



Performance evaluation of the parallel processing producer–distributor–consumer network architecture

Grzegorz Polaków*, Mieczysław Metzger

Department of Automatic Control, Electronics and Computer Science, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

ARTICLE INFO

Article history:

Received 4 August 2009

Received in revised form 29 January 2013

Accepted 12 April 2013

Available online 23 April 2013

Keywords:

Real-time Ethernet

switched Ethernet

producer–distributor–consumer

process automation

ABSTRACT

The CSMA/CD access method is no longer invoked in switched, full-duplex Ethernet, but the industrial protocols still take the presence of the method into account. The parallel processing producer–distributor–consumer network architecture (ppPDC) was designed specifically to actively utilize the frame queuing. The network nodes process frames in parallel, which shortens the time needed to perform a cycle of communication, especially in cases when frame processing times within the nodes are not uniform. The experiments show that the achievable cycle times of the ppPDC architecture are an order of magnitude shorter than in the well-known sequential PDC protocol.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In teaching and research on control systems there often is a need to setup an ad-hoc computer network for process data exchange. In this application domain, particularly valuable features of the network are flexibility, compatibility with a broad base of hardware and ease of configuration. These needs arise from the very dynamic environments of research and teaching, where tasks, instrumentation and experimenters change frequently. Quite similar application domain involving distributed application with multiple participants was considered for example in [1].

Retail industrial networks and protocols are not suitable in the case. They are designed in such a way that they require detailed time consuming configuration process, after which they can operate unattended reliably over long periods of time. In the considered domain a network protocol is required, configuration of which is quick and easy, while long-term reliability is not crucial.

An obvious solution is to use communication channels based on the well-known TCP or UDP to transmit the data in the binary form. The situation, however, is not simple, when there are many communication channels required between multiple devices. This is exactly the situation in the described application domain involving activities of many students and researchers with differing tasks and agendas. However, the task can be accomplished by using a data exchange architecture designed for retail industrial networks (e.g. producer–

distributor–consumer, producer–consumer, master–slave, client–server) in order to organize communicational channels and provide real-time in an off-the-shelf Ethernet [2] network.

An interesting description of historical background and resulting solutions supporting RT in Ethernet is presented by Moraes et al. in [3]; an example of performance analysis is available in [4]. For a certain period of time, the situation in the industrial Ethernet-based networks market recalled the struggle for dominance in the fieldbus market [5,6]. The products resulting from the race included EtherCAT, Ethernet Powerlink, Foundation Fieldbus High Speed, Modbus/TCP, Profinet, and Ethernet/IP. Currently, most of industrial Ethernet definitions are included in the IEC61784-2 standard (see [7] for a brief survey) as so called communication profiles. Typical approaches to industrial Ethernet include encapsulation of pre-existing industrial protocols in the Ethernet frames (as in Ethernet/IP), use of TDMA method to provide real-time access (Ethernet Powerlink), and sometimes even breaking compatibility with IEEE802.3 MAC layer (as in the RT Class 2 traffic of Profinet).

Parallel, further development of Ethernet market rendered the CSMA/CD obsolete, due to the growing popularity of the switched Ethernet (IEEE802.1D standard [8]). Huge potential of the switched Ethernet in industrial applications has been widely recognized, and multiple studies of performance and constraints were conducted (for examples see [9–11]). All these publications emphasize the improved performance and reliability of the switched networks and focus on determining the dependence of network efficiency on number of nodes and switching parameters. However, literature does not indicate that switching technology enables new possibilities in the field of communicational protocols and data distribution schemes due to the fact, that Ethernet switches are actually fast computers

* Corresponding author. Tel.: +48 32 237 2354; fax: +48 32 237 2127.

E-mail addresses: grzegorz.polakow@polsl.pl (G. Polaków), mieczyslaw.metzger@polsl.pl (M. Metzger).

built to fulfill one task: reliable queuing and delivering Ethernet frames. There are no proposals for the use of this low-level queuing functionality in order to relieve the network nodes of some of the communication tasks. On the contrary, there are still protocols in use, in which the use of switching technology disrupts the frame scheduling and is not recommended [12]. The parallel processing producer–distributor–consumer (ppPDC) scheduling model proposed in this paper attempts to fill this gap and create a new network architecture, which is in contrast to the already existing ones (e.g. producer–distributor–consumer, producer–consumer, master–slave, client–server) and is designed to exploit the capabilities of the underlying switched network.

The ppPDC model is based on the well-known producer–distributor–consumer network architecture (PDC), which is used in industrial communication networks to provide the desired characteristics of cyclic data exchanges in process control systems. The use of the PDC architecture in real life will be presented here with the FIP protocol. The FIP protocol (field instrumentation protocol) was developed in the early 1980s. In 1996 it was included, with minor modifications under the WorldFIP name, in the EN50170 standard, among the P-NET and Profibus networks. This European standard, in turn, was incorporated into in 2000 into the IEC61158 international standard [13] as Type 7. This description of the WorldFIP protocol is referred to by the IEC61784 standard [14] as the CPF-5 profile [13].

The PDC network architecture is based on frames broadcasting. Contents of messages (e.g. numerical values of variables in case of process control) produced by the network nodes (i.e. *producers*) are broadcasted over the network in response to specific requests broadcasted by the designated privileged network node, referred to as a *distributor*. Nodes of network waiting for a particular unit of data (i.e. *consumers*) listen to the network traffic for the occurrence of two broadcasted messages: request issued by the distributor (*ID_DAT* frame) containing identifier of the data unit, and the answer sent by the producer (*RP_DAT* frame) containing the requested value (Fig. 1a). The distributor node broadcasts the requests for the variables according to a schedule, pre-planned at the stage of network configuration. The schedule is designed according to specific rules, resulting from requirements of a given control system (e.g. required sampling periods of the physical signals), and physical network

capabilities (e.g. transmission rate). The communication is performed in sequences, where the main cycle of the schedule is called a *macrocycle*, which in order consists of smaller *microcycles*. For a more detailed description of the FIP protocol please refer to [13,14].

The PDC network architecture is designed to be used in a simple network built of a single shared transmission medium. Therefore the PDC scheduling performs two roles: on one hand it serves as a medium access control mechanism, which prevents collisions from occurring, and on the other hand it serves as a method to access a database distributed in the producer nodes in hard real-time. In case of modern switched networks, built otherwise than with shared medium, each of the network nodes has its own collision domain. Therefore, the functionality of the medium access control is superfluous and may be dropped in order to improve the performance of the database mechanism.

The proposed modification of the PDC architecture is designed to work in a switched network, in which there is no need for an additional medium access control functionality at the application level. Moreover, presence of the switching queue allows for existence of multiple frames sent destined to the same network node at the same moment. As long as the volume of frames does not exceed the switch queue capacity, frames will be reliably queued and delivered. Therefore, it is reasonable to depart from the sequential model of communication employed in the traditional bus topology. The proposed network architecture stresses the need of time synchronization of the microcycles, but it allows for the nodes to work in parallel.

