

This proposal should be reexamined for two reasons: first, in light of a future RTE protocol, and second, in light of the real-time mechanisms introduced in the Internet stack needed to implement such applications as phone over IP, video transmission, videoconferencing, etc. With the capability for the user to control protocol behavior, according to TCCA recommendations, we could hope for a common, general purpose, and real-time communication architecture.

Two methods are possible for solving this problem: one is based on the encapsulation of IP datagrams in the TC-DLL frames, the other is based on the modification of Ethernet frame scheduling to meet real-time constraints. The former was chosen for years by several fieldbus vendors, and the latter was supported by the defenders of Ethernet (or Ethernet variants) as the data link layer for fieldbuses.

IV. CONCLUSION

The standardization of protocols is far from being finished. The needs of the end-users expressed in the European MAP user group are still more or less valid. Maybe the new working group of IEC TC65 on RTE will take into account the TCCA recommendations, and contribute to the design of a common architecture, which could improve the interoperability of heterogeneous components. Another challenge could be the definition of a common time-critical data link service and protocol, with the right parameters to dynamically tune the protocol to the application needs (QoS required versus possible QoS).

The fieldbus technology covers a very large spectrum of techniques and applications. The fieldbus is present everywhere. This phenomenon may explain the diversity and the lack of a real standard, but it is not the only reason.

One could write a paper making a parody of Pouzin’s well-known paper [132], entitled “Virtual circuits versus datagrams: Technical and political issues,” written when

IP and X25 (in the mid-1970s) were fighting as network protocol standard candidates in the standardization bodies. Such a paper could now be entitled “Client–server versus publisher–subscriber: Technical and political issues” or “Token bus versus bus arbitration: Technical and political issues” to explain the importance of political or economical and strategic aspects in choosing a standard or not.

This paper has tried to explain the different approaches and solutions in order to give the reader the most complete overview on the history of the fieldbus and on its current situation.

Not all the aspects have been treated, for different reasons:

- different physical layers, the powering of devices by the network, the intrinsic safety; because the solutions are numerous and, if important concerning the applications, these points are not really strategic;
- network management, which is out of the scope of standardization and covered by proprietary solutions;
- conformance testing, which is very closely attached to each solution;
- the problem of interoperability and interchangeability, which was (and is) an open problem until now;
- the problem of scheduling policies, which are the basic element of solution for the real-time constraints management.

And to conclude, going back to the title of this paper, is fieldbus a technology? Fieldbus may be considered as a technology for the design of automation systems, like any other component or artifact. It is an essential component of any automation system, and a major component of a lot of systems. Several solutions have been promoted, implemented and tested in real industrial applications. All the recent power plants, new factories, trains, cars, new buildings, etc., include fieldbuses, even if they are invisible to the user. The technology may then be considered as mature.

But it is also more than a technology.

Fieldbuses in industrial automation represent more than a technology because they are, today, true real-time communication networks. And as such, they are also relevant to time modeling, time management, and to the sciences that have time as an object of study.

Fieldbuses represent more than a technology because they are the basis for the emergence of new paradigms for communication and cooperation between agents. The difference with the normal OSI world comes from the different expressions of qualities of service when considering the applications. New communication paradigms [1], [155] have been created with the development of sensor networks, of ad hoc networks, of ambient intelligence. They are an extreme mobility, a variable connectivity, a great number of stations, an opportunity to discover new stations, the autonomy of agents, and the list goes on. A convergence could perhaps be found with the definition of a real time-critical data link service and protocol.

Fieldbuses represent more than a technology because they have provided an opportunity for extensive research, although we did not consider this point, this research concerns:

- the protocol verification;
- the performance evaluation;
- the distributed application design methods;
- the scheduling, and especially the joint scheduling of tasks and of messages;
- the joint modeling of the system and of the network, in order, for example, to analyze the impact of network behavior on the system itself;
- the modeling of devices with different objectives (proof of interoperability, documentation, configuration, maintenance, etc.).

The road was long to arrive at the Internet solution as a common communication stack for a large spectrum of applications. This choice was accompanied by a drastic reduction in the number of operating systems. Will we see the same evolution for fieldbuses and for automation operating systems in the near future?

