Article citation info:
Walczak M, Hetmanczyk M. The performance of IIoT communication standards. Diagnostyka. 2023;24(3):2023307.
https://doi.org/10.29354/diag/169033.

DIAGNOSTYKA, 2023, Vol. 24, No. 3

e-ISSN 2449-5220
DOI: 10.29354/diag/169033

THE PERFORMANCE OF IIOT COMMUNICATION STANDARDS

Maciej WALCZAK [1,*], Mariusz HETMANCZYK [2]

[1] APA Group, Tarnogorska 251 St., 44-105 Gliwice, Poland
[2] Silesian University of Technology, Faculty of Mechanical Engineering, Konarskiego 18A St., 44-100 Gliwice, Poland
[*] Corresponding author, e-mail: maciej.walczak@apagroup.pl

Abstract

The requirements of Industry 4.0 determine the necessity to change thinking in the field of production development, adopted management methods and modernisation of production resources. When planning the implementation of a new production system (or retrofit), it is possible to use the RAMI 4.0 reference model, which was published in April 2015 by the VDI/VDE Society Measurement and Automatic Control. A key aspect of modern industrial systems is connectivity and trouble-free data exchange. In the case of data exchange, the basic element holding back the development of Industry 4.0 is the lack of standardisation, as well as the lack of interoperability between IIoT network nodes. Modern IIoT applications require high network throughput, low latency and reliability. In view of such guidelines, efficient communication standards and specialised equipment are required. Edge Computing is one of the most important technology trends of the 21st century that will play a key role in the IIoT market. Due to the diversity of available technologies and solutions, no universal standards have been developed to date that can be referred to when planning, building and implementing new applications. The article presents an overview of the most popular industrial communication protocols and their systematisation in terms of meet the requirements for IIoT devices.

Keywords: Industry 4.0, Edge Computing, IIoT

List of Symbols/Acronyms

IIoT - Industrial Internet of Things
INT - Integer
IP - Internet Protocol
KB - Kilobyte
M2M - Machine to Machine
MQTT - Message Queue Telemetry Transport
OPC - Open Platform Communications
PLC - Programmable Logic Controller
QoS - Quality of Service
RTU - Remote Terminal Unit
SCADA - Supervisory Control And Data Acquisition
SSL - Secure Socket Layer
TCP - Transmission Control Protocol
TLS - Transport Layer Security
UA - Unified Architecture

1. INTRODUCTION

Industry 4.0 is a term that has been on the market for many years, and yet there are still many different ways of interpreting it. The idea is to use digitization, processing and exchange of huge amounts of data. It is important to note that their implementation should not be a goal, but only a tool to achieve business benefits [1]. Modern industry should connect people and machines and be based on the automatic collection and processing of large data sets, enabling automatic reporting and optimization of production processes [2].

Improving the production cycle requires permanent monitoring of dynamic processes. The fourth industrial revolution involves the transition from estimating the condition of devices (visual analysis at a time defined by a schedule) without taking into account historical data, to monitoring the condition of equipment by collecting data and making data-driven decisions [3]. This takes into account a much larger amount of data, results of previous inspections, additional variables (for instance environmental conditions, vibrations, energy consumption).

The amount of data generated by some IIoT devices and applications can be astonishing. However, not all of them are crucial for managing the production process. There is a common belief in industry that collecting large amounts of data guarantees success in subsequent analysis. The first stage is therefore the creation of a Big Data repository to which all potential information carriers will go, and in the second stage the number of recorded and analysed parameters is reduced.

Due to the variety of available technologies and solutions, so far no universal standards have been developed that can be referred to when planning,

Received 2023-03-18; Accepted 2023-06-29; Available online 2023-07-19

building and implementing new projects. Research shows that in the case of data exchange between industrial network nodes, the lack of standardization and interoperability is the primary element holding back the development of Industry 4.0 in enterprises. [4].

Most of the machines and production lines are controlled by the use of industrial computers or PLCs, and the configuration structure is strictly hierarchical. Unified communication protocols for data exchange used in enterprises, provide great opportunities for data access. However, they turn out to be insufficient for the advanced distribution of large data sets, not used strictly in the control process [5, 6].