The modified architecture is the result of on-going research; the first draft of the design was described in [15]. The original idea was to develop a network protocol in order to support time-determined and distributed simulation of dynamical systems with the LabVIEW software development system. The architecture proved to be useful, and during the last years it was employed in a few applications, for example as a software agent communication framework [16].

A single cycle of communication according to the modified schedule consists of the following stages:

- The distributor broadcasts a message (*BROAD* frame) containing consolidated data units received from the producers in a previous

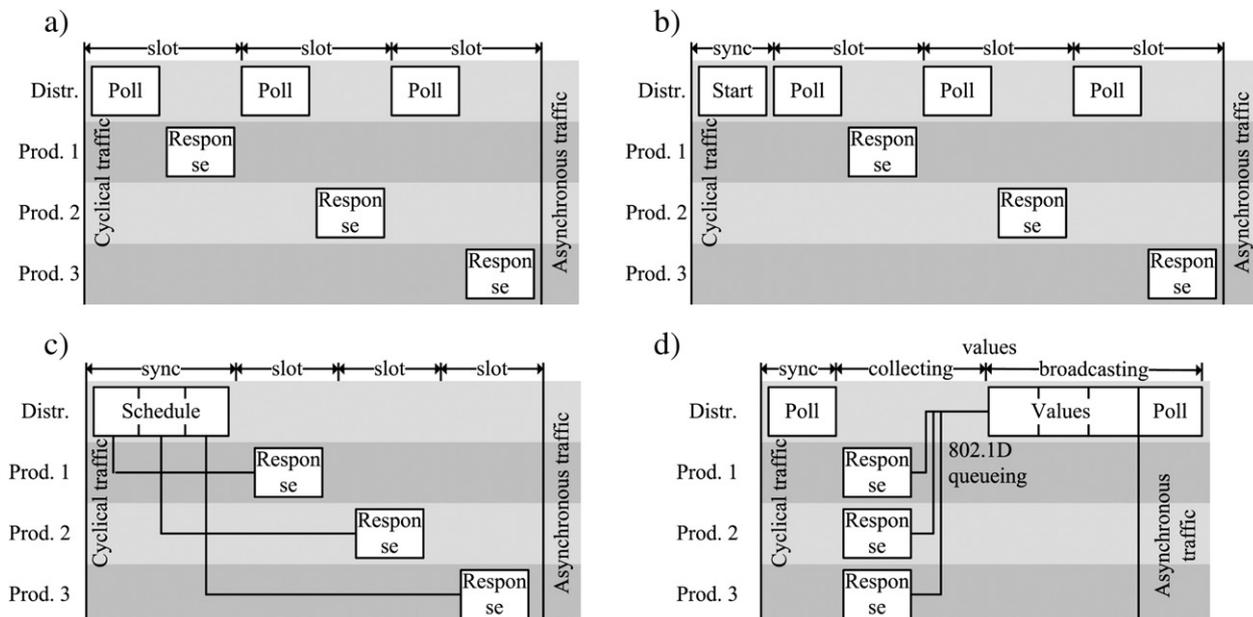


Fig. 1. Comparison of the data scheduling in (a) FIP (PDC), (b) Ethernet Powerlink, (c) FTT Ethernet, and (d) proposed ppPDC.

cycle. This message is received by the consumers, and at the same time, it is a call for the producers to send new data unit to the distributor.

- The producers send new data units to the distributor using the typical point-to-point messages (*RP_DAT* frames).

Due to the parallelism present in the proposed communication scheduling (all the producers are requested to send the messages at the same moment, thus their distributed threads start computing in parallel) the proposed network architecture was designated as parallel processing producer–distributor–consumer (ppPDC). In Fig. 1 a comparison is presented between the proposed ppPDC, classical PDC, and two most similar industrial Ethernet implementations, i.e. Ethernet Powerlink, and FTT Ethernet.

The main benefit of the ppPDC approach is the previously mentioned parallelization of the producer processing threads distributed among the network nodes. The processing times are the higher the more complicated protocol stacks of the network nodes are, and usually the processing times dominate significantly over the physical transmissions times [9,10,17]. Therefore, it is expected that the parallelization of the processing times will significantly shorten the period of time needed to perform all transmissions scheduled for a cycle. This hypothesis is evaluated theoretically and proved experimentally in the following sections of the paper. In a result, the ppPDC is proved to be a very efficient protocol, easy to implement, and provided a performance comparable to retail industrial protocols. Moreover, the only requirement of the ppPDC is the Ethernet and UDP support, which makes it compatible with nearly all the instrumentation used in the modern automation. Therefore it is a perfect tool for use in the application domain of multiuser dynamical systems, but it could also replace currently used cyclical data traffic mechanisms in some of the retail Ethernet-based industrial protocols to improve their performance.

2. Materials and methods

The experimental setup consisted of four industrial real-time PLCs connected to a PC. Three of the controllers were National Instruments cRIO-9076, and one was cRIO-9074; the PC was Lenovo Thinkpad R61 laptop. The evaluated Ethernet network was built using the cat. 5 cables, which connected the devices with the unmanaged National Instruments UES-3880 switch. Network interfaces were configured to auto-negotiate; resulting link parameters were full-duplex mode at the speed of 100 Mbit/s. The schematic diagram of the experimental setup is shown in Fig. 2.

2.1. Description of the developed and tested protocols

For the purpose of this performance evaluation, computer software was developed with the LabVIEW 2011, which implemented

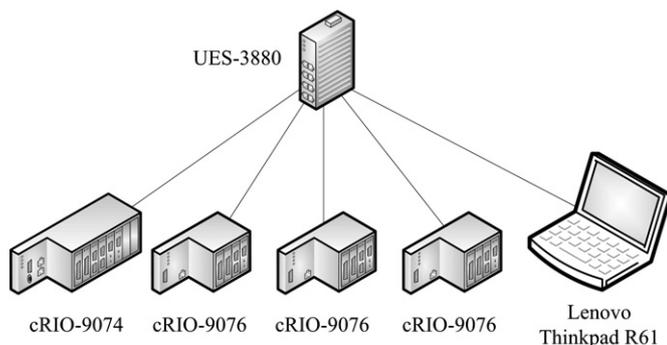


Fig. 2. The experimental setup for performance evaluation.

simplified protocols working according to PDC and ppPDC architectures. Standard LabVIEW function blocks for the UDP/IP communication were used, which were extended with the custom application-level protocols. Suitability of the UDP/IP protocol for encapsulation of industrial data is discussed for example by Vitturi in [18], while working implementation of FIP PDC model over TCP/UDP/IP is investigated in [19]. The protocol design assumes the following:

- IDs of exchanged variables are numeric and have a size of 8 bits – this means that the maximum number of variables exchanged via the protocols is limited to 256. The number is big enough for the performance evaluation, although in the industrial applications the bit size of numeric ID would certainly need to be increased.
- All variables are of the same type with a length of 16 bits – this makes processing of frames by the network nodes simpler by minimizing the time required for the processing and eliminates the need for otherwise necessary type/length fields.
- Microcycles are identical to macrocycles – during the performance evaluation stage, there is no need for the specific scheduling of the specific variables, because the experiments determine the performance of the microcycles only, depending on the contained number of variables.

