MÜHENDİSLİK FAKÜLTESİ

Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi
International Journal of Engineering Research and Development

UMAGD, (2020) 12(1), 317-324.

10.29137/umagd.654056

Cilt/Volume:12 Sayı/Issue:1 Ocak/January 2020

Araştırma Makalesi / Research Article

IoT based Smart Home Testbed using MQTT Communication Protocol

Fikret Yalçınkaya[1], Hüseyin Aydilek[1], Mustafa Yasin Erten[*1], Nihat İnanç[1]

[1]Kırıkkale University, Department of Electrical & Electronics Engineering, 71450, Kırıkkale, TURKEY

***Başvuru/Received**: 04/11/2019* ***Kabul / Accepted**: 11/01/2020* ***Çevrimiçi Basım / Published Online:** 13/01/2020*

***Son Versiyon/Final Version**: 31/01/2020*

Abstract

Technology is developing day by day in the world. In addition, developing technologies bring innovations and conveniences to all areas of life. However, ensuring the continuity of these innovations brought by technology reveals different problems. Smart home approach, which increases the quality of human living areas, is one of the most popular working subjects of recent times. In a smart home, built with the Internet of Things technology products, it is very important that the sensors and control devices communicate in a safer way and work in a coordinated manner to ensure the ecosystem's continuity in a safer way. In this study, an IoT based smart home testbed was realized by using MQTT communication protocol which one of the most used IoT communication protocols. With the developed system, the control of the smart home and the operating performance of the system were controlled with the mobile application. The results obtained in the light of the data provided by the test show that the system developed with the MQTT communication protocol can successfully ensure data flow and control in Smart Home applications.

Key Words

"Smart Home, IoT, MQTT, Testbed"

***Corresponding Author:** mustafaerten@kku.edu.tr*

1. Introduction

Technology is developing rapidly on a daily basis. Together with the developing technology, various technological devices that provide innovation and convenience come into our lives. Automation systems, free of human intervention, have started to take place in human life regularly with the development of technology.

Mankind has needed shelter starting with the creation and has begun to create spaces associated with it. Natural environments forced human to do that and, mankind began to construct their own structures that show the characteristics of the era in which they live and their social and cultural change. Nowadays, the concept of "smart home" has started to emerge with the dominant effect of modern-era technology. With smart home automation concept, it is aimed to provide both energy-time efficiency as well as security, comfort and saving benefits for housing needs.

Smart houses, the first example encountered with the Push Button Mansion in the 1950s, continue to gain various features over time. Push Button Mansion was the first smart home application, it had many features such as windows that close when it rains, lighting that opens automatically to the bell press, garage door that can open and close according to the driving path, fire and burglar alarm systems (Railton, 1950). Studies in recent years are not only about adding features to smart homes, but also energy and waste management (Han et al., 2014), (Zhou et al., 2016), (Anvari-Moghaddam et al., 2016), system security (Kumar, 2014), (Komninos, 2014), (Jacobsson et al., 2016), monitoring (Liu et al, 2016), (Vanus et al., 2017), application development (Aminikhanghahi et al., 2018), (Feng et al., 2017), (Juwan et al., 2019) and the operation of new communication protocols on the system.

Smart home kind of automation can manage our spaces from a single point and gain (add) various security features with the help of smart sensors and smart home assistants used inside or outside the house. All devices and sensors used in the smart home applications are controlled by the master home automation controller, generally called a smart home system. All devices within the smart home system should communicate with each other, by definition, and with the main controller when necessary. The communication of smart devices with each other over a network revealed the concept of Internet of Things (IoT).

IoT can be defined as a network system that allows electronic devices to communicate with each other using various communication protocols. It was first put forward in a presentation by Kevin Ashton in 1999 and it has become the present form with the developing technology. According to the research, the number of devices interacting with each other in 2003 was 500 million but today it has reached to 14 billion and it is estimated that this number will increase to 50 billion by 2020.

