Journal of KONBiN 49(2019)
e-ISSN 2083-4608
DOI 10.2478/jok-2019-0020

Marcin BEDNAREK[1], Tadeusz DĄBROWSKI[2], Wiktor OLCHOWIK[2]
[1] *Rzeszow University of Technology (Politechnika Rzeszowska)*
[2] *Military University of Technology (Wojskowa Akademia Techniczna)*

SELECTED PRACTICAL ASPECTS OF COMMUNICATION DIAGNOSIS IN THE INDUSTRIAL NETWORK

Wybrane praktyczne aspekty diagnozowania komunikacji w sieci przemysłowej

***Abstract**: Industrial networks combine elements of distributed control systems: process stations, operator stations and engineering stations. DCS stations often communicate using a dedicated, closed communication protocol. The industrial networks can be also used to manage communication between the stations of various systems, separate in terms of configuration. The process station communicates here with an external operator station, that is the SCADA system. For this purpose, the process stations and the SCADA systems can also communicate according to standard communication protocols, e.g. Modbus TCP. The paper examines the selected variants of diagnosing the communication status between the process station and the external operator station conducted according to the Modbus TCP protocol. The practical methods of finding the communication system unfitness causes are discussed.*
Keywords: communication diagnosis, data transmission system, industrial network

***Streszczenie:** Sieci przemysłowe łączą elementy rozproszonych systemów sterowania: stacje procesowe, stacje operatorskie oraz stacje inżynierskie. Stacje komunikują się często za pomocą dedykowanego protokołu komunikacyjnego. Za pomocą sieci przemysłowych może być także prowadzona komunikacja pomiędzy stacjami odrębnych systemów. Stacja procesowa komunikuje się tu z zewnętrzną stacją operatorską, czyli systemem SCADA. Stacje procesowe oraz systemy SCADA mogą komunikować się wg standardowych protokołów komunikacyjnych, np. Modbus TCP. W referacie rozpatruje się wybrane warianty diagnozowania stanu komunikacji pomiędzy stacją procesową i zewnętrzną stacją operatorską prowadzonej wg protokołu Modbus TCP. Dyskutowane są praktyczne sposoby znalezienia przyczyn niezdatności systemu komunikacji.*
Słowa kluczowe: diagnozowanie komunikacji, system transmisji danych, sieć przemysłowa

1. Introduction

Industrial networks combine elements of distributed control systems (*DCS* – fig. 1). These elements are process stations, operator stations, engineering stations and other elements which enable the exchange of data between industrial systems [1]. The process stations are industrial controllers which manage the control of processes, the operator stations are used for visualisation and operator interactions, and the engineering stations are designed for configuration, programming and activation of control and visualisation programmes.

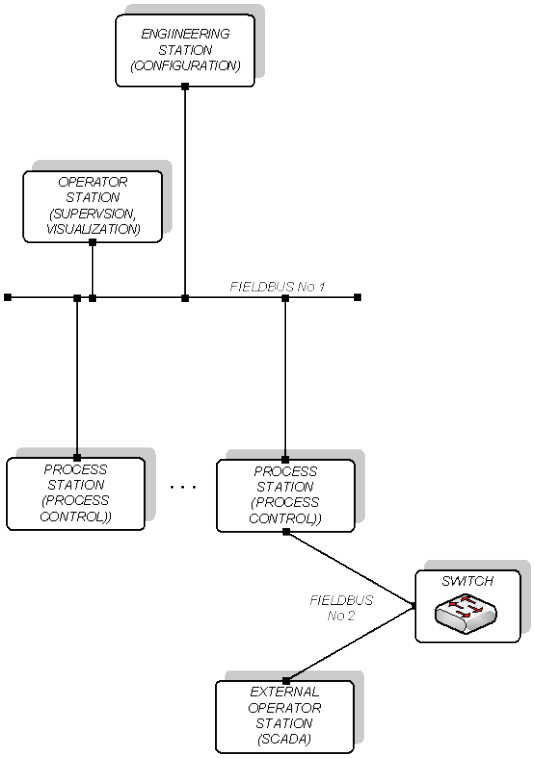


Fig. 1. Studied system

With the use of the industrial network, the process variable values and other data exchanged between the stations are transferred. The DCS stations often communicate using a dedicated, closed communication protocol [2]. It should be noted that the industrial networks can combine stations of not only one control system (fig. 1 – industrial network 1). They can be also used to manage communication between the stations of various control systems, separate in terms

of configuration (fig. 1 – industrial network 2). The process station communicates here with an external operator station, that is the SCADA (*Supervisory Control And Data Acquisition*) system performing the data visualisation and acquisition [3]. For this purpose, the process stations and the SCADA systems can also communicate according to standard communication protocols, e.g. Modbus TCP [4]. This paper examines the selected variants of diagnosing the communication status between the process station and the external operator station conducted according to the Modbus TCP protocol. The practical methods to help in potential finding of the communication system unfitness cause with the use of the following are discussed:

- communication status window of the communication driver,
- Quality field of process variables,
- protocol analyser software,
- system variables of the FBD language communication blocks.

