Time Analysis of Data Exchange in Distributed Control Systems Based on Wireless Network Model

B. Twaróg, Z. Gomółka and E. Żesławska

Abstract The paper presents studies concerning time analysis of data exchange in distributed control based on Ethernet wireless network model and relevant Modbus protocol. Presented control system operated in a real-time system that was responsible for continuous-time and periodic event processing. The events appeared at the input of system which had to generate answers (events) at the output. External event response time is closely conditioned by time which means that such systems have to ensure that response time is not exceeded, regardless of the sequence of object events. It is essential to keep the time determinism in the functioning of all real-time, regardless of whether it is a single PLC driver or their group of distributed control system.

Keywords Microprocessor controllers · Distributed wireless networks · Data transmission

1 Introduction

Distributed control systems may exist when all devices are connected to the computer network that enables communication and data transfer between them. Communication in distributed control system is used both for the exchange of essential process data (discrete and analogous) between control devices and actuators and for the exchange of data between control apparatus and measuring-control sta-

B. Twaróg (✉) · Z. Gomółka
Department of Computer Engineering, Faculty of Mathematics and Natural Sciences,
University of Rzeszow, ul. Pigonia 1, 35-959 Rzeszow, Poland
e-mail: btwarog@ur.edu.pl

Z. Gomółka
e-mail: zgomolka@ur.edu.pl

E. Żesławska
Department of Applied Information, Faculty of Applied Informatics,
University of Information Technology and Management in Rzeszow,
ul. Sucharskiego 2, 35-225 Rzeszow, Poland
e-mail: ezeslawska@wsiz.rzeszow.pl

D. Mazur et al. (eds.), *Analysis and Simulation of Electrical and Computer Systems*, Lecture Notes in Electrical Engineering 452,
https://doi.org/10.1007/978-3-319-63949-9_21

tions. Locally placed signal conditioning and data acquisition are the basis of this architecture. In addition, it is necessary to consider the features like synchronized measurement, environmental resistance and the selection of bus and communications protocol with their speed, functionality, capacity and determinism that have to meet particular requirements. Sometimes, it is not required to use intelligent units in distributed control systems. However, this solution is beneficial when some of the logic is directly placed in the measuring nodes. It enables local data analysis and nondependent subsystems control reducing the central processor load. Pre-processing of information reduces the amount of transferred data. Instead of transferring full packages of downloaded samples, only the measurement results are sent, as they are needed for decision-making and saving processes and for further analysis. It is always possible to transfer the whole package of data if necessary. Implementation of the above features suggests the selection of real-time operating system (RTOS) due to its reliability and deterministic nature. By having the execution of commands and measurements in particular time-guaranteed, real-time systems are a suitable solution for the projects that require precise definition of the moment of operation execution and high level of reliability. Synchronization in centralized measurement systems is relatively easy to be executed, as it's particular elements are located in common housing. Synchronization in distributed control system, in turn, involves significant impediments which often result from large distances between devices. This problem may be resolved, for example, by time synchronization which provides all the system elements with an access to a common reference source of information on time. It can be used for generating the event, trigger pulses and clock signals. When the distance between units is significant, it is popular to apply currently available techniques that provide information about the running time, such as GPS, IEEE 1588 or IRIG-B. Time synchronization is used most often in order to avoid problematic physical connections, replacing them with existing network infrastructure or wireless communication [1–4]. The network which links particular parts of distributed control system is an equally evaluative element as the applied measurement equipment. The solution should be selected after the analysis of basic parameters of the system, such as the distance between nodes, network capability, synchronization technique and transmission speed. The applied Modbus is a nondeterministic protocol with basic client-server architecture and TCP/IP layer. It is an ideal solution for the applications that work with a large number of devices from different suppliers. Moreover, it is well adapted to critical data transfer as it caches information and implements the process which controls the correctness of received data between the sender and the recipient, the so-called handshake.

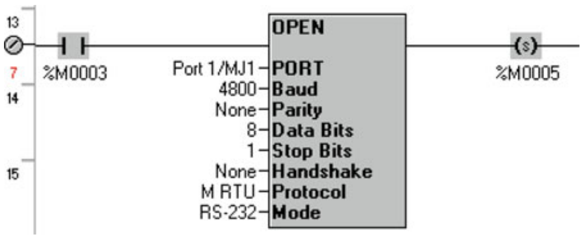
2 System Components and Configuration

The system elements used for the flagship research solution were installed and run in the laboratory of Interdisciplinary Centre for Computational Modelling at the University of Rzeszow.