IoT systems started to work under many topics. These studies can be classified as the security of a system established with IoT, performances of communication protocols and applications that are included in various fields. (Alrawais et al., 2017), (Nurse et al., 2017), (Dabbagh & Rayes, 2019) have worked on IoT security (Lee et al., 2013), (Johnsen et al, 2013), (Luzuriaga et al ., 2014), (Luzuriaga et al., 2015). The literature review of the communication protocols used in IoT systems and the comparison of their performances is given in Table 1.

Table 1. Comparison Works about IoT Communication Protocols

Author	Application Area	Methods	Performance Comparision
Chen & Kunz, 2016	Healthcare Applications	MQTT, DDS	Latency: DDS (better) Bandwith Cons: MQTT (better)
Mun, D. H., Le Dinh, M., & Kwon, Y. W., 2016, June.	Comparison of protocols in terms of transmission time and energy consumption	CoAP, MQTT, MQTT-SN, Websocket	Web socket is better than other protocols
Kayal, P., & Perros, H. (2017, March.	Smart Parking	CoAP, MQTT, XMPP, Websocket	Lower Server Utilization: CoAP (Better) Multi-threading Support: XMPP (better) Higher Server Utilization: Web socket (better)
Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y., 2014, April.	Comparison	CoAP, MQTT	MQTT lower delay (at lower packet loss) MQTT higher delay (at higher packet loss)

In an IoT system, the communication of devices/things (Publisher / subscriber) with each other or with the broker can only be possible by using the same network protocol. The communication of the devices with each other raises different problems. Network protocol used for communication of devices; network traffic, the amount of data transmitted, the transmission of data, as well as problems with the limited hardware features of the IoT devices with effective hardware and power problems in the issue of important hardware. For this reason, various network protocols have been developed by companies to enable IoT devices to communicate more efficiently.

2. Material and Methods

2.1. IoT

Electronic devices have gained the ability to communicate and share data directly with each other or by creating an ecosystem with the development of technology in recent years, The ability of electronic devices to communicate with each other and to share data using various network structures and protocols has been named as the Internet of Things (IoT) over time.

All kinds of smart electronic devices with connection capacity can be a part of the IoT ecosystem. With the increasing number of these devices, we encounter applications in many fields such as smart home, smart environment, smart city, smart health services, smart agriculture and animal husbandry, and finally smart industrial and military applications.

In an IoT ecosystem built with constrained hardware features, the communication protocol has gained particular importance due to reasons, such as increased communication traffic, power consumption, insufficient processor, memory and storage capacity. For this reason, network protocols such as MQTT, CoAP, AMQP, SOAP have emerged.

2.2. MQTT

Message Queue Telemetry Transport (MQTT) is a message-based communication protocol introduced by IBM in 1999. It is the most common protocol used to exchange data between machines within the Internet of Things platform. This protocol is preferred because the size of the communication packets is small, easy to use and implement, and the network bandwidth is large.

Table 2. Properties Comparison of IoT Communication Protocols

Protocols	CoAP	HTTP	MQTT	WebSocket
Transport	UDP	TCP	TCP	TCP
Messaging	Request/ Response	Request/ Response	Publish/ Subscribe	Asynchronous
Community Support	IPSO, OMA, IETF	IETF, oneM2M	IBM	IETF
Delay	Fair	Fair	Good	Good
Cloud Support	Good	Good	Good	Fair
Power Usage	Good	Good	Fair	Good

MQTT is a protocol based on the publish / subscribe system. This system consists of Client, Broker and Topic. The client represents the devices which are connected to the network and sending and receiving data such as sensors, cell phones, and computers. An MQTT client can be both a publisher or a subscriber. The publisher is the source of the data on the system to be sent. A subscriber is a client who wants to receive messages from the publisher. A MQTT broker is a device that connects clients in the system. The broker is responsible for collecting and maintaining that data when the client connected to it publishes data and processing it properly. It then transmits this data to the subscribers connected to it. The broadcaster and subscriber do not need to work and know each other at the same time. The broker retrieves the data into the buffer in sequence when the client is offline and forwards it when the client is online. It has a queuing system in this way.

The MQTT has a three-stage control package to establish safer communication: a mandatory fixed header, an optional variable header, and a load. Optional fields often complicate the operation of the protocol. CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, SUBSCRIBE, SUBACK, and MQTT are some of the MQTT control packages that are exchanged between MQTT clients and the MQTT broker.