2. Communication between the process station and the external operator station

As mentioned in the introduction, the article focuses on practical ways of diagnosing the status of communication conducted in accordance with the Modbus TCP protocol [4]. Before they are discussed, it is worth briefly characterising the way of the controller (process station) communication with the external operator station. This protocol is directly associated with its predecessor, which is Modbus RTU [5] mainly using the RS-485 differential serial standard for operation. The idea behind the Modbus protocol operation is communication on the basis of polling slave devices by master ones. In Modbus RTU, there are two types of devices – *master* and *slave* ones. The initiator of the message exchanges is always *master, slave* has no right to independently "speak". Communication is carried out according to a specific scenario and a question-answer scheme. Each slave device has a strictly defined address (number) and responds only to messages addressed to it. Similarly, in Modbus TCP, with the only difference that a completely different transmission medium, which is a computer network and a *TCP/IP* protocol, is applied for communication. Due to a change in the transmission medium and communication protocol in relation to the RTU version, device addressing – IP addressing standard for computer networks is used. In addition, within the framework of one IP address, several *slave* devices can be connected, therefore, space was left in the transmitted frame to enter the *slave* device number. A change in the medium and transport protocol also affected the nomenclature modification. The role of *master* in Modbus TCP is played by the *client* that polls the *server*

(instead of *slave*). The question may be in the nature of requesting certain values or setting the values of process variables. Usually, the SCADA system with an appropriate communication driver (here called the external operator station) is a client, and the server – that is the industrial controller (process station). There are also visualisation packages that can act as master and slave devices [6]. Then, the roles may be reversed: the operator station will become the server, and the process station will be the client.

The further practical ways of diagnosing communication conducted by Modbus TCP are useful at the stage of activating the communication between the stations. Especially, when the task of the engineer activating the communication system is the configuration of both communicating parties. The possible lack of communication between the stations or the occurring transmission errors are the prerequisite for the use of one of the presented methods. The symptom of the mentioned situations related to the transmission system unfitness is most often the lack of appropriate indications of the alphanumeric graphic displays or interruption of the animation liquidity of dynamic elements changing the location (depending on the transmitted variable values) on the synoptic display (sliders, bargraphs, filled areas, etc.).

3. Use of the protocol analysers

The use of a sniffer [7] for the communication diagnosis seems to be the easiest method. With its help, it is possible to observe the messages from both communicating parties. Nothing simpler cannot be apparently imagined. An additional diagnosing station is connected to the system and the incoming messages are observed by activating the software for capturing and analysing the network packages in it (fig. 2). But even here, there are some difficulties.

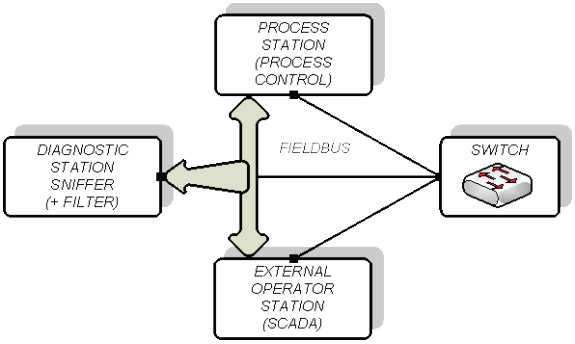


Fig. 2. Analyser use – option 1

Firstly, in order to make the analyser intercept all messages, the additional diagnosing station should receive all packages directed to each other by the process and operator stations. While several years ago – the devices connecting hosts in the computer network included hubs, which transmitted the data received from one port (input) to all other – it was a simple task because by connecting the diagnosing station to the next hub input, it was possible to observe complete network traffic without problems; the current separation devices include switches, or the next generation hubs, which send messages only between the communicating ports on the basis of the analysis of the passing network traffic. Therefore, it is important to put some effort to "cheat" the ARP table [8] of the device in order to impersonate the selected device and redirect the network traffic to the sniffer port. The more and more perfect safeguards applied in the devices do not allow to do it in a simple way or block this type of activities completely. Therefore, a compromise solution is to activate the analysis software, not as a separate position, but placing it in the operator station (fig. 3). Such a solution can be applied, when we do not deal with a built-in system, and the operator station is e.g. an industrial computer. The activation of the interception of messages is then associated with a slight modification of the ARP entries in the operator system of the operator station.

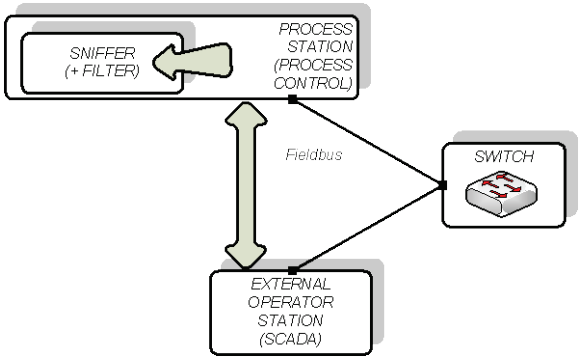


Fig. 3. Analyser use – option 2

Secondly, even if the diagnosis copes with the interception of messages of the TCP/IP protocol, it must somehow extract the ones that are directly related to the transmission of the Modbus TCP commands and responses. Therefore, a very useful tool in this case includes appropriate filters that allow to display only the Modbus TCP messages. For the popular Wireshark sniffer, they can be found in [9].

The third and final difficulty in using the discussed method of diagnosing the Modbus TCP communication are limited cognitive abilities of an engineer-operator. Despite having the log (recording of the intercepted messages with a time

stamp) for reliable diagnostic inference based on the communication recording, knowledge on the importance of subsequent fields in the Modbus TCP messages and permissible (or expected) values of these fields is required. Therefore, this method is recommended for advanced, experienced engineers activating or looking for causes of the emerging communication unfitness. It provides as much data on the sent messages as possible, but it requires the operator to significant erudition.