Transmitted data are formatted into frames, the format of which results from the assumptions stated above. In the cyclic communication of the PDC network architecture there are two types of frames: the frames (designated as ID_DAT according to the FIP protocol) containing identifiers of variables, and the frames (*RP_DAT*) containing values of variables. For the purpose of this performance evaluation, formats of the frames have been designed to be as simple as possible, with the fixed number of fields and their sizes (Table 1). Both types of frames, formatted according to the specification, are encapsulated in multicast UDP messages. In the ppPDC network architecture there are also two types of frames (Table 2): the frames containing the set of all values of variables (designated as BROAD), and the frames sent from producers to a distributor containing a single value of a variable (designation *RP_DAT* borrowed from the PDC architecture). The length of a BROAD frame depends on the number of variables exchanged during the microcycle. For a microcycle containing only one variable, the length is three octets, for the largest possible microcycle, i.e. the one containing 256 variables, the length is 513 octets, which still does not exceed the maximal allowed payload for UDP packets in Ethernet networks. The BROAD frames are encapsulated in multicast UDP messages, and the *RP_DAT* are sent as point-to-point ones.

2.2. The method of time measurements

The studies conducted were intended to determine the minimal achievable duration of microcycle in the function of used network architecture, number of variables and number of network nodes. In order to measure the duration of a microcycle it is necessary to determine exact moments in time, in which specific frames arrive to their destinations. To be able to compare these measured moments between various network nodes, it becomes necessary to develop an absolute method of synchronizing the nodes, in reference to which a duration of data exchange may be measured. Typical approaches to this task include precise synchronization of computers internal clocks, or introduction of the additional reference channel of

Table 1
The formats of the frames used during the PDC evaluation.

Frame	ID_DAT		RP_DAT	
Content (hex)	01	xx	02	xx xx
Size (octets)	1	1	1	2
Description	Frame identifier	Variable identifier	Frame identifier	Value of variable

Table 2

The formats of the frames used during the ppPDC evaluation.

Frame	BROAD					RP_DAT			
Content (hex)	01	xx xx	xx xx	...	xx xx	02	xx	xx xx	
Size (octets)	1	2	2	...	2	1	1	2	
Description	Frame identifier	Value of 1st variable	Value of 2nd variable	...	Value of nth variable	Frame identifier	Variable identifier	Value of variable	

communication (as in [12]). Problems of the software execution measurement are described for example in [20]. In the considered case of PDC and ppPDC network architectures, the task is easier due to the cyclic nature of protocols and due to the presence of privileged network supernode. All the data exchanges in the network are performed in response to requests sent by the distributor node. The requests are usually synchronized to a predefined scheduling of variables. Removing the synchronization results in total occupation of the media, since the microcycles are performed one after another, as quick as physically possible. In such situation, it is enough to observe intervals between starts of successive microcycles. Each of the intervals is the duration of the microcycle. This method was used in the described experiments.

2.3. The experiments planned

The purpose of the experiments was to verify the hypothesis on better performance of the ppPDC architecture when compared to the classical PDC architecture. In order to fulfill this task, series of tests were conducted in which durations of microcycles were measured for the various numbers of variables in the system. The durations of microcycles containing 4, 8, 16, 32, 64, 128, and 256 variables were measured. In all the experiments the producers of the variables were equally distributed among all the PLCs, and the PC performed the role of the distributor.

It must be noted that, the proposed method of performance evaluation, takes the performance of the whole of the operating systems into account. The application layer was chosen for implementation, because this level of implementation was already verified in research and experimentation conducted earlier by authors. Due to the use of UDP/IP protocols it proved to be compatible with a wide range of software and hardware environments [15,16,21]. Obviously, implementation of the ppPDC at the MAC layer level should also be considered; complexity of such the task made it unattainable for now, but it is a task for the future. Still, the performance achieved by ppPDC even at the software level is comparable with the protocols based on a modified MAC.

3. Theory and calculations

The main feature of the proposed ppPDC network architecture is parallel processing of network messages in connected nodes which is the main modification when compared to the PDC architecture (see Fig. 1). As the literature states [9,10,17], the dominant time components of the communication process are the duration of the message processing in the network protocol stack of an operating system and the duration of message interpretation in applications. It would appear that parallelization of these processing times should exercise a positive influence on durations of microcycles. To determine the scale of the impact, a detailed analysis of timings of all the stages of message propagation has to be performed, for both network architectures. The following analysis is done with the optimistic assumption, that the network operates correctly with no transmission errors, and that there are no malicious network nodes. Issues of pessimistic cases, e.g. when microcycle is not realizable within given time constraints or when the network performance is reduced due to the errors in the communication links are complex and go beyond the intended scope of this work. Additionally, the analysis below assumes that both the architectures are based on the Ethernet, as

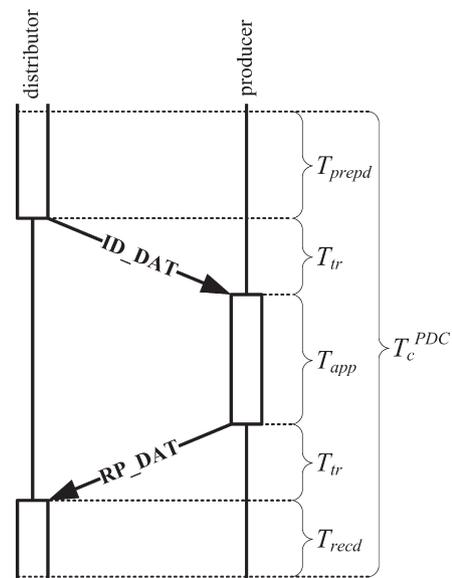
that was the case which was experimentally evaluated. Due to this assumption, several other assumptions were made, which are valid only for the case. For other media and other physical layers the calculations will be very different.

3.1. Analysis of cyclic communication in the PDC architecture

A sequence of transmissions which forms a microcycle in the PDC architecture consist of an atomic data exchange repeated several times. The atomic exchange, in which a value of a variable is transmitted to a consumer, consists of broadcasting the identifier of the variable by the distributor, followed by broadcasting the value of the variable by the producer. A detailed timing analysis of such an atomic operation has to be performed at first, which can be later used to assess whole microcycle timings.

