FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 7, N° 1, 2008, pp. 1 - 26

Invited Paper

FIELDBUSSES IN MANUFACTURING AUTOMATION

UDC 007.5:621.39

Žarko Čučej, Karl Benkič

University of Maribor, Slovenia

Abstract. *Remote control, as well as remote supervising systems, are the backbone of todays manufacturing automatization. Both are enabled by different kinds of computers, programmable logic controllers, etc., connected by different data networks. These networks differ from other data networks, at least in regard to the following (i) shape of traffic, (ii) any available processing capabilities for data interpretation and validation are usually very limited, and (iii) most importantly - data error renaissance, as well as "on-time" delivering of correct data, is expected from them. This article contains a brief overview of those data communications networks used in manufacturing, as well as in remote sensing and monitoring applications, with emphasis on CAN, TTCAN, TTP and Wireless Sensor Network fieldbusses.*

Key words: *Fieldbusses, Wireless Sensor Networks, Automatization*

1. INTRODUCTION

Nowadays many industries technologies can be designated according to shortness, such as CAD (Computer-Aided Design), CCQ (Computer-Controlled Quality) … CIM (Computer-Integrated Manufacturing). They are based on:

- computers used as information processing machines,
- networks linking all processes in a factory,
- new knowledge,

giving synergy in competitiveness, more humane production, lower pressure on the environment, etc.

CIM has the key position among them. Today, CIM is not just a dream, it is becoming - step by step - the key of all modern organization, not only manufacturing. Electronically integrated information necessary for manufacturing (Fig. 1) using different communication systems, allows every database to be accessed by authorized users (computers, peripherals, manufacturing devices) anywhere in the company. This makes possible produc-

Received January 21, 2008

tion models such as "just-in-time" from the logistic side to mass production by tracing any single item considered in manufacturing. In essence, it builds on the distributed environment in which most companies already operate, and aggregates various constituents into a single data base, driving a variety of systems.

Employing computers in different aspects of manufacturing happened almost at the same time as the development of miniframes/minicomputers such as PDP-11, and others. In early applications they were directly linked with peripheral devices. Later on, with the development of first data networks capable of working in real-time, the door opened to the fast growing development of integrated computers in automatization, by linking those computers used at different levels of a manufacturing organization, the periphery and manufacturing devices such as CNC machines, as well as manufacture management. This involved a serious step toward CIM.

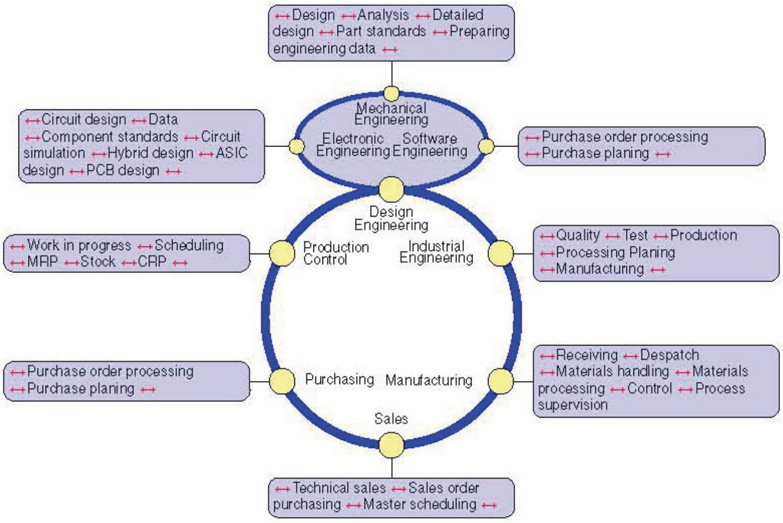

↔Design ↔Analysis ↔Detailed design ↔Part standards ↔Preparing engineering data ↔
↔Circuit design ↔Data ↔Component standards ↔Circuit simulation ↔Hybrid design ↔ASIC design ↔PCB design ↔
Mechanical Engineering
Electronic Engineering
Software Engineering
↔Purchase order processing ↔Purchase planing ↔
Design Engineering
↔Work in progress ↔Scheduling ↔MRP ↔Stock ↔CRP ↔
Production Control
Industrial Engineering
↔Quality ↔Test ↔Production ↔Processing Planing ↔Manufacturing ↔
↔Purchase order processing ↔Purchase planing ↔
Purchasing
Manufacturing
↔Receiving ↔Despatch ↔Materials handling ↔Materials processing ↔Control ↔Process supervision
Sales
↔Technical sales ↔Sales order purchasing ↔Master scheduling ↔

Fig. 1 Information flow in computer integrated manufacturing.

1.1. Today's trends in automation

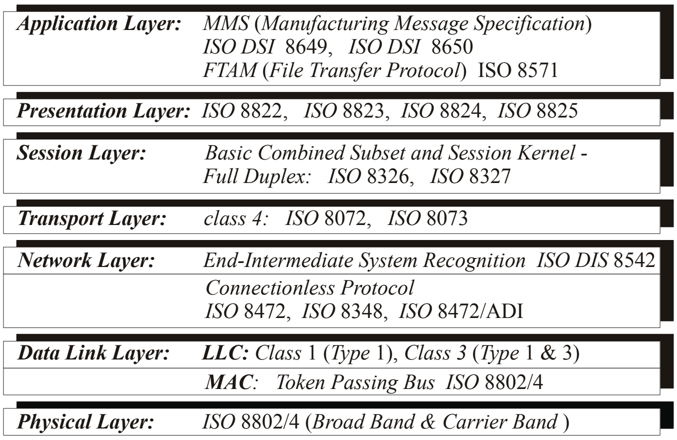
Advances in networking and information technologies are transforming factory-floor communication systems into a mainstream activity within industrial automation. It is now recognized that future industrial computer systems will be intimately tied to real-time computing and to communication technologies. For this vision to succeed, complex heterogeneous factory-floor communication networks (including mobile/wireless components) need to function in a predictable, flawless, efficient and interoperable way [1]. The future of distributed automation requires more mechanisms for distribution of the automation functions for various reasons [2]:

- centralized supervision and control of decentralized technological plants,
- remote control, configuring, commissioning, parameterizations, maintenance of distributed automation systems,
- including remote experts or external knowledge for the plant's operation and maintenance.

The resulting automation system has to offer location-based and context-sensitive services to guarantee suitable local and remote functions for different user needs, and requires real-time data transmission, safety and security mechanisms [3].

1.2. Manufacturing Automatization Protocol: MAP

Over recent decades, there have been many attempts to develop a universal data network to serve different requests during manufacturing data exchange. Many of them, like MAP (Manufacturing Automatization Protocol), where developers led by the General Motors Company, have developed numerous standards based on Open Systems Interconnection (OSI) reference model. This reference model (Fig. 2) failed, because:



Application Layer:	*MMS* (*Manufacturing Message Specification*) *ISO DSI* 8649, *ISO DSI* 8650 *FTAM* (*File Transfer Protocol*) ISO 8571
Presentation Layer:	*ISO* 8822, *ISO* 8823, *ISO* 8824, *ISO* 8825
Session Layer:	*Basic Combined Subset and Session Kernel -* *Full Duplex:* *ISO* 8326, *ISO* 8327
Transport Layer:	*class 4:* *ISO* 8072, *ISO* 8073
Network Layer:	*End-Intermediate System Recognition* *ISO DIS* 8542
	Connectionless Protocol *ISO* 8472, *ISO* 8348, *ISO* 8472/ADI
Data Link Layer:	***LLC:*** *Class* 1 (*Type* 1), *Class 3* (*Type* 1 & 3)
	MAC*:* *Token Passing Bus* *ISO* 8802/4
Physical Layer:	*ISO* 8802/4 (*Broad Band & Carrier Band*)

Fig. 2 MAP 3.0 suite, 1988.

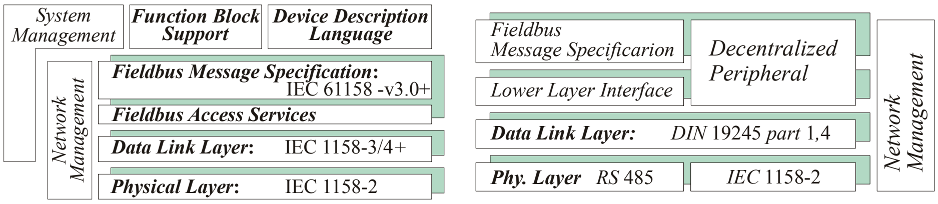
- suggested solutions were too complex and, consequently, too expensive,
- presumption that the ISO/OSI reference model would prevail in data communications was wrong (recently, IP technologies with different upper layers have become the mainstream of any kind of digital communications),
- pushing new technologies instead of integrating the existing ones diminished interests of the companies which already produce different automation equipment capable of communicating over special local networks – fieldbusses,
- most of the proposed technology was immature,
- communication issues were clearly misunderstood,
- general, unified approach such as MAP requires enormous effort from companies in regard to top-down implementation.