4. Communication status window of the communication driver

The use of the communication status window of the communication driver is one of the easiest ways providing the answer to the question concerning the communication fitness. The diagnosis is carried out from the operator station. The visualisation software of the station to communicate with the devices (controller – operator station) needs the appropriate software communication interface – communication driver. It is not required to equip the station with hardware interface other than a network card of the popular standard of local Ethernet computer networks [10]. The communication is carried out in accordance with the diagram of fig. 4.

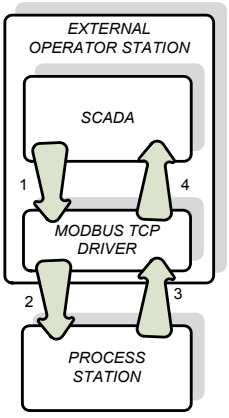


Fig. 4. Communication system elements

By calling up the communication status window (fig. 5), is possible to read the basic parameters of the Modbus TCP communication:

- delay of messages,
- received data quality,
- transaction type (reading/recording) – for each type of variables,

– type of the used registry (needed, e.g. to determine the function number for reading/recording of the desired variable value in the process station),
– registry addresses in the process station.

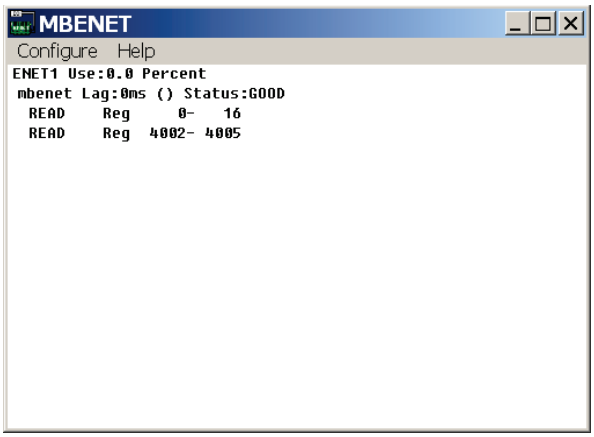


Fig. 5. The communication status window [11]

In case of the example taken from the original application made in the Intouch system with the use of the *mbenet* driver, presented in fig. 5 [11], in case of temporary unfitness of the communication system, the window status elements are displayed in red informing about the emerging transmission errors.

The communication diagnosis method presented here on the basis of the status window allows to obtain only rough data on the selected communication parameters. More data can be extracted by adding the *Quality* variable field test [12].

5. *Quality* variable field

The use of this method allows to obtain much broader information on the communication system fitness. In part – among the available SCADA packages – additional fields of the I/O (input/output – type of variables used in communication) variables are applied. The standard reference in the visualisation application to the variable name results in the performance of operation on the value assigned to the variable [11]. It is possible to refer to the variable with the use of the syntax: *variable_name.field*. For the variables of the I/O type, several fields, on the basis of which it is possible to draw a conclusion about the operating status of the communication system, are available. The *Quality* field values usually comply with the OPC standard [13].

Table 1

Possible meanings of the Quality field values, study based on [12]

No.	Quality status	Quality substatus
1.	Good	Clamped High
		Clamped Low
		Cannot Convert
		Cannot Access
		Communications Failed
2.	Uncertain	Non-specific
		Last Usable Value
		Sensor Not Accurate
3.	Bad	Non-specific
		Configuration Error
		Not Connected
		Device Failure
		Sensor Failure
		Last Known (Received) Value
		Com Failure
		Out of Service

In table 1, a simplified interpretation of the *Quality* field values was provided for better readability. Of course, some values represented by 8 last, least significant bits correspond to each of the mentioned tables of operating situations determined by the value of status and substatus fields. The complete information on the field values can be found, among others, in [12]. Information on the connection quality with the data read from the communication driver status window provides a much broader view of the data transmission system state.

6. System variables of the FBD language communication blocks

The deliberations presented so far on the communication system diagnosis process concern – except for the analyser option as a separate diagnostic stand – operation carried out from the operator station. So, what diagnosing options are made available by the process station?

At the level of a task of the process station programmed in the language of function blocks [1], it is possible to use the block providing some statistical data (much simplified) on communication, i.e. the number (fig. 6) of:

- sent messages (if the process station is a client),
- received messages,
- messages, at the delivery of which the permissible response time (timeout) was exceeded,
- message transmission repetitions (repetitions),
- number of messages received with an incorrect checksum value (CRC).

The value of the provided parameters is set to zero at the communication task restart.

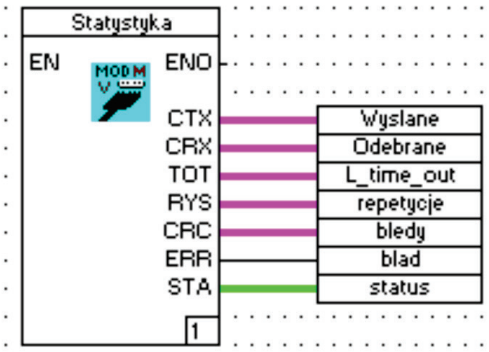


Fig. 6. Reading of the simplified communication statistics

7. Conclusion

The paper presents a discussion on the probability of using several various methods for obtaining diagnostic information on communication carried out in accordance with the Modbus TCP protocol. The practical communication diagnosis methods with the use of the external diagnosing station (type I), from the operator station (type II) and the process station (type III) were presented on the basis of (type indicating the place of using the procedure was provided in brackets):

- logs obtained with the use of the protocol analyser software (I or II),
- data presented in the communication status window of the communication driver (II),
- *Quality* field values of the process variables (II),
- values of the system variables of the communication blocks (III).