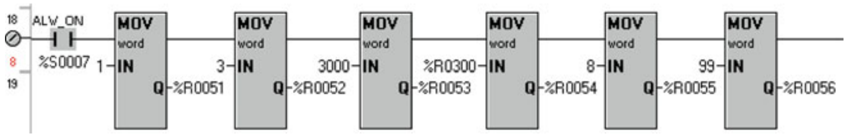
- The control units HEXT251C112 (an integrated whole driver's function and operator touch panel, equipped with 12 discrete inputs (four inputs can be configured so that they can work as high-frequency counters), six relay outputs max. load of 2A and four analogous inputs operating in current standard (0–20, 4–20 mA) or voltage standard (0–10 V), ports RS232 and RS485, USB, Ethernet, CAN).

The central unit has 1 MB of memory for software and powerful processor that executes algorithms with the speed of 0.013 ms/kB. The 3.5″ operator touch screen supports resolution of 320 × 240 pixels and 32,000 colours depth. The applied LED backlight technology ensures high contrast ratios and light display. The screen is equipped with five keys where four of them are freely programmable function keys. 130 MB of memory is dedicated to visualization application which allows creating 1024 operator panels. A driver is programmed from a professional level device C-scape which allows the device configuration, control algorithm creation, operator panel's creation and configuration of communication and extension modules. Creating sophisticated control algorithms is possible thanks to a rich selection of ready function blocks. In order to facilitate driver's configuration, one can use ready wizards which organize the procedure of the ports', protocols' and communication networks' configuration. Additional tools that the module for driver's programming is equipped with enable verification of the correctness of the written application in debug mode, a status display of controller's operation or saving current records in a file. Driver's configuration to operate in Modbus RTU master mode allows cyclic polling of devices operating as slaves and can be executed in two ways: from the level of wizard and a drop down list of available protocols in the menu *Program/Protocol Configuration,* or from the level of ladder language using available function blocks. Ladder program configuration enables the replacement of communication ports' parameters during operation of the driver and closing of communication from the level of control program [5, 6].

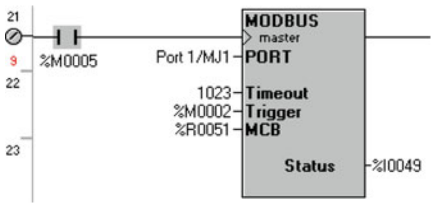
When selecting the block **Open Communication Port** from the menu *Comm operation* it is possible to create a channel for the demanded communication port. Channel's operating parameters can be set by this element. The channel remains open until it is closed by the *Close Port* element, or it is closed when a controller disables the RUN mode.



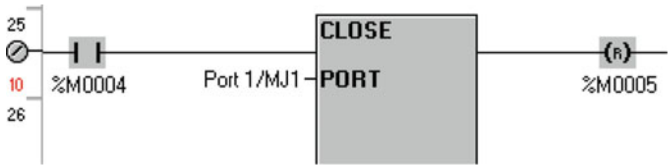
Data block for the Master device is a group of six words, configured for each RTU command. All the *Master Control Blocks* are stored in a programmable controller's memory. A word MCB is created using *Data Move* blocks that consist of six successive records where information about saving and reading of variables is stored.



- Word 1—ID Slave address we want to establish communication
- Word 2—Performed function
- Word 3—Offset Slave
- Word 4—Length of block being read
- Word 5—Type of read variables
- Word 6—Destination address of the read variables



We close communication with the block *Close Communication Port* and close the open channel.



- The network switches Korenix JetNet 2005 (5-port 10/100 TX Compact Fast Ethernet, 3.2 Gbps Switch Fabric with excellent data exchange performance, support 1.5 kV Hi-Pot isolation protection, dual mode for power input, AC18-27V/DC18-32V, fault relay alarm).

Industrial Astraada Net switches are the elements which allow data exchange between the devices that communicate via Ethernet. Unmanageable switches are the compact solutions which have to ensure stable and efficient communication in demanding industrial environment. Additionally, to increase the efficiency of data transfer and using the full capacity of all ports (without their blocking and losing packages), the devices use the filtering—Broadcast Storm, prioritizing of packages —Quality of Service and the Flow Control function [7–9].

