The Concept of Using Multi-protocol Nodes in Real-Time Distributed Systems for Increasing Communication Reliability

Andrzej Kwiecień[1], Marcin Sidzina[2], and Michał Maćkowski[1]

[1] Silesian University of Technology, Institute of Informatics
[2] University of Bielsko-Biala, Department of Mechanical Engineering Fundamentals
andrzej.kwiecien@polsl.pl,
msidzina@ath.bielsko.pl,
michal.mackowski@polsl.pl

Abstract. The paper presents the considerations on the method which enables to accelerate the exchange of data in distributed control system, using multi-protocol nodes. The method presented in this paper is related to hardware and software, and based on new original communication protocol, which is used by the node for supporting various communication protocols to simultaneous data exchange via different buses. This method will enable to increase not only network bandwidth, but also the security of data transmission.

Keywords: distributed control system, redundancy, network node, industrial networks, distributed real-time systems, industrial protocols, medium access, acceleration of data exchange.

1 Introdution

Industrial real-time distributed systems are built on the basis of several models of access ([1–6]) to the communication link:

- Master-Slave,
- Token-Bus, Token-Ring,
- PDC (Producer-Distributor-Consumer),

or other being a combination of the above. There are many commercial products, which implement partly or entirely models mentioned above ([7–9]). It is also necessary to mention that there are many attempts and many works which refers to the use of Ethernet in industrial communication, such as in [10–13]. However the mentioned scope, is out of the authors' interest. Thus it was not discussed in the paper. Taking into consideration the reliability of the system (which can be increased thanks to the redundancy) or network overload (permanent or temporary) there is a problem how to increase the flexibility of network connections in order to improve the time of data exchange. The mentioned redundancy can also be used ([14, 15]) as a way of improving the transmission time parameters [14]

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2013, CCIS 370, pp. 177–188, 2013.

and it also may increase the bandwidth. However, it should be noted that the greatest advantages of using redundant communication link to increase the performance of transmission time, occur when the redundancy already exists in the system (for example because of the reliability) ([16, 17]). This is due to the high cost of system with redundancy. Based on the above discussion some questions arise. First of all, whether there are other possibilities capable to unload intensive data traffic, especially when temporary overload occurs ([18–20])? Next, is there a possibility to take advantage of multi-protocol communication that uses specialized nodes of a distributed system (maintaining the rules of real-time system) in case of system failure, such as: damage of transmission system (cable, optical fiber, network coprocessor, etc.), events in industrial area (communication failure) or other reasons, the security of transmission can be ensured? It would be interesting to use multiprotocol communication [20] using specialized nodes of distributed system according to rules of real time system.

The rules of real-time system means that the global time response to any event or serious of events cannot be exceed.

$$T_{\mathrm{RE}} < T_{\mathrm{G}} \tag{1}$$

where:

T_{RE} – the result of design and is the maximum time of system reaction to an event or the sequence of events,

T_{G} – is the maximum of acceptable response time of system to the sequence of events and it is usually force by the technology of system.

2 The Concept of Multi-protocol Node

Figure 1 presents the concept of multi-protocol network with one highlighted node $A0$, which additionally includes three communication interfaces with A, B and C networks. Subscribers $B1$ and $B4$ are connected to network A, subscribers $B2$, $B3$ and $B4$ to networks B as well as C. It is assumed that node $A0$ is the master station to networks A, B and C. The idea is that each node of network S may be a node for other network. Obviously, the networks A, B and C may use various protocols and transmission media. For these considerations some simplifications have been made, which in the authors opinion do not limit the scope of current discussion.

Figure 1 presents network S (“ring”) which can be defined as follows:

$$S\{N\langle *A0\rangle \gg N\langle A1\rangle \gg N\langle A2\rangle \gg N\langle A3\rangle \gg N\langle A4\rangle \gg N\langle A5\rangle \gg N\langle *A0\rangle\} \tag{2}$$

where S stands for network name, and $N\langle NAME\_OF\_NODE\rangle$ is node name $NAME\_OF\_NODE$.