It is worth noting that the main task of the communication diagnosis process is to obtain the complete information on the situation existing in the system – with the use, among others, of the discussed diagnosing methods. However, in order to automate the diagnosis process, it should be remembered that the primary task of the operator station is visualisation and the potential operator impact on the running process, while the process station task is to control the process. Therefore, the activities related to obtaining the necessary diagnostic information should be performed in such a way as not to hinder the implementation of standard functions of the station.

The ability to automate the diagnosis process with the use of the protocol analyser is problematic. It is related to the necessity to involve substantial computing power in the process of analysing the recorded logs, and also to the fact that the diagnostic inference process based on the network traffic analysis requires a thorough knowledge of the diagnostician. Therefore, this process cannot take place without the current control and the operator's intervention.

From the operator station, in order to automate the diagnosing process, it is possible to use, e.g. window, event and application scripts, that is programmes written in the internal script language of a given SCADA package, performed during calling up and closing, etc. respectively of: windows, certain activities or visualisation applications. The methods for programming these scripts provide limited possibilities of building the advanced programming constructions, but they can successfully refer to the variables and their fields. Due to the above-mentioned restrictions, the most convenient and useful way of obtaining diagnostic data is, therefore, the use of information contained in the *Quality* field of variables. It provides the possibility of obtaining fairly detailed diagnostic information on the quality of communication, at the same time, with the moderate involvement of computing power of the operator station.

8. References

1. Bednarek M.: Wizualizacja procesów. Laboratorium [Visualisation of processes. Laboratory], Oficyna Wydawnicza Politechniki Rzeszowskiej [Publishing House of Rzeszów University of Technology], Rzeszów 2004.
2. Dokumentacja elektroniczna systemu Freelance 2000 [Electronic documentation of the Freelance 2000 system].
3. Informator techniczny Wonderware: Komunikacja OPC w pakiecie Wonderware FactorySuite 2000 [Wonderware technical information guide: OPC communication in the Wonderware FactorySuite 2000 package], No. 24 30-10-2001.

4. MODBUS Messaging on TCP/IP Implementation Guide V1.0, October 24, 2006 http://www.Modbus-IDA.org.
5. MODBUS over Serial Line Specification and Implementation Guide, Modbus.org, 2005, [http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf].
6. https://digitalsupport.ge.com/en_US/Documentation/iFIX-58-Ebooks-in-PDF-Format.
7. https://www.wireshark.org.
8. RFC 826, STD 37: An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, November 1982 [https://www.rfc-editor.org/info/rfc826].
9. https://www.wireshark.org/docs/dfref/m/mbtcp.html.
10. Standard IEEE 802.3 [https://standards.ieee.org/].
11. Dokumentacja elektroniczna pakietu Wonderware Intouch [Electronic documentation of the Wonderware Intouch package].
12. Wonderware FactorySuite InTouch, Opis funkcji, pól i zmiennych systemowych [Description of functions, fields and system variables], Invensys Systems, Inc., 2005. [Wonderware FactorySuite InTouch Reference Guide, September 2002, Invensys Systems, Inc.]
13. Mahnke W., Leitner S.H., Damm M.: OPC Unified Architecture, Springer Science & Business Media, 2009.

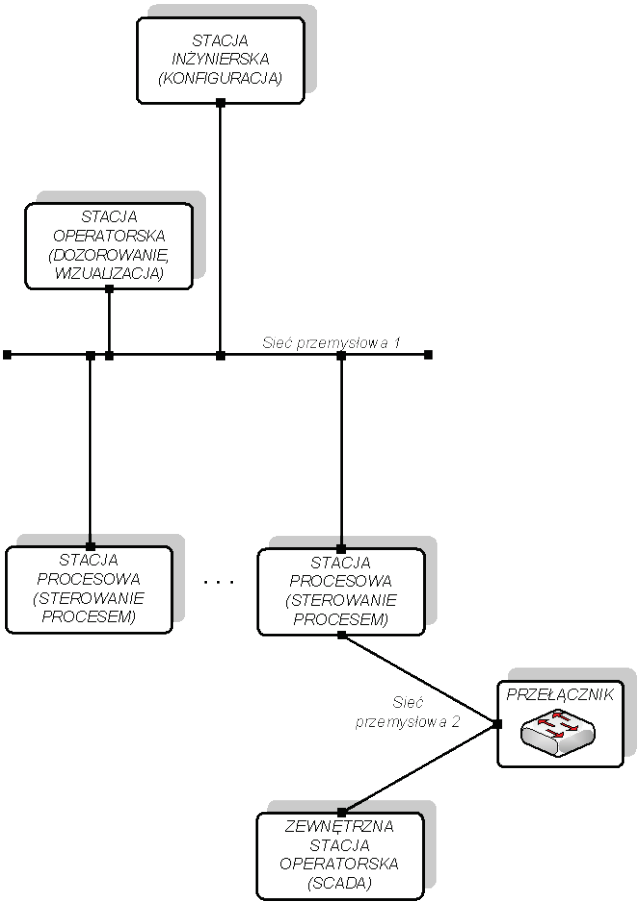
WYBRANE PRAKTYCZNE ASPEKTY DIAGNOZOWANIA KOMUNIKACJI W SIECI PRZEMYSŁOWEJ

1. Wprowadzenie

Sieci przemysłowe łączą ze sobą elementy rozproszonych systemów sterowania (DCS – ang. *Distributed Control System* - rys. 1). Elementami tymi są stacje procesowe, stacje operatorskie, stacje inżynierskie oraz pozostałe elementy pozwalające na wymianę danych pomiędzy systemami przemysłowymi [1]. Stacje procesowe są sterownikami przemysłowymi prowadzącymi sterowanie procesami, stacje operatorskie służą do wizualizacji i oddziaływań operatorskich, stacje inżynierskie przeznaczone są do konfiguracji, programowania i uruchamiania programów sterowania oraz wizualizacji.