- Radiomodems SATELLINE-1870 (band 868–870 MHz, transmission speed 1200–9600 bps, 80 channels, channel spacing 25 kHz, output power 100 mW, coverage up to 2 km, RS232)

Radiomodems replace cable connections (depending on the topography and output power) up to the distance of tens of kilometres. It is possible to increase transmission coverage by using signal transmitters (any modem can be used as a transmitter). Popular applications of radiomodems are telemetry, as well as remote control and monitoring. For the range of frequency 868–870 MHz, wireless devices can operate without special permits and extra costs. This band is divided into the ranges depending on the output power and the operating cycle of the transmitter. For industrial applications the band excerpt from 869,400 to 869, 650 MHz is most interesting as it enables data transfer with max. power up to 500 mW. The applied omnidirectional antennas in the discussed model are characterized by the fact that the waves they create disperse in each direction with the same strength. In the case of small systems and mobile applications, it is advised that quarter-wave and half-wave antennas should be installed vertically, at the height of at least 0.5 m above the surrounding objects.

In order to change the determined options of radiomodems a terminal configuration application SaTerm.exe was used, to allow modelling of characteristic values by selecting a relevant number in the menu. The frequency of 869.6375 MHz has been set for all the radiomodems (Fig. 1).

In the case of transferring data via radiomodem there are some delays caused by a radio link and by the circuits of radiomodem itself. Such delays appear during switching from standby mode to data transfer mode and also when data is received and transmitted. In the data transfer mode, the radiomodem monitors both the radio channel and the serial port. A synchronization signal is sent at the beginning of every transmission. When the signal is detected by radiomodem, the mode switches to data receiving. When radiomodem synchronization signal is being transmitted, radiomodem caches data in memory. Transmission is closed after the cache is emptied and break in data sent by the terminal is detected. If the speed of serial port is the same or lower than the speed of radio link, it is impossible to overfill internal transmit cache (Table 1).

Delay from the time the transmission was completed to the time the reception on the serial link was finished is presented in Fig. 2 (Table 2).

```
------------------------------------------------------------------------------
                    ***** SATEL, SATELLINE - 1870  *****

        SW Version 2.12 HW Version uCTC8L.f0 Serial no. no entry
------------------------------------------------------------------------------
Current settings
----------------
1) Radio frequency     869.6375 MHz, Band 6 (869.4000-869.6500, 500 mW)
2) Radio settings      TX power 100 mW, RSSI-threshold -113 dBm, TX delay 0 ms
3) Addressing          RX address OFF/0000/0000, TX address OFF/0000/0000
                       Protocol 1: OFF, Start char. 00, Offset 0, 1 BYTE
                       Protocol 2: OFF, Start char. 00, Offset 0, 1 BYTE
                       Hop-count 15, TX address bitmapping OFF
                       RX address masking OFF, Repeater address bypassing OFF
                       Subnet mask 1 0000, Subnet mask 2 0000
4) Serial port         9600 bit/s, 8 bit data, Even parity, 1 stop bit
                       Pause length 3
5) Handshaking         CTS Clear to send, RTS Ignored
6) Additional setup    Repeater OFF, SL OFF, Block-CRC OFF, TX priority ON,
                       Power save OFF, Frame limit OFF, SL Extended OFF
                       Full frame CRC - OFF
7) Tests               Test mode Inactive
8) Restore factory settings
```

Fig. 1 Values of configuration options for the modem SATEL1870

Table 1 Time of delays

Functions	Delay (ms)	Comment
Time of moving from standby time to activation state (switching controlled by DTR line)	700	Activation state, radiomodem is ready to receive data
Time of moving from deactivation state (economical) to activation state (switching controlled by SHDN line)	120	
Time of moving from activation time to economical mode (switching controlled by SHDN line)	1000	
Time of delay of the RS serial connector	0	
Delay between characters	Max. 6–7 characters	Speed of data transfer using a communication link is 9600 bps or less

- The Astraada Drive GD10 inverter (input voltage 1 × 230VAC ± 15% or 3 × 400VAC ± 15%, port RS485, Modbus RTU communication, LED panel, integrated brake module up to 15 kW, integrated PID controller).

Astraada Drive GD10 is a series of frequency converters that have a separate LED control panel with a potentiometer (can be at a distance of 20 m), enabling drive system control, with the power between 0.2 and 2.2 kW for the single-phase power supply and between 0.75 and 2.2 kW for the three-phase power supply [10, 11].