An attempt was made to minimize some MAP by downsizing the MAP to miniMAP, which uses only four layers from ISO/OSI reference model, but proved too complex and demanding, and still required the introduction of new technology instead of integrating the existing.

The efforts put into MAP and miniMAP development, although they had both been non-functional for more than a decade, still has an influence on today's research and development, for example the effort to unify planning and design to use fieldbusses. Such an effort was the InterOperability System Project, evolved through the FOUNDATION^{TM} Fieldbusses Initiative.

2. Industrial Local Area Networks: Fieldbusses

The development of industrial networks started around 1980 with the development of smart sensors for use in digital control. The main concerns for industrial networks are reliability, responsiveness and predictability, and not only data throughput as the case is in general data local area networks [4], [5]. Since industrial networks are usually used in some application field, for example, cars, buildings, motion control, process control etc, the coined name for them is a fieldbus. Many fieldbus standards for industrial networks developed by different companies compete on the market today [6]. Some of them are internationally standardized, and some are de-facto standard for dedicated application field.


System Management
Function Block Support
Device Description Language
Network Management
Fieldbus Message Specification: IEC 61158 -v3.0+
Fieldbus Access Services
Data Link Layer: IEC 1158-3/4+
Physical Layer: IEC 1158-2
Fieldbus Message Specificarion
Lower Layer Interface
Decentralized Peripheral
Data Link Layer: DIN 19245 part 1,4
Phy. Layer RS 485
IEC 1158-2
Network Management

(a) Profibus protocol stack (b) FieldbusTM protocol stack

Fig. 3 Protocol stacks of two prominent fieldbusses.

All fieldbusses are based on the ISO/OSI reference model. Initially, they have defined only a physical and data link layer. Experiences from MAP as well from usage in applications, where devices from different producers are linked, show that the application layer is very important in automatization design. The application layer – where Fieldbus Message Specification (Fig. 3) derived from MMS in MAP has the main role – makes it clearer and more uniform. Further improvement came with the introduction of a user layer, where the framework is defined for performing control functions in the field devices (Fig. 3b). In general, fieldbusses can be divided into "event driven" (Table I) and "time triggered" (Table II). Only CAN, TTCAN, TTP and FOUNDATION^{TM} fieldbusses are briefly reviewed from fieldbusses mentioned in the table I and II.

3. Controller Area Network: CAN

CAN is a real-time, serial fieldbus first introduced by BOSCH in the early 1980's. Today, CAN is one of the most popular fieldbus protocols [5], [6]. CAN bus was developed for use in cars. It was developed as an open system (as almost all other buses of this new generation). Its open-system allowed CAN to become widely used in other industrial automation applications. CAN was internationally standardized in 1993 by ISO (ISO 11898-1). Since then, ISO have introduced two CAN standard extensions: ISO 11898-2 defining high speed can and ISO 11898-3 which defines fault tolerance. CAN protocol is based on the ISO/OSI reference model. It corresponds to the first two layers (Fig. 4).

The medium used for carrying information is not part of any standardization (which corresponds to the open system; and the user can use mediums allowing higher transmitting speeds). In general, twisted pair or optic fibre is used. CAN define a carrying medium where two stages can be clearly defined: recessive (corresponds with bit-state 1) and dominant (corresponds with bit-state 0) states. CAN bus can operate from 1 Mb/s inside a 40 m bus length to 5 kb/s achieved inside a 10 km bus length.

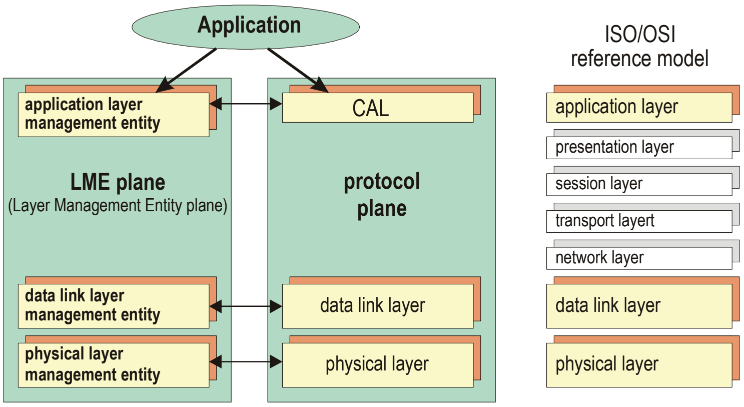
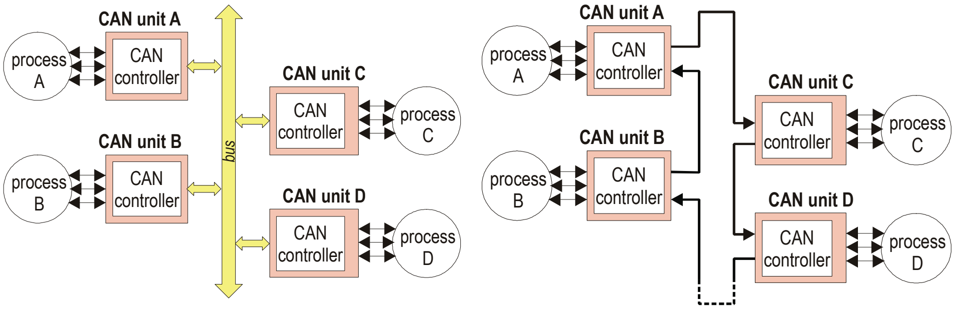


Fig. 4 CAN protocol.

As previously mentioned, physical bus is not part of the CAN standard. However, today we can find two main variants of physical bus used for CAN. These are twisted pair, where the carrying medium is an electrical current, and fibre optic, where the medium represents the light. Two topologies are commonly used with these two physical busses: bus topology for twisted pair (Fig. 5a) and token ring used with optic fibre (Fig. 5b).



(a) twisted pair bus topology

(b) fibre optic ring topology

Fig. 5 Possible network topology at CAN.

Twisted pair bit is defined by differential voltage and fibre with light (on or off). CAN defines message addressing which means CAN entity cannot be directly addressed. Message address is often called message ID. Message ID can be 11 bit (standard ID) or 29 bit (extended ID) long. In terminology there are two "types" of message. Standard message uses 11 bit ID and extended message uses 29 bit ID [7]. Corresponding to message (not station) addressing, CAN bus can be very easily upgraded with new devices. Adding devices or removing them does not affect devices already on the bus. No re-programming or new settings are needed. CAN uses frames for data output and internal communications. There are 4 types of frames known in CAN. The data frame used for data conveyance, the remote frames for data request, Error frame for error signalling and Overload frame. Data and remote frame are used in standard or extended CAN messages while error and overload frame are used only in standard messages (Fig. 6).

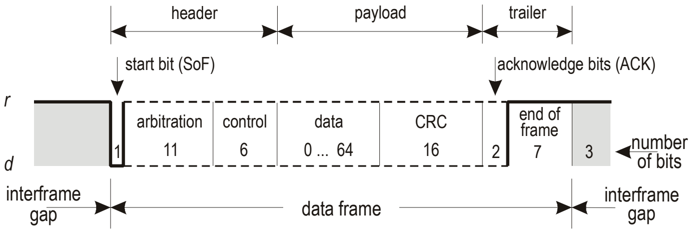


Fig. 6 Data and remote frame.

The data-frame can be filled-up with a maximum of 8 data bytes protected by CRC. The remote frame is basically a data-frame without data (0 data bytes). When data is needed, remote frame is sent for querying the data. Entity responds to the remote frame with data-frame using remote-frame's ID, and containing data requested by the remote frame.

CAN is event-triggered, which means that the times when the messages are sent are unpredictable. This can result in message collisions, because every node, in the worst-case scenario, could send a message to the bus at the same time. In order to cope with this, CAN uses the CSMA/AMP (Carrier Sense Multiple Access with Arbitration by Message Priority) technique for media access. In CSMA/AMP, every message to be sent is assigned with priority. A node can only begin sending message if the bus is free. If collision is detected when the message is already being transmitted, any collision is resolved via message priority. All, except the node sending the highest prioritized message (lower ID), must stop sending. This means that the most important message never gets interrupted (Fig. 7). In the worst case, it must wait for a free bus [8], [9].

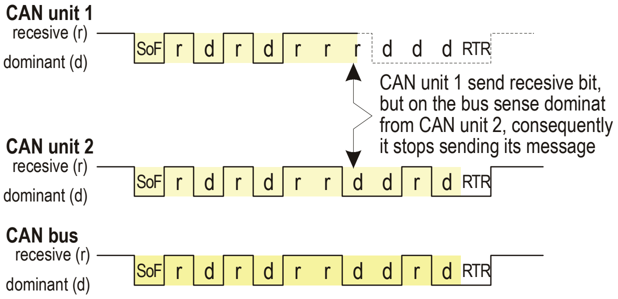

CAN unit 1
recesive (r)
dominant (d)
SoF r d r d r r r d d d RTR
CAN unit 1 send recesive bit, but on the bus sense dominat from CAN unit 2, consequently it stops sending its message
CAN unit 2
recesive (r)
dominant (d)
SoF r d r d r r d d r d RTR
CAN bus
recesive (r)
dominant (d)
SoF r d r d r r d d r d RTR

Fig. 7 Arbitration by Message Priority.