The diagram shown in Fig. 3 illustrates sequential stages of a variable exchange. It should be noted that, apart from the distributor and the involved producer, there are also other nodes (consumers and producers) connected to the shared medium. However, their role in the data exchange is strictly passive, as they only monitor the medium activity, so they are not included in Fig. 3. The duration of the atomic variable exchange consists of the following elements:

- T_{prepd} – preparation time of the ID_DAT message containing the identifier of the variable. It is the time, it takes for the distributor application, to construct the message and pass it down through the networking protocol stack of the underlying operating system. The message eventually reaches the network adapter, where it is formed into the network frame injected into the physical medium.
- T_{tr} – propagation time of the ID_DAT frame on the network.
- T_{app} – processing time at the producer node. It consists of the postprocessing time of the ID_DAT frame and preprocessing time of the RP_DAT message. The tasks involved include passing the ID_DAT frame up through the network protocol stack, interpreting

**Fig. 3.** Timing diagram of a single exchange according to the PDC model.

the frame and formulating the answer by the producer application, passing the answer through the network protocol stack and formatting the RP_DAT network frame.

- T_{tr} – propagation time of the RP_DAT frame on the network medium. It is assumed that the time it takes for the RP_DAT frame to propagate is the same as the propagation time of the ID_DAT frame, due to the same physical distance of propagation and nearly no difference in frame length (in case of Ethernet media, small amounts of data are padded to fit the minimal payload size anyway). Due to this assumption, both the propagation times of the ID_DAT and RP_DAT frames are designated the same as T_{tr} .
- T_{recd} – receiving time of the RP_DAT frame by the distributor node, including the postprocessing time of the frame by the network protocol stack and the time it takes for the distributor application to interpret the message.

In sum, the total time of the atomic exchange of variable is expressed as $T_c^{PDC} = T_{prepd} + T_{app} + T_{recd} + 2T_{tr}$.

Processing times of the network stacks and applications are non-determinable and have to be treated as variables of unknown values. The propagation times, on the other hand, are easily and relatively precisely determinable. The frame propagation time on the Ethernet link involving a single switching queue consists of stages illustrated in Fig. 4.

The following stages of the frame propagation in a network can be distinguished:

- Transmitting the frame in the link between the sender and the switch, which takes $T_{prop} + T_{dur}$, where T_{prop} is the propagation time of the frame on the medium, depending on the length of the medium; T_{dur} is the duration time of frame transmission, which depends on the length of the frame and the bitrate of the transmission.
- Processing of the frame in the switching hardware, which takes T_{mux} . During this time, the switch analyzes the destination address of the frame and decides to which port (or multiple ports – in case of multicasting and broadcasting) the frame should be delivered, then the frame is forwarded.
- Transmitting the frame in the link between the switch and the receiving node, which takes $T_{prop} + T_{dur}$. The same designation of the times as in the first stage is the result of the assumption taken, that all the physical links between the switch and the nodes have the same length. The assumption is generally valid, as in real life, the lengths of links are usually similar.

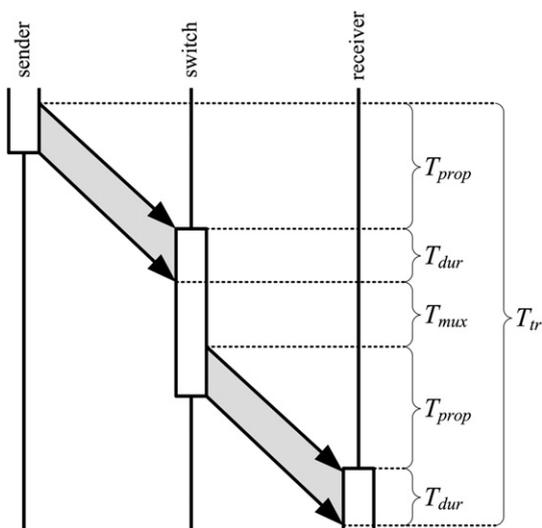


Fig. 4. Timing diagram of an Ethernet frame transmission through the switched link.

In sum, the transmission time of the frame through the switched link may be expressed as $T_{tr} = 2(T_{prop} + T_{dur}) + T_{mux}$. Values of the components of the expression can be approximated with some typical values found in the literature, or resulting from the standard [2]. Assuming that the payload of the ID_DAT and RP_DAT frames is relatively small (below 46 octets), Ethernet frames containing the messages will be 576 bits long (minimum size of an Ethernet frame, including a preamble). For the bitrate of the network equal to 100 Mbit/s, it results in T_{dur} equal to 5.76 μ s. The propagation time T_{prop} can be assumed to be about 0.1 μ s for the wires 20–30 m long [9]. The reaction time T_{mux} of a switch is characteristic to the specific manufacturer and model of the switch (or even to the specific unit), but Loeser and Haertig [10] provided a comprehensive set of values of the parameter for several cases. The representative value of the parameter for popular 100 Mbit/s Ethernet switches can be assumed to be about 45 μ s. In conclusion, the frame transmission time T_{tr} may be approximated with the value of about 55–60 μ s; for the following calculations, the worst case value of 60 μ s is used.

The microcycle performed according to the PDC model in the Ethernet network is a simple sequence of atomic variable exchanges. The duration of the cycle will therefore be equal to the variable exchange time T_c multiplied by n , where n is the number of variables exchanged during the cycle:

$$T_{cycle}^{PDC} = n(T_{prepd} + T_{app} + T_{recd} + 2T_{tr}). \quad (1)$$

Taking all the expressions and values found earlier, the same time may be expressed as $T_{cycle}^{PDC} = n(T_{ap1} + 120 \mu\text{s})$, where T_{ap1} is the sum of all undeterminable processing times within the applications and operating systems: $T_{ap1} = T_{prepd} + T_{app} + T_{recd}$ (assuming that the processing times T_{app} are equal in all of the network nodes). It is worth of noticing that, no matter what the ratio of the T_{ap1} to the T_{tr} is, the T_{cycle}^{PDC} is the linear function of the number of variables.

3.2. The analysis of cyclic communication timings in the ppPDC architecture

In Fig. 5 stages of a microcycle performed according to the ppPDC architecture and their durations are shown.

The ppPDC schedule involves the phenomena far more complicated than the PDC architecture, due to the active exploitation of the switching queue. Total duration of a microcycle consists of the following components:

- T_{prepd} – preparation time of the content of the BROAD frame. The preparation involves the distributor application formulating the content of the frame and passing the message through the networking protocol stack of the operating system, which results in constructing the frame and injecting it into the physical media.
- T_{tr2} – propagation time of the BROAD frame on the network. This time is similar to the T_{tr} time of the PDC architecture, but the length of the message transmitted differs, so the T_{dur} component changes. It has to be assumed that the payload of the BROAD message is relatively large, so the safe assumption regarding the Ethernet frame containing it, will be that the frame has maximal allowed length, i.e. 1526 octets (including the preamble). In effect it may be concluded that $T_{tr2} = 2(T_{prop} + T_{dur2}) + T_{mux}$, where T_{dur2} is the time needed to propagate the 1526 octets over the medium, which equals to 122.08 μ s for 100 Mbit/s rate. In effect, the value of T_{tr2} may be considered to be about 290 μ s.
- T_{app} – processing time at the producer node. It consists of processing time of the BROAD frame and preparation time of the RP_DAT message, including the processing times of the messages in the networking protocol stack and computing time of the producer application.
- T_{que} – delivering time of the RP_DAT frames. The delivery of the RP_DAT frames consists of the physical transmission of the frame between the producer and the network switch, processing of the