Za pomocą sieci przemysłowej przesyłane są wartości zmiennych proceso-wych oraz inne dane wymieniane pomiędzy stacjami. Stacje DCS komunikują się często za pomocą dedykowanego, zamkniętego protokołu komunikacyjnego [2]. Należy także zauważyć, iż sieci przemysłowe mogą łączyć stacje nie tylko jednego systemu sterowania (rys. 1 – sieć przemysłowa 1). Za ich pomocą może być także prowadzona komunikacja pomiędzy stacjami różnych, odrębnych pod kątem konfiguracyjnym systemów (rys. 1 – sieć przemysłowa 2). Stacja procesowa komunikuje się tu z zewnętrzną stacją operatorską, czyli systemem SCADA (ang. Supervisory Control And Data Acquisition) prowadzącym wizualizację i akwizycję danych [3]. Do tego celu stacje procesowe oraz systemy SCADA mogą komunikować się wg standardowych protokołów komunikacyjnych, np. Modbus TCP [4]. W niniejszym opracowaniu rozpatruje się wybrane warianty diagnozowania stanu komunikacji pomiędzy stacją procesową i zewnętrzną stacją operatorską prowadzonej wg protokołu Modbus TCP. Dyskutowane są praktyczne sposoby pomagające w ewentualnym znalezieniu przyczyny niezdatności systemu komunikacji z wykorzystaniem:

- okna statusu komunikacji drivera komunikacyjnego,
- pola *Quality* zmiennych procesowych,
- oprogramowania analizatora protokołów,
- zmiennych systemowych bloków komunikacji języka FBD.



Rys. 1. Rozpatrywany system

2. Komunikacja pomiędzy stacją procesową i zewnętrzną stacją operatorską

Jak wspomniano we wstępie, autorzy skupili się na praktycznych sposobach służących diagnozowaniu stanu komunikacji prowadzonej wg protokołu Modbus TCP [4]. Zanim zostaną one omówione, warto krótko scharakteryzować sposób komunikacji sterownika (stacji procesowej) z zewnętrzną stacją operatorską. Protokół ten bezpośrednio związany jest ze swoim protoplastą, jakim jest Modbus RTU [5], wykorzystujący do działania głównie różnicowy standard szeregowy RS-485. Ideą działania protokołu Modbus jest komunikacja na zasadzie

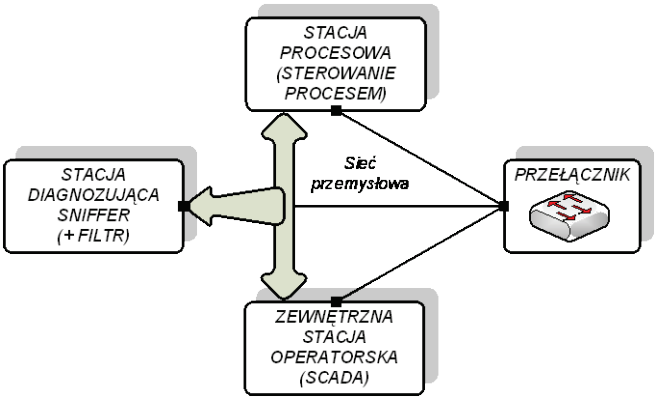
odpytywania urządzeń podrzędnych przez nadrzędne. W Modbus RTU występują dwa typy urządzeń – *master* i *slave*. Inicjatorem wymian komunikatów jest zawsze *master*, *slave* nie ma prawa samodzielnie się „odzywać". Komunikacja realizowana jest wg określonego scenariusza i schematu pytanie-odpowiedź. Każde z urządzeń podrzędnych ma ściśle określony adres (numer) i odpowiada tylko na adresowane do niego komunikaty. Podobnie w Modbus TCP, z tą tylko różnicą, że do komunikacji stosuje się zupełnie inne medium transmisyjne, jakim jest sieć komputerowa oraz protokół *TCP/IP*. Ze względu na zmianę medium transmisyjnego i protokołu komunikacji w stosunku do wersji RTU, wykorzystuje się standardową dla sieci komputerowych adresację urządzeń – adresację IP. Dodatkowo w ramach jednego adresu IP może być podłączonych kilka urządzeń *slave*, dlatego w transmitowanej ramce pozostawiono miejsce na wprowadzenie numeru urządzenia *slave*. Zmiana medium i protokołu transportowego wpłynęła także na modyfikację nazewnictwa. Rolę *mastera* pełni w Modbus TCP *klient*, który odpytuje *serwer* (zamiast *slave*). Pytanie może mieć charakter żądania pewnych wartości lub ustawienia wartości zmiennych procesowych. Przeważnie klientem jest system SCADA z odpowiednim driverem komunikacyjnym (tutaj nazywany zewnętrzną stacją operatorską), a serwerem – sterownik przemysłowy (stacja procesowa). Są też pakiety wizualizacji mogące pełnić funkcję urządzenia nadrzędnego i podrzędnego [6]. Wtedy role mogą zostać odwrócone: serwerem stanie się stacja operatorska, klientem – stacja procesowa.