Notation (2) should be then interpreted as follows: Network S (here “ring”) consists of six nodes called: $A0$, $A1$, $A2$, $A3$, $A4$, $A5$, which are connected together in the following way: $A1$ with $A2$; $A2$ with $A3$; $A3$ with $A4$; $A4$ with $A5$; $A5$ with $A0$ and $A0$ node is additionally the node of other or others networks.

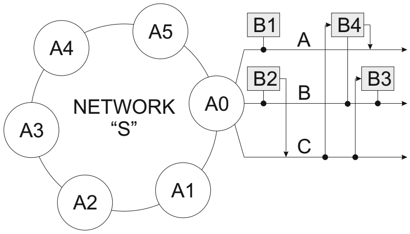

Fig. 1. The idea of network with multi-protocol nodes

The connections between the nodes are arranged as an example, because there are many possible ways to connect the nodes. A given example is only to keep order. If the node name is preceded by * it means that a particular node is at the same time a node for other networks.

As the above notation indicates, this is the network of "ring" type with undefined protocol. The only requirement is that it should be a deterministic protocol that guarantees to each subscriber the access to the communication medium within a maximum time T_{ACC}. For further discussion network S will not be taken into account. The only node taken under discussion is node $A0$ in order to describe the phenomena which can be reproduced on other nodes in network S. However, it is important to remember that networks A, B and C do not have to be isolated from network S, which means that any subscriber of network S can transmit data to/from networks A, B and C via node $A0$. As already mentioned, at the current stage of considerations these data transfers are not taken into account.

The considered network is a Master-Slave network in which the node $A0$ is the Master station. Choosing the Master-Slave model to these considerations was intentional, but it in no way limit the scope of discussion. It intends only to make a practical study designed to answer a number of questions concerning the possibility to reduce transmission time and increase its reliability. Thus, the network with node $A0$, can be described as follows:

$$\mathrm{Master}A0\,\underbrace{\{N\langle B1\rangle N\langle B4\rangle\}}_{\text{Network }A}\,\underbrace{\{N\langle B2\rangle N\langle B3\rangle N\langle B4\rangle\}}_{\text{Network }B}\,\underbrace{\{N\langle B2\rangle N\langle B3\rangle N\langle B4\rangle\}}_{\text{Network }C} \quad (3)$$

where $A0$ is simultaneously master station for networks A, B and C.

And node $B1$, $B2$, $B3$ and $B4$ are slave stations in networks A, B and C.

It should be stated that Master-Slave networks are characterized by the fact that Slave devices are of secondary nature and they cannot make a communication in network system. This is due to the fact that all exchange scenario is supervised by Master station. The only one station which is able to establish network communication is Master. It sends a request for the answer to other nodes and request to write data into memory address space of Slave station. Triggering the communication by the Slave station, is not very often discussed in literature. However if this problem is mentioned then it is usually limited to the possibility of system node failure or system crash but outside the computer system. The implementation of special procedures, which are prepared for such situations is constrained with many restrictions. These procedures are only performed in critical situations, for example when the communication system tends to fall.

The Master-Slave protocol has been defined for a single communication bus. The device coprocessor which operates in the network normally does not support two communication buses. In order to increase the reliability and security of the network by introducing a redundant system it is necessary to equip the nodes in this system with additional interfaces (communication coprocessors). It is also possible to use for example "sleep" Master stations which take control over the network on case of failure of communication for the main Master station [14, 15].

Thus $A0$ node is Master station. According to the idea of such network, the so called exchange scenario is placed in Master station. It determines the order of exchanges between Master and Slave stations. In a classic Master-Slave network, where Master is equipped with one communication interface, the analysis of data flow is relatively simple. The concept contains also one exchange scenario and the Master station has to choose via which bus the request should be sent in order to minimize the exchange time.