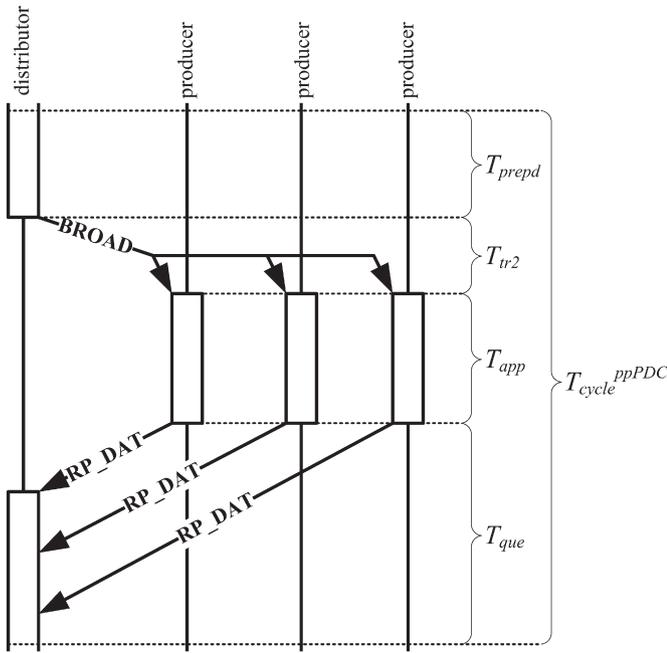


Fig. 5. Timing diagram of a microcycle according to the ppPDC model.

frame within the switching hardware (forwarding, queuing), physical transmission between the switching unit and the distributor node, and processing of the frame at the distributor node.

It should be noted that some of the symbols, which were used during the analysis of the PDC architecture, are used again to denote timings of the ppPDC model. The repetition is intentional, because some of the timings in both the architectures correspond with each other, and their durations will be the same, regardless of the architecture exploited. The same denotations of these similar durations will allow to compare the architecture performances later:

- T_{prepd} – preparation times of the BROAD and ID_DAT frames are comparable, because both the processes take place in the same network node, i.e. the distributor. The only difference between both the cases is the content of the messages being prepared, but still, the varying contents have to be processed by the same networking protocol stack and the same hardware network adapter of the same physical network node.
- T_{app} – processing times at the producer applications do not change, because the main principle of producer nodes operation stays the same. The nodes receive the request (either the ID_DAT or the BROAD frame), which has to be processed, and the answer has to be prepared, within the same hardware bounds (the protocol stacks and the hardware are the same).

The queuing and forwarding time is determined by many phenomena and requires separate detailed explanation. The sequence of events involved in the process is shown in Fig. 6.

To simplify the analysis of phenomena constituting this stage of the microcycle, it is assumed that the processing times T_{app} for all the producing nodes are the same. This assumption does not affect the accuracy of the considerations, because these times are in fact nearly the same due to the symmetry of the star-shaped network. If the processing times vary significantly, for example due to highly varying processing power of the nodes, it should be assumed that the time used for the analysis is the largest one. While the considerations would be pessimistic in this case, the results would still be valid. The additional assumption is that the wires connecting all the

nodes to the switching unit are of the same length, so the propagation times on the different network segments are the same.

The following stages of the RP_DAT frames collecting by the distributor can be distinguished:

- The producer nodes transmit the formulated RP_DAT frames over the physical links. An assumption that all the RP_DAT frames reach the switching unit at the same time results from the previous assumption that the propagation times on all links are equal. Nevertheless, the following analysis is also valid for the cases when frames do not reach the network switch at the same moment, but are delivered fast enough to not allow the switch's queue to empty, i.e. for cases when intervals between successive frames reaching the switch do not exceed $T_{prop} + T_{dur} + \min(T_{mux}, T_{gap})$.
- The RP_DAT frames (or the first of the frames, if they are not delivered simultaneously) propagate on the links between the producers and the network switch. The process takes $T_{prop} + T_{dur}$.
- The switching unit processes the first received frame, determines the destination port and forwards the frame. The duration of this stage is designated as T_{mux} .
- The first forwarded frame propagates on the link between the switch and the distributor node, which takes $T_{prop} + T_{dur}$.
- When the distributor node begins to process the received frame (the processing time denoted as T_{recd}^1), the switch sends another queued frame to the distributor (after waiting for T_{gap} , which is the minimal Ethernet inter frame gap).

The whole process finishes after the distributor node processes last received RP_DAT frame. In summary $T_{que} = 2(T_{prop} + T_{dur}) + T_{mux} + (n - 1) \max((T_{gap} + T_{prop} + T_{dur}), T_{recd}) + T_{recd}$. The expression can be significantly simplified when the arguments of the maximum function are evaluated. The T_{gap} value is defined by the Ethernet standard and equals to $0.96 \mu s$ for the 100 Mbit/s rate, therefore the $T_{gap} + T_{prop} + T_{dur}$ expression has value of about $7 \mu s$. The value of the second argument of the maximum function, i.e. T_{recd} , is unknown,

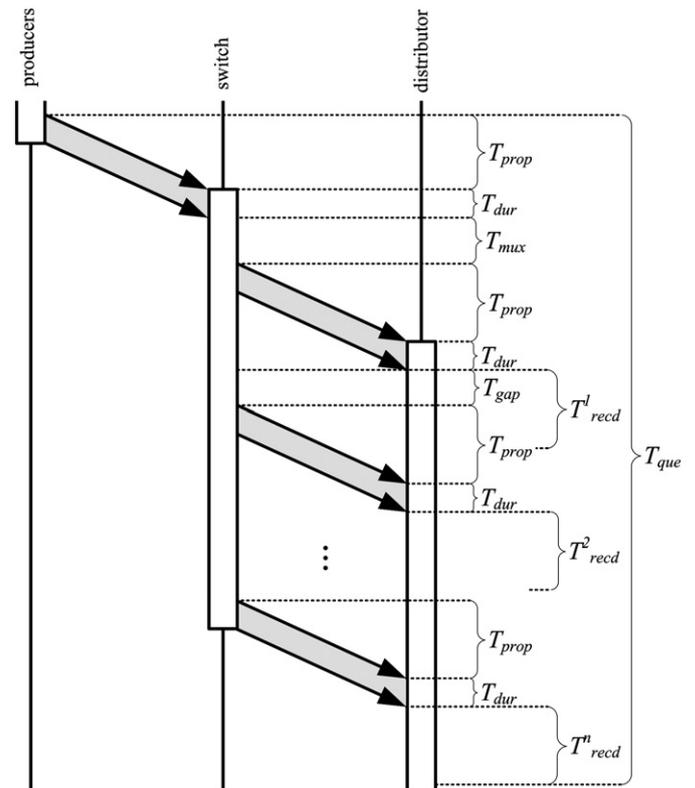


Fig. 6. Timing diagram of the RP_DAT frames queuing in the ppPDC model.

since it is the result of many processes taking place in the network protocol stack of the operating system of the distributor node. However, taking into account that the processing times in network node software are much greater than the physical propagation times [17], it may be safely assumed that the T_{recd} processing time is greater than the $T_{gap} + T_{prop} + T_{dur}$ expression. The validity of the assumption is verified by the experiments conducted, which showed that a duration of a microcycle was always at least two orders of magnitude greater than the mentioned 7 μ s. Taking the assumption into account, the T_{que} may be reformulated as $T_{que} = 2(T_{prop} + T_{dur}) + T_{mux} + nT_{recd}$.

