DOI: https://doi.org/10.15276/hait.07.2024.4
UDC 004.7

Study of the method of controlling the compatibility of Internet of Things devices based on the MQTT application layer protocol

Artur V. Timenko[1)]
ORCID: https://orcid.org/0000-0002-7871-4543; timenko.artur@gmail.com. Scopus Author ID: 56503994400
Vadym V. Shkarupylo[2)]
ORCID: https://orcid.org/0000-0002-0523-8910; shkarupylo.vadym@nubip.edu.ua. Scopus Author ID: 57189326576
Nataliia A Kulykovska[1)]
ORCID: https://orcid.org/0000-0003-4691-5102; natalya.gontar@gmail.com. Scopus Author ID: 57208667683
Svitlana S. Hrushko[1)]
ORCID: https://orcid.org/0000-0002-0064-408X; grushko_ss@i.ua. Scopus Author ID: 57202232710
[1)] National University "Zaporizhzhia Polytechnic", 64, Zhukovsky Str. Zaporizhzhia, 69063, Ukraine
[2)] National University of Life and Environmental Sciences of Ukraine, 15, Heroyiv Oborony Str. Kyiv, 03041, Ukraine

ABSTRACT

Amid the rapid development of the Internet of Things and its impact on various areas of life, ensuring compatibility between different system components is becoming an urgent task. This is especially important in the context of developing and integrating Internet of Things systems with a high level of diversity and dynamism. In this article, we consider the problem of interoperability of Internet of Things components, focusing on application layer protocols that are key to ensuring intercomponent interaction. The main purpose of the article is to develop and validate a model that will optimize the processes of interaction between system components, taking into account the specifics of protocols. The model is based on the use of temporal action logic, which provides formal verification of interactions between components and allows identifying potential compatibility problems at the early stages of development. The developed model has been tested using a software simulator that allows simulating various scenarios of interaction in the Internet of Things network. The experimental results demonstrate the effectiveness of the proposed methodology in increasing the level of interoperability between system components, which in turn reduces the risks of data loss and ensures the stability of Internet of Things systems. Due to the in-depth analysis and development of specialized methods and tools, this study makes a significant contribution to the development of theoretical and practical aspects of interoperability. However, to further improve the accuracy and versatility of the model, additional empirical studies with a larger data set are recommended.

Keywords: Internet of Things; device compatibility; interaction protocol; temporal logic of actions

***For citation*: Timenko A. V., Shkarupylo V. V., Kulykovska N. A., Hrushko S. S. "Study of the method of controlling the compatibility of Internet of Things devices based on the MQTT application layer protocol". *Herald of Advanced Information Technology*. 2024; Vol. 7 No. 1: 48–58. DOI: https://doi.org/10.15276/hait.07.2024.4**

INTRODUCTION

The emergence of the Internet of Things (IoT) has heralded a transformative era in industrial environments, enabling unprecedented levels of device and system interaction. A key element of seamless integration and interoperability in these complex ecosystems is device interoperability, which is achieved through standardized communication protocols. However, careful analysis reveals a significant divergence in existing approaches in terms of direction and implementation, with a notable lack of methods that specifically address IoT service interoperability in terms of interoperability and scalability.

This gap is particularly critical given the enormous scale of IoT systems, characterized by the number of devices involved, their geographic distribution, and the diversity of corporate ownership.

Ensuring the interoperability of IoT system components, especially at the interaction protocol levels, is a primary concern. Although most existing interoperability approaches are aimed at ensuring data integrity, the need to ensure consistent interaction between IoT service components requires further research.

This article proposes an approach to addressing this problem, focusing on verification of the model and method of IoT device interoperability. Software component verification takes advantage of the expressive capabilities of Temporal Logic of Actions (TLA), using the TLA+ formalism and the TLA Checker (TLC) method for model checking.

Thus, the developed method covers the verification of software interoperability of IoT system components, taking into account aspects of functional integrity and protocol-level interoperability. This research contributes to the field by providing a novel method for ensuring interoperability of IoT devices, focusing on both interoperability and scalability within a large and diverse IoT system environment.

LITERATURE REVIEW

The proliferation of the Internet of Things (IoT) has highlighted the paramount importance of device interoperability in these multifaceted systems. The seamless interoperability of IoT devices, which is crucial for the realization of complex IoT ecosystems, is hindered by many challenges, including different communication standards, heterogeneous device capabilities, and diverse security protocols [1, 2].

Recent research has delved into these issues, proposing innovative solutions to enhance interoperability and collaboration in IoT systems. Haugen and Delsing [3, 4], [5] highlighted the interoperability challenges in industrial IoT systems and proposed a decentralized approach using semantic technologies to facilitate real-time device synchronization, a method proven in the chemical industry. This approach emphasizes the need for semantic interoperability, where devices understand and interpret data in a context-sensitive manner, enabling more consistent and functional interactions.

In addition, Malamas, Dasaklis, Voutsinas, and Kotzanikolaou [6, 7], [8, 9] have explored the integration of blockchain architecture as a service layer into existing ERP systems, improving access control and data integrity to enhance interoperability between supply chain stakeholders. This proposal reflects the growing trend of using distributed ledger technologies to ensure data integrity and trust in IoT ecosystems, addressing issues related to data tampering and unauthorized access.

In the healthcare sector, Kuo and Cheng [10, 11], [12] explored the use of smart contracts and blockchain technologies to enhance data integrity and authenticate user interactions, potentially improving interoperability in healthcare interoperability systems using the FHIR standard. Their work highlights the critical role of secure and reliable data exchange in healthcare IoT systems, where patient privacy and accuracy are of utmost importance.

The IoTAttest framework, introduced by Dirin, Oliver, and Lyne [13, 14], [15], focuses on device interoperability through Trusted Platform Module 2.0 and remote attestation technologies, ensuring the integrity of devices and data in IoT ecosystems. This framework addresses interoperability at the hardware level, ensuring that devices are not only interoperable but also protected from tampering and cyber threats.