The identifier (ID) of the message defines the message's priority. The lower the value the identifier has, the higher the message priority. As previously mentioned, bit 0 is dominant and 1 is a recessive bit. If the node sends a 1 on the bus and senses a collision with a 0, it stops sending. But if it senses a 1, it does not know if it is lower or higher than the other, therefore, it sends the next bit of the identifier. At some point there will only be one node still sending, and the message will be sent [8], [9].

Table 1 Summary of the basic characteristics of some popular event fieldbusses

Network	Speed	# of nodes	Arbitration	Cable type	Primary application
World FIP	31.2Kb/s, 1Mb/s, 5Mb/s	64 to 256	Bus arbiter	Twisted pair, optic fibre	Real time control, Process/Machine
Lon Works	1.25Mb/s	32000	Predictive CSMA/CA	Twisted pair, optic fibre	Appliance control
MACRO	100Mb/s	256	Ring management	Twisted pair, optic fibre	Motion control
Profibus	9.6Kb/s - 12Mb/s	32 or 127	Hybrid MAC	Twisted pair, optic fibre	Factory automation
CAN	1 Mb/s	type a: 2048	CSMA/CD enhanced	Twisted pair, optic fibre	Sensor/actuators automotive

Table 2 Summary of the basic characteristics of some popular time driven fieldbusses

Network	Speed	# of nodes	Arbitration	Cable type	Primary application
SERCOS	2-4Mb/s	256	Ring management	optic fibre	Motion control
TTCAN	1 Mb/s	type a: 2048	CSMA/CD enhanced	Twisted pair, optic fibre	Sensor/actuators Automotive
TTP	25Mb/s	--	TDMA	Twisted pair, optic fibre	Automotive, hard-real time

Sometimes a lot of messages with high ID need to be transmitted. Cases like this affect messages with lower IDs (which can wait indefinitely). It is up to the system's designer as to how to resolve conflicts like this, but the general practice is to reserve some especially low IDs for these cases. When the message is "waiting too long" the system assigns new low ID to the message, and sends it.

CAN specification also states how the error detection and management should be implemented. There are five different methods for discovering an incorrect message:

- *Monitoring*. The sender checks that the bit sent is the same as on the bus. If a 0 is sent and a 1 is detected, an error has occurred.
- *Bit stuffing*. CAN uses NRZ coding for efficient error frames detection. An error frame starts with either six 1's or six 0's. No other messages will contain these strings as the transmitter inserts a stuff bit with the complementary value after five consecutive equal bits. The receivers then remove this stuff-bit.
- *Frame check*. Every node checks the received frame against the specification. If an error is found, the node signals a "format error".

- *ACK errors*. All nodes should report if the reception of the message was correct or not by writing a 0 (ok) or a 1(not ok) in the ACK field of the message. A 0 overrides a 1, so the transmitter will only know if at least one other node received a correct message.
- *Cyclic Redundancy Check (CRC)*. A checksum is calculated over the message before sending and is transmitted together with the message. The receiver does the same calculations. If the checksum calculated is equal to the checksum received it is likely that the message has arrived correctly [9].

The CAN protocol is good enough for most real-time applications but, for safety-critical systems, the event-triggered approach is too weak.

4. TTCAN

TTCAN is a higher layer protocol, above the unchanged CAN protocol, that synchronizes communication and provides a global time-base. With synchronized nodes, messages can be sent at specified times, and known by all. This leads to a non-competitive environment and hence the latency is predictable. The TTCAN protocol is divided into two levels and is standardized as part of the CAN standard in ISO 11898-4. In the standard, four major parts are defined:

- Time-triggered execution of CAN with a central Time-master, periodic messages, event-triggered messages and Time-master synchronization.
- Fault tolerance of this Time-master.
- Drift correction and Global-time.
- Event synchronized – time-triggered communication.

TTCAN implements the time-triggered architecture; hence the time-triggered schedule is determined prior to system start. The schedule consists of many time-slots. In each slot only one message can be sent, so the competition for the bus is avoided and the message latency is not dependent of the number of nodes connected to the network. In the worst case, latency can be calculated and the protocol is predictable [10]–[14].

4.1 Medium Access

TTCAN uses a version of the TDMA access method when sending or receiving messages [14]. The Time-division Multiple Access (TDMA) bandwidth allocation scheme divides the time domain into smaller time slots, or time windows. Network nodes are assigned different slots when they have the right to use the bus for transmission. The sequence of time slots when all nodes can send their messages forms a basic cycle. Several different basic cycles can be forming a matrix cycle. All basic cycles are of the same size in the temporal domain, but can differ in the layout. When a matrix cycle is ended, the transmission scheme starts over again by repeating the matrix cycle.

4.2. Elements of the protocol

The TTCAN protocol consists of three core elements [10], [14]:

- *The Reference Message*. The CAN communication is initiated by a Reference-message. This Reference-message is the basis for any protocol activity. The Reference message is sent periodically by the Time-master. The transmission pattern is defined by a sequence of so called "Time windows".

- *The basic cycle*. The period between two consecutive reference messages is called a basic cycle (Fig. 8).

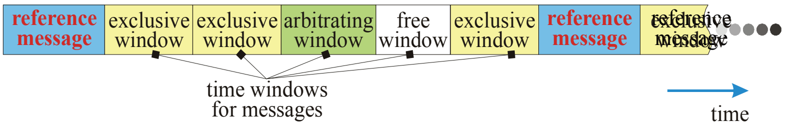
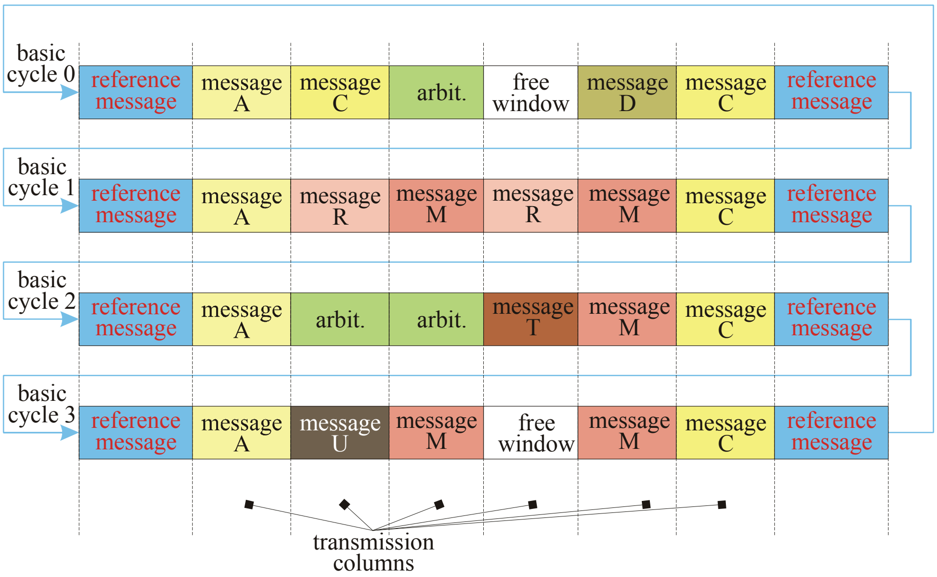


Fig. 8 The Basic Cycle.

The basic cycle contains Time-windows pre-defined for the transmission of messages in the system. The windows within a basic cycle can be of varying sizes but the length of a basic cycle is always the same. The time windows can be of different types to allow the transmission of both types of messages: periodic state messages and spontaneous states event messages.

- *The matrix cycle*. In order to offer a more flexible time triggered system, several basic cycles can build a communication matrix (Fig. 9). If only one basic cycle were allowed, it would be hard to adapt applications with different control loops and different deadlines to the communications protocol. The existence of a system matrix allows for individual patterns among the applications, running on the same network. The matrix cycle defines a message transmission schedule.



basic cycle 0	reference message	message A	message C	arbit.	free window	message D	message C	reference message
basic cycle 1	reference message	message A	message R	message M	message R	message M	message C	reference message
basic cycle 2	reference message	message A	arbit.	arbit.	message T	message M	message C	reference message
basic cycle 3	reference message	message A	message U	message M	free window	message M	message C	reference message

Fig. 9 The Matrix Cycle: Transmission Columns and System Matrix.

When the protocol has reached the end of the message transmission schedule, it immediately starts over at the beginning of the schedule.

4.3. Time windows type

In TTCAN three different types of time windows are defined: *exclusive*, *arbitrating* and *free* time-windows. It allows different kinds of messages to be sent:

- Periodic messages within Exclusive time-windows. No other message can be scheduled in the same window. The automatic retransmission of CAN messages is not allowed in an exclusive time-window.
- Sporadic messages within Arbitrating time-windows (Fig. 10). It can be assigned to more than one node. The possible bus conflicts are resolved using the CAN arbitration mechanism.