Using the value of T_{que} , total duration of a microcycle in the ppPDC architecture is expressed as $T_{cycle}^{ppPDC} = T_{prepd} + T_{tr2} + T_{app} + 2(T_{prop} + T_{dur}) + T_{mux} + nT_{recd}$, which can be rewritten as

$$T_{cycle}^{ppPDC} = T_{prepd} + T_{tr} + T_{tr2} + T_{app} + nT_{recd}. \quad (2)$$

Taking all the expressions and values found earlier, the same time may be expressed as $T_{cycle}^{ppPDC} = T_{ap2} + 350 \mu\text{s} + nT_{recd}$, where T_{ap2} is the sum of undeterminable processing times: $T_{ap2} = T_{prepd} + T_{app}$. It is worth of noticing that the T_{cycle}^{ppPDC} is the linear function of the number of variables with the non-zero constant term.

To compare the magnitudes of the microcycle duration times obtained in the two communication scenarios, expression (1) and expression (2) should be compared. By subtracting (2) from (1) the following is obtained:

$$T_{cycle}^{PDC} - T_{cycle}^{ppPDC} = (n-1)(T_{prepd} + T_{app}) + (2n-1)T_{tr} - T_{tr2}.$$

This expression defines the microcycle time gained by replacing the PDC architecture with the ppPDC one. It is important to note, that the gain consists in large part of the undeterminable and relatively large processing times. The expression is also a linear function, so the more variables are contained in the microcycle, the more time is gained.

In case of a microcycle containing only one variable, the gain equals to $T_{tr} - T_{tr2}$. Since it was previously shown that T_{tr2} is greater than T_{tr} , it could be inferred that for the one variable microcycle the ppPDC architecture is less efficient than the PDC. This is however an incorrect inference, since it is simply the effect of the safe assumption taken when evaluating the values of T_{dur} and T_{dur2} (which are the components of T_{tr} and T_{tr2} respectively). In fact, in case of the one variable microcycle, due to small length of the BROAD frame, the T_{dur2} will be equal to the T_{dur} , so both the architectures are equally efficient in the case. In all the other cases, the ppPDC microcycles are shorter than the respective PDC ones. This conclusion was verified by the conducted experiments.

4. Results/Discussion

A result of a single experiment in which durations of consecutive microcycles were measured is a set of samples, which can be presented as a simple graph. An example of such graph is provided in Fig. 7, which shows the results of the single experiment for the ppPDC architecture and microcycles containing eight variables.

Attention should be paid to the very specific distribution of the samples, as they tend to group into clusters. It is clear that the distribution of the data is multimodal. Multiple modes are explained by the very complex nature of the processes, which are investigated. A duration of a microcycle depends on many factors, as it involves processing the messages by complex multilayered protocol stacks of operating systems. NI cRIO runs VxWorks operating system, which is a real-time OS, known in automation and robotics. The jitter is most likely caused by the work mode of network adapters of cRIOs (i.e. interrupt mode – in contrast to the strict real-time polled mode), and the OS of the computer (i.e. MS Windows 7) used as the

distributor node. Therefore, the microcycle times contains, among others, the execution times of applications, services, and cores of operating systems. Since the main property of the ppPDC architecture is the minimization of the impact of these times on the microcycle time, they had to be taken into account in the performance evaluation.

Anyhow, the resulting distributions of microcycle times have to be characterized by some measures, which would allow to compare the network performances in various cases. Due to multimodality of the distributions, the most popular measures, such as arithmetic mean, standard deviation, maximum, and minimum, are not representative due to their sensitiveness to the multiple modes, skewness of distributions, and sample size. Some robust measures were chosen to represent the characteristics of data gathered, i.e.: first decile, first quartile, median, third quartile, and ninth decile. The collection of the measures calculated is presented in Table 3. The arithmetic mean is also provided in the table for comparison with the median.

In Fig. 8 the deciles of the gathered data are illustrated, in function of the number of variables exchanged in the microcycle. In effect, the figure shows the expected duration time of the microcycle with the probability of 80%. Similar presentation of the max–min range would be pointless due to the susceptibility of the range to the sample size. Consequently, Fig. 9 presents the medians of the gathered data as functions of the number of variables. Both the figures represent the robust measures of location and dispersion of distributions of microcycle times.

Charts in Fig. 9 can be fitted with the method of the least squares. The resulting functions are defined by the following equations (values in seconds):

$$\begin{aligned} T_{fit}^{PDC} &= 0.0009n - 0.0007, \\ T_{fit}^{ppPDC} &= 0.00006n + 0.0009. \end{aligned}$$

The negative value of one of the parameters is a side effect of fitting the function to the specific realization of the random variable. It is hard to determine the values of variables of Eqs. (1) and (2) using the fitted parameters, but it is worthwhile to compare the two obtained slope factors. As it is seen, in case of the proposed ppPDC architecture, the value of slope constant is about an order of magnitude smaller than in the PDC model. It is the confirmation of the hypothesis on the better performance of the modified ppPDC scheduling when compared to the classical PDC one.

5. Conclusions

Although the performance of the ppPDC is evidently better than PDC, there is a price for the increased performance of the ppPDC architecture, which is the limit on the amount of data carried by a BROAD frame, resulting from the maximum allowable payload of an Ethernet frame. Using the previously presented simple formats of the frames (Table 2), the maximum amount of the process data possible to be carried by a BROAD frame was derived. Using these quantitative limits, an exact maximum number of variables can be determined, assuming that the size of a variable is known. For example, assuming that all the variables are of 8-bit type, there can be maximally 1472 of them in a microcycle. The limits determined for the other sizes of the variables are shown in Table 4. When the limits are reached, it should be considered to divide the long microcycle into few shorter consecutive microcycles. The jumbo frame functionality of modern Ethernet networks could also provide solution to the problem.