Klimushin, Roe, and Kolisnyk [16, 17], as well as Boer, Bosco, Ugarelli, and Yaatun [18, 19], [20] have called for standardization and the development of a robust certification infrastructure to increase interoperability of IoT devices, thereby reducing security risks. Their discussions emphasize the need to establish industry-wide standards and certification processes to ensure that devices from different manufacturers can communicate and operate seamlessly in the same ecosystem.

Together, these studies emphasize the complexity of achieving device interoperability in IoT systems, pointing to the need for a multidimensional approach that encompasses not only technical solutions but also regulatory frameworks and industry standards. Ongoing research in this area indicates the dynamic nature of the IoT, where innovations in interoperability and interconnection mechanisms are key to the sustainable growth of IoT networks.

PURPOSE OF THE ARTICLE

The presented work offers a comprehensive way to solve the problem of ensuring the compatibility of IoT system components by developing a method and means of compatibility control at the program level. The purpose of the article is to develop a mathematical model that will increase the efficiency of compatibility control of IoT system components based on the MQTT application layer protocol

To achieve this goal, it is necessary to:

– to develop a model for formalizing intercomponent interaction links, which will automate the process of controlling the compatibility of these components at the level of interaction protocols;

– based on the above model, to develop an appropriate method for automated control of the compatibility of IoT system components, which would allow formulating judgments regarding the compatibility of these components at the level of the applied interaction protocols;

– check the model for adequacy using IoT network modeling software.

MAIN PART
1. PROTOCOLS OF IOT DEVICES

Component interoperability can be achieved through the use of standardized protocols and interfaces that allow system components to "understand" each other and cooperate seamlessly. This implies that each component of the system can receive, process, and send data according to defined rules and formats that ensure that messages transmitted between components are correctly interpreted and processed. The interoperability model should assume that the behavior of each component in any possible state of the system remains within the expected parameters, thus ensuring that the systems as a whole function harmoniously.

At the initial stage of researching IoT device interoperability, the key is to define the fundamental concepts. This involves an in-depth analysis of the principles of IoT devices, the characteristics of the communication protocols used, the areas of application, and the potential benefits and limitations in the context of interoperability. Particular attention is paid to the study of the interaction between different components of the IoT system at the protocol level, using a detailed comparison to identify common states and possible transitions based on the types of messages supported by each component.

The IoT can be described as a network of complex sensors and devices embedded in physical objects and connected via the Internet to collect data and manage these objects. IoT functionality is based on sensors, identification, communication, computing, services, and semantics. IoT devices collect data from the environment, recognize it according to application requirements, and use appropriate methods to process and analyze it. Computing is distributed between network devices and cloud data centers, depending on the requirements for functionality and quality of service [21, 22].

To simulate various practical IoT applications, it is necessary to integrate a large number of sensors, actuators, and edge devices operating in a variety of environments. This presents researchers with challenges due to the heterogeneity of IoT device characteristics and the need to adapt systems to ensure optimal performance, including resource allocation, task migration, and fault tolerance. The main challenges are related to modeling application architecture and network protocols. There are many protocols for transferring data between sensors, networks, and cloud servers, but none of them can fully meet all the needs of various IoT use cases. Therefore, it is important to consider a combined Modeling the interaction between these protocols is a complex task that requires careful analysis and optimization [24].

Table 1 provides a comparative analysis of various IoT protocols such as MQTT, CoAP, AMQP, WebSocket, and XMPP, focusing on the main parameters: transport layer, quality of service (QoS), and security [25].

This comparison demonstrates the unique features of each protocol and their suitability for specific IoT use cases. Based on this analysis, it can be argued that the study using the MQTT protocol is justified due to its effectiveness in the conditions of limited bandwidth and power consumption requirements that are typical for IoT devices. MQTT is distinguished by its scalability, providing the ability to work with a large number of devices, and supports different levels of QoS, which ensures reliable message delivery.

In addition, the protocol offers solid security mechanisms, including SSL/TLS encryption and authentication, making it suitable for applications where data security is critical.

Table 1. **Messaging protocols in IoT systems**

Title	Transportation level	Quality of service (QoS)	Security.
MQTT (Message Queuing Telemetry Transport)	TCP/IP	3 levels: At most once (0), At least once (1), Exactly once (2)	SSL/TLS, Authentication and Authorization
CoAP (Constrained Application Protocol)	UDP	4 levels: Confirmation, Non-confirmation, Confirmation, Reset	DTLS, Pre-shared keys, Raw public keys, Certificates
AMQP (Advanced Message Queuing Protocol)	TCP/IP	Multiple levels, including At least once, At most once, Exactly once	SASL TLS for Authentication and Encryption
WebSocket	TCP/IP	Not defined; relies on the reliability of the underlying TCP	SSL/TLS
XMPP (Extensible Messaging and Presence Protocol)	TCP/IP	Not specifically defined; relies on extension	SASL/TLS Authentication and Encryption

Source: **compiled by the authors**

2. SYSTEM COMPONENT INTEROPERABILITY MODEL AT THE PROTOCOL LEVEL

Establishing the compatibility of system components at the level of interaction protocols can be reduced to determining the truth of the following statement:

$$M, \sigma | = \phi, \quad (1)$$

where: M is the system model on the basis of which we check the compatibility of components at the application layer of the OSI network model; σ is computation as a sequence of states that reproduces the behavior of the system on the model M; ϕ is a temporal formula that must take a true value for each element of the sequence σ.

For the model M we take the Kripke structure on the set of atomic statements AP:

$$M = \langle S,\ S_0,\ R,\ L \rangle, \quad (2)$$

where S is the total set of states of the model; $S_0 \subset S$ is the set of initial states; $R \subseteq S^2$ is set of transitions between states; $L: S \to 2^{AP}$ is function of marking states.