ACKNOWLEDGMENT

The author would like to thank all his colleagues and the Ph.D. students who have contributed to the fieldbus concept development and its story. The author would also like to thank the reviewers whose remarks have contributed to improve the first version, and Mrs. T. Wagner for the linguistic aspects.

Special thanks are addressed to R. Berthoumieux and M. Desjardins without whose confidence this adventure would never have happened.

REFERENCES

- [1] E. H. Aarts, “Ambient intelligence: calming, enriching and empowering our lives,” *Password*, no. 8, Jul. 2001.
- [2] *French Standards NF C46601 to C46607: FIP Bus for Exchange of Information Between Transmitters, Actuators and Programmable Controllers*, 1989–1992.
- [3] *French Standard C46-638, Système de Communication haute performance pour petits modules de données (WorldFIP Profil 1)*, 1996.
- [4] L. Almeida, “Flexibility and timeliness in fieldbus based real time system,” Ph.D. thesis, Univ. Aveiro, Aveiro, Portugal, 1999.
- [5] R. Alur and T. A. Henzinger, “Logics and models of real time: a survey,” in *Lecture Notes in Computer Science, REX Workshop*. Heidelberg, Germany: Springer-Verlag, 1991, vol. 600, pp. 74–106.
- [6] B. Armitage, G. Dunlop, D. Hutchinson, and S. Yu, “Fieldbus: an emerging communications standard,” *Microprocess. Microsyst.*, vol. 12, no. 10, pp. 555–562, Dec. 1988.
- [7] *AS-i, Actuator and Sensor Interface*, CENELEC Std. EN 50295-2, 1998.
- [8] *French Standards AFNOR, NFC 46621 to 623 and 629. Un réseau pour la gestion technique et administrative des bâtiments*, 1991.
- [9] K. Bender, *PROFIBUS, the Fieldbus for Industrial Automation*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [10] *IEEE 1118, Standard Microcontroller System Serial Control Bus*, 1991.
- [11] L. Borsi and E. Pavlik, “The concepts and structures of distributed process automation systems,” *Process Autom.*, no. 2, pp. 63–70, 1980.
- [12] J. Brandes *et al.*, “A real time programming language and its application for measuring processes,” presented at the IFAC 5th World Congr., Paris, France, 1972.
- [13] P. F. Brown and C. R. Mac Lean, “The architecture of the NBS factory automation,” presented at the IFAC Congr., München, Germany, 1987.
- [14] A. Burns, “Scheduling hard real time systems: a review,” *Softw. Eng. J.*, vol. 6, no. 3, p. 116, 128, 1991.