Another significant disadvantage of the model described is the delay between the data production by producers and the delivery to consumers. In the presented case of application-level implementation, the delay in ppPDC model is more than an order of magnitude smaller than the duration of a whole microcycle in PDC model, so it

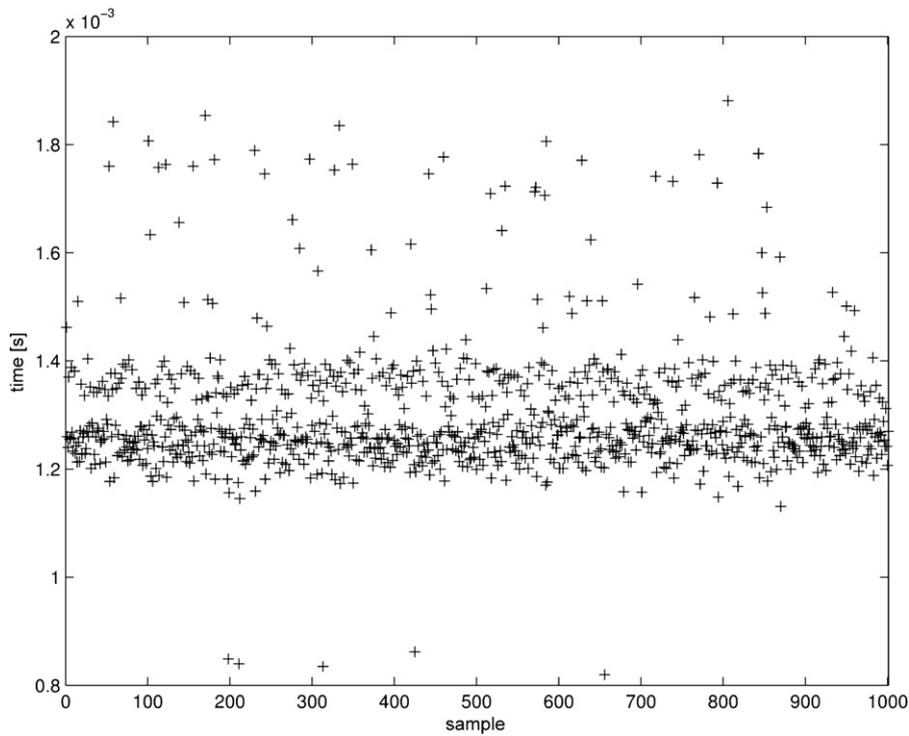


Fig. 7. The example of data gathered during the experimentation.

is negligible. However, in case of low-level ppPDC implementation for quick systems, the delay could be significant, and its impact should be carefully assessed.

There are also other limitations resulting from the maximum capacity of a network link [9], which result in the relationship between the demanded period of variable exchanges and the possible maximum number of the variables. However, because of the high bit-rate of Ethernet (100 Mbit/s in the presented case) these imposed time limits are not significant, since the most important restrictions are imposed by the layer of applications and operating system. These latter limits are not possible to be determined analytically due to the non-determinism introduced by the layer. The limits can be assessed only by using the results of the performance evaluation presented in this paper.

An important aspect of the implementation of the ppPDC architecture in place of the PDC is the change of a network topology. The PDC architecture assumes that a network is based on a shared medium, so the PDC is by definition intended for the networks employing the bus topology. The ppPDC architecture assumes that the medium is divided into separate zones of collisions, which are connected to the frame

queuing and forwarding unit. In effect, the ppPDC is intended for the network topology of a star. The bus topology is currently well established in the industrial reality, since virtually all of today's industrial networks use this topology, while the star topology is nowadays dominant in the field of LANs, due to the popularity of the switched Ethernet. The change of topology may be unwelcomed by engineers and practitioners, but the idea can be easily promoted, since the star topology is convenient in many applications, and often it is even more justified than the bus topology (see [21]).

Implementation of the ppPDC at the MAC level is currently considered by authors. In case of such implementation, some of the considerations provided in the paper lose validity. In particular, assumptions regarding relation of processing times to physical transmission times should be re-evaluated. Anyway, physical implementation is expected to be particularly valuable, as it will allow to compare the performance of ppPDC, PDC, and FIT in real application contexts, since theoretical comparisons are often flawed [22].

Table 3
Statistical measures (in milliseconds) of observed microcycle durations in both models.

		Number of variables						
		4	8	16	32	64	128	256
ppPDC	Arith. mean	1.12	1.30	1.84	2.85	4.75	8.55	16.20
	1st decile	0.82	1.20	1.67	2.63	4.52	8.27	15.94
	1st quartile	0.91	1.23	1.70	2.81	4.59	8.36	16.07
	Median	1.17	1.27	1.81	2.86	4.73	8.57	16.20
	3rd quartile	1.24	1.35	1.97	2.92	4.90	8.72	16.33
	9th decile	1.33	1.39	2.07	3.02	5.00	8.80	16.44
PDC	Arith. mean	3.49	5.72	13.93	27.95	56.01	112.32	226.96
	1st decile	3.26	5.29	13.45	27.32	55.19	110.94	224.79
	1st quartile	3.41	5.55	13.73	27.63	55.56	111.60	225.95
	Median	3.50	5.72	13.98	27.96	56.00	112.32	227.18
	3rd quartile	3.60	5.87	14.15	28.21	56.42	113.03	228.13
	9th decile	3.69	6.07	14.31	28.46	56.81	113.69	229.20

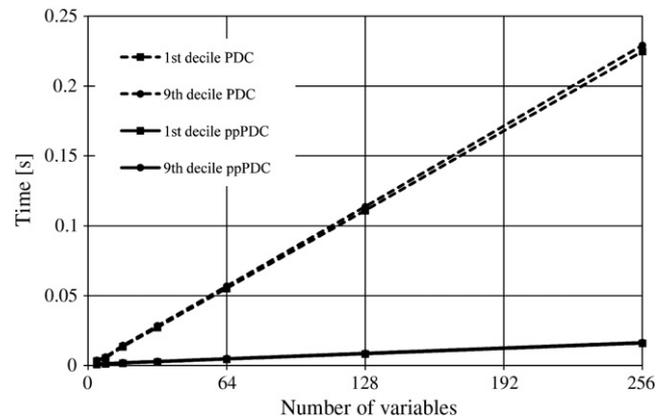


Fig. 8. Deciles of the observed microcycle time distributions.

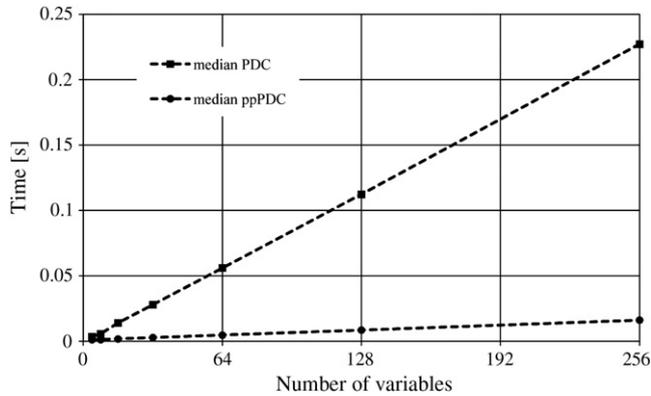


Fig. 9. Medians of the observed microcycle time distributions.

Table 4

Maximum number of variables possible to be broadcasted at once in a BROADCAST frame.