Figure 2 presents the exchange scenario. It requires the data transfer to/from specific nodes ($B1$, $B2$, $B3$, $B4$). This transfer may occur regardless of situation in network S. However, the data transmitted by a node $A0$ may not be local value essential for networks A, B and C. Then, the essential will be "time stamp" or in other words "recent indicator", which will be transmitted by $A0$ node:

– At the time of receiving them from network S (data from the network S will be forwarded to the network A, B and C).
– At the time of receiving them from any A, B or C network (data will be transmitted to the network S).

Therefore, assume the exchange scenario as in Fig. 2 (realized from top).

The first data to be transmitted are data to/from node $B1$. It can be realized only with the use of network A. Next transaction is to/from node $B2$. This can be done in three ways:

– with network B (if the network is not busy),
– with network C if the network B is busy, or
– waiting until one of them is not busy.

The next data to be transmitted are to/from the node $B3$. Three cases can be considered:

– the network B will be used when it is not busy, or
– data will be transmitted with network C, in case when network B is busy and C is free, or
– waiting until one of them is not busy.

In the following step, data to be transmitted are data to/from node $B4$. In this case, the transfer can be accomplished in the following way:

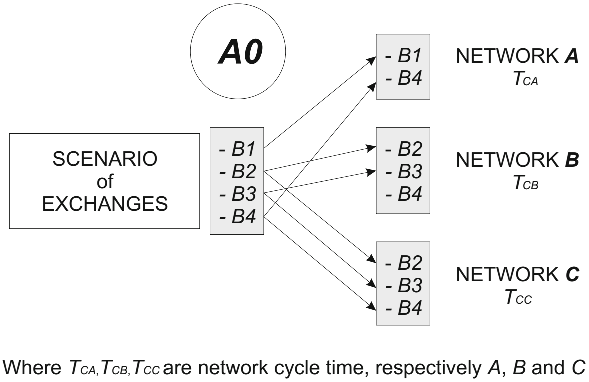

Fig. 2. The exchange scenario in networks A, B and C

- with the network A (if free), or
- with network B, if A is busy, or
- with network C, if the both above are busy, or
- waiting until one of them is not busy.

3 Criterion of Network Selection

With this proposed sequence of exchanges, there are some problems with choosing, which of the available network will be used for data transmission. This will depend on a number of factors that are associated with both transmission parameters as well as differences in transmission protocols. A network designer will be faced to the problem related to the criterion of network selection. The more differences in time cycle of particular network (A, B or C), the more important the problem will be. Such situation occurs when the protocols are significantly different from each other because of using various transmission media. The criterion of network selection may be described in several steps as follows:

1. Determining the fastest network based on the test data, (T_{TR}).
   In this point the static calculation of T_{TR} parameter is done, which corresponds to the transmission time of a particular data block or even a single frame as a function of data transfer rate, protocol overhead (number of bits per single transmitted character, numbers of control bit, etc.) and the size of data block. Having calculated T_{TR} it is possible to order the priority of using particular network as a function of its speed.
   The aim of the basic research was to determine the duration of data transmission depending of the type of transmission protocol, duration of application of master station and coprocessor type (Sect. 4).
2. Preparing the information exchange.
3. Checking the busy flag for each network FL_i.

4. Selecting network A, B or C based on the T_{TR} value and network busy flag Fl_i.
5. Running the transmission on a particular data bus (order of transmission to the selected coprocessor).
6. Fetching the next exchange transaction (B_i) and return to step 2.

After completing all exchanges, the realization of exchange scenario starts from the beginning. It is possible that data between networks are exchange asynchronous. Which means that the flow in the network S is totally independent from flow in networks A, B, C.

It should be noted that the configuration of the network, the selection of nodes and assigning them to particular networks will be a time consuming process. A necessary condition to obtain the proper effects is to provide the nodes with a sufficient and independent communication interfaces (network coprocessors). Additionally, some problems appears which are the object of the research. The above discussion assumed that the network operates without any failure. This group may consists of:

– If at least one of the networks (A, B, C) will be a token ring network, it will be necessary then to calculate the cycle time of the network in order to eliminate the phenomenon of monopolizing the communication link;
– The necessity of estimating the maximum time of exchange in each node of A type, in order to calculate the bandwidth of entire network;
– If the protocols will vary significantly, then the algorithms of network failure detection will be getting more complicated.