The idea of the approach used in this work, which distinguishes it from alternative solutions, is as follows: two aspects are covered, namely, checking the correctness of the protocol specification for the interaction of system components and checking the correctness of the protocol implementation used according to the specification. The first verification is due to the constant development of protocol specifications in terms of acquiring new functionality. This process is accompanied by the influence of the human factor in terms of incompleteness and inaccuracy of the concepts used. The second check is focused on establishing compliance of the protocol implementation with the specification. In case of successful completion of each of the above checks, statements are made regarding the compatibility of IoT system components at the application layer of the OSI network model.

Thus, the proposed approach can be characterized as follows:

– create a protocol model based on the PlusCal tool according to expressions (1)-(2), where each label represents the corresponding property – the upper plane (the level of atomicity can be shifted);

– on the basis of the manually created PlusCal model, synthesize the corresponding detailed model in TLA+ using the tools of the TLA Toolbox (Table 2).

Table 2. **Specification of the protocol for interaction of system components in the PlusCal language**

No. s/n	Excerpt from the specification	Commentary
1	(* --algorithm spec variables v_1 \in {0,1,2}, v_2 \in {0,1,2}, v_3 \in {0,1,2}, v_4 \in {0,1,2}	Define the state variables and their valid values.

2	begin
	v_1 := 0; v_2 := 0;
	v_3 := 0; v_4 := 0;
		Set initial values for the variables.

3	while v_4 <= 2 do
	if v_1 < 2
	then v_1 := v_1+1;
	else
	if v_2 < 2
	then v_2 := v_2+1;
	else
	if v_3 < 2
	then v_3 := v_3+1;
	else
	if v_4 < 2
	then v_4 := v_4+1;
	end if;
	end if;
	end if;
	end if;
	end while;
	end algorithm*)
		An algorithmic component that determines the sequence of changes in the values of variables

Source: **compiled by the authors**

The use of PlusCal makes it easier to understand the algorithmic component at the heart of the specification.

Table 3 shows the TLA+ specification based on the content of Table 2.

The adequacy of the developed model was tested on the example of the specification of the protocol of the application layer of interaction of IoT system components – MQTT. For this purpose, the number of states and the depth of traversals for the transition systems based on the Kripke structure and the corresponding specification in the TLA+ language were compared. Number of states: 9, traversal depth: 8.

3. METHOD OF CONTROLLING THE COMPATIBILITY OF SYSTEM COMPONENTS AT THE PROTOCOL LEVEL

The essence of the developed method is as follows [26, 27]: the components of the IoT system are considered in pairs. A common set of state variables of the transition system is formed – based on the types of messages supported by each of the elements of the pair; on the basis of the above transition systems, formal specifications are synthesized to be tested by the model checking

Table 3. **A fragment of the TLA+ system component interaction protocol specification**

No. s/n	Fragment of the specification	Commentary
1	* BEGIN TRANSLATION	Label of the built-in translation tool from PlusCal to TLA+
2	VARIABLES v_1, v_2, v_3, v_4	List of state variables - elements of the set
3	Invar == /\ v_1 \in {0,1,2} /\ v_2 \in {0,1,2} /\ v_3 \in {0,1,2} /\ v_4 \in {0,1,2}	Valid values of variables
4	Start == /\ v_1 = 0 /\ v_2 = 0 /\ v_3 = 0 /\ v_4 = 0	Initial state
5	Nxt == /\ IF v_4 <= 2 THEN /\ IF v_1 < 2 THEN /\ v_1' = v_1+1 /\ UNCHANGED << v_2, v_3, v_4 >> ELSE /\ IF v_2 < 2 THEN /\ v_2' = v_2+1 /\ UNCHANGED << v_3, v_4 >> ELSE /\ IF v_3 < 2 THEN /\ v_3' = v_3+1 /\ v_4' = v_4 ELSE /\ IF v_4 < 2 THEN /\ v_4' = v_4+1 ELSE /\ TRUE /\ v_4' = v_4 /\ v_3' = v_3 /\ v_2' = v_2 /\ v_1' = v_1 ELSE UNCHANGED <<v_1, v_2, v_3, v_4>>	An algorithm that determines the next 8 states
6	Sc == In /\ [][Nxt]_<<v_1, v_2, v_3, v_4>>	The resulting formula

Source: **compiled by the authors**

method; each of these specifications is tested by the model checking method in an automated mode. The application of the developed method involves organizing the interaction of system components according to the message exchange model. The developed method is based on the use of the developed model for checking the compatibility of IoT system components as a means of synthesizing input data for the method.

The essence of the method is revealed in the following steps:

Step 1. For each of the interacting components, a formal specification of the interaction protocol is synthesized according to the developed model.

Step 2. Each of the synthesized specifications is verified by model checking. The results of such verification are compared by the positions of the number of states and depths of traversals of the corresponding graphs (transition systems), analytically defined by the Kripke structure.

The above means that the compatibility of components at the level of interaction protocols is performed on the basis of topological verification, namely, the transition systems that are the basis of formal specifications of interaction protocols are considered as graphs. These graphs are compared by the following parameters: traversal depth, number of vertices. And if the specifications of the interaction protocols for all components coincide in these parameters, then such components are characterized as compatible with each other at the level of interaction protocols. If a certain specification(s) do not match, then the corresponding components are characterized as incompatible.

Let's say we have n system components. Then the statement about the compatibility n of system components at the level of interaction protocols can be formalized as follows:

$$\phi_1 \equiv \phi_2 \equiv \ldots \equiv \phi_n, \qquad (3)$$

where $\phi_j (j = 1,2,\ldots,n)$ is the temporal formula defined by the formal specification of the interaction protocol j-of the system component; $\equiv$ means that all temporal formulas from ϕ_1 to ϕ_n are equivalent to each other.

In the context of the developed model, to control the compatibility of IoT system components, they must interact according to the same interaction protocol specification. That is, each component must meet the same requirements and rules of interaction, which ensures compatibility between them at the level of interaction protocols.