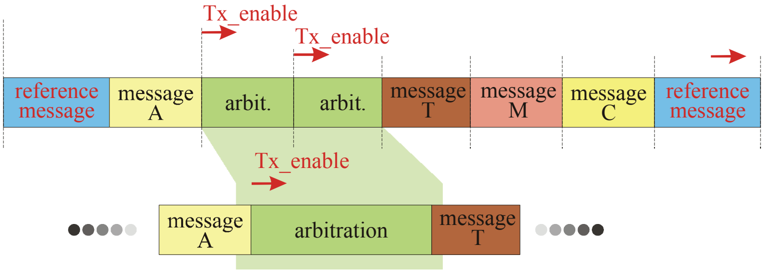


Fig. 10 Merged Arbitrating Windows.

Consecutive Arbitrating time-windows may be merged to allow longer periods for arbitrating messages. The transmission of a spontaneous message in an Arbitrating time-window can only be started if there is sufficient time remaining. The retransmission is enabled inside the merged arbitrating window.

- Free space for extensions within Free-time window. During the design of a transmission schedule, it is also possible to reserve time windows for future use. It could be either for new nodes or for giving existing nodes more bandwidth. A free time window cannot be assigned to a message unless it is converted to either an exclusive or an arbitrating time window first.

4.4. Sending, receiving and network time unit

Within a basic cycle, time-triggered actions are driven by the progression of time. This perception of time is called "Cycle-time" and is reset at the beginning of each basic cycle. The connections between the Cycle-time and the matrix time are called time marks. A time mark specifies the beginning of a time window and could be $Tx_{Trigger}$, for transmitting, or $Rx_{Trigger}$, for receiving. A time-mark consists of a base mark that determines the number of the first basic cycle after the beginning of the matrix when the messages must be processed, and a repeat-count done (Fig. 6). The repeat-count specifies the number of basic cycles between two successive transmissions/receptions of the message.

The granularity of the cycle time in TTCAN network is the network time unit (NTU). The calculation of NTU is different in the two levels of TTCAN. In level one NTU is equal in duration to the nominal CAN bit time and is a fraction of the physical second (2^{-2}) in TTCAN level two.

4.5. Level differences

It is mandatory that each TTCAN controller be equipped with a local time counter. The local time is incremented at each NTU. Another timer is necessary to achieve the cycle-based sending/receiving of the messages and the synchronization to the Time Master of the TTCAN network – *Cycle-time*. *Global-time* becomes necessary for distributed system design.

Level one is a less demanding version of TTCAN where timing is only based on the transmissions of reference messages. There is no clock correction between reference messages to keep the clock at the same updating frequency, which means that there can only be low precision schedules. This level can be implemented as a software layer on top of CAN hardware.

The synchronization at level two is at a finer level. The NTU is a fraction of a physical second and is updated 2^2-times per second. This level also provides a global time, which is continuously updated at the nodes and all nodes adapt their updating frequency to that of the Time-master, by the use of a drift compensation algorithm. The global time can, in turn, be synchronized to an external clock, to allow TTCAN to be interfaced with other networks.

4.6. Event synchronized - time triggered communication

It is unnecessary for the reference message to be sent periodically – in some applications it can be advantageous to trigger its transmission by an external event to the bus. In this case the application has to signal this to all other bus members in the reference message preceding this event's synchronization by setting one bit in the previous reference message, the Next_is_Gap bit. The maximum length of the gap must be defined in every application (Fig. 11). A Frame-synchronization pulse is generated in each node for each frame in the CAN network. This pulse is, disregarding signal propagation times, synchronous in the whole network [13], [14].

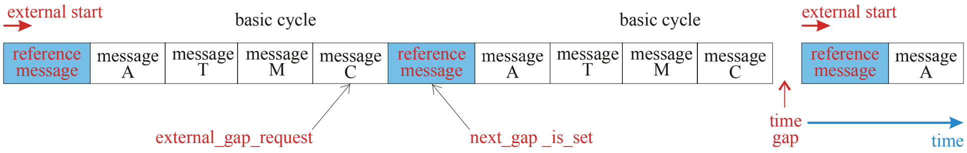


Fig. 11 Event synchronization.

4.7. Event synchronized - time triggered communication

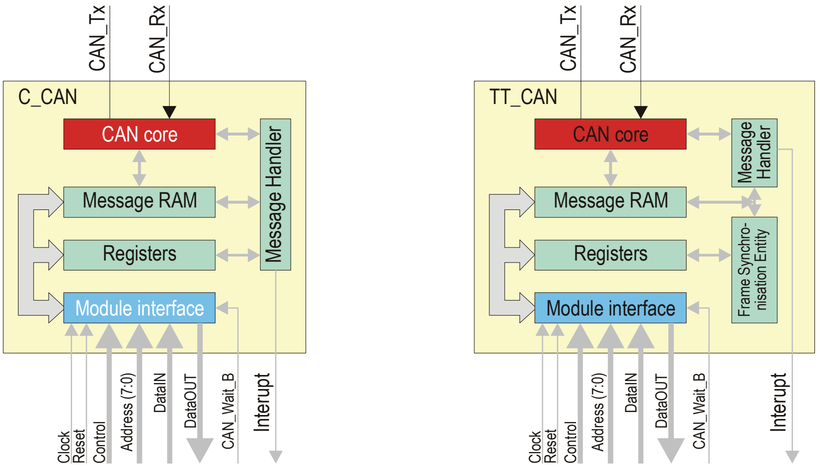
In TTCAN, each valid reference message generated by time master starts a new basic cycle. A new basic cycle resets the node Cycle time in each node. The basic cycle starts when synchronization of the cycle time is finished. Basic cycle consists of time windows for messages (Fig. 11). The starts of time windows (exclusive or arbitration time windows) are defined off-line and must be known to all interested nodes in the network. Several different basic cycles can form a cycle matrix. For TTCAN node, it is unnecessary to know the whole system cycle matrix. It is enough to know only those cycle times needed for its message transmitting.

4.8. TTCAN controllers in the network

TTCAN implementation can be made as software module implementation on existing CAN hardware modules or the TTCAN hardware modules can be used (Fig. 12). It depends on the bit-rate and the number of messages in system cycle matrix, and how high TTCAN loads CPU. In high bit-rates and many messages, software implementation is not recommended, or it can be done only on high performance CPU's (ARM7 or 9, DSP,…). TTCAN does not need any special bus. It can run on conventional CAN buses (twisted pair or optic fibre).

5. Time-Triggered Protocol

TTP protocol contains a time-triggered architecture where the communication system decides when to exchange data autonomously, according to a static schedule. Every controller contains its own schedule control data, which is stored in a personalized message descriptor list (MEDL), that specifies at which instant a data transmission must be performed or accepted by the controller. This requires a paradigm shift from a loosely coupled ECU (Electronic Control Unit) to integrated electronic architecture, a distributed safety-critical real-time platform. There are four reasons for this:



(a) structure of CAN controller

(b) structure of TTCAN controller

Fig. 12 Internal CAN and TTCAN structures.

- *cost*: an integrated architecture allows for distributing and adding functions across the network at optimal cost,
- *stability and reliability*: design faults in software and hardware are major source for instability and low reliability,
- *safety*: new functions, by-wire and collision avoidance, are safety relevant,

- *multiple use of sensors*: in an integrated architecture, sensors can be used by several systems.

There are two version of TTP: a TTP/A version of the protocol, developed for not so strict real-time systems and the TTP/C version, which is developed for safety critical systems. The C stands for the SAE classification of the protocol. This class contains protocols with a speed of 125 kb/s – 1 Mb/s, which are used for real time control [19].

5.1. Time-Triggered Architecture

The basic building block of the TTA is a node (Fig. 13). A node comprises a processor with memory, input-output subsystem, time-triggered communication controller, operating system and relevant application software in a self-contained unit, if it is possible, on a single silicon die. Two replicated communication channels connect the nodes in order to build a cluster (Fig. 14). The communication subsystem reads a date frame with state information from the CNI (Communication Network Interface) as *a priory* known fetch instant and delivers it to the CNIs of all nodes of the cluster as a priory delivery instant. Delivery instants are contained in a scheduling table within the communication controller called "MEDL" (Message Descriptor List) consistently known to all communication controllers in a cluster.

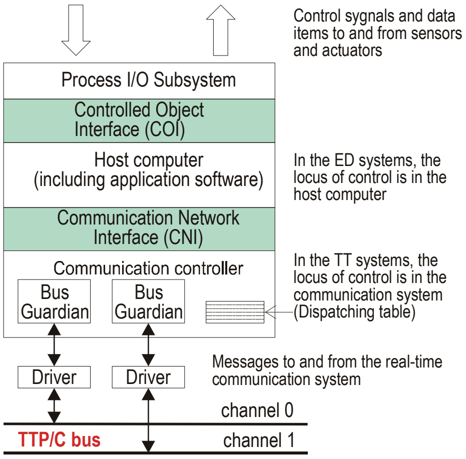

Control sygnals and data items to and from sensors and actuators
Process I/O Subsystem
Controlled Object Interface (COI)
Host computer (including application software)
In the ED systems, the locus of control is in the host computer
Communication Network Interface (CNI)
Communication controller
Bus Guardian
Bus Guardian
In the TT systems, the locus of control is in the communication system (Dispatching table)
Driver
Driver
Messages to and from the real-time communication system
channel 0
TTP/C bus
channel 1

Fig. 13 Structure of node.