- [15] P. Burton, "FieldBus: an overview of current proposals," in *IEE Colloq. Industrial LANs: The Real Issues*, 1987, pp. 5/1–5/3.
- [16] *Bosch CAN Specification—Version 2.0 Part A*, 1991.
- [17] C. Cardeira and Z. Mammeri, "Scheduling in fieldbus based real-time systems," in *Real-Time Computing*, W. A. Halang and A. D. Stoyenko, Eds. Berlin, Germany: Springer-Verlag, 1994, pp. 568–573.
- [18] —, "A schedulability analysis of tasks and network traffic in distributed real-time systems," *Measurement*, vol. 15, pp. 71–83, 1995.
- [19] R. P. Carson, "Distributed data analysis in computer networks," *Ind. Res./Develop.*, vol. 23, no. 5, pp. 130–135, May 1981.
- [20] S. Cavalieri, A. Consoli, and O. Mirabella, "Adding multimaster capabilities to Interbus-S," in *Fieldbus Technology, Systems Integration, Networking and Engineering: Proc. Fieldbus Conf. (FeT'99)*, pp. 30–37.
- [21] *European Standard EN 50170. Fieldbus. Volume 1: P-Net, Vol. 2: PROFIBUS, Vol. 3: WorldFIP*, 1996.
- [22] *High Efficiency Communications Subsystems for Small Data Packages, CLC TC/65CX*, CENELEC Std. EN 50254, 1996.
- [23] *Criteria for 'High Efficiency Communications Subsystems for Small Data Packages' CLC/TC65CX(SEC)040*, CENELEC Std. EN 50254, 1996.
- [24] J. M. Chang and N. F. Maxemchuck, "Reliable broadcast protocols," *ACM Trans. Comput. Syst.*, vol. 2, no. 3, pp. 251–272, 1984.
- [25] A. Chatha and C. Polsonetti, "Fieldbus standard: We need a winner," *Instrum. Control Syst.*, vol. 65, no. 10, pp. 29–31, 1992.
- [26] N. Collins, "Boeing architecture and TOP (technical and office protocol)," in *Proc. Int. Conf. Networking: A Large Organization Perspective*, 1986, pp. 49–54.
- [27] *ControlNet Specification, Release 2.0*, 1997.
- [28] L. Costrell, "CAMAC instrumentation system—introduction and general description," *IEEE Trans. Nucl. Sci.*, vol. NS-18, no. 2, pp. 3–8, Apr. 1971.
- [29] Y. Dakroury and J. P. Elloy, "A new multiserver concept for the MMS environment," presented at the 9th IFAC Workshop DCCS, Tokyo, Japan, 1989.
- [30] B. Dasarathy, "Timing constraints of real time systems: Constructs for expressing them, methods for validating them," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 1, pp. 80–86, Jan. 1985.
- [31] J. D. Decotignie, P. Raja, G. Noubir, L. Ruiz, and J. Hernandez, "Analysis of polling protocols for fieldbus networks," *Comput. Commun. Rev.*, vol. 23, no. 3, pp. 69–90, 1993.
- [32] J. D. Decotignie and R. Prasad, "Die Feldbusse-der grosse Bazar," *Bull. SEV/VSE*, no. 3, pp. 26–29, 1993.
- [33] J. D. Decotignie, "Some future directions in fieldbus research and development," in *Fieldbus Technology, Systems Integration, Networking and Engineering: Proc. Fieldbus Conf. (FeT'99)*, pp. 308–312.
- [34] P. Deschizeaux, R. Griesner, and P. Ladet, "A real time operating system for microcomputer network," in *Proc. 2nd IFAC/IFIP Symp. Software for Computer Control (SOCOCO 1979)*, Prague, Czechoslovakia, pp. MIII-1–MIII-4.
- [35] D. Dietrich and T. Sauter, "Evolution potentials for fieldbus systems," in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS 2000)*, pp. 343–350.
- [36] S. R. Dillon, "Manufacturing automation protocol and technical and office protocols—success through the OSI model," in *Proc. 32nd IEEE Computer Society Int. Conf.*, 1987, pp. 80–81.
- [37] *German Standards 19245-1 to 19245-3, PROFIBUS, Process Fieldbus*, 1990.
- [38] *German Draft Standard 19 258-1. Interbus-S, Sensor and Actuator Network for Industrial Control Systems for CENELEC*, 1996.
- [39] *German Draft Standard 19245-4 PROFIBUS-PA, Profibus for Process Automation*.
- [40] *Danish Standard, DS 21906. P-Net, MultiMaster, MultiNet Fieldbus for Sensor, Actuator and Controller Communications*.
- [41] J. Dwyer, "Why General Motors' manufacturing automation protocol is here to stay," *Automation*, vol. 21, no. 5, pp. 19–21, May–Jun. 1985.
- [42] Eur. Conf. Integrated Home Applications, Amsterdam, Netherlands, Jan. 13–15, 1991.
- [43] *French Standard NF C46 624 à 628, Un réseau pour la gestion technique et administrative des bâtiments*, 1991.
- [44] J. P. Elloy, P. Molinaro, and R. Ricordel, "A temporal extension to the MMS protocol with the Kernext tool," in *Proc. IEEE Int. Workshop Factory Communication Systems*, J. D. Decotignie, Ed., 1995, pp. 225–234.