If, for example, the following situation occurs: $\phi_j \neq \phi_k$, where $j \neq k$, $k \in \{1,2,\ldots,j-1,j+1,\ldots,n\}$, i $\forall k,l \in \{1,2,\ldots,j-1,j+1,\ldots,n\}$, $k \neq l: \phi_k \equiv \phi_l$ we say that the j -component of the system is incompatible.

4. STUDY OF MEANS OF VERIFICATION OF THE METHOD OF CONTROLLING THE COMPATIBILITY OF IOT DEVICES

The process of managing the compatibility of IoT devices based on the MQTT protocol is as follows:

– the paper presents a mathematical model for formalizing intercomponent interaction relationships, which allows automating the process of controlling component compatibility at the level of interaction protocols;

– based on this model, a method for automated control of the compatibility of IoT system components is presented. The method is based on

checking the compliance of the protocol implementation with its specification using the model checking method;

– IoT system components are considered in pairs. For each pair of components, a common set of state variables is formed based on the types of supported messages;

– for each component, a formal specification of the interaction protocol is synthesized based on the generated set of state variables;

– formal specifications are verified by model checking in an automated mode;

– the results of the model check for both specifications are compared by the parameters of the number of states and traversal depth. If these parameters match, the components are considered compatible at the level of interaction protocols.

The proposed method and model are validated using the IoTSim-Edge software simulator, which allows simulating various scenarios of interaction in the IoT network.

In the context of IoT research and development, it is important to recognize the impact of cloud and edge computing integration on system performance. The model and method we have developed is proposed to be integrated into IoTSim-Edge, which provides a unique opportunity to simulate the interaction between cloud services and edge devices. This allows not only to reproduce real-world usage scenarios, but also to analyze the impact of different data management strategies on overall system performance.

Fig. 1 shows the architecture of IoT-Edge computing. Sensor nodes will collect information about the environment using sensors and send information for processing and storage. Actuators will be activated based on data analysis. The communication layer is responsible for transferring data from IoT devices, peripherals, and the cloud.

The next level is for the network infrastructure, which consists of various types of peripheral devices, such as Arduino, Raspberry Pi. These devices can be accessed transparently using various types of virtualization and containerization mechanisms. It provides the infrastructure to deploy the raw data generated by the sensor nodes. In many cases, when the edge is active enough to process the data, it does not need to send the data to the cloud for further processing. Finally, the result is sent back to the actuator to perform a specific action.

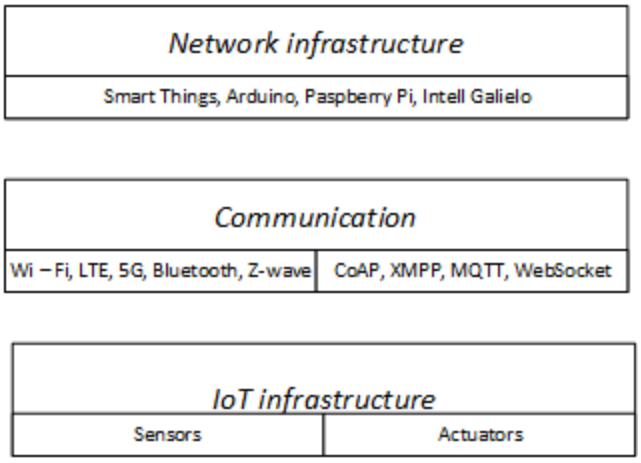

Fig. 1. **Architecture of the modeling environment**
Source: **compiled by the authors**

Therefore, a simulation framework such as the IoTSim-Edge simulator that supports the deployment of an application that evaluates the performance of different methods, scenarios under different conditions is well suited for method and interoperability model validation. In addition, evaluating methods under different scenarios and conditions can be done cost-effectively in a simulation environment.

The architecture of the proposed simulator consists of several levels [10].

Fig. 2 shows the elements that were selected for the experiment. Here is a brief description of each.

IoTSim-Edge, developed on the basis of CloudSim, improves its functionality for modeling IoT edge infrastructure [30]. It integrates a variety of IoT devices that produce data and respond to events, and supports the processing of this data at the edge for faster response using devices such as Raspberry Pi [31]. A central control layer manages the data processing, and flexible configuration management via GUI allows for system customization. Key controls include policies, mobility, quality of service (QoS), and security protocols, with an emphasis on standardized communication protocols such as MQTT and CoAP to ensure interoperability between components.

The IoTSim-Edge software is freely distributed and open source. A software module has been developed for it that integrates into the IoTSim-Edge system, which implements the Temporal Logic of Action (TLA+) model and a method of interoperability control for IoT devices.