Another element is the interoperability of the devices used. The communication protocol is largely defined by the enterprise control system. If there are Rockwell Automation controllers, the network is based on the EtherNet/IP standard [7]. In the case of SIEMENS S7 controllers, Profinet is used [8], and configurations with Mitsubishi controllers are based on CC-Link IE. Nowadays, due to economic crises and limited equipment availability, industrial plants cannot rely on a single technology provider. The industry expresses the need for open industrial communication technologies [9].

The communication protocol is a set of standards and rules that allow for the exchange of data between devices. At the same time, it defines the method of establishing a connection, sending information packets, as well as interpreting the received data. In the context of Industry 4.0, the term is often used to describe the communication capabilities of various devices [10]. There is no doubt that openness to TCP/IP and Ethernet standards [11] is a forward-looking approach and compatible with Industry 4.0.

OPC UA [12] and MQTT [13,14] are often mentioned among the advanced communication systems that aspire to take over the role of a universal way of information exchange in Industry 4.0.

However, they should not be understood only as protocols, but rather as end-to-end communication solutions. Additionally, a very popular standard that enables data acquisition from devices is Modbus TCP [11], which also finds numerous applications in IIoT applications [15-18].

2. THE SCOPE AND PUPROSE OF THE RESEARCH

This article compares the communication between the SIEMENS S7-1200 PLC controller and the IIoT Platform - Nazca 4.0 (Fig. 1). Both solutions: enable data exchange (OPC, MQTT and Modbus TCP protocols), use the TCP/IP interface, do not require additional communication modules and the libraries for their operation (available natively - Nazca 4.0, or on the manufacturer's website - Siemens S7).

NAZCA 4.0 is a cutting-edge, full-featured application for integrating production management systems in factories and enterprises. It allows combining the operation of machines within a single platform, optimizing the usage of electricity and other utilities and identifying production points that generate unnecessary costs [19].

The research carried out concerned only the reading of data from the controller as the main method of communication between the supervisory system and the process infrastructure in order to analyse the data.

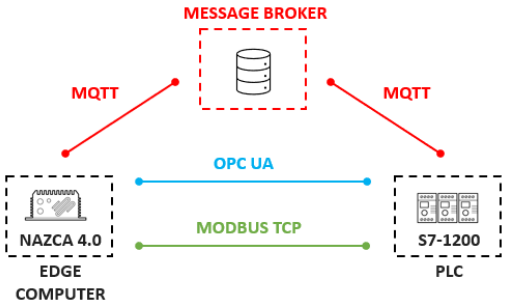

Fig. 1. Diagram of the configuration of data exchange devices and communication protocols within the performed tests

For the purposes of the research, the following factors were taken into account: communication network model and bandwidth usage, stability and security of communication.

2.1. OPC UA protocol

The first generation of the OPC protocol (OPC classic), was released in 1996. It was designed to abstract PLC-specific protocols in a standardized interface to allow SCADA systems to access data from controllers of different manufacturers.

OPC UA (developed by the OPC Foundation in 2008) is the next generation standard as an update to the original for safe and reliable data exchange in industrial automation. Information sharing in OPC UA is based on the Client-Server model (Fig. 2).

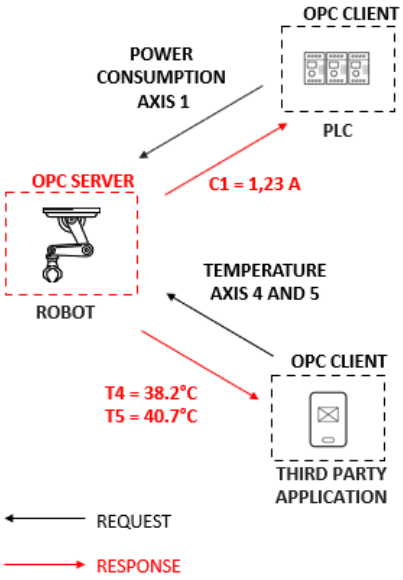

Fig. 2. Schematic representation of the data exchange using the OPC UA protocol

During the tests, temperature parameters and energy consumption (Tab. 1) of devices configured as OPC CLIENT were recorded.

Table 1. List of OPC UA objects

Object	Variable ID
TEMPERATURE	T4, T5, T6
POWER CONSUMPTION	C1, C2

In this solution, one of the devices is a server with appropriately modelled data and functionalities and after establishing connection, the client obtains read or read-write access.