- [45] "User requirements for communications in time critical applications," European MAP Users Group (EMUG), ISO/TC 184/SC5/WG2-N180, Feb. 1989.
- [46] D. Fantoni, "A never-ending story: The Fieldbus standardization," *Automazione e Strumentazione*, vol. 47, no. 11, pp. 69–75, 1999.
- [47] M. Felser and T. Sauter, "The fieldbus war: History or short break between battles," in *Proc. IEEE Int. Workshop Factory Communication Systems*, 2002, pp. 73–80.
- [48] R.-D. Floyd, "Manufacturing automation protocol," in *IEEE Int. Conf. Communications*, 1985, pp. 620–624.
- [49] G. Fohler, "Realizing changes of operational modes with a pre-run time scheduled hard real time system," in *Dependable Computing and Fault Tolerant Systems*. Berlin, Germany: Springer-Verlag, 1993, pp. 287–300.
- [50] D. Galara and J. P. Thomesse, *Groupe de réflexion FIP: Proposition d'un système de transmission série multiplexée pour les échanges d'informations entre des capteurs, des actionneurs et des automates réflexes (Ministère de l'industrie et de la recherche)*. Paris, France: Editions KIRK, 1991.
- [51] M. Gault and J. P. Lobert, Contribution to the fieldbus standard IEC/TC65/SC65C/WG6, 1985.
- [52] R. W. Gellie, "PROWAY—A standard for distributed control systems," in *Proc. Specialists' Meeting on Distributed Systems for Nuclear Power Plants*, 1980, pp. 65–100.
- [53] R. Genier, "The THEMIS solar power plant," *Epure*, no. 5, pp. 3–17, Jan. 1985.
- [54] W. George, "A good year for fieldbuses," *Control Instrum.*, vol. 28, no. 8, pp. 26–28, Aug. 1996.
- [55] J. Gertler and J. Sedlak, "Software for process control—A survey," *Automatica*, vol. 11, pp. 613–625, 1975.
- [56] C.-A. Gifford, "A military standard for multiplex data bus," in *Proc. IEEE 1974 National Aerospace and Electronics Conf.*, pp. 85–88.
- [57] K. Grant, Users requirements on time critical communications architectures: Technical report, 1992.
- [58] M. Graube, "The carrier band network and Mini-MAP: low-cost solutions," *Control Eng.*, vol. 33, no. 11, pp. 30–31, Oct. 1986.
- [59] P. Griem and J. W. Bernard, "Considerations for distributed industrial control systems," in *Proc. Int. Telemetering Conf.*, 1976, pp. 686–691.
- [60] G. H. Gürtler, "Fieldbus standardization, the European approach and experiences," in *Feldbustchnik in Forschung, Entwicklung und Anwendung*. Berlin, Germany: Springer-Verlag, 1997, pp. 211–216.
- [61] D. Harrold, "Ethernet everywhere," *Control Eng.*, vol. 46, no. 6, pp. 46–52, 1999.
- [62] E. Hassler, "Industrial master/slave system," *Technische Rundschau*, vol. 72, no. 14, pp. 17–18, Apr. 8, 1980.
- [63] —, "An industrial master-slave system: Organization of communications for decentralized freely-programmable controls," *Technische Rundschau*, vol. 72, no. 17, pp. 25–26, Apr. 29, 1980.
- [64] W. Hodson, "Will fieldbus kill the DCS?," *Control Syst.*, vol. 15, no. 2, pp. 21–23, 1988.
- [65] IEE Colloq. Distributed Process Control: Today and Tomorrow, London, U.K., Mar. 1, 1982.
- [66] *IEC 870-5. Telecontrol Equipment and Systems—Part 5: Transmission Protocol—Section 1: Transmission Frame Formats, Section 2: Link Transmission Procedures, Section 3: General Structure of Application Data, Section 4: Definition and Coding of Application Information Elements*, 1990.
- [67] *IEC Standard 1158-2, Fieldbus Standard for Use in Industrial Control Systems—Part 2 Physical Layer Specification and Service Definition + AMD1*, 1995.
- [68] *IEC. TC44 (Sec)148 Draft Standard for Electrical Equipment of Industrial Machines—Serial Data Link for Real Time Communications Between Controls and Drives, (SERCOS)*.
- [69] *IEC/TC57. TR 870-1-4. Telecontrol Equipment and Systems—Part 1: General Considerations: Basic Aspects of Telecontrol Data Transmission and Organization of Standards IEC 870-5 and 870-6*, 1994.
- [70] *TC65/SC65C. Digital Data Communications for Measurement and Control-Fieldbus for Use in Industrial Control Systems. Part 3 Data Link Service Specification, IEC 1158-3, IEC 65C/160/CDV*, 1996.
- [71] *TC65/SC65C. Digital Data Communications for Measurement and Control-Fieldbus for Use in Industrial Control Systems. Part 4 Data Link Protocol Specification, IEC 1158-4, IEC 65C/161/CDV*, 1996.
- [72] *IEC 61 804-1: Function Blocks (FB) for Process Control—Part 1: General Requirements. 65C 269 CDV*, 2002.