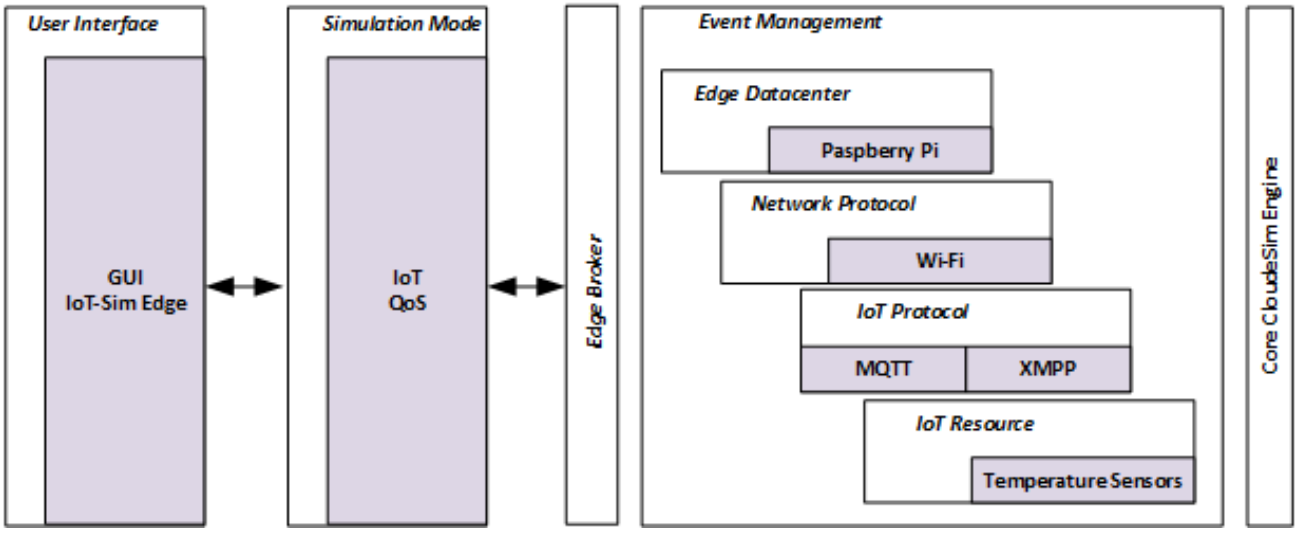

Fig. 2. **Parameters of the study**
Source: **compiled by the authors**

5. CHECKING THE MODEL FOR ADEQUACY

To describe the proposed approach, let's consider the scenario of the second QoS layer of the MQTT protocol.

There are four types of messages. The set of state variables representing message types is formed as follows:

$$V = \{v_1, v_2, v_3, v_4\}, \qquad (4)$$

where $v_1 \in V$ is the message "publish QoS 2"; $v_2 \in V$ is a PUBREC; $v_3 \in V$ is PUBREL message; $v_4 \in V$ is PUBCOMP message. The sequence diagram is shown in Fig. 3.

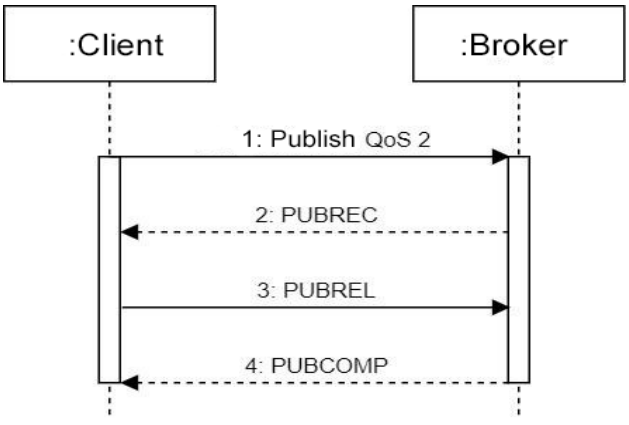

Fig. 3. **Sequence diagram of the QoS 2 scenario**
Source: **compiled by the authors**

In this case, the format of messages between devices looks like the one shown in Fig. 4.

```
"iotDeviceEntitites":[
    {
        "mobilityEntity":{
            "movable":  {
                "x":0.0,
                "y":0.0,
                "z":0.0
                        }
                        }
    },
    "assignmantId"^1,
    "iotClassName":"org.edge.core.iot.TemperatureSensor",
    "iotType":"environmental",
    "name":"temperature",
    "complexOfDataPackage":1,
    "networkModelEntity";{
        "networkType":"wifi",
        "communicationProtocol": "mqtt",
        "versionProtocol":"v.2",
        "messageProtocol":1
                        }
    "numberOfEntity":3
    }
```

Fig. 4. **Messages from the device**
Source: **compiled by the authors**

For the experiment, we modeled a network of different temperature sensors in the amount of 10 to 50 units. Each device had the same settings, only the protocol versions differed (Fig. 5).

Compatibility calculation formula:

$$Compatibility = \frac{N\ pos}{N\ sum}, \qquad (5)$$

where $Npos$ is the number of compatible devices; $N\ sum$ is the total number of selected devices.

Analyzing the chart, we see the following for the MQTT protocol:

– compatibility increases when the number of devices increases from 10 to 20;

– peak compatibility is observed on about 30 devices, where it reaches just over 90 %;

– beyond 30 devices, compatibility decreases dramatically, suggesting that MQTT may have limitations in dealing with more devices in this context.

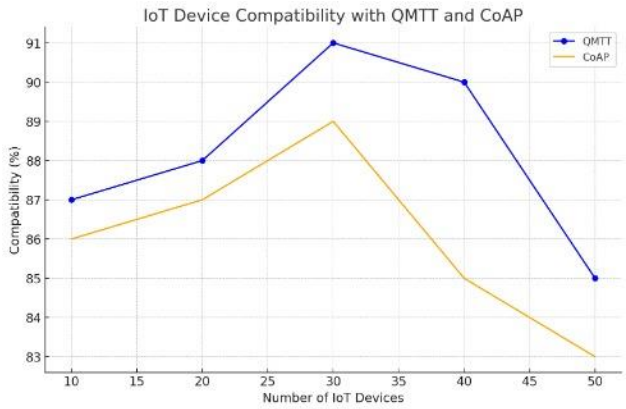

Fig. 5. **Results of the study of device compatibility for MQTT and CoAP protocols**
Source: **compiled by the authors**

The following conclusions can be drawn for the CoAP protocol:

– compatibility is relatively lower than MQTT and does not show a significant increase with the number of devices;

– the highest level of compatibility for CoAP is approximately 0 devices, and the percentage is just below 88 %;

– similar to MQTT, there is a sharp decline in compatibility as the number of devices continues to increase, which becomes more noticeable after 30 devices.

This pattern suggests that while both protocols work well up to a certain number of devices, their ability to maintain a high level of interoperability decreases as more devices are added to the network. The steeper decline for CoAP compared to MQTT may indicate that CoAP is less suitable for networks with a large number of devices.