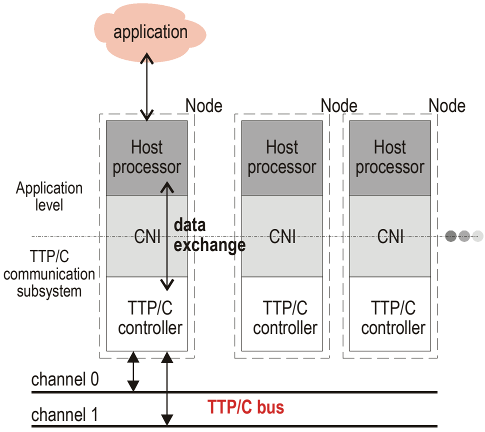


Fig. 14 Cluster

5.2. Characteristics of TTP/C

The design of the TTP/C protocol was guided by the following principles:

- *Time-Triggered Bus Access*. TTP/C communication is organized according to a static TDMA schedule: each node has a sender slot and knows the points-in-time when to send and to receive (Fig. 15). Due to this access strategy bus collisions are avoided, transmission delays are known, and the jitter is minimal.
- *Temporal Firewall*. The communication network interfaces between the host computer and the controller rules out control error propagation by design. TTP/C systems are compassable from the point of view of their temporal characteristics because the properties of this interface are fully specified in the temporal domain.

- *Fault-Tolerant Global Time Base*. The TTP/C controllers process a fault-tolerant clock synchronization that establishes a sparse global time base without relying on a central time server. The time-base, with a precision in the microsecond range, is provided to all host computers.
- *Consistent Membership Service*. The TTP/C controller informs its host computer about the state of every other computer in the cluster with a latency of less than two TDMA rounds. The membership service employs a distributed agreement algorithm to determine, in case of a failure, whether the outgoing link of the sender or the incoming link of the receiver has failed.
- *Single Hardware Failure*. Any single hardware failure must be tolerated in a properly configured TTP/C system.
- *Malicious Software*. A maliciously faulty host computer can produce erroneous data outputs, but cannot interfere in any other way with the correct operation of the rest of a TTP/C system, if it is protected by a central guardian. Fail-silence behaviour in the temporal domain is within the scope of the TTP/C system (controllers + guardian), while fail-silence in the value domain is within the scope of the host.
- *Clique Avoidance*. Clique avoidance is used to detect those faults outside the fault hypothesis which cannot be tolerated at the protocol level.
- *Confidence Principle*. A TTP/C controller will always assume that it is operating properly unless protocol mechanisms explicitly state otherwise. In this way it is assured that, upon disagreement between nodes, the node that finally wins has never made a wrong decision.

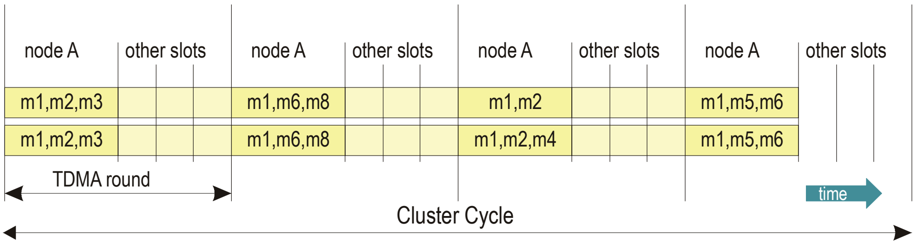


Fig. 15 Frames, messages, slots, TDMA round and cluster cycle.

5.3. Time-Triggered Protocol

TTP is a fault-tolerant protocol that provides the following services [15]:

- Autonomous fault-tolerant message transport at known times and with minimal jitter between the CNIs of a cluster's nodes of (by employing a TDMA strategy on replicated communication channels).
- Fault-tolerant clock synchronization that establishes the global time base without relying on a central time server.
- Membership services to inform every correct node about consistency of data transmission. This service can be viewed as a distributive acknowledgement service that informs the application promptly if an error occurred in the communication system.
- Clique avoidance to detect faults outside the fault hypothesis, which cannot be tolerated at the protocol level.

5.4. TTP's TDMA concept

As TTCAN, TTP/C uses the medium access method TDMA. The time is divided into slots, which are allocated to the smallest replaceable units (SRUs). A replaceable unit is an electronic module that is the replaceable smallest unit in case of a fault. It is connected to the TTP/C network and can only access the bus for transmission in a time slot exclusively allocated for it. A SRU can always monitor and receive a message from the bus [19].

- Clock synchronization is necessary to provide all nodes with equivalent time-concepts.
- Each node measures the difference between the known priory expected and the observed arrival time of a correct message.
- A fault-tolerant average algorithm uses this time difference for periodical calculation of local clock correction terms for keeping all clocks in the cluster synchronized.

5.5. Frame formats

This protocol contains two types of frames. I-frames (initialization frames) are used for system initialization and contain the internal state of the TTP controller. This state is called the "controller state" (C-state) and is described below. A node can be integrated into the membership service after receiving I-frame. The frames are sent both at the start-up of the system and at a predefined time interval during normal operation. The latter is used to re-integrate nodes into the system after they have failed and restarted.

N-frames (normal frames) are used during normal operation and contain the data sent between nodes in the system. The destination address of the frame is not stated in the frame, but in the MEDL. This reduces the overhead and allows a more efficient bus usage. A node that knows that it is supposed to receive the message does it, and the other nodes ignore the message. They do not even have to check if the message is for them, as they know it is not [18].

6. FOUNDATION^{TM} FIELDBUS

FOUNDATION^{TM} fieldbus is an open, fieldbus architecture for plant information integration. It contain H1 bus running at 31.25 kbit/s, which is optimized for integration of field instrumentation and High Speed Ethernet (HSE) running at 100 Mbit/s intended for integration of H1 and other control subsystems into a high performance control backbone. This architecture is intended for reducing the number of different networks, gateways and systems in the plant hierarchy (Fig. 16) while, at the same time, increasing information integration between automation systems, plant application packages, and Management Information systems (MIS). A cost-effective system can be built on standard, high-volume, low-cost networking technology. Replacing proprietary automation systems and networks with a single, open, integrated fieldbus architecture enables direct integration of sensors, devices, and high speed subsystems, resulting in lean hierarchy. The open specification allows a complete range of interoperable products from many suppliers. Two level communications, H1 and HSE, are based on OSI reference model (Fig. 17). The H1 physical layer and communication stack is part of the IEC 61158 International Standard. HSE uses the IEEE 802.3 and Internet Engineering Task Force (IETF) standards for the physical layer and network oriented layers for the communication stack. The Fieldbus Message Specification (FMS) for H1 and the Field Device Access (FDA) for HSE application layers are included in the International Electrotechnical Commission (IEC) 61158 fieldbus standard.

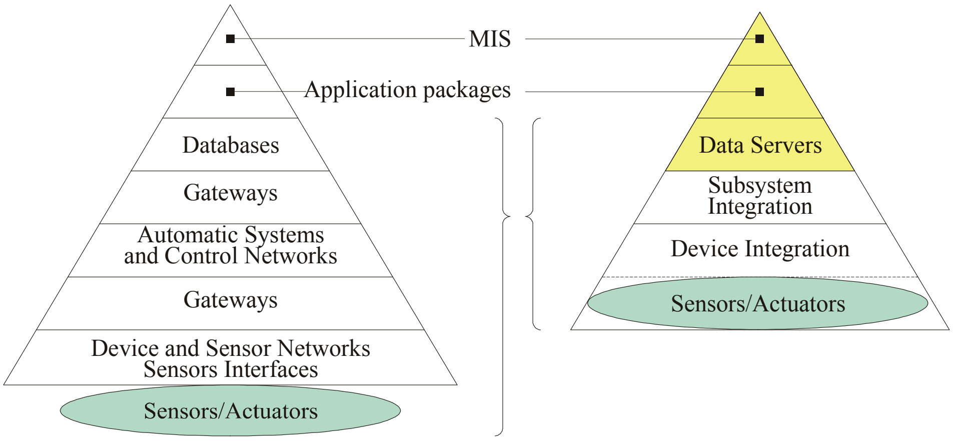


Fig. 16 Integration in Fieldbus. MIS: Management Information Systems.

Te User layer is added to the ISO/OSI reference model. It is the main contribution of ISP, now standardized by IEC 61804-2.