4 The Initial Research

The method discussed in the paper, or in other words a schema of operations, has been developed on the basis of many tests, which aimed at shortening the duration of data exchange in distributed real time communication system. The research results confirm a lot of relationships:

– Duration of the basic cycle of a node has a significant influence on duration of data exchange (Fig. 3–6).
– There is a possibility to shorten effectively the duration of automation cycle, both static and dynamic, in order to reduce the duration of data exchange, e.g. [21].
– Generally, each type of coprocessor and its localization (e.g. embedded in central unit, installed in PLC slot) is characterized by different exchange durations. It is advised to conduct separate tests for each device, which check the duration of information exchange (comparison of Figures 3 with 4).

As it was mentioned in Sect. 3, the main goal of research was to determine the duration of data transmission (T_{TR}), depending on the type of transmission protocol, the duration of application of Master station and type of coprocessor

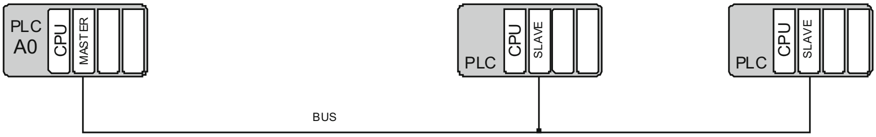

Fig. 3. The schema of test bench for Modbus/RTU (separate network coprocessor)

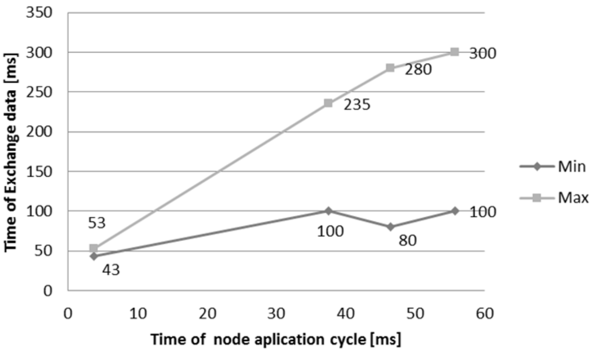

Fig. 4. Test bench 1. Coprocessor of Modbus/RTU protocol placed in PLCV Reading of ten words for automat cycle is as follows: 3.7 ms, 37.6 ms, 46.5 ms, and 55.8 ms.

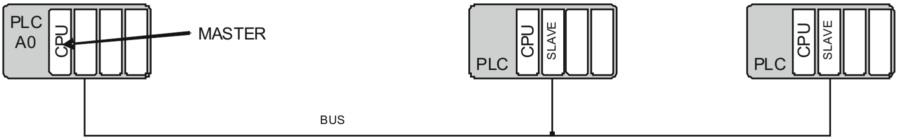

Fig. 5. The schema of test bench for MODBUS/RTU (network coprocessor embedded in central unit)

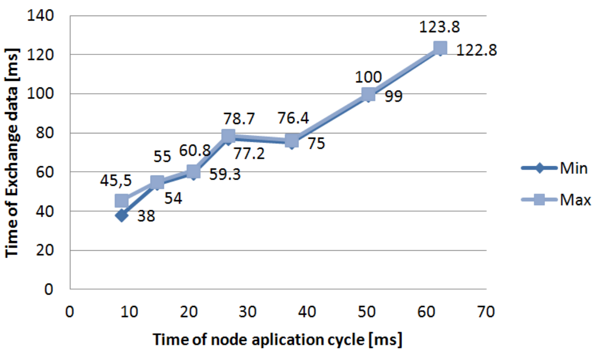

Fig. 6. Test bench 2. Coprocessor for Modbus/RTU placed in the central unit. Reading of ten words for automation cycle is as follows: 8.6 ms, 14.6 ms, 20.7 ms, 26.6 ms, 37.2 ms, 50.2 ms and 62.3 ms.