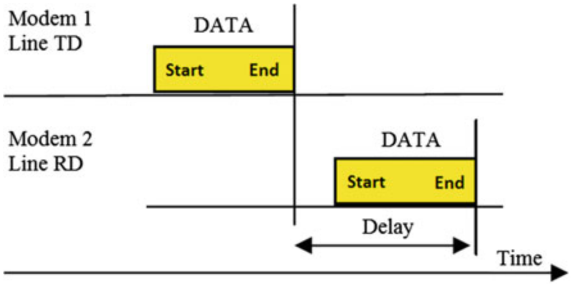


Fig. 2 Delay in data transfer via radiomodem

Table 2 Delay [in (MS)] of transmission for the radio channel 25 kHz numbers of the transferred bytes

Bps	1	10	100	500	1000
1200	70	70	70	70	70
4800	70	70	70	70	70
9600	70	70	86	120	160
19200	70	75	126	370	670

3 Master Communication—Modbus Protocol

Master/Slave networks, which allow direct information exchange between Master and Slave stations, have two basic types of triggered exchanges:

- *request—response* information exchange with a selected Slave subscriber station; in this case, if the exchange includes command, the response must follow,
- *publishing—no response* Master unit sends a publishing frame to all subscribers.

Realization of all configuration exchanges occurs in Master station, which is a significant feature of the network with Master/Slave access. The so-called exchange scenario, created in the station, is then realized by the coprocessor. It can be assumed that the duration of data exchange is calculated in the dependence as follows:

$$D_{E_{qr}} = D_{NM} + D_{AM} + 2(D_{PF} + D_{TF} + D_{DF} + D_{AF}) + D_{AS} \tag{1}$$

where:

- $D_{E_{qr}}$ duration of request–response exchange execution
- D_{NM} time measured from the moment of reporting the exchange during cycle phase to the moment of communication phase start
- D_{AS} duration of Slave station cycle
- D_{AM} time from the moment of decoding of information by Master station processor to the moment of receiving and using it in the Master station application
- D_{PF} duration of frame preparation

- D_{TF} duration of frame transmission
- D_{DF} duration of frame detection
- D_{AF} duration of frame analysis

Modbus/TCP protocol is based on TCP protocol stacks and Modbus/RTU protocol frame is included in TCP/IP frame. This protocol works in the client-server architecture where the Client station (Master from the serial networks) is the one that initializes communication. The implementation of Modbus/TCP protocol discussed here includes split operations of recording and reading since the moment of starting the port. Each open port command is triggered from the application level and includes information about the device the transaction will be executed with. All messages are triggered from the level of PLC driver's application. In order to connect with another device it is necessary to open the port for communication, trigger a read or record functions, and when the communication is over, use the close connection function.

$$D_{\mathrm{EM}} = \sum_{i}^{n} D_{\mathrm{MOP}_i} + \sum_{j}^{m} D_{\mathrm{MWR}_j} + \sum_{k}^{w} D_{\mathrm{MCP}_k} \tag{2}$$

where:

- D_{EM} duration of exchanges in Modbus/TCP protocol over a certain time period
- D_{MOP} duration of open port operation
- D_{MWR} duration of exchange, record or read
- D_{MCP} duration of close port operation
- n number of open port operations
- m number of record and read operations
- w number of close port operations

The final time evaluation should also include delays that result from the application of radiomodems as a communication link.

4 Time Dependencies in the Process of Data Exchange

The presented measurements were executed with the use of devices equipped exclusively with communication interface after RS-485 in Modbus/RTU protocol, and further measurements were done using devices that enable communication in the Ethernet standard with Modbus/TCP protocol via radiomodems. In the experimental verification of the duration of data exchange for distributed communication nodes, there was use of the developed software which initiated communication between the system's nodes and then realized the measurement of data exchange duration. This software uses embedded time read functions from PLC driver. The time is measured from the moment of starting central supply unit to the moment of measurement. Starting and ending the measurement of the duration of a particular

Table 3 Duration of the request–response exchange for Modbus RTU protocol

Number of read registers	Initial register	Min. time (ms)	Max. time (ms)	Number of tests
1	100	33	51	200
32	100	67	83	200
64	100	105	121	200

Table 4 Duration of the request–response exchange for Modbus/TCP protocol via SATEL radiomodem link

Number of read registers	Min. time (ms)	Max. time (ms)	Number of tests
1	225	272	300
32	227	268	300
64	230	271	300

exchange is a substantial problem. In the study, Master station individually queried Slave station—triggered by bit. The read number of registers is 1, 32, 64 (Table 3).