Type/length of variables	Maximum number of variables
8 bits	1472
16 bits	736
32 bits	368
64 bits	184
128 bits	92

Because an experimental verification is the most recognized method of the performance evaluation of a communication network and the method is widely used, there is a broad base of the results for similar protocols and applications available in the literature, to which results gathered in the paper are directly comparable – see for example [23–26]. This proves that ppPDC, while simple to develop, provides very attractive properties comparable to low-level hardware industrial protocols even when implemented at the application layer.

Acknowledgements

This work was supported by the National Science Centre under grant no. 2012/05/B/ST7/00096.

References

- [1] J.A. Dienes, M. Díaz, B. Rubio, Using standards to integrate soft real-time components into dynamic distributed architectures, *Computer Standards & Interfaces* 34 (2012) 238–262.
- [2] IEEE802.3, IEEE standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—Part 3: carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2002.
- [3] R. Moraes, F.B. Carreiro, P. Bartolomeu, V. Silva, J.A. Fonseca, F. Vasques, Enforcing the timing behavior of real-time stations in legacy bus-based industrial Ethernet networks, *Computer Standards & Interfaces* 33 (2011) 249–261.
- [4] S. Vitturi, L. Peretti, L. Seno, M. Zigliotto, C. Zunino, Real-time Ethernet networks for motion control, *Computer Standards & Interfaces* 33 (2011) 465–476.
- [5] L. Kren, Look out factory floor—here comes Ethernet/IP, *Machine Design* (2000) 82–86, (15/6/2000).
- [6] M. Babb, The race for real-time Ethernet, *IEEE Computing and Control Engineering* 15 (1) (2004) 8.
- [7] L. Winkel, Real-Time Ethernet in IEC 61784-2 and IEC 61158 series, 4th IEEE International Conference on Industrial Informatics, 2006, pp. 246–250.
- [8] IEEE802.1D, IEEE standard for local and metropolitan area networks—Media Access Control (MAC) bridges, 2004.
- [9] K.C. Lee, S. Lee, Performance evaluation of switched Ethernet for real-time industrial communications, *Computer Standards & Interfaces* 24 (2002) 411–423.
- [10] J. Loeser, H. Haertig, Low-latency hard real-time communication over switched Ethernet, *Proceedings of 16th Euromicro Conference on Real-Time Systems*, Catania, Sicily, 2004, pp. 13–22.
- [11] T. Skeie, S. Johannessen, O. Holmeide, Timeliness of real-time IP communication in switched industrial Ethernet networks, *IEEE Transactions on Industrial Informatics* 2 (1) (2006) 25–39.

- [12] G. Cena, L. Seno, A. Valenzano, S. Vitturi, Performance analysis of Ethernet Powerlink networks for distributed control and automation systems, *Computer Standards & Interfaces* 31 (2009) 566–572.
- [13] International Electrotechnical Commission, Digital data communications for measurement and control—fieldbus for use in industrial control systems—parts 2 to 6, IEC 61158, 2000.
- [14] International Electrotechnical Commission, Digital data communications for measurement and control—part 1: profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems, IEC61784-1, 2003.
- [15] G. Polaków, M. Metzger, Design and implementation of Ethernet-based horizontal communication scheme using LabVIEW, *Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics*, 2006, pp. 829–834.
- [16] G. Polaków, M. Metzger, Agent-based framework for distributed real-time simulation of dynamical systems, in: A. Håkansson, et al., (Eds.), *KES-AMSTA 2009*, LNAI 5559, Springer, Heidelberg, 2009, pp. 213–222.
- [17] J.-D. Decotignie, Ethernet-based real-time and industrial communications, *Proceedings of the IEEE* 93 (6) (2005) 1102–1117.
- [18] S. Vitturi, On the use of Ethernet at low level of factory communication systems, *Computer Standards & Interfaces* 23 (2001) 267–277.
- [19] S. Zakaria, L. Abdenour, S. Aomar, Ethernet based implementation of a periodic real time distributed system, *Electrical Engineering and Applied Computing* 90 (2011) 337–350.
- [20] P. Bilski, W. Winięcki, Methods of assessing the time efficiency in the virtual measurement systems, *Computer Standards & Interfaces* 34 (2012) 485–492.
- [21] G. Polaków, M. Metzger, ppPDC communication framework—a new tool for distributed robotics, in: S. Carpin, et al., (Eds.), *SIMPACT 2008*, LNAI 5325, Springer, Heidelberg, 2008, pp. 191–202.
- [22] L. Seno, F. Tramarin, S. Vitturi, Performance of industrial communication systems: real application contexts, *IEEE Industrial Electronics Magazine* 6 (2) (2012) 27–37.
- [23] S. Cavaliere, Modelling and analysing congestion in KNXnet/IP, *Computer Standards & Interfaces* 34 (2012) 305–313.
- [24] L. Chen, Z. Shan, T. Tang, H. Liu, Performance analysis and verification of safety communication protocol in train control system, *Computer Standards & Interfaces* 33 (2011) 505–518.
- [25] D.-S. Kim, S. Lee, Feasibility analysis of hybrid control networks based on common industrial protocol, *Computer Standards & Interfaces* 33 (2011) 357–366.
- [26] H. Pérez Tijero, J.J. Gutiérrez, On the schedulability of a data-centric real-time distribution middleware, *Computer Standards & Interfaces* 34 (2012) 203–211.



Grzegorz Polaków graduated with a Master's Degree in Control Science and Robotics from Silesian University of Technology in 2005. After three years of doctoral studies he received his PhD in automation. Currently he is an assistant professor at the Institute of Automatic Control. His research interests include all aspects of the IT presence in industrial automation, e.g., communication in distributed control systems, industrial networks, PLCs programming languages, multiagent systems in control.



Mieczysław Metzger received his M.Sc degree in automatic control from the Silesian University of Technology Gliwice, Poland in 1971. In 1978 he received his Ph.D., and the D.Sc. (habilitation) in 1989 in modelling, simulation and control of industrial processes. He is currently the titular professor and a full professor at the Silesian University of Technology and since 1992 Vice Director for Research of the Institute of Automatic Control at the University. He is an author or co-author of about 200 research papers in international journals as well as conference proceedings and 11 paper books for students. Some of the papers published in journals "Simulation", "Simulation - Practice and Theory", "Control-Theory and Advanced Technology", "Water Science and Technology", "Systems Analysis, Modelling, Simulation", "Chemical Engineering and Processing", "AICHE Journal", "Applied Thermal Engineering", "Control and Cybernetics". He is author of the book: *Modelling, Simulation and Control of Continuous Processes* (2000). He is a member of IWA and IEEE.

His current research interests include modelling, simulation (conventional and real-time training simulators) and control of continuous industrial processes, including agent-based and network-based distributed control systems. The subjects of his studies are dynamic properties and control of processes in environmental protection, in chemical and power plants and in biotechnology. He was the moving spirit behind the development of the industrial-scale pilot plants treated as real-world control plants. All of these plants improve experimental basis for research in the industrial automation and in the industrial informatics.