Based on the results of the analysis, the relevance of solving the problem of ensuring the compatibility of IoT system components based on the proposed method and model is established and substantiated.

CONCLUSIONS

The method and model proposed in this article for controlling the compatibility of IoT system components are based on the use of temporal logic of actions (TLA+) and the PlusCal toolkit. The approach is to create an initial protocol model using PlusCal, where each label represents a separate property or aspect of the system's behavior. Then, based on this initial model, a detailed model is synthesized in TLA+ using the TLA Toolbox tools.

The adequacy of this model was verified by analyzing the MQTT protocol specification, comparing the number of states and traversal depths in the Kripke structure underlying the model. Specifically, the model contained 9 states with a traversal depth of 8.

The study of the method and model for controlling the compatibility of IoT system components described in this paper included the following steps:

– it was found that most approaches focus on ensuring the integrity of data exchanged between system components through the control of interaction protocols. However, the issue of consistency of interaction between components from the point of view of analyzing the interaction protocol has not been considered;

– an approach is developed that includes software compatibility testing using temporal logic of actions (TLA+). The algorithmic language PlusCal is used to simplify the model implementation;

– describes a method of compatibility verification based on standardization of IoT device data transfer protocols;

– a software module has been developed that integrates into the IoTSim-Edge system, which implements the Temporal Logic of Action (TLA+) model for IoT device interoperability;

– model checking confirms the correctness of the model and method implementation based on the MQTT protocol, emphasizing the consistency of the interaction of components.

Thus, the developed approach covers the verification of the software compatibility of IoT system components, taking into account the aspects of interaction at the level of interaction protocols.

The IoTSim-Edge software application was used for the experimental study. For the experiment, we modeled a network of different temperature sensors in the amount of 10 to 50 units. Each device had the same settings, only the protocol versions differed. The MQTT compatibility improves as the number of devices increases from 10 to 20, peaks at about 30 devices with just over 90 %, and then drops sharply, indicating the limitations of MQTT in large networks. Conversely, CoAP interoperability, which is lower than MQTT, does not increase significantly with more devices and drops noticeably after 30 devices, indicating that CoAP may be less suitable for large networks. These trends emphasize the need to address IoT interoperability issues, confirming the proposed method and the relevance of the model.

REFERENCES

1. Bour, G., Bosco, C., Ugarelli, R. & Jaatun, M. G. "Water-Tight IoT-just add security". *Journal of Cybersecurity and Privacy*. 2023; 3 (1): 76–94. DOI: https://doi.org/10.3390/jcp3010006.

2. Dirin, A. & Saballe, C. A. "Machine learning models to predict students' study path selection". *International Journal of Interactive Mobile Technologies (iJIM).* 2022; 16 (01): 158–183. DOI: https://doi.org/:10.3991/ijim.v16i01.20121.

3. Gopika Rajan, J. & Ganesh, R. S. "Hardware based data security techniques in Internet of Things: a review". *3rd International Conference on Smart Electronics and Communication (ICOSEC)*. 2022. p. 408–413. DOI: https://doi.org/10.1109/icosec54921.2022.9952021.

4. Shaporin, R., Hodovychenko, M. & Melnyk, R. "Research of LoRaWAN productivity performance models for building IoT networks". *Herald of Advanced Information Technology*. 2022; 5 (2); 123–132. DOI: https://doi.org/10.15276/hait.05.2022.10.

5. Gopika, D. & Panjanathan, R. "Energy efficient routing protocols for WSN based IoT applications: a review". *Materials Today: Proceedings*. 2020. p. 214–278. DOI: https://doi.org/10.1016/j.matpr.2020.10.137.

6. Gnusov, Y. V., Klimushyn, P. S., Kolisnyk, T. P. & Mozhaiev, M. O. "Analysis of systems of modeling of microcontrollers with additional modules of cryptographic information protection". *Bulletin of National Technical University "KhPI". Series: System Analysis/ Control and Information Technologies/* 2020; (1 (3)): 79–84. DOI: https://doi.org/10.20998/2079-0023.2020.01.14.

7. Klimushyn, P., Solianyk, T., Mozhaev, O., Nosov, V., Kolisnyk, T. & Yanov, V. "Hardware support procedures for asymmetric authentication of the internet of things". *Innovative Technologies and Scientific Solutions for Industries.* 2021; 4 (18): 31–39. DOI: https://doi.org/10.30837/itssi.2021.18.031.

8. Dung, L. M., Truong, Q. B., Nguyen, H. D. & Lam, V. V. "Application of IoT technology on control system and monitoring for grown in greenhouse". *CTU Journal of Innovation and Sustainable Development*, 2023; 15 (ISDS): 76–82. DOI: https://doi.org/10.22144/ctujoisd.2023.037.

9. Lam, A. N. & Haugen, O. "Supporting IoT semantic interoperability with autonomic computing". *Industrial Cyber-Physical Systems*. 2018. 1–18. DOI: https://doi.org/10.1109/icphys.2018.8390803.

10. Lam, A. N., & Haugen, O. "Applying semantics into Service-oriented IoT Framework". *IEEE 17th International Conference on Industrial Informatics.* 2019. p. 206–213. DOI: https://doi.org/10.1109/indin41052.2019.8972295.

11. Lam, A. N., Haugen, O. & Delsing, J. "Dynamical orchestration and configuration services in industrial iot systems: an autonomic approach". *IEEE Open Journal of the Industrial Electronics Society*. 2022; 3: 128–145. DOI: https://doi.org/10.1109/ojies.2022.3149093.

12. Zhang, C. "Intelligent Internet of things service based on artificial intelligence technology". *IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering*. Nanchang: China. 2021. p. 731–734. DOI: https://doi.org/10.1109/ICBAIE52039.2021.9390061.

13. Alturki, R., Gay, V., Awan, N., Alshehri, M., J. M. & Ur, A." Privacy, security and usability for IoT-enabled weight loss apps". *International Journal of Advanced Computer Science and Applications.* 2020; 11 (4): 1–34. DOI: https://doi.org/10.14569/ijacsa.2020.0110435.