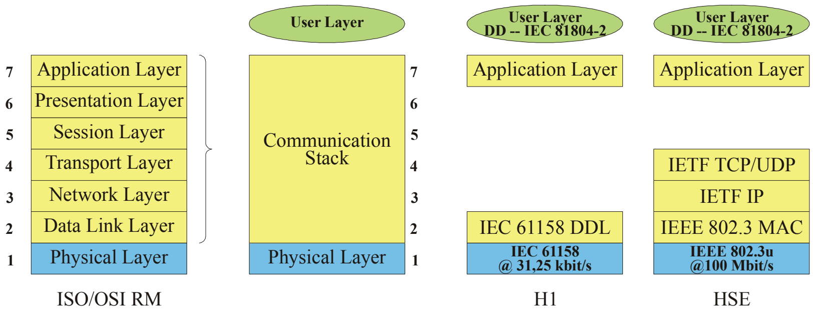


Fig. 17 User Layer: Distributed Control.

6.1. Physical layer

For Intrinsically Safe applications H1 supports bus powered installations using both the entity and Fieldbus Intrinsically Safe Concept (FISCO) models. H1 twisted-pair cables can run up to 1900 meters without a repeater and up to four repeaters can be used to increase this distance.

HSE is designed as a high performance, completely redundant, control backbone for the integration of H1 subsystems and high density data generators, such as batch controllers and analyzers. For HSE, standard twisted-pair Ethernet cables can run up to 100 m between an Ethernet switch and the device; full-duplex fibre optic cable can run up to 2000 m.

6.2. User layer

In order to achieve true interoperability FOUNDATION^{TM} fieldbus has a "User Layer" that defines the framework for performing control functions in the field devices. Device Descriptions (DD) – defined by the IEC 61804-2 International Standard – define the meaning of the parameter information to allow any host to interpret and display the process data independently of the device supplier. Device Descriptions provide uniform integration, configuration/setup, operational and diagnostics/maintenance.

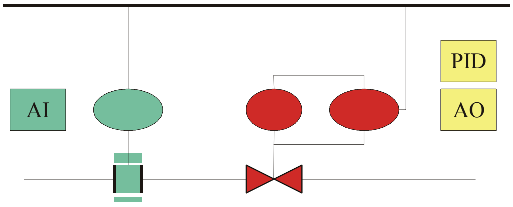


Fig. 18 Function blocs in use: example of the PID and AO blocks running in the valve.

The Fieldbus Foundation, HARTR Communication Foundation and PROFIBUS Nutzerorganisation e.V. (PNO) established a joint working group for extending Device Description standard.

In order to achieve distributed control, functions such as an Analogue Input (AI) in a flow transmitter or an Analogue Output (AO) in a valve are implemented as function blocks. Proportional Integral Derivative (PID) control functions can also be built into Function Blocks (FB) and run in a field device (Fig. 18). For basic function blocks the number and type of inputs and outputs are pre-defined by the FOUNDATION^{TM} fieldbus specification. A special Flexible Function Block (FFB) is prepared for more complex functions such as batch control, I/O gateways, and sequencing.

FFBs are a key component of open, integrated fieldbus architecture for information integration. They are application specific blocks with the I/O and algorithm configured by the end user. FFBs enable control strategies such as supervisory data acquisition, batch control and sequencing, and discrete and analogue I/O interfacing, including gateways to other plant device networks. FFBs allow end users to replace many of their existing, incompatible sensor busses with an open fieldbus architecture, and in doing so, achieve plant wide distributed control and seamless, enterprise-wide information integration through a common function block model. The future technical direction includes extension of DD capabilities and support for Safety Instrumented Systems (SIS).

6.3. Server integration

Modern data servers are built on high performance PC platforms with Ethernet access (Fig. 19). The data server provides a common interface for applications. Typical application packages are:

- Human/Machine Interface (HMI),
- Configuration Tools,
- Diagnostic Programs,
- Enterprise Resource Planning (ERP) program, and
- Management Information System (MIS) databases.

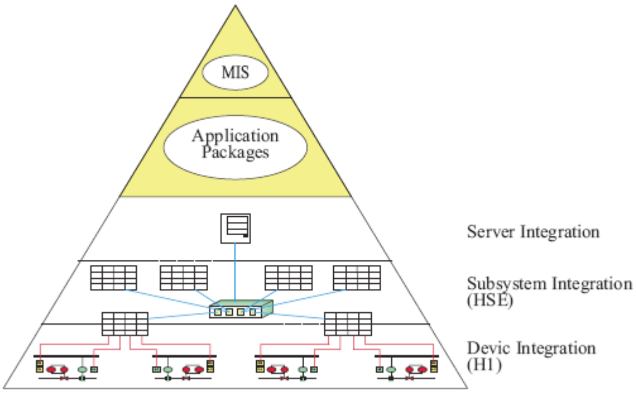


Fig. 19 Open, integrated Fieldbus architecture.

7. Safe and Secure Issues

Standard IEC 61508 [20] defines Safety Integrity Levels (SIL) for electrical, electronic and programmable electronic devices. SIL define permissible residual error probabilities (REP) and error detection rates. New IEC standardization activities: "Profiles for functional safe and secure communications in industrial networks" [21]. SIL 3: REP should be ≤ 10^{-7} errors/h [3], [20]. Where:

- *communication part*: REP ≤ 10^{-9} errors/h (1% of 10^{-7})
- *sensors, PLCs, actuators, etc.*: REP ≤ 10^{-2} errors/h (99% of 10^{-7})

Special safety busses were developed for the functionality of an automation system, existing fieldbusses were or are going to be extended with safety layers [22], [23]. To meet IEC 61508 requirements, FOUNDATION^{TM} launched the SIS project for the extension of function block specifications by adding additional error checks and new function blocks such as "Write Lock" function for preventing unauthorized changes. Leading companies in automation ABB, Invensys/Triconex, Emerson Process Management, ExxonMobil, DuPont, Flowserve, HIMA, Honeywell, Metso Automation, Rockwell Automation, Rotork Control Systems, Saudi Aramco, Shell Global Solutions, Smar, Softing, TÜV, Westlock Controls and Yokogawa joined together and completed draft specifications. In 2003 the TÜV Anlagentechnik GmbH, Automation, Software and Information Technology, a global, independent and accredited testing agency approved the FF-SIS system concept.

One principle for making safe communication is to consider the communication channel as a *Black Channel* [2]. The safety measures would be realised in a separated safety layer that is situated between the communication protocol and the application. Using this principle, an existing communication system can be used as it is - and with existing components such as ASICs, cables, connectors, repeaters, links etc. as far as they meet the electrical requirements.

8. Security

The protection of safety-critical and infrastructure systems (such as automation systems for utilities, but also for process industries and manufacturing plants) against electronic and communication network based attacks has become more and more important. Security criteria are:

- confidentiality of information, i.e. protection against access by unauthorized third parties,
- information integrity, i.e. detection of unauthorized modification of the message contents with a specified level of confidence;
- timeliness of the message delivery, i.e. detection of any unauthorized message re-timing, re-sequencing or replay of prior messages;
- Authentication and authorization of communication peers.

Nowadays, two common defence approaches are discussed: hard perimeter, and defense-in-depth (several zones/shells are placed around the automation object, which has to be protected). Many arguments are given why defence-in-depth with multiple, staged, complementary security mechanisms is the more suitable approach.

9. Wireless Sensors Network

Advances in science and technology have made dreams come true over last decade given that computing capabilities will become exponentially cheaper with each passing year. Chips are getting smaller and smaller and mainly cheaper. New types of devices have been born with cheap microprocessors, sensors and FM radio modules such as wireless sensor nodes [24]. If we combine a few of these devices together they form the wireless sensor network (WSN). Wireless sensor nodes (we will address them as nodes) have limited processing speed, storage capability, and communication bandwidth. The main problem is node power supply. Nodes are powered by batteries, so low power consumption is crucial.

In most applications or deployment WSN networks must operate over a long period of time. In scenario like this most of the device components will be turned off (radio, sensor) or put to low power consumption mode (microprocessor) for most of the time [24]. WSN sensor nodes are mainly deployed randomly. They are scattered many times from the airplane, boat or thrown out of the vehicle. Communication capabilities must allow randomly deployed sensors to connect to each other and communicate through harsh environments. Each of these factors (power consumption, unknown positions; reduced power consumption) complicates the network protocols (topology, routing, MAC …).

9.1. WSN sensor node

WSN sensor node is an entity that consists of four (five) main blocks (Fig. 20).

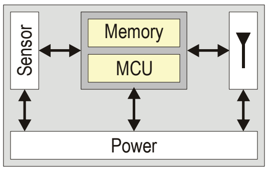


Fig. 20 Sensor node.

The core block is MCU with memory. The memory module can be part of a MCU or separated and connected via a bus to MCU. This is why we have four or five modules. The next module is a sensor which is connected to the MCU via bus or AD module. Module three represents an RF communication module. The wireless communication module is connected to MCU, which communicates with other devices through this RF module. All modules are powered by battery (power) which can be counted as the fourth module in sensor node.

Sensor nodes form a wireless-sensor network, which at one point, stores data or it is connected to more powerful network (Fig. 20).