Servers can also provide information about data (structure, names) directly after being connected to the network. In addition, OPC UA has built-in cybersecurity functionalities and enables simultaneously connection encryption (Table 2).

Table 2. List of advantages and disadvantages of the OPC UA protocol

Disadvantages	Advantages
High bandwidth usage for full communication.	Data encryption and certificates of reliability.
The specification is very extensive and includes a number of functions, the implementation of which is optional.	The ability to view OPC objects provided by the server (without the need to know the server configuration, which makes the implementation of the OPC client very easy).
	Possibility of using cyclic (not recommended) and acyclic communication.
	The ability to create separate subscriptions with different publishing times.
	Possibility of connecting multiple clients at the same time.

The OPC UA protocol requires a large number of packets to establish communication. Figure 3 shows the packet exchange between the server and the client during the connection establishment phase. This is the main shortcoming of the protocol in question, but a solution to the problem has now been developed.

Before the first update of the OPC UA object state, the devices sent 23 packets and approximately 7 KB of data. By configuring communication on the OPC UA client side, it is possible to group objects in different subscriptions (sending notifications about a change in object data value).

OPC UA optimizers are also available on the market to improve the performance of certain operations. Optimizers are implemented as a configurable plugin in each instance of the web object.

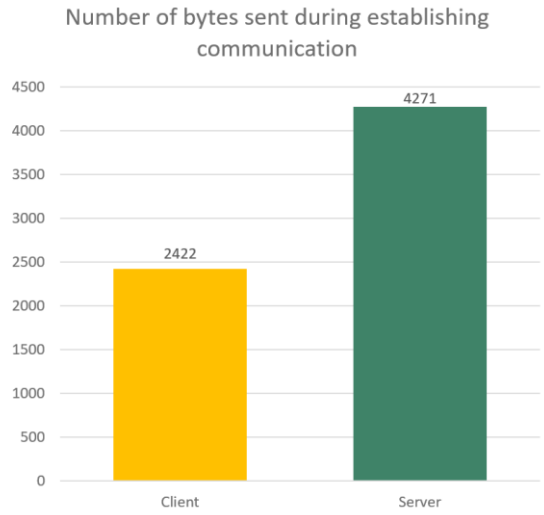

Fig. 3. Data packets exchange in the phase of establishing the connection of the OPC UA protocol

This optimizes the bandwidth usage is presented in Figure 4.

Protocol	Length	Info
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	188	UA Secure Conversation Message: PublishResponse

Fig. 4. Exchange of data in the case of reading one variable

The notifications are transported in OPC UA Publish Response packets and contain only those subscription objects that have changed the value.

A single change of 500 variables, within a single subscription, requires new values to be queried twice (Fig. 5).

For one variable of the INT type (2 bytes), the response frame contains 188 bytes, and for 500 variables of this type (1000 bytes), the double response frame contains 2208 bytes. The queries are again 128 bytes and the full communication is 2.5 KB.

Protocol	Length	Info
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	864	UA Secure Conversation Message: PublishResponse
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	1344	UA Secure Conversation Message: PublishResponse

Fig. 5. Exchange of data in the case of reading 500 variables.

OPC UA clients can use the ReadRequest and ReadResponse packets for cyclical communication (not recommended), but also for testing the connection to the server.

These packets are small and sent at relatively long intervals (in the example below, 10 seconds), without significant impact on bandwidth usage.

Figure 6 shows the exchange of data using ReadRequest and ReadResponse packages.

A high level of cybersecurity is ensured by encryption (SSL/TLS) and authentication (username and password) from the level of data acquisition and public key.