Przedstawione dalej praktyczne sposoby diagnozowania komunikacji prowadzonej wg Modbus TCP są przydatne na etapie uruchamiania komunikacji pomiędzy stacjami, szczególnie wtedy gdy zadaniem inżyniera uruchamiającego system komunikacji jest konfiguracja obydwu komunikujących się stron. Ewentualny brak komunikacji pomiędzy stacjami lub pojawiające się błędy transmisji są właśnie przesłanką do zastosowania jednej z przedstawionych metod. Objawem wymienionych sytuacji dotyczących niezdatności systemu transmisji jest najczęściej brak odpowiednich wskazań na graficznych wyświetlaczach alfa-numerycznych lub też zaburzenie płynności animacji dynamicznych elementów zmieniających położenie (w zależności od wartości przesyłanych zmiennych) na ekranie synoptycznym (suwaki, bargraphy, obszary wypełniane itp.).

3. Wykorzystanie analizatora protokołów

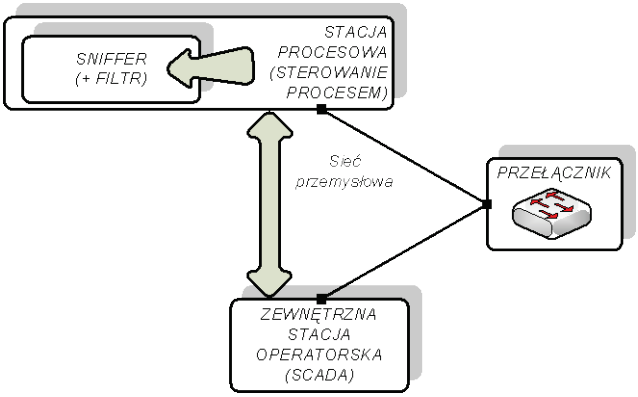
Wykorzystanie sniffera [7] do diagnozowania komunikacji wydaje się najprostszym sposobem. Za jego pomocą można obserwować komunikaty pochodzące od obydwu komunikujących się stron. Pozornie niczego prostszego nie

można sobie wyobrazić. Dołącza się dodatkową stację diagnozującą do systemu i, uruchamiając na niej oprogramowanie do przechwytywania i analizy pakietów sieciowych, obserwuje napływające komunikaty (rys. 2). Jednak i tu pojawiają się pewne trudności.



Rys. 2. Wykorzystanie analizatora – wariant 1

Po pierwsze, aby analizator właściwie przechwycił wszystkie komunikaty, dodatkowa stacja diagnozująca powinna otrzymywać wszystkie pakiety wzajemnie kierowane do siebie przez stację procesową i operatorską. O ile kilkanaście lat temu – kiedy urządzeniami łączącymi hosty w sieci komputerowej były huby, które wysyłały dane otrzymane z jednego portu (wejścia) na wszystkie pozostałe – było to zadanie proste, ponieważ podłączając stację diagnozującą do kolejnego wejścia huba, można było bez problemów obserwować pełen ruch sieciowy, o tyle współcześnie urządzeniami rozdzielającymi są przełączniki (switche), czyli huby kolejnych generacji, które na podstawie analizy przechodzącego ruchu sieciowego wysyłają komunikaty tylko pomiędzy komunikującymi się portami. Trzeba więc poświęcić nieco wysiłku, aby „oszukać" tablicę ARP [8] urządzenia, aby podszyć się pod wybrane urządzenie i przekierować ruch sieciowy na port sniffera. Coraz doskonalsze zabezpieczenia stosowane w urządzeniach nie pozwalają tego zrobić w prosty sposób lub blokują tego typu działania całkowicie. Dlatego pewnym kompromisowym rozwiązaniem jest uruchomienie oprogramowania analizującego, nie jako oddzielnego stanowiska, lecz umieszczenie go w stacji operatorskiej (rys. 3). Takie rozwiązanie może być zastosowane wówczas, gdy nie mamy do czynienia z systemem wbudowanym i stacją operatorską jest np. komputer przemysłowy. Uruchomienie podsłuchu komunikatów wiąże się wtedy z niewielką modyfikacją wpisów ARP w systemie operacyjnym stacji operatorskiej.



Rys. 3. Wykorzystanie analizatora – wariant 2

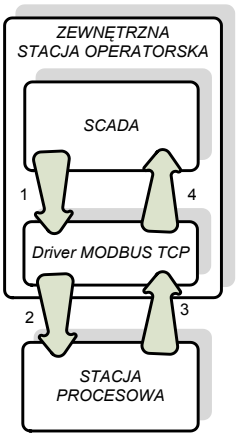
Po drugie, nawet jeśli diagnosta poradzi sobie z przechwyceniem komunikatów protokołu TCP/IP, musi w jakiś sposób wyłuskać te, które bezpośrednio wiążą się z przesyłem poleceń i odpowiedzi Modbus TCP. Dlatego bardzo pomocnym narzędziem są w tym przypadku odpowiednie filtry umożliwiające wyświetlenie tylko komunikatów Modbus TCP. Dla popularnego sniffera Wireshark można je znaleźć w [9].

Trzecią, ostatnią trudnością w użyciu omawianego sposobu diagnozowania komunikacji Modbus TCP, są ograniczone możliwości poznawcze inżyniera-operatora. Mimo posiadania logu (zapisu przechwyconych komunikatów ze stemplem czasowym) do wiarygodnego wnioskowania diagnostycznego na podstawie zapisu komunikacji wymagana jest jeszcze wiedza o znaczeniu kolejnych pól w komunikatach Modbus TCP i dopuszczalnych (czy też oczekiwanych) wartościach tych pól. Dlatego metoda ta jest polecana dla zaawansowanych, doświadczonych inżynierów uruchamiających lub poszukujących przyczyn pojawiającej się niezdatności komunikacji. Dostarcza najwięcej danych dotyczących zawartości przesyłanych komunikatów, ale wymaga od operatora gruntownej wiedzy.