Basic research area for Modbus/TCP comprised two stations. Each node had a unique IP number. With regard to the measurement of data exchange duration in the system of a deterministic access to the link, only one node served as client station and sent data only through one communication channel. The read number of registers is 1, 32, 64 (Table 4).

The executed measurements for Modbus/TCP protocol show that maximum time of sending a small data package lasts longer than in Modbus/RTU network. However, increasing data package fails to cause a significant increase in transaction duration. These results may suggest that this standard suits very well when larger amounts of information are sent.

5 Conclusions

The first part of the studies involved the analysis of Modbus/RTU protocol where the time between reporting the demand of executing the exchange and its termination was measured. The exchange is defined as an order to read or record a certain amount of information given in 16-bit or 32-bit words. The read information block had a size of 1–64 words for Modbus/RTU protocol and 128 words for Modbus/TCP protocol. The obtained minimum and maximum transaction durations are the results of the measurement. For the real-time systems, the worst case seems to be the most significant. It can be assumed that the worst case of read or record for the studied area would be the situation where there would be no common data area and the communication system would have to read every register through separate transactions. The obtained results show that the selection of devices working in particular network loops is a very important element of the system. Transaction scenario has the greatest impact on the duration of network cycle and the system reaction time depends on it. That is why it is crucial to separate groups of devices during the designing of the system. Application of the Ethernet technology in the low-level communication gave an opportunity to send larger blocks of information at a minimum time when compared to a smaller amount of data. The executed measurements show that the system which sends information blocks larger than 32 words should use Modbus/TCP protocol. This technology is faster especially in the case of monitoring the status of multiple parameters of the control device. When small data package exchange is concerned, communication is faster with the use of the RS-485 interface. Delays connected with the radiomodems had a significant influence on the times of Modbus/TCP transmissions. It was concluded that it is better to employ communication in Modbus/RTU when there is need to manage controlling devices or security devices in Modbus/RTU, especially when the size of data block does not exceed a few registers.

Acknowledgements This work has been supported by the National Science Centre, Poland, based on decision DEC-2012/05/B/ST7/01183.

References

1. Bolanowski, M., Paszkiewicz, A.: The use of statistical signatures to detect anomalies in computer network, Analysis and simulation of electrical and computer systems. Lecture Notes in Electrical Engineering, vol. 324, pp. 251–260. Springer (2015)
2. Grabowski, F., Paszkiewicz, A., Bolanowski, M.: Wireless networks environment and complex networks, analysis and simulation of electrical and computer systems. Lecture Notes in Electrical Engineering, vol. 324, pp. 261–270. Springer (2015)
3. Flaga, S.: Programowanie sterowników PLC w języku drabinkowym. BTC, Warszawa (2014)
4. Mackay, S., Wright, E., Reynders, D., Park, J.: Practical Industrial Data Networks: Design, Installation and Troubleshooting. Elsevier, Great Britain (2004)
5. Kasprzyk, J.: Programowanie sterowników przemysłowych. WNT, Warszawa (2012)
6. Sacha, K.: Systemy czasu rzeczywistego. Politechnika Warszawska, Warszawa (2006)
7. Rinaldi, J.: Modbus: The Everyman's Guide to Modbus. John S Rinaldi, Mechanicville (2015)
8. Gaj, P.: Zastosowanie protokołu TCP/IP do transmisji informacji dla potrzeb przemysłowych systemów kontrolno-nadzorczych. Rozprawa doktorska, Gliwice (2003)
9. Sidzina, M., Kwiecień, A.: Systemy czasu rzeczywistego: projektowanie i aplikacje/ praca zbiorowa pod red. Piotra Gaja, WKiŁ, Warszawa (2005)
10. Cupek, R., Gaj, P., Kwiecień, A.: Współczesne problemy systemów czasu rzeczywistego. Praca zbiorowa, WNT, Warszawa (2004)
11. Reynders, D., Wright, E.: Practical TCP/IP and Ethernet Networking for Industry. Elsevier, Great Britain (2003)