14. Afsarimanesh, N., Mukhopadhyay, S. C. & Kruger, M. "IoT-Enabled microcontroller-based system". *Electrochemical Biosensor: Point-of-Care for Early Detection of Bone Loss. Cham: Springer International Publishing.* 2020. p. 93–103. DOI: https://doi.org/10.1007/978-3-030-03706-2_6.

15. Yang, Y. "Multi-tier computing networks for intelligent IoT". *Nature Electronics.* 2019; 2 (1): 4–5. DOI: https://doi.org/10.1038/s41928-018-0195-9.

16. Yang, P. & Xu, L. "The Internet of Things (IoT): Informatics methods for IoT-enabled health care". *Journal of Biomedical Informatics*. 2018; 87: 154–156. DOI: https://doi.org/10.1016/j.jbi.2018.10.006.

17. Yang, Y., Luo, X., Chu, X. & Zhou, M.-T. "IoT technologies and applications". *Fog-Enabled Intelligent IoT Systems. Cham: Springer International Publishing.* 2019. p. 1–37. DOI: https://doi.org/10.1007/978-3-030-23185-9_1.

18. Conceicao, A. A. et al. "Internet of things environment automation: A smart lab practical approach". *2nd International Conference on Information Technology and Education.* Malang: Indonesia. 2022. p. 1–6. DOI: https://doi.org/10.1109/ICITE54466.2022.9759899.

19. Yang, Y. “Problems and countermeasures of coal mine engineering management under the background of big data and IoT”. *Wireless Communications and Mobile Computing*. 2022. p. 1–7. DOI: https://doi.org/10.1155/2022/1725741.

20. Zhang, Y. & Chen, J.-L. “Declarative construction of distributed event-driven IoT services based on iot resource models”. *IEEE Transactions on Services Computing*. 2019; 1: 125–140. DOI: https://doi.org/10.1109/tsc.2017.2782794.

21. Desbiens, F. “MQTT.” *Building Enterprise IoT Solutions with Eclipse IoT Technologies Berkeley, CA: Apress*. 2022. p. 67–101. DOI: https://doi.org/10.1007/978-1-4842-8882-5_4.

22. Korzukhin, S. V., Khaydarova, R. R. & Shmatkov, V. N. “Configurable IoT devices based on ESP8266 SoC system and MQTT protocol”. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*. 2020; 20 (5): 722–728. DOI: https://doi.org/10.17586/2226-1494-2020-20-5-722-728.

23. Desbiens, F. “CoAP.” *Building Enterprise IoT Solutions with Eclipse IoT Technologies. Berkeley, CA: Apress*. 2023. p. 25–44. DOI: https://doi.org/10.1007/978-1-4842-8882-5_2.

24. Lee, C. & Fumagalli, A. “Internet of Things Security – Multilayered method for end to end data communications over cellular networks”. *IEEE 5th World Forum on Internet of Things (WF-IoT)*. Limerick: Ireland. 2019. p. 24–28. DOI: https://doi.org/10.1109/WF-IoT.2019.8767227.

25. Zeng, X., Garg, S. K., Strazdins, P., Jayaraman, P. P., Georgakopoulos, D. & Ranjan, R. “IOTSim: A simulator for analyzing IoT applications”. *Journal of Systems Architecture*. 2017; 72: 93–107. DOI: https://doi.org/10.1016/j.sysarc.2016.06.008.

26. Timenko, A. V. “On the aspects of IoT protocols specification and checking”. *Shipbuilding & Marine Infrastructure*. 2019; 2 (12): 35–41. DOI: https://doi.org/10.15589/smi2019.2(12).4.

27.Mazur, D. S., Timenko, A. V., Kudermetov, R. K. & Shkarupylo, V. V. “Approach to open flow-compatible switches implementation on the basis of raspberry pi” . *Scientific Notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences*. 2019; 6 (1): 112–116. DOI: https://doi.org/10.32838/2663-5941/2019.6-1/20.

28. Jha, D. N., Alwasel, K., Alshoshan, A., Huang, X., Naha, R. K., Battula, S. K. & Ranjan, R. “IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments”. *Software: Practice and Experience*. 2020; 50 (6): 844–867. DOI: https://doi.org/10.1002/spe.2787.

29. Al-Rubaie, N. R. O., Kamel, R. N. N. & Alshemari, R. M. “Simulating fog computing in OMNeT++”. *Bulletin of Electrical Engineering and Informatics*. 2023; 12 (2): 979–986. DOI: https://doi.org/10.11591/eei.v12i2.4201.

30. Kulykovska, N., Timenko, A., Hrushko, S. & Ilyashenko, M. “A semantic chatbot for Internet of things management”. *IEEE 9th International Conference on Problems of Info Communications, Science and Technology (PIC S&T)*. 2022. p. 246–250. DOI: https://doi.org/10.1109/picst57299.2022.10238683.

31. Beskorovainyi, V. V., Petryshyn, L. B. & Honcharenko, V. O. “Mathematical models of a multi-criteria problem of reengineering topological structures of ecological monitoring networks”. *Applied Aspects of Information Technology*. 2021; 5 (1): 11–24. DOI: https://doi.org/10.15276/aait.05.2022.1.