After a successful network has been established between the MQTT client and the MQTT broker, the control packages are exchanged between the client and the broker. The client that wants to connect to the MQTT broker sends a CONNECT packet specifying its identity, flags, protocol level, and other fields. The server confirms the client through the CONNACK packet with a return code indicating the connection status. This ensures a connection between the client and the broker.

If the client wants to send data as a publisher, it sends a PUBLISH packet to the broker. This package includes QoS transmission level, subject name, load capacity, etc. If the data to be sent is transmitted in QoS 0, the broker does not send any confirmation packet for the broadcast packet. If the data is transmitted to QoS 1, the broker confirms to the client that the packet has been broadcast with the PUBACK packet. In QoS 2, four packages are exchanged. The broker confirms that the PUBLISH package was received with the PUBREC package. The MQTT client then sends a request for the data that it wants to publish with a PUBREL packet. The broker then sends the fourth packet of PUBCOMP, confirming that the message is complete and published once on the given subject, indicating that the transaction is complete.

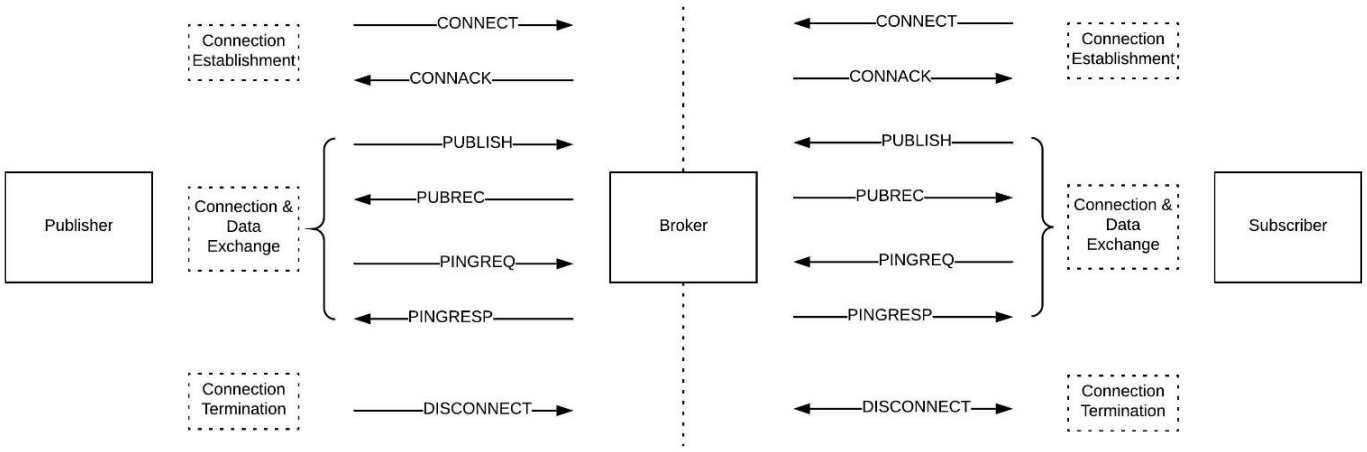

Figure 1. Structure of MQTT Communication

If the MQTT client wants to receive the published message, it sends the SUBSCRIBE packet to the broker. The broker confirms the subscription to the request with the SUBACK package. When the subscription is successful, messages on the specified subject are transmitted to the subscriber with the QoS transmission level. To unsubscribe a topic, the subscriber sends the UNSUBSCRIBE package to the broker and the broker confirms the cancellation of the subscription with the UNSUBACK package.
After a certain timeout period, the connection between the client and the broker is terminated. The client transmits the PINGREQ packet to the server to maintain the connection, indicating that it is active. The MQTT broker responds to the subscriber with the PINGRESP packet and ensures that the connection continues.

The MQTT client sends a DISCONNECT packet to the broker to terminate the connection. The broker does not send a reply packet to the subscriber in response to this packet, but the client then deletes all messages related to the subscriber and the subscriber is disconnected from the broker.