Protocol	Length	Info
OpcUa	150	UA Secure Conversation Message: ReadRequest
OpcUa	120	UA Secure Conversation Message: ReadResponse
OpcUa	644	UA Secure Conversation Message: PublishResponse
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	644	UA Secure Conversation Message: PublishResponse
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	644	UA Secure Conversation Message: PublishResponse
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	476	UA Secure Conversation Message: PublishResponse
OpcUa	128	UA Secure Conversation Message: PublishRequest
OpcUa	150	UA Secure Conversation Message: ReadRequest
OpcUa	120	UA Secure Conversation Message: ReadResponse

Fig. 6. Exchange of data using ReadRequest and ReadResponse packages

2.2. MQTT protocol

MQTT is a communication protocol developed in 1999, based on the Public-Subscribe model and it enables the effective exchange of information between different devices. MQTT is designed as a lightweight protocol - which means that it does not require a large amount of computing resources to process received and sent data. This is especially important in automation components with a low processor power (e.g. sensors, actuators). In the Public-Subscribe model, devices that receive and publish data are unaware of each other's presence. Each device only connects to a communication broker, which is the central element of communication. The information is then published on special channels called topics.

Additionally, the communication can be secured by TLS and the connection to the broker can be realized using a static IP address or a domain name.

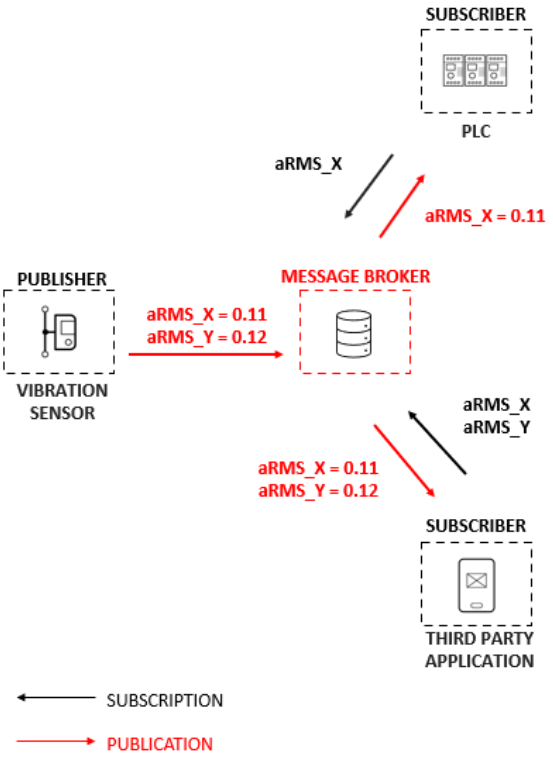

Fig. 7. Schematic representation of the data exchange using the MQTT protocol

After establishing a connection via TCP/IP, a session is created at the level of the MQTT protocol between the subscriber and the broker, using two connect commands (Command and Ack), 174 bytes in total (Figure 8).

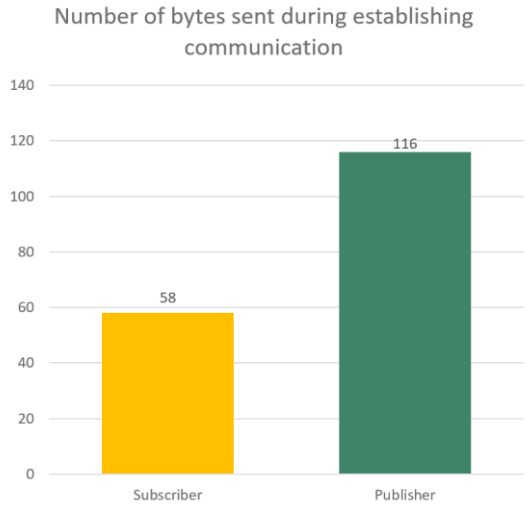

Fig. 8. Data packets exchange in the phase of establishing the connection of the MQTT protocol

Data within the MQTT protocol is sent with messages of the Publish type. First from server to broker, then from broker to subscribers. Each package is associated with a topic name. This is a hierarchical namespace that defines the information and therefore the subscribers who will receive the data package. Figure 9 shows the publication of one INT variable with a transfer of only 64 bytes. The packet is sent acyclically, only when the variable value is changed.

Protocol	Length	Info
MQTT	64	Publish Message [test]

Fig. 9. Exchange of data in the case of reading one variable

The transmission of 1000 bytes within one topic was associated with the transmission of one packet of 1062 bytes (Fig. 10). Compared to sending one variable of type INT, only the data field has increased (from 2 to 1000 bytes), and the rest of the packet format remains the same.