(Sect. 3). It is worth to mention once again the fact that networks connected to $A0$ node can be networks with various protocols (including Token-passing).

The first phase of empirical test consisted of measurements whose aim was to determine the relationships between various network coprocessors and central unit. The basic parameters, which were taken into account included: duration of cycle of application node and the type of used transmission protocol. Therefore, there were four test benches prepared for measuring those parameters. The duration of information exchange was measured with the use of communication protocols: Modbus/RTU, Token-Passing, SRTP (based on TCP), for different durations of cycle of node application.

Test bench 1 for Modbus/RTU protocol consisted of one Master station and two Slave stations. The stations (Slave) were given unique numbers of ID station. Coprocessor of Master station was placed in PLC, and it was the only one stations authorized to establish communications.

Figure 4 presents the research results, which indicate that the maximum time of information exchange for determined cycles' duration of network applications was 300 ms. What is more, it was noticed that there was a significant difference between the minimum and maximum time of information exchange. The longer the duration of application, the greater difference between times is. It is a consequence of network structure, in which a coprocessor is an independent device that works on the same bus as the rest I/O modules of PLC controller.

Test bench 2 for Modbus/RTU protocol consisted of one Master station and two Slave stations. As previously, Slave stations were given unique ID numbers, but coprocessor of Master station was placed in PLC central unit. This station was the only one authorized to establish communication.

Figure 6 presents the results, which clearly show, that in the presented node solution there are not such significant differences between the minimum and maximum time for information exchange during the realization of node application. In addition, it can be seen that with the adopted node architecture the time for data exchange is much shorter than in the previous solution.

Test bench 3 (Fig. 7) used Toke-Passing protocol (GENIUS). All nodes were given unique SBA address (Serial Bus Address) and were authorized to transmit and receive data in the network. All network coprocessors were placed on basic cassettes of particular PLC controllers, and they realized only periodical transactions.

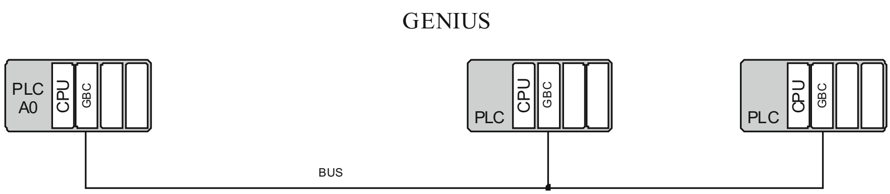

Fig. 7. The schema of test bench for Token-Passing – GENIUS network

Figure 8 presents the results, which indicate that the maximum time of information exchange for determined cycles durations of node application was 46.2 ms, with the duration of node application 15.8 ms. As in case of test bench 1, it was also noticed that there was a significant difference between the minimum and maximum time of information exchange. The longer the duration of application, the greater difference between times is.

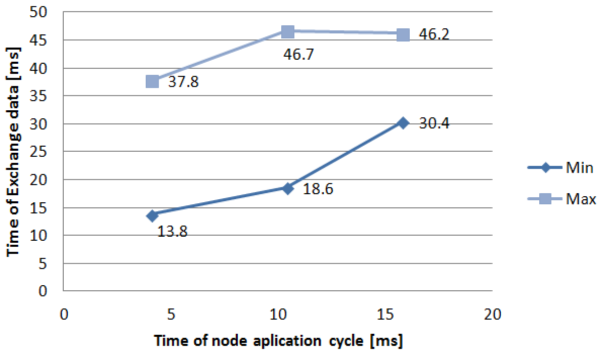

Fig. 8. Test bench 3. Coprocessor for Token-Passing protocol placed in the main cassette of controller. The reading of eight words for automation cycle is as follows: 4.1 ms, 10.4 ms and 15.8 ms.

Test bench 4 for SRTP protocol (Service Request Transfer Protocol) consisted of three PLC equipped with modules of Ethernet communication network connected to switch in star architecture. Each node was given unique IP number. In such configuration each node may be a "client". However, because of the measurement of time for data exchanges in a system with a deterministic access to the link, only the one node can be a "client" station, being authorized to establish communication with the "server" stations.