4. Okno statusu komunikacji drivera komunikacyjnego

Wykorzystanie okna statusu komunikacji drivera komunikacyjnego jest jednym z prostszych sposobów dających odpowiedź na pytanie dotyczące zdatności komunikacji. Diagnozowanie prowadzone jest z poziomu stacji operatorskiej. Oprogramowanie wizualizacyjne stacji do komunikacji z urządzeniem (sterownikiem – stacją operatorską) potrzebuje odpowiedniego

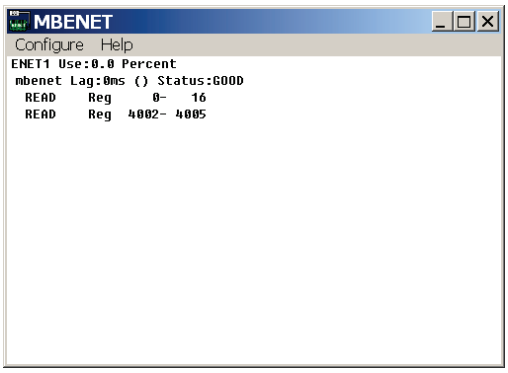
programowego interfejsu komunikacyjnego – drivera komunikacyjnego. Nie jest wymagane wyposażenie stacji w interfejs sprzętowy inny niż karta sieciowa popularnego standardu lokalnych sieci komputerowych Ethernet [10]. Komunikacja odbywa się wg schematu z rys. 4.



Rys. 4. Elementy systemu komunikacji

Wywołując okno statusu komunikacji (rys. 5), można odczytać podstawowe parametry dotyczące komunikacji Modbus TCP:

- opóźnienie komunikatów,
- jakość otrzymanych danych,
- typ transakcji (odczyt/zapis) – dla każdego rodzaju zmiennych,
- rodzaj wykorzystywanego rejestru (potrzebny np. do określenia numeru funkcji do odczytu/zapisu żądanej wartości zmiennej w stacji procesowej),
- adresy rejestrów w stacji procesowej.



Rys. 5. Okno statusu komunikacji drivera komunikacyjnego [11]

Dla przykładu zaczerpniętego z autorskiej aplikacji wykonanej w systemie *Intouch* z wykorzystaniem drivera *mbenet*, przedstawionego na rys. 5 [11], w przypadku chwilowej niezdatności układu komunikacji elementy okna statusu wyświetlane są kolorem czerwonym, informującym o pojawiających się błędach transmisji.

Przedstawiony tu sposób diagnozowania komunikacji na podstawie okna statusu pozwala na uzyskanie tylko zgrubnych danych dotyczących wybranych parametrów komunikacji. Więcej danych można wyłuskać, dołączając badanie pola zmiennej *Quality* [12].

5. Pole *Quality* zmiennej

Wykorzystanie tej metody pozwala na uzyskanie znacznie szerszej informacji na temat zdatności systemu komunikacji. W części – spośród dostępnych pakietów SCADA – wykorzystuje się dodatkowe pola zmiennych I/O (input/output – typ zmiennych wykorzystywanych do komunikacji). Standardowe odwołanie się w aplikacji wizualizacyjnej do nazwy zmiennej powoduje wykonanie operacji na wartości przypisanej do zmiennej [11]. Do zmiennej można się odwołać za pomocą składni: *Nazwa_zmiennej.pole*. Dla zmiennych typu I/O jest dostępnych kilka pól, na podstawie których można wnioskować o stanie eksploatacyjnym układu komunikacji. Wartości pola *Quality* jest zgodny zazwyczaj ze standardem OPC [13].

W tabeli 1 dla lepszej czytelności podano uproszczoną interpretację wartości pola *Quality*. Oczywiście, każdej z wymienionych w tabeli sytuacji eksploatacyjnych, określonych wartością pól status i substatus, odpowiadają pewne wartości reprezentowane przez 8 ostatnich, najmniej znaczących bitów. Pełną informację nt. wartości pól można znaleźć m.in. w [12]. Informacje o jakości połączenia wraz z danymi odczytanymi z okna statusu drivera komunikacyjnego dają znacznie szerszy obraz stanu systemu transmisji danych.

Tabela 1

Możliwe znaczenia wartości pola *Quality*, opracowanie na podstawie [12]

Lp.	Status jakości	Możliwy substatus jakości
1.	Dobry	Obcięcie wartości od góry
		Obcięcie wartości od dołu
		Brak możliwości konwersji
		Brak dostępu
		Brak komunikacji
2.	Niepewny	Nieokreślona wartość
		Ostatnio odebrana wartość
		Niedokładny czujnik
3.	Zły	Nieokreślona wartość
		Błąd w konfiguracji
		Brak połączenia
		Uszkodzenie sprzętowe
		Uszkodzenie czujnika
		Ostatnio odebrana wartość
		Uszkodzenie portu
		Wyłączenie

6. Zmienne systemowe bloków komunikacji języka FBD

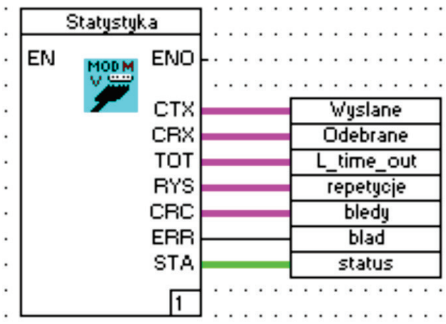
Przedstawione dotychczas rozważania na temat procesu diagnozowania sytemu komunikacji dotyczą, z wyjątkiem opcji analizatora jako oddzielnego stanowiska diagnostycznego, operacji przeprowadzanych z poziomu stacji operatorskiej. Jakie zatem możliwości diagnozowania są udostępniane przez stację procesową?