- [73] IEC 61 158-3. TC65/SC65C. *Digital Data Communications for Measurement and Control-Fieldbus for Use in Industrial Control Systems. Part 3 Data Link Service Specification*, 2003.
- [74] IEC 61 158-5. TC65/SC65C. *Digital Data Communications for Measurement and Control-Fieldbus for Use in Industrial Control Systems. Part 3 Application Service Specification*, 2003.
- [75] IEC 61 499-1 *Function Block for Industrial Process Measurement and Control; Part 1 Architecture CDV 65-338*, 2004.
- [76] IEC 61 499-2 *Function Block—Part 2 Software tool requirements—CDV 65-339*, 2004.
- [77] IEC 61 804-2: *Function Blocks (FB) for Process Control—Part 2: Specification of FB Concept and Electronic Device Description Language (EDDL)*, 65C 324 FDIS, 2004.
- [78] *IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, 2002.
- [79] *The MAP Book: An Introduction to Industrial Networking*. Santa Clara, CA: Industrial Networking, 1987.
- [80] F. Inose, K. Takasugi, and M. Hiroshima, "A data highway system," *Instrum. Technol.*, vol. 18, no. 1, pp. 63–67, Jan. 1971.
- [81] "Proway LAN Industrial Data Highway," *Instrum. Soc. Amer.*, ISA-S72.01, 1985.
- [82] Field bus standard for use in industrial control systems: Discussion draft and questionnaire for functional requirements, *Instrum. Soc. Amer.*, 1986.
- [83] Field bus standard for use in industrial control systems: Functional guidelines (various versions), *Instrum. Soc. Amer.*, 1986.
- [84] Fieldbus functional guidelines, *Instrum. Soc. Amer.*, Sep. 1987.
- [85] *ISA-SP50-1987, Fieldbus—Draft Standard*, Sep. 1987.
- [86] *Data Communications, Open System Interconnection—Basic Reference Model*, ISO 7498, 1982.
- [87] *Information Processing Systems, Local Area Networks—Part 3: Carrier Sense Multiple Access-Collision Detection*, 1990.
- [88] *Information Processing Systems, Local Area Networks—Part 4: Token Bus Access Method*, 1990.
- [89] *ISO/IEC IS 9506 Manufacturing Message Specification*, 1990.
- [90] *TR 12178, Industrial Automation, Time Critical Communication Architectures—User Requirements. TC184/SC5/WG2*, 1994.
- [91] *IS 11 898 Road Vehicle-Interchange of Digital Information-Controller Area Network for High Speed Communication*, 1995.
- [92] *DIS 13 236 Information Technology-Quality of Service-Framework*, 1996.
- [93] *WD 13283, User Requirements for TCCS and Network Management Requirements, TC184/SC5/WG2-N582*, 1996.
- [94] H. Kopetz, "Event triggered vs time triggered real time systems," in *Lecture Notes in Computer Science, Operating Systems of the 90s and Beyond*. Heidelberg, Germany: Springer-Verlag, 1990, vol. 563, pp. 87–101.
- [95] H. Kopetz and G. Gründsteidl, "TTP, a time triggered protocol for fault tolerant real time systems," *Computer*, vol. 27, no. 1, pp. 14–23, 1994.
- [96] H. Kopetz, W. Elmenreich, and C. Mack, "A comparison of LIN and TTP/A," in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS 2000)*, pp. 99–107.
- [97] M. Kronental, "Toward the standardization of real time operating system kernels," presented at the 2nd IFAC/IFIP Symp. Software for Computer Control (SOCOCO 1979), Prague, Czechoslovakia.
- [98] L. Lamport, "Time, clocks and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978.
- [99] N. Laurance, *MMS Over the Three Layer Stack*, Oct. 1992.
- [100] G. Le Lann, "Distributed systems, toward a formal approach," in *Proc. IFIP World Congr.*, 1977, pp. 155–160.
- [101] G. Le Lann and N. Rivierre, "Real time communications over broadcast networks: The CSMA-DCR and the DOD/CSMA-CD protocols," in *Proc. RTS'94*, pp. 67–84.
- [102] M. Leon Chavez M, "Fieldbus and real time MAC protocols," presented at the IFAC Conf. SICICA 2000, Buenos Aires, Argentina.
- [103] P. Leviti, "Tutorial on the Data Link Service Specification, Part A and B, Digital Data Communications for Measurement and Control," IEC, Doc. 65C/166/INF, Jun. 1966.
- [104] —, "IEC 61 158: An offence to technicians," in *Proc. IFAC Int. Conf. Fieldbus Systems and Their Applications (FET 2001)*, pp. 9–16.
- [105] LonWorks, Documentation Echelon Corp., Palo Alto, CA, 1995.
- [106] G. Loose, "A user view of the user layer," *C&I*, vol. 28, no. 5, pp. 60–61, May 1996.