Protocol	Length	Info
MQTT	1062	Publish Message [test]

Fig. 10. Exchange of data in the case of reading 1 kb of data

To test the connection to the broker, MQTT subscribers use the PingRequest and PingResponse packages. These packets are small and transmitted at relatively long intervals (less than 18 seconds in the example in Figure 11) with no significant impact on network bandwidth. Table 3 summarizes the

advantages and disadvantages of the MQTT protocol.

Protocol	Length	Info
MQTT	60	Ping Request
MQTT	56	Ping Response
MQTT	108	Publish Message [test]
MQTT	60	Ping Request
MQTT	56	Ping Response
MQTT	60	Ping Request
MQTT	56	Ping Response

Fig. 11. Exchange of data using PingRequest and PingResponse packets

Table 3. Summary of advantages and disadvantages of the MQTT protocol

Disadvantages	Advantages
The broker is a communication bottleneck (for larger scale solutions, broker hubs can be used).	MQTT is a very lightweight protocol with a low transport load and low demand for network bandwidth.
Vulnerability to failures (a centralized broker is a key communication element).	Security - TLS / SSL support for encrypting connections between devices and the broker.
No search function for the broker or available topics - it is necessary to know the structure of access to data.	The Quality of Service (QoS) parameter, which is used to guarantee the reliability of communication.
The names of the topics are arbitrary - in the absence of standardization / documentation, it is difficult to define what is behind them.	The ability to assigning permissions (on the broker side) for each topic separately.
	Acyclic communication.
	Publish-subscribe model enabling high scalability.

The basic element securing communication is the authentication mechanism (publication, subscription) with the use of username and password, as well as encrypts messages by SSL/TLS and QoS (different levels of message delivery reliability):

- QoS 0 (at most once) - the receiver of the message does not send the acknowledgement packet; the message is delivered once or not at all (in case of communication problems) and is not saved on the sender's side,
- QoS 1 (at least once) - after receiving the packet, the receiver sends an acknowledgment packet; the message is stored on the sender's side and sent until the sender receives an acknowledgment that the packet was received; the service guarantees the delivery of each package at least once (it can be delivered multiple times when there is a problem with the confirmation package),
- QoS 2 (exactly once) - after receiving the packet, the receiver sends an acknowledgement packet; the message is stored on the sender's side and sent until the sender receives an acknowledgment that the packet was received; a more advanced acknowledgment sequence for receiving packets ensures that each packet is published exactly once.

In the event of loss of communication, the functionality of notifying interested subscribers is available, as well as automatic reconnection after the return of the subscriber and sending the lost messages.

2.3. Modbus TCP protocol

The Modbus TCP protocol was developed in 1979 by Modicon, and thus is one of the oldest digital transmission protocols used in industrial and building automation. It is the equivalent of Modbus RTU, but for communication it uses the TCP protocol (on port 502). It belongs to the master-slave family, which means that only one device can generate network traffic, and the other devices can only respond or take actions related to transmitted request. Variables are mapped as registers, where the device manufacturer defines the table according to which the appropriate parameters are assigned registers with read or read-write permissions. It is possible to group registers.

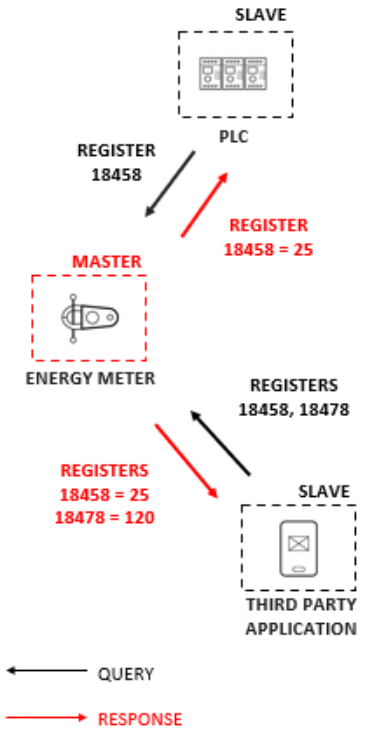

Fig. 12. Schematic presentation of the data exchange using the Modbus TCP protocol

Table 4. List of MODBUS TCP protocol map variables

Address	Description	Variable ID
18458	Current intensity	I1
18460	Current intensity	I2
18462	Current intensity	I3
18476	Total Active Power	-
18478	Total Reactive Power	-

Slave devices are passive, sends responses only when a value is queried. Variables are read cyclically using Query and Response packets containing information about registers. Although the query for a single parameter of the INT type requires only 131 bytes to be transmitted, in the case of slow-changing registers, cyclic communication significantly affects high usage of bandwidth. This is due to the continuous acquisition of the same value.