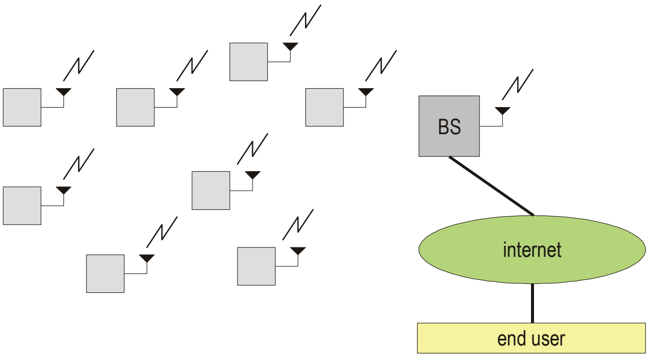


Fig. 21 Wireless sensors network: base station.

The entity that connects WSN to other network is often called the base-station (BS) [25]. The base-station is rarely powered by batteries so power consumption is not a BS concern. Its primary WSN goal is sensing or monitoring something, reacting to some event(s), and reporting it. In respect of that, almost all data flow goes from nodes to the base station. In some research over 95% of traffic was towards the base-station [30]. The base-station forwards data to the end-user via a network. The end-user collects and manipulates the data collected in WSN.

9.2. Applications

Applications for WSN are only limited within your imagination. But let us examine a few potential applications using WSN:

- Habitat and space monitoring, where we can monitor large areas of climate and animal behaviour over a long period of time.
- Chemical and biological sensors can be used where the human body would suffer serious damage.
- An army uses WSN on the battlefield for surveillance, soldier positioning and health monitoring.
- Traffic monitoring and control, where WSN can detect hot-spots and warn approaching drivers. Even road tolling can be achieved using WSN.
- In emergency scenarios, such as early signs of forest fire or fire direction early prediction can be made with WSN. WSN can help fire-fighters from being trapped.

Of course, every application needs a special type of WNS (or special type of nodes), which monitors different things in different time-frames. WSN is application dependent, which means, there is no multipurpose WSN and no multipurpose MACs, routing protocols etc.

9.3. Organization

We have in mind two basic ideas when we talk about WSN organization: MAC protocols, routing protocols and topology control. MAC protocols are commonly independent of upper routing protocols but, in WSN routing, protocols can be part of MAC protocol. Medium access protocol (MAC) regulates common channel access. In the worst-case scenario, every node in radio sight wants to communicate. There has to be a mechanism which defines who communicates and when. MAC can be defined as a set of rules for when a frame is sent, when the node listens for the frame and when the node is put to sleep. MAC ensures low error rates, energy saving, and power control [27]. There are several MAC protocols proposed for WSN.

9.4. Medium Access

When the receiver receives more than one packet at a time, they are called "collided packets". Packets collide even if they coincide partially. When a packet collides, data is lost and packets have to be retransmitted. All collisions and retransmitting cost energy and because energy is important, MAC protocols should prevent such energy waste [27].

There are three types of communication patterns known in WSNs [28]: broadcast, convergecast and local gossip. The broadcast type is usually used by a base station that wants to transmit some information to all sensor nodes. Broadcast communication pattern should not be confused with broadcast packet type. When a group of sensors communicate with one particular sensor, the communication pattern is called "convergecast". Sometimes sensors communicate with each other. This communication pattern is known as "local gossip". MAC protocols can be divided into two main groups [29]:

- Scheduled protocols, where a common channel is divided into sub channels in a time-base (TDMA), frequency-base (FDMA) or code-base (CDMA) manner.
- Connection based protocols, which do not divide channels into sub-channels. A common channel is shared by all nodes and it is allocated only on demand. A connection mechanism must be implemented to decide which node has the right to access the channel and when.

There are currently many MAC protocols known for the WSN, but we let the interested reader explore [27]–[29]. Some of the commonly used MAC protocols in WSN can be examined in Table 3.

Table 3 MAC protocols used in sensor networks

MAC	Time sin. needed	Supporting comm. pattern	Type	Adaptivity to changes
S-MAC, T-MAC, DSMAC	No	All	CSMA	Good
WiseMAC	No	All	CSMA	Good
TRAMA	Yes	All	TDMA/CSMA	Good
SIFT	No	All	CSMA/CA	Good
DMAC	Yes	Just Convergecast	TDMA/ Slotted Aloha	Week

9.5. Routing protocols

WSN routing protocols have been a target for intensive research over the past few years. WSN network differs from other wireless networks such as WiFi, ad-hoc networks and cell networks in many aspects. Let's look at a few of them:

- WSN, in most applications, only gathers data; MANET networks, in many cases, work as computer distributed systems.
- WSN is usually used by one user. MANET uses more independent users at the same time.
- WSN networks are usually bigger (up to a few thousand nodes) than MANET networks.
- Many entities in WSN can use the same ID (address). In MANET this is impossible.
- WSN are application-dependent.
- Traffic in WSN is mainly towards BS. In MANET there is no pre-determined traffic flow.

Many users think that MANET and WSN are the same networks but, as shown previously, they differ in many aspects [26]. Unique networks such as WSN need special routing protocols. Why?

- Due to the large number of nodes it is impossible to develop a global addressing scheme. Mainly because the ID overhead would be too large, and secondly, in sensor nodes there is insufficient memory for such a large addressing scheme.
- Sensor nodes in WSN are constrained by energy consumption, computing capabilities, and memory. Thus, special careful resource management is needed.
- WSN nodes are stationary in most applications and scenarios.

Many routing protocols are developing to cope with the strict demands of WSN. Some tasks, like finding and maintaining communication routes are serious in the case of sudden node failure or while taking into consideration low power consumption restriction. There are several routing challenges and open design issues while developing new routing algorithms [31]:

- *Network deployment*. Nodes can be deployed deterministically or randomly. Protocol aspect deterministic deployment is simpler for routing - the routing paths are known in advance. When deploying nodes randomly, we need Ad Hoc routing protocol consisting of many wireless hops. Lowering the number of hops means lower power consumption. A routing algorithm must aspire to this attribute.
- *Energy consumption without losing accuracy*. In multihop networks, nodes forward messages received from a network (router). So, regarding energy consumption, the power used for routing actions must be as low as possible.
- *Fault tolerance*. While WSN operates many sensors, it may fail because of power lack or physical damage. In cases like this, routing protocol must be capable of establishing new routes to BS. Multiple redundancies must be used in fault-tolerant sensor networks.
- *Scalability*. Routing protocols must be capable of dealing with large a number of sensor nodes and guarantee message delivery to BS, even if most of the nodes are in sleep-mode.
- *Network dynamic*. Some nodes may not be stationary. Even BS can move.

- *Quality of Service*. This has become a more and more desirable feature. Real-time sensor networks [30] are desired in many applications. Therefore, bounded latency could be a condition.

In general there are three types of routing protocols in WSN: Flat-based, where all sensor nodes perform the same operations. Hierarchical-based routing, where some nodes perform higher operations than others (routing) and finally, location-based, where sensor node position is used to optimize routing paths (Table IV).

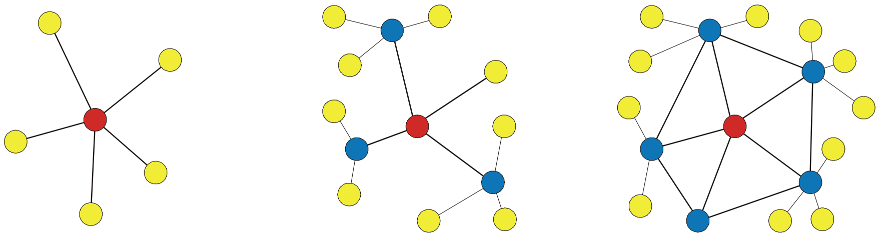
Table 4 Overview of MAC protocols used in sensor networks.

	Mobility	Energy needed	Location aware	Scalability Classification
LEACH	Fixed BS	High	Good	Cluster
PEGASIS	Fixed BS	High	No	Reactive
SPIN	Yes	Average	Limited	Proactive
Direct diffusion	Limited	Average	Limited	Proactive
Rumor	Very limited	Not specified	Good	Hybrid
MCFA	Fixed BS	Not specified	Good	Proactive

We will not go into details as to how each WSN routing protocol works. We address interested reader to [26], [31] and [32] where additional information can be found.

9.6. Topology

Topology is a set of "communications links between entities used by routing protocol" in multi-hop network. Routing protocols can use topology links explicitly or implicitly [33]. Topology (or available links) depends on many factors such as node mobility, noise, transmitting power, etc. Correct topology control can optimize network power usage by keeping the network connected using minimal possible resources, and helps reduce redundancy in network communications [26].



(a) star (b) cluster tree (c) mesh

Fig. 22 Example of topologies in ZigBee (802.15.4).

9.7. Power consumption

Power consumption is one of the main issues in WSN. When nodes are deployed (dropped from aircrafts or vehicles, especially on a battlefield) there is practically no option to replace or recharge the batteries, so low power consumption is one of the most wanted attributes of WSN nodes. Many routing and MAC protocols are designed to maintain battery power as long as possible [29], [31], but there is always a compromise between power consumption and performance.