- [107] Z. Mammeri and P. Lorenz, "Integration of temporal mechanisms in communication protocols for time critical distributed systems," presented at the 12th IFAC Workshop on Distributed Computer Control Systems, Toledo, Spain, 1994.
- [108] J. Marx, "MAP and EPA: on the road to connectivity," *Instrum. Control Syst.*, vol. 60, no. 12, pp. 28–30, Dec. 1987.
- [109] S. McClelland, "A Hart to Hart with Rosemount," *Sens. Rev.*, vol. 9, no. 2, pp. 71–74, Apr. 1989.
- [110] M. J. McGowan, "Process bus protocol orchestrates distributed or centralized control," *Control Eng.*, vol. 27, no. 9, pp. 129–132, Sep. 1980.
- [111] G. A. Mitchell, "Ethernet's in control," *Control Eng.*, vol. 47, no. 5, pp. 46–54, May 2000.
- [112] H. M. Morris, "Distributed system makes wide use of bubble memories," *Control Eng.*, vol. 29, no. 1, pp. 68–70, Jan. 1982.
- [113] L. Motus L., "Time concepts in real-time programming," in *Proc. IFAC/IFIP Workshop Real Time Programming*, 1992, pp. 1–10.
- [114] J. A. Murphy, "Token-passing protocol boosts throughput in local networks," *Electronics*, vol. 55, no. 18, pp. 158–163, Sep. 8, 1982.
- [115] N. Nakano, "Time critical communication architecture in factory automation," in *IFIP Information Infrastructure Systems for Manufacturing*, H. Yoshikawa and J. Goosenaerts, Eds. New York: Elsevier, 1993, pp. 363–374.
- [116] N. Navet, "Evaluation de performances temporelles et optimization de l'ordonnement de tâches et de messages," Ph. D. dissertation, INPL, Nancy, France, Nov. 1999.
- [117] P. Neumann, "Locally distributed automation—but with which fieldbus system?," *Assembly Autom.*, vol. 19, no. 4, pp. 308–312, 1999.
- [118] D. Norris, "Smart distributed system: distributed control for factory floor automation," presented at the FieldComms'95 Making the Most of the Fieldbus Conf., vol. 1, Titchfield, U.K..
- [119] M. Ochsuer and M. Schrier, "To fieldbus or not to fieldbus," *InTech*, vol. 44, no. 10, pp. 44–48, 1997.
- [120] Open DeviceNet Vendor Assoc. (ODVA), *DeviceNet Specification Release 2.0*, 2001.
- [121] J. S. Ostroff, "Formal methods for the specification and design of real time safety critical systems," *J. Syst. Softw.*, vol. 18, no. 1, pp. 12–24, 1992.
- [122] D. Pancucci, "Peace breaking out on fieldbus," *Manuf. Comput. Solutions*, vol. 5, no. 7, pp. 48–51, 1999.
- [123] D. Paret, *Le réseau CAN, Controller Area Network*. Paris, France: Dunod, 1996.
- [124] D. Paret and C. Fenger, *The I2C Bus From Theory to Practice (book and disk)*. New York: Wiley, 1997.
- [125] M. Pearson, "Implementing MAP/EPA in the manufacturing cell," *FMS Mag.*, vol. 5, no. 1, pp. 15–18, Jan. 1987.
- [126] J. F. Peyrucat, "Interautomaton communication via networks," *Mesures, Regulation, Automatisme*, vol. 48, no. 1, pp. 35–37, 39, 41, Jan. 24, 1983.
- [127] T. P. Phinney, D. Brett, D. McGowan, and Y. Kumeda, "FieldBus—Real time comes to OSI," in *Proc. 10th Annu. Int. Phoenix Conf. Computers and Communications*, 1991, pp. 594–599.
- [128] R. Piggin, K. Young, and R. McLaughlin, "The current fieldbus standards situation—a European view," *Assembly Autom.*, vol. 19, no. 4, pp. 286–289, 1999.
- [129] J. Pimentel, "Fieldbus application layer: functionality and models," in *Proc. 8th IFAC Distributed Computer Control Systems*, 1988, pp. 15–20.
- [130] P. Pleinevaux and J. D. Decotignie, "Time critical communication networks: Field busses," *IEEE Network*, vol. 2, no. 3, pp. 55–63, May 1988.
- [131] K. Pluhar, "Introducing four more new integrated distributed control systems," *Control Eng.*, vol. 27, no. 8, pp. 45–57, Aug. 1980.
- [132] L. Pouzin, "Virtual circuits versus datagrams—technical and political problems," in *Proc. Nat. Computer Conf.*, 1976, pp. 483–494.
- [133] L. R. Qualls, "Advantages of a time division multiplex data bus for remotely piloted vehicle built-in test," in *Proc. IEEE 1976 Nat. Aerospace and Electronics Conf.*, pp. 203–207.
- [134] N. Raja, "Static and dynamic polling mechanisms for fieldbus networks," *Oper. Syst. Rev.*, vol. 27, no. 3, pp. 34–45, 1993.
- [135] A. Reeve A, "Which fieldbus will you use—and when?," *Control Instrum.*, vol. 25, no. 5, pp. 67–70, 1993.
- [136] C. W. Rose and J. D. Schoeffler, "Microcomputers in instrumentation and data acquisition systems," *Anal. Instrum.*, vol. 12, pp. 157–163, 1974.