Protocol	Length	Info
Modbus…	66	Query: Trans: 58; Unit: 1, Func: 3
Modbus…	65	Response: Trans: 58; Unit: 1, Func: 3

Fig. 13. Exchange of data in the case of reading one variable

Observing 250 REAL variables (1000 bytes) within one group, required querying four times for successive registers. Each response packet can contain a maximum of 250 bytes. One-time refresh of the variable values, in this case, is 8 packets and 1.5kB of data (Fig. 14). In the case of the Modbus TCP protocol, it is only possible to limit the communication to the specified IP address, but it is not possible to encrypt data.

Protocol	Length	Info
Modbus…	66	Query: Trans: 24; Unit: 1, Func: 3
Modbus…	313	Response: Trans: 24; Unit: 1, Func: 3
Modbus…	66	Query: Trans: 25; Unit: 1, Func: 3
Modbus…	313	Response: Trans: 25; Unit: 1, Func: 3
Modbus…	66	Query: Trans: 26; Unit: 1, Func: 3
Modbus…	313	Response: Trans: 26; Unit: 1, Func: 3
Modbus…	66	Query: Trans: 27; Unit: 1, Func: 3
Modbus…	313	Response: Trans: 27; Unit: 1, Func: 3

Fig. 14. Exchange of data in the case of reading 1 kb of data

Table 5 summarizes features of the Modbus TCP protocol.

Table 5. Summary of advantages and disadvantages of the Modbus TCP protocol

Disadvantages	Advantages
Addressing limited to 247 devices, which reduces the number of devices that can be connected to the Master.	Interoperability - a very large number of measuring devices supporting the Modbus TCP/IP protocol.
Slaves cannot report information without being queried by the Master device.	The specification is available for download free of charge and no license fees are required for using the Modbus TCP/IP protocol.
No protection against unauthorized commands or data interception.	Built-in diagnostics - a defined communication timeout that occurs between the client and the server; an error occurs when communication is broken.
No data encryption.	Ease of implementation and commissioning.

3. SUMMARY

The ongoing technological transformation forces a change in the production management architecture and the transition from linear communication processes to networks of connections of devices and systems. It is necessary to ensure a high level of operational autonomy of system elements and the possibility of distributed decision-making based on the current state of production [20]. The use of the IIoT, as well as Big Data processing mechanisms, is key factor in digital transformation. Production becomes more flexible and manufacturers are able to fulfil customized orders faster and cheaper than before [2].

Many communication protocols can be used in similar a scenario which does not mean that they are ideal for a given application. The problem is not in choosing a specific protocol, but in understanding the differences in data exchange.

The issue of the number of packets and bytes is relevant for heavily loaded networks with low bandwidth. When using high-speed networks, it is only necessary to analyse the number of data bytes (not the number of packets sent) as they are a critical indicator in slow networks.

Figure 15 shows a comparison of the number of packets sent by two devices involved in communication during sending one variable of the integer type. In the case of MQTT, the publication of a single value is only 64 bytes, while in the case of OPC it is almost 5 times more.

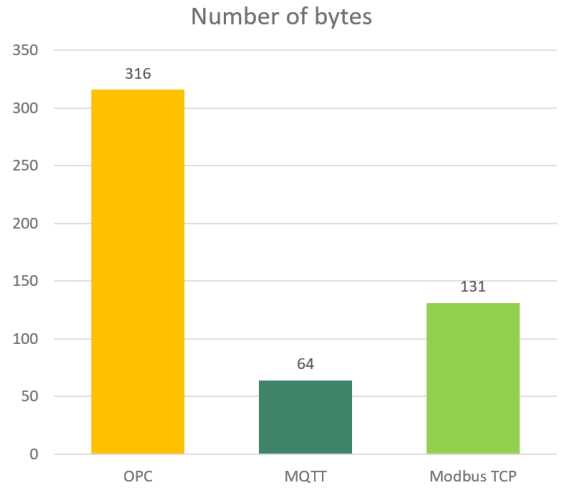

Fig. 15. Number of bytes transferred between devices when sending a single integer value

Figure 16 shows a comparison of the number of packets sent by two devices participating in communication when sending 1000 bytes of data (e.g. 250 floating point values). Again, the OPC protocol requires the largest number of communication packets to be sent, but in this case it is only 60% more packets than in the MQTT protocol.