Figure 9 illustrates the research results, which indicate that the maximum time of information exchange for determined cycles durations of node application was 177.4 ms (Fig. 10), with the duration of node application 66.5 ms. Such as in case of test bench 2, there is a difference between the minimum and maximum time of information exchange.

The empirical research confirmed the thesis, which refers to the necessity of determining the duration of cycles for information exchange in the network supervised by node $A0$ (see also Sect. 3.1). Determining the time T_{TR} is necessary before the network installation is turned on, because only then the algorithm of network controlled by node $A0$ will be work correctly. A very interesting is the fact, that during the research there were the differences between minimum and maximum time of information exchange during the realization of node application. It is also advised to take into account the network architecture in case of choosing a network controlled by node $A0$ (which means position of coprocessor, model of protocol, etc.). One of the factor which increase duration of node

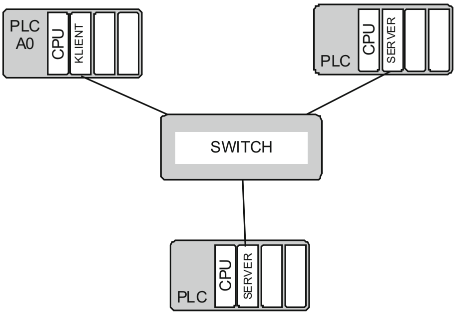

Fig. 9. The schema of test bench for SRTP

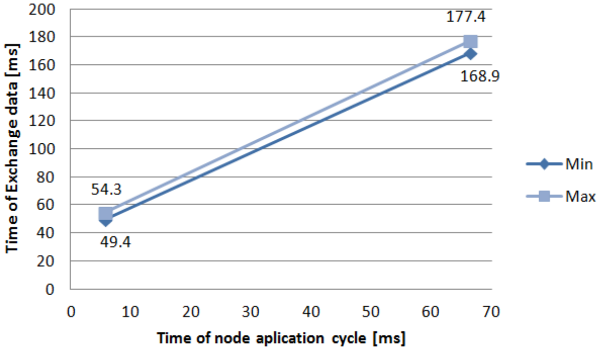

Fig. 10. Test bench 4. Coprocessor of SRTP protocol placed in main cassette of controller. The reading of eight words for automat cycle is as follows: 5.6 ms and 66.5 ms.

cycle is the number of network coprocessors. The bigger number of coprocessors means that the expenditure for system diagnostic are increased, and it influences on the duration of automat cycle. The research results are very important base for further research of shortening data exchange time, and yet they are used to determine the reliability and effectiveness of presented method.

5 Conclusions

This paper presents the concept of creating complex network, which are based on a model of multiprotocol nodes. If the presented idea turn out to be verified by positive results, then it will be the beginning of work on formal structure and definition of a new communication protocol. What is more, it will be possible to use such solution in designing the industrial computing real time systems. It was created on the basis of already tested redundant structures ([22]) with the use of existing MODBUS RTU protocol. The idea of using this protocol aimed at creating a new model of protocol in order to test and determine the practical usefulness of a suggested solution for designing industrial applications.

The already obtained results of practical verification and theoretical background described in Sect. 3, tend to encourage further tests and research. The expected results will be the base of new controller prototype equipped with new integrated software which realizes an algorithm of new protocol. The authors intended to build a new controller which will be able to work with existing hardware solutions. An additional advantage of the proposed solution is the fact that it will be able to work in existing installations, only by installing additional specialized software procedures into the existing system. Currently, the research on developing new software for the presented communication system are being conducted. The aim of this research is to verify the reliability and efficiency of the system. Thanks to the analysis of fundamental time assumptions, it is possible to extend the work of some communication systems with various architecture.