Z poziomu zadania stacji procesowej programowanej w języku bloków funkcyjnych [1] istnieje możliwość wykorzystania bloku dostarczającego pewnych danych statystycznych (w dużym uproszczeniu) dotyczących komunikacji, tj. liczby (rys. 6):

- wysłanych wiadomości (jeśli stacja procesowa jest klientem),
- odebranych wiadomości,
- komunikatów, przy których dostarczaniu przekroczono dopuszczalny czas odpowiedzi (*timeout*),

– powtórek transmisji komunikatów (repetycji),
– liczby komunikatów odebranych z niepoprawną wartością sumy kontrolnej (CRC).

Wartość podanych parametrów jest zerowana przy restarcie zadania komunikacyjnego.



Rys. 6. Odczyt uproszczonej statystyki komunikacji

7. Podsumowanie

W opracowaniu przedstawiono dyskusję dotyczącą możliwości wykorzystania kilku różnych sposobów pozyskania informacji diagnostycznej dotyczącej komunikacji prowadzonej wg protokołu Modbus TCP. Przedstawiono tu praktyczne metody diagnozowania komunikacji z użyciem zewnętrznej stacji diagnozującej (typ I), z poziomu stacji operatorskiej (typ II) oraz stacji procesowej (typ III) na podstawie (w nawiasie podano typ wskazujący na miejsce zastosowania procedury):

– logów uzyskanych przy pomocy oprogramowania analizatora protokołów (I lub II),
– danych prezentowanych w oknie statusu komunikacji drivera komunikacyjnego (II),
– wartości pola *Quality* zmiennych procesowych (II),
– wartości zmiennych systemowych bloków komunikacji (III).

Warto podkreślić, iż głównym zadaniem procesu diagnozowania komunikacji jest pozyskanie pełnej informacji o sytuacji istniejącej w systemie, za pomocą m.in. omówionych sposobów diagnozowania. Chcąc jednak zautomatyzować proces diagnozowania, należy pamiętać, że podstawowym zadaniem stacji operatorskiej jest wizualizacja i ewentualne oddziaływanie operatorskie na przebiegający proces,

natomiast zadaniem stacji procesowej – sterowanie procesem. Dlatego czynności związane z pozyskiwaniem niezbędnej informacji diagnostycznej powinny być wykonywane w taki sposób, aby nie utrudniały realizacji standardowych funkcji stacji.

Możliwość automatyzacji procesu diagnozowania z wykorzystaniem analizatora protokołów jest problematyczna. Wiąże się to bowiem z koniecznością zaangażowania sporej mocy obliczeniowej w procesie analizy zapisanych logów, a także z tym, że proces wnioskowania diagnostycznego na podstawie analizy ruchu sieciowego wymaga gruntownej wiedzy diagnosty. Proces ten nie może zatem odbywać się bez bieżącego nadzoru i ingerencji operatora.

Z poziomu stacji operatorskiej, w celu automatyzacji procesu diagnozowania, można natomiast wykorzystać np. skrypty okienne, zdarzeniowe, aplikacyjne, czyli programy napisane w wewnętrznym języku skryptowym danego pakietu SCADA, wykonywane podczas wywoływania, zamykania itp. odpowiednio: okien, pewnych czynności lub aplikacji wizualizacyjnych. Sposoby programowania tych skryptów dają dość niewielkie możliwości budowy zaawansowanych konstrukcji programistycznych, ale z powodzeniem mogą odwoływać się do zmiennych i ich pól. Z powodu tych ograniczeń najbardziej wygodnym, przydatnym sposobem pozyskiwania danych diagnostycznych jest więc wykorzystanie informacji zawartych w polu *Quality* zmiennych. Daje to możliwość uzyskania dosyć szczegółowej informacji diagnostycznej dotyczącej jakości komunikacji, przy jednoczesnym umiarkowanym zaangażowaniu mocy obliczeniowej stacji operatorskiej.

8. Literatura

1. Bednarek M.: Wizualizacja procesów. Laboratorium, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2004.
2. Dokumentacja elektroniczna systemu Freelance 2000.
3. Informator techniczny Wonderware: Komunikacja OPC w pakiecie Wonderware FactorySuite 2000, nr 24 30-10-2001.
4. MODBUS Messaging on TCP/IP Implementation Guide V1.0, October 24, 2006 http://www.Modbus-IDA.org.
5. MODBUS over Serial Line Specification and Implementation Guide, Modbus.org, 2005, [http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf].
6. https://digitalsupport.ge.com/en_US/Documentation/iFIX-58-Ebooks-in-PDF-Format
7. https://www.wireshark.org.

8. RFC 826, STD 37: An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, November 1982 [https://www.rfc-editor.org/info/rfc826].
9. https://www.wireshark.org/docs/dfref/m/mbtcp.html.
10. Standard IEEE 802.3 [https://standards.ieee.org/].
11. Dokumentacja elektroniczna pakietu Wonderware Intouch.
12. Wonderware FactorySuite InTouch, Opis funkcji, pól i zmiennych systemowych, Invensys Systems, Inc., 2005.
13. Mahnke W., Leitner S.H., Damm M.: OPC Unified Architecture, Springer Science & Business Media, 2009.