- [137] G. Saba, J. P. Thomesse, and Y. Q. Song, "Space and time consistency qualification in a distributed communication system," in *Proc. IMACS/IFAC Int Symp. Mathematical and Intelligent Models in System Simulation*, vol. 1, 1993, pp. 383–391.
- [138] T. Sauter and M. Wollschlaeger, "Feldbussysteme-Historie, Eigenschaften und Entwicklungstrends," *Informationstechnik und Technische Informatik*, vol. 42, pp. 7–16, 2000.
- [139] V. Schiffer, "The CIP family of fieldbus protocols and its newest member—Ethernet/IP," in *Proc. Conf. Emerging Technologies and Factory Automation (ETFA)*, 2001, pp. 377–384.
- [140] T. H. Schwalenstocker, "A process control system using multibus," in *Proc. 1st Annu. Control Engineering Conf.*, 1982, pp. 133–137.
- [141] "Working Draft 0.2, Low voltage switchgear and controlgear—Part 5," CENELEC, EN 60947-5, 1997.
- [142] Y. Q. Song, F. Simonot, and J. P. Thomesse, "Message sojourn time for TDM schemes with any buffer capacity," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1013–1021, Feb./Mar./Apr. 1995.
- [143] M. Soutif, "Rapport sur l'Industrie des Instruments de Mesure," Ministère de la recherche, Paris, France, 1982.
- [144] J. A. Stankovic, "Misconceptions about real time systems," *Computer*, vol. 21, no. 10, pp. 10–19, Oct. 1988.
- [145] K. M. Sturgis, "GM's manufacturing automation protocol," in *Proc. Conf. Local Net '84*, pp. 13–22.
- [146] J. W. Szymanski, "Embedded internet technology in process control devices," in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS 2000)*, pp. 301–308.
- [147] J. P. Thomesse, "A new set of software tools for designing and realizing distributed systems in process control," in *Proc. IFAC/IFIP Workshop Real Time Programming*, 1977, pp. 47–54.
- [148] J. P. Thomesse and P. Noury, "Communication models client-server vs producer-distributor-consumer," ISO, TC 184/SC5/WG2-TCCA, 1989.
- [149] J. P. Thomesse and T. Lain&é, "The field bus application services," in *Proc. 15th Conf. IEEE-IES Factory Automation (IECON'89)*, pp. 526–530.
- [150] J. P. Thomesse, P. Lorenz, J. P. Bardin, P. Leterrier, and T. Valentin, "Factory instrumentation protocol: model, products and tools," *Control Eng.*, vol. 38, no. 12, pp. 65–67, 1991.
- [151] J. P. Thomesse, "Le réseau de terrain FIP," *Revue Réseaux et Informatique Répartie*, vol. 3, no. 3, pp. 287–321, 1993.
- [152] —, "Time and industrial local area networks," in *Proc. COMPEURO'93 Paris*, pp. 365–375.
- [153] —, "A review of the fieldbuses," *Annu. Rev. Control*, vol. 22, pp. 35–45, 1998.
- [154] —, "Fieldbuses and interoperability," in *Control Engineering Practice*. New York: Pergamon, 1999, vol. 7, pp. 81–94.