Conflicts of Interest: The authors declare no conflict of interest

Received 22.12.2023
Received after revision 04.03.2024
Accepted 15.03.2024

DOI: https://doi.org/10.15276/hait.07.2024.4
УДК 004.7

Дослідження методу контролю сумісності пристроїв Інтернету речей на основі протоколу прикладного рівня MQTT

Тіменко Артур Валентинович[1)]
ORCID: https://orcid.org/0000-0002-7871-4543; timenko.artur@gmail.com. Scopus Author ID: 56503994400
Шкарупило Вадим Вікторович[2)]
ORCID: https://orcid.org/0000-0002-0523-8910; shkarupylo.vadym@nubip.edu.ua. Scopus Author ID: 57189326576

Куликовська Наталія Анатоліївна[1)]
ORCID: https://orcid.org/0000-0003-4691-5102; natalya.gontar@gmail.com. Scopus Author ID: 57208667683
Грушко Світлана Сергіївна[1)]
ORCID: https://orcid.org/0000-0002-0064-408X; grushko_ss@i.ua. Scopus Author ID: 57202232710
[1)] Національний університет «Запорізька політехніка», вул. Жуковського, 64. Запоріжжя, 69063, Україна
[2)] Національний університет біоресурсів і природокористування України, вул. Героїв Оборони, 15. Київ, 03041, Україна

АНОТАЦІЯ

На тлі стрімкого розвитку Інтернету речей та його впливу на різноманітні сфери життя, забезпечення сумісності між різними компонентами систем стає актуальним завданням. Особливо важливим це стає у контексті розробки та інтеграції систем Інтернету речей з високим рівнем різноманіття та динамічності. У цій статті ми розглядаємо проблему сумісності компонентів Інтернету речей, акцентуючи увагу на протоколах прикладного рівня, які є ключовими у забезпеченні міжкомпонентної взаємодії. Основна мета статті полягає у розробці та валідації моделі, яка дозволить оптимізувати процеси взаємодії між компонентами систем з урахуванням специфіки протоколів. Запропоновано модель, яка базується на використанні темпоральної логіки дій, що забезпечує формальну верифікацію взаємодій між компонентами та дозволяє виявляти потенційні проблеми сумісності на різних етапах розробки. На базі цієї моделі розроблено метод автоматизованого контролю сумісності компонентів системи. Метод базується на перевірці відповідності реалізації протоколу його специфікації за допомогою методу перевірки на моделі. Компоненти системи Інтернет речей розглядаються попарно. Для кожної пари компонентів формується загальна множина змінних стану на основі типів підтримуваних повідомлень. Для кожного компонента синтезується формальна специфікація протоколу взаємодії на базі сформованої множини змінних стану. Ці формальні специфікації перевіряються методом перевірки на моделі в автоматизованому режимі. Результати перевірки на моделі для обох специфікацій співставляються за параметрами кількості станів і глибини обходу. Якщо ці параметри співпадають, компоненти вважаються сумісними на рівні протоколів взаємодії.. Розроблена модель була перевірена за допомогою програмного симулятора, який дозволяє моделювати різноманітні сценарії взаємодії в мережі Інтернету речей. Результати експериментів демонструють ефективність запропонованої методології у підвищенні рівня сумісності між компонентами системи, що в свою чергу знижує ризики втрати даних та забезпечує стабільність роботи систем Інтернету речей. Завдяки глибокому аналізу та розробці спеціалізованих методів та інструментів, це дослідження вносить значний вклад у розвиток теоретичних та практичних аспектів забезпечення сумісності. Однак, для подальшого підвищення точності та універсальності моделі, рекомендується проведення додаткових емпіричних досліджень з більшим набором даних.

Ключові слова: Інтернет речей; сумісність пристроїв; протокол MQTT; темпоральна логіка дій; моделювання взаємодії

ABOUT THE AUTHORS

Artur V. Timenko - Senior Lector of the Department of Computer Systems and Networks, National University “Zaporizhzhia Polytechnic”. 64, Zhukovsky Str. Zaporizhzhia, 69063, Ukraine
ORCID: https://orcid.org/: 0000-0002-7871-4543; timenko.artur@gmail.com. Scopus Author ID: 56503994400
***Research field*:** Development of IoT projects; computer networks

Тіменко Артур Валентинович - старший викладач кафедри Комп'ютерних систем і мереж. Національний університет «Запорізька політехніка», вул. Жуковського, 64. Запоріжжя, 69063, Україна

Vadym V. Shkarupylo - PhD, associate professor of Computer Systems, Networks and Cybersecurity Department, Kyiv National University of Life and Environmental Sciences of Ukraine. 15, Heroyiv Oborony Str. Kyiv, 03041, Ukraine
ORCID: https://orcid.org/0000-0002-0523-8910; shkarupylo.vadym@nubip.edu.ua. Scopus Author ID: 57189326576
***Research field*:** Formal methods; safety critical systems

Шкарупило Вадим Вікторович - канд. техн. наук, доцент кафедри Комп’ютерних систем, мереж та кібербезпеки Національного університету біоресурсів і природокористування України, вул. Героїв Оборони, 15. Київ, 03041, Україна

Nataliia A Kulykovska - Senior Lector of the Department of Computer Systems and Networks. National University “Zaporizhzhia Polytechnic”, 64, Zhukovsky Str. Zaporizhzhia, 69063, Ukraine
ORCID: https://orcid.org/0000-0003-4691-5102; natalya.gontar@gmail.com. Scopus Author ID: 57208667683
***Research field*:** Distributed computer systems; semantic web; knowledge engineering, IoT

Куликовська Наталія Анатоліївна - старший викладач кафедри Комп’ютерних систем і мереж. Національний університет «Запорізька політехніка», вул. Жуковського, 64. Запоріжжя, 69063,Україна

Svitlana S. Hrushko - PhD, associate professor of the Department of Computer Systems and Networks. National University “Zaporizhzhia Polytechnic”, 64, Zhukovsky Str. Zaporizhzhia, 69063, Ukraine
ORCID: https://orcid.org/0000-0002-0064-408X; grushko_ss@i.ua. Scopus Author ID: 57202232710
***Research field*:** Design of FPGA control devices; computer system interfaces

Грушко Світлана Сергіївна - канд. техн. наук, доцент кафедри Комп’ютерних систем та мереж. Запорізький національний університет «Запорізька політехніка», вул. Жуковського, 64. Запоріжжя, 69063, Україна