Communication is almost always the most energy-intensive operation. Transmitting and receiving (or listening to a channel) rapidly increases power consumption, and because of this almost all radios have a sleep-function. One of the most common power saving techniques is powering radio down when there is no activity on the communication or sensing side [24]. Of course, the radio must be turned on immediately when an event occurs. When activity is spotted on the sensing side of the node, the node radio is awakened by the MCU controlling the radio, and a message about the activity is sent over the network. When the activity happens on the radio side (message is transmitted over network), there is no way for MCU to know that if not listening. MAC protocol must be designed in such a fashion that it implements time-scheduling for waking up the radio and listening to the medium. When radio is awaking and there is some activity on the medium, MCU can react to it.

Required computer power is the second energy consumer. When CPU works at full speed there can be high-power consumption. In this regard there are some "processor stages" for operating. These are mainly: full speed, reduced speed and sleep, when the processor is put into sleep mode (Low power consumption).

Table 5 presents the power consumption of a LPC2138 ARM microcontroller. It clearly shows that, when CPU is operating at high performance energy, consumption is almost 4 times greater than when operating on lower performance (40 MHz vs. 10 MHz). The idea for WSN is to put the processor on reduced speed or sleep mode whenever possible.

Table 5 Overview of MAC protocols used in sensor networks.

Core mode	1 Mhz	10 Mhz	60 Mhz
Active (Flash ARM)	6,4 mA	11,6 mA	40,6 mA
Idle	1,3 mA		11,3 mA

Power consumption can be a complex problem and must be approached systematically. Sensors, microcontroller peripheral, radios, environment and even modulation are used in radio affect power consumption. There is no general recipe on how to reduce power consumption.

9.8. Security

Security has become more and more important in WSN, especially for security-critical applications (army, monitoring, intruder detection). There are two ways of achieving security in WSN: Security on the physical layer [30] and security on the higher layers (MAC, encrypting data). WSN networks are unlike traditional computer networks, and standard network security approaches ca not be simply applied to WSN. There are some constrains in WSN which must be taken into consideration while implementing security [35].

10. CONCLUSIONS

Today, no automatization can be imagined without the use of communication and information technology. At the sensor and actuator levels fieldbusses have prevailed and at higher levels, busses based on the Ethernet.

The important field of sensing and automatization is where objects are mobile or difficult to wire. The wireless communications is the right solution. Another advantage of their fast growing usage is also low installation cost, and simple usage.

REFERENCES

1. Mário Alves, Eduardo Tovar, Luís Marques, Steve Behaeghel, and Katrijn Van Nieuwenhuyse. Engineering Hybrid Wired/Wireless Fieldbus Networks - a case study
2. Neumann, P.; Virtual Automation Network, IEEE conference ICIT'03, Maribor 2003, Proceeding, pp. 994 – 999
3. Neumann, P.: Merging Fieldbus and Telecommunication Systems in the Industrial Automation Domain. IEEE AFRICON 2002, 6th AFRICON conference in George, ZA, Proceedings Vol.1, pp.197
4. Frequently Asked Questions about LonWorks Networks, http://www.lonworks.echelon.com, Echelon - The LonWork Company, 1997.
5. The History of Fieldbus, Fieldbus Tutorial-History, http://rolf.ece.curtin.edu.au/-clive/Fieldbus, 1997.
6. Selecting the Right Fieldbus, Gespac, 1997. http://www.gespac.com/html
7. Delta-Tau Data Systems, Inc.: Motion and Control Ring Optical, Specification, May 1998. http://www.macro.org
8. W. Stalings. "Network standards", Addison-Wesley, 1996.
9. CAN Specification, version 2.0, Robert Bosh GmbH, 1991.
10. T. Fuehrer, B. Mueller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther: Time Triggered Communication on CAN (Time Triggered CAN – TTCAN). http://www.can.bosch.com/docu/CiA2000Paper_1.pdf
11. T. Fuehrer, B. Mueller, F. Hartwich, R. Hugel: Timing in the TTCAN Network. http://canopen.de/can/ttcan/hartwich2.pdf
12. T. Fuehrer, B. Mueller, F. Hartwich, R. Hugel: CAN Network with Time Triggered Communication. http://canopen.de/can/ttcan/hartwich1.pdf
13. T. Fuehrer, B. Mueller, F. Hartwich, R. Hugel, H. Weiler: Fault tolerant TTCAN networks. http://canopen.de/can/ttcan/mueller.pdf
14. G. Leen, D. Heffernan: TTCAN: a new time-triggered controller area network. Microprocessors and Microsystems, Volume 26, Issue 2, 17 March 2002.
15. TTP/C specification http://www.ttagroup.org
16. TTTech Computertechnik AG. Specification of the TTP/C protocol. 1999. http://www.tttech.com
17. H. Kopetz. A comparison of CAN and TTP (TTTech promotion material, 2001)
18. H. Kopetz, R. Hexel, A. Krueger, D. Millinger, R. Nossal, A. Steininger, C. Temple, T. Fuerher, R. Pallier, M. Krug. A Prototype Implementation of a TTP/C Controller. http://www.vmars.tuwien.ac.at/php/pserver/extern/searchpaper.php
19. H. Sivencrona, J. Hedberg, H. Röcklinger. Comparative Analysis of Dependability Properties of Communication Protocols in Distributed Control Systems. 2001. http://www.sp.se/electronics/rnd/palbus/Reports/PALBUS_10_2.pdf
20. IEC 61508: Functional Safety of Electrical/Electronic/Programmable El. Safety-Related Systems
21. IEC 65C/307/NP: Digital data communications for measurement and control – Profiles for functional safe and secure communications in industrial networks, New Work Item Proposal, 2003
22. Redmill, F.; Anderson, T.: Components of System Safety. Springer Verlag, Berlin-Heidelberg-New York, 2002.
23. Reinert, D.; Schaefer, M.: Sichere Bussysteme für die Automation. Hüthig Verlag Heidelberg, 2001.
24. David Culler, Deborah Estrin, Mani Srivastava. "Guest Editors' Introduction. Overview of Sensor Networks", IEEE Computer Society, August 2004.
25. F.L. Lewis. Wireless Sensor Networks. Smart Environments. Technologies, Protocols, Applications, ed. D.J. Cook and S.K. Das, Wiley, New York, 2004

26. Qiangfeng Jiang, D. Manivannan. "Routing Protocols for Sensor Networks", Department of Computer Science, University of Kentucky, Kentucky.
27. Ilker Demirkol, Cem Ersoy, and Fatih Alagöz. "MAC Protocols for Wireless Sensor Networks. A Survey", IEEE Communications Magazine, 2006.
28. S.S., Kulkarni, "TDMA services for Sensor Networks", Proceedings of 24th International Conference on Distributed Computing Systems Workshops, Pages:604 - 609, 23-24 March 2004.
29. Wei Ye and John Heidemann, "Medium Access Control in Wireless Sensor networks", USC/ISI TECHNICAL REPORT ISI-TR-580, OCTOBER 2003
30. Karl Benkic. "Proposed use of a CDMA technique in wireless sensor networks", IWSSIP 2007, Slovenija, 2007.
31. Jamal N. AL-Karaki, Ahmed E. Kamal. "Routing techniques in wireless sensor networks. A survey", The Hashemite University, Iowa State University, IEEE Wireless Communications, December 2004.
32. Kemal Akkaya, Mohamed Younis. "A survey on routing protocols for wireless sensor networks", Ad Hoc Networks, Elsevier, Volume 3, Issue 3, May 2005
33. Ramanathan R. and Rosales-Hain R., "Topology control of multihop wireless networks using transmit power adjustment," in Proc. IEEE INFOCOM, 2000.
34. Gaurav Srivastava, Paul Boustead, Joe F.Chicharo. "A Comparison of Topology Control Algorithms for Ad-hoc Networks", University of Wollongong, NSW, Australia, 2003
35. Aurebach. Security in Distributed, Grid, Mobile, and Pervasive Computing, ISBN-13. 978-0849379215

INDUSTRIJSKI MREŽNI PROTOKOLI U INDUSTRIJSKOJ PROIZVODNJI

Žarko Čučej, Karl Benkič

Daljinsko upravljanje, kao i sistemi nadgledanja su osnova današnje automatizovane prozivodnje. Daljinsko upravljanje kao i nadgledanje su realizovani različitim vrstama računara, PLC sistemima, povezani različitim mrežama. Ove mreže se razlikuju od ostalih mreža podataka bar u sledećem: (i) organizacija saobraćaja (ii) nivo procesiranja za interpretaciju i validaciju podataka je ograničen I (iii) najbitnije – otpornost na greške kao i slanje ispravnih podataka u pravom trenutku je nešto što se očekuje od ovakvih mreža. Ovaj članak sadrži kratak pregled mreža za komunikaciju koje se sreću u proizvodnji, kao i u daljinskom nadgledanju i detekciji, sa posebnim naglaskom na CAN, TTCAN, TTP i bežični senzorski industrijski mrežni protokol.

Key words: *Industrijski mrežni protokoli, bežične sensorske mreže, automatizacija*