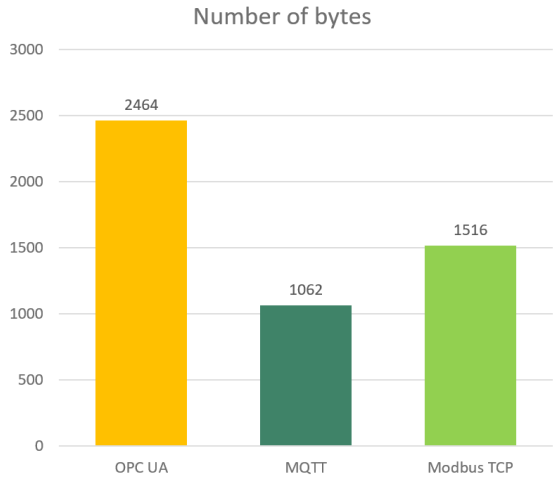

Fig. 16. Number of bytes transferred between devices when sending 250 floating point values (1000 bytes)

Given that the Fourth Industrial Revolution requires the collection and analysis of large data sets, it is very important to reduce bandwidth requirements. As far as this is concerned, the MQTT protocol is the best solution, no matter if single values or large packets of data from measurement devices are sent.

MQTT is designed to be lightweight and energy efficient. For this reason, there is no implemented search for the topic or broker where the message is published. However, it is necessary for the subscriber to know the structure before establishing the communication. MQTT is suitable for transmitting data between devices with minimal functionality and for transmitting over unreliable networks with low bandwidth and high latency. Thanks to these features, MQTT plays a key role in IIoT and M2M communication [21].

OPC UA is a complete architecture in which the communication protocol is only part of it. The OPC UA allows all network nodes, methods and data structures to be monitored. Communication can be based on both a subscription model and a client-server model and is compatible with various operating systems. It allows data encryption and supports certificates of reliability. If the goal is to share data from a PLC or a robot, it is easiest to do with OPC.

The Modbus TCP protocol is the most widely used communication protocol in the case of measuring devices (the first widely accepted communication standard). It is very easy to understand and implement also for machine builders who are not programmers. It is perfect for reading fast-changing values, where cyclical data collection is required (e.g. electrical network analysers, compressed air flow meters).

Many manufacturing plants have chosen a protocol based on the architecture that exists in their production environment. If they have a SCADA system, they usually use OPC UA. However, when purchasing new equipment or looking to make a digital transformation, it is worth considering devices with the MQTT protocol.

Author contributions: *research concept and design, M.H.; Collection and/or assembly of data, M.H., M.W.; Data analysis and interpretation, M.H., M.W.; Writing the article, M.H., M.W.; Critical revision of the article, M.H.; Final approval of the article, M.H.*

Declaration of competing interest: *The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