- [155] J. P. Thomesse and M. Leon Chavez, "Main paradigms for current fieldbus concepts," in *Proc. Fieldbus Conf. (FeT'99)*, pp. 2–15.
- [156] K. Tindell, A. Burns, and J. Welling, "Analysis of real time communication," *J. Real Time Syst.*, vol. 9, pp. 147–171, 1995.
- [157] —, "Calculating controller area network message response times," *Control Eng. Practice*, vol. 3, no. 8, pp. 1163–1168, 1995.
- [158] P. H. Troutman, "A digital link for controllers and valves," *Instrum. Technol.*, vol. 25, no. 7, pp. 55–57, Jul. 1978.
- [159] Y. Tsukada, "Trends in process control systems," *Jpn. Electron. Eng. (JEE)*, no. 110, pp. 24–27, Feb. 1976.
- [160] "UML profile for schedulability, performance, and time specification," Object Management Group (OMG), Needham, MA, Jan. 2002.
- [161] F. Vasquez and G. Juanole, "Pre run time schedulability analysis in fieldbus networks," in *Proc. IEEE Conf. Industrial Electronics (IECON'94)*, pp. 1200–1204.
- [162] L. Vega-Saenz and J. P. Thomesse, "Time in distributed systems—cooperation models and communication types," in *Proc. 5th Workshop Future Trends of Distributed Computing Systems*, 1995, pp. 41–49.
- [163] M. Welburn, "Take the bus. . . but don't get on the wrong one," *Control Instrum.*, vol. 30, no. 8, pp. 41–42, 1998.
- [164] G. G. Wood, "Fieldbus services under MAP," in *Proc. ISATA 17th Int. Symp. Automotive Technology and Automation*, vol. 1, 1987, pp. 87 134/1–87 134/9.
- [165] —, "Survey of LAN's and standards," *Comput. Stand. Interfaces*, pt. 6, no. 1, pp. 27–36, 1987.
- [166] —, "Current fieldbus activities," *Comput. Commun.*, vol. 11, no. 3, pp. 118–123.
- [167] —, "International standards emerging for fieldbus," *Control Eng.*, pt. 2, vol. 35, no. 10, pp. 22–25.
- [168] H. Zimmermann, "OSI reference model. The ISO model of architecture for open system interconnection," *IEEE Trans. Commun.*, vol. COM-28, no. 4, pp. 425–432, Apr. 1980.



Jean-Pierre Thomesse was born in Neufchâteau, France, in 1947. He received the DEA degree (Postgraduate Diploma, preparation for doctoral studies) in computer science from the University of Nancy, Nancy, France, in 1971 and the Ph.D. and Doctorat d'Etat (national diploma giving the right to lead research projects) degrees from the Institut National Polytechnique de Lorraine (INPL), Nancy, in 1974 and 1980, respectively.

He has been Professor at the INPL since 1981.

He was an active participant in the development of the WorldFIP fieldbus, which began in 1982. He was then involved in its standardization. The current application of his knowledge and experience is in real-time communication and embedded intelligent systems with different kinds of application in industry, cars, building automation, and health care.