Acknowledgments. This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

References

1. Kwiecień, A.: The improvement of working parameters of the industrial computer networks with cyclic transactions of data exchange by simulation in the physical model. In: The 29th Annual Conference of the IEEE Industrial Electronics Society. Conference Center, Roanoke, Virginia, USA (November 2003)
2. Modbus-IDA. Modbus Application Protocol Specification V1.1b3. (December 2006), `http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf`
3. Modbus-IDA. Modbus Messaging on TCP/IP Implementation Guide V1.0b. (October 2006), `http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf`
4. Miorandi, D., Vitturi, S.: Analysis of master-slave protocols for real-time-industrial communications over IEEE802.11 WLAN. In: 2nd IEEE International Conference on Industrial Informatics 2004, pp. 143–148 (June 2004)
5. Conti, M., Donatiello, L., Furini, M.: Design and Analysis of RT-Ring: a protocol for supporting real-time communications. IEEE Transactions on Industrial Electronics 49(6), 1214–1226 (2002)
6. Raja, P., Ruiz, L., Decotignie, J.D.: On the necessary real-time conditions for the producer-distributor-consumer model. In: IEEE International Workshop on Factory Communication Systems, WFCS 1995, pp. 125–133 (October 1995)
7. Genius® I/O System and Communications. GE Fanuc Automation, doc. no: GEK-90486f1 (November 1994)
8. PACSystemsTM Hot Standby CPU Redundancy, GE Fanuc Intelligent Platforms. doc. no: GFK-2308C (March 2009)
9. Siemens. Simatic Profinet description of the system. Siemens, doc. no: A5E00298288-04, Warsaw (2009)
10. Decotignie, J.-D.: Ethernet-Based Real-Time and Industrial Communications. Proceedings of the IEEE 93(6), 1102–1117 (2005)
11. Felser, M.: Real-Time Ethernet-Industry Prospective. Proceedings of the IEEE 93(6) (June 2005)

12. Jestratjew, A., Kwiecień, A.: Performance of HTTP Protocol in Networked Control Systems. IEEE Transactions on Industrial Informatics 9(1), 271–276 (2013)
13. Jestratjew, A., Kwiecień, A.: Using Cloud Storage in Production Monitoring Systems. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2010. CCIS, vol. 79, pp. 226–235. Springer, Heidelberg (2010)
14. Sidzina, M., Kwiecień, B.: The Algorithms of Transmission Failure Detection in Master-Slave Networks. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2012. CCIS, vol. 291, pp. 289–298. Springer, Heidelberg (2012)
15. Kwiecień, A., Sidzina, M.: Dual Bus as a Method for Data Interchange Transaction Acceleration in Distributed Real Time Systems. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2009. CCIS, vol. 39, pp. 252–263. Springer, Heidelberg (2009)
16. IEC 62439, Committee Draft for Vote (CDV): Industrial communication networks: high availability automation networks. Chap. 6: entitled Parallel Redundancy Protocol (April 2007)
17. IEC 62439, Committee Draft for Vote (CDV): Industrial communication networks: high availability automation networks. Chap. 5, entitled Media Redundancy Protocol based on a ring topology (April 2007)
18. Gaj, P., Jasperneite, J., Felser, M.: Computer Communication Within Industrial Distributed Environment – a Survey. IEEE Transactions on Industrial Informatics 9(1), 182–189 (2013), doi:10.1109/TII.2012.2209668
19. Stój, J.: Real-Time Communication Network Concept Based on Frequency Division Multiplexing. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2012. CCIS, vol. 291, pp. 247–260. Springer, Heidelberg (2012)
20. Gaj, P.: The Concept of a Multi-Network Approach for a Dynamic Distribution of Application Relationships. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 328–337. Springer, Heidelberg (2011)
21. Kwiecień, A., Sidzina, M.: The Method of Reducing the Cycle of Programmable Logic Controller (PLC) Vulnerable "to Avalanche of Events". In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 379–385. Springer, Heidelberg (2011)
22. Kwiecień, A., Stój, J.: The Cost of Redundancy in Distributed Real-Time Systems in Steady State. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2010. CCIS, vol. 79, pp. 106–120. Springer, Heidelberg (2010)