REFERENCES

1. Mychlewicz C, Piątek Z. From industry 4.0 to smart factory. Siemens 2017; 5: 32.
2. Dziurdzia M, Janiszewski M, Gęborys P, Gawrysiak M, Jankovic-Żelazna A. Industry 4.0, the challenges of modern production. PwC; 4.
3. Chunquan L, Yaqiong C, Yuling S. A review of industrial big data for decision making in intelligent manufacturing. Engineering Science and Technology, an International Journal 2022; 29: 1-16. https://doi.org/10.1016/j.jestch.2021.06.001.
4. Pollak A, Hilarowicz A, Walczak M, Gąsiorek D. A framework of action for implementation of industry 4.0. An empirically based research. Sustainability 2020; 12.
5. Baptista LF, Barata J. Piloting industry 4.0 in SMEs with RAMI 4.0: an enterprise architecture approach. Procedia Computer Science 2021; 192: 2826-2835. https://doi.org/10.1016/j.procs.2021.09.053.
6. Benešl T, Kaczmarczyk V, Sýkora T, Arm J, Dvorský P, Husák M, Marcoň P, Bradáč Z. Asset administration shell - manufacturing processes energy optimization. IFAC – PapersOnLine 2022; 55(4): 334-339. https://doi.org/10.1016/j.ifacol.2022.06.055.
7. Alessandria E, Seno L, Vitturi S. Performance analysis of Ethernet/IP networks. IFAC Proceedings Volumes 2007; 40(22): 391-398. https://doi.org/10.3182/20071107-3-FR-3907.00054.
8. Ferrari P, Flammini A, Vitturi S. Performance analysis of PROFINET networks. Computer Standards & Interfaces 2006; 28(4): 369-385. https://doi.org/10.1016/j.csi.2005.03.008.
9. Dahlman E, Parkvall S, Sköld J. The next generation wireless access technology. 5G NR (Second Edition): Springer Elsevier; 2021. https://doi.org/10.1016/B978-0-12-822320-8.00005-2
10. https://elearning.przemyslprzyszlosci.gov.pl/slownik-pojec/protokol-komunikacyjny/.
11. Elamanov S, Son H, Flynn B, Ki Yoo S, Dilshad N, Song JS. Interworking between Modbus and internet of things platform for industrial services. Digital Communications and Networks; 2022. https://doi.org/10.1016/j.dcan.2022.09.013.
12. Schleipen M, Gilani SS, Bischoff T, Pfrommer J. OPC UA & Industrie 4.0 - Enabling technology with high diversity and variability. Procedia CIRP 2016; 57: 315-320. https://doi.org/10.1016/j.procir.2016.11.055.
13. Mishra B, Kertesz A. The use of MQTT in M2M and IoT systems: a survey. IEEE Access 2020; 8: 201071-201086. https://doi.org/10.1109/ACCESS.2020.3035849
14. Atmoko RA, Riantini R, Hasin MK. IoT real time data acquisition using MQTT protocol. Journal of Physics:

Conference Series 2017; 853: 1-6. https://doi.org/10.1088/1742-6596/853/1/012003.
15. Bhardwaj A, Kaushik K, Bharany S; Rehman AU, Hu YC, Eldin ET, Ghamry NA. IIoT: traffic data flow analysis and modeling experiment for smart IoT devices. Sustainability 2022; 14(14645): 1-18. https://doi.org/10.3390/su142114645.
16. Tashtoush YM, Darweesh DA, Husari G, Darwish OA, Darwish Y, Issa LB, Ashqar HI. Agile approaches for cybersecurity systems, IoT and intelligent transportation. IEEE Access 2022; 10: 1360-1375. https://doi.org/10.1109/ACCESS.2021.3136861.
17. Zhang T, Li Y, Philip Chen CL. Edge computing and its role in industrial internet: methodologies, applications, and future directions. Information Sciences 2021; 557: 34 - 65. https://doi.org/10.1016/j.ins.2020.12.021.
18. Yi S, Li C, Li Q. A survey of fog computing: concepts, applications and issues. Mobidata '15: Proceedings of the 2015 Workshop on Mobile Big Data 2015; 15: 37-42. https://doi.org/10.1145/2757384.2757397.
19. https://nazca40.pl/en/nazca-4-0-industry-iiot-platform-english/.
20. Mychlewicz C, Piątek Z. Siemens report - from industry 4.0 to smart factory. Siemens 2017; 22-24
21. Van Glabbeek R, Deac D, Perale T, Steenhaut K, Braeken A. Flexible and efficient security framework for many-to-many communication in a publish/subscribe architecture. Sensors 2022; 22: 7391.

Maciej WALCZAK,
Master of Science in Automation and Robotics at the Silesian University of Technology. Now a PhD student in the field of distributed edge processing algorithms in Industry 4.0 systems. Employee of APA company as a project manager.
E-mail: maciej.walczak@apagroup.pl

Mariusz HETMANCZYK,
PhD, Associate Professor. Employee of the Department of Automation of Technological Processes and Integrated Manufacturing Systems, Faculty of Mechanical Engineering of the Silesian University of Technology. Scientific interests focused on the following issues: diagnostics and industrial networks, RAMI 4.0 in the architecture of production systems, Asset Administration Shell, electric drive systems.
E-mail: mariusz.hetmanczyk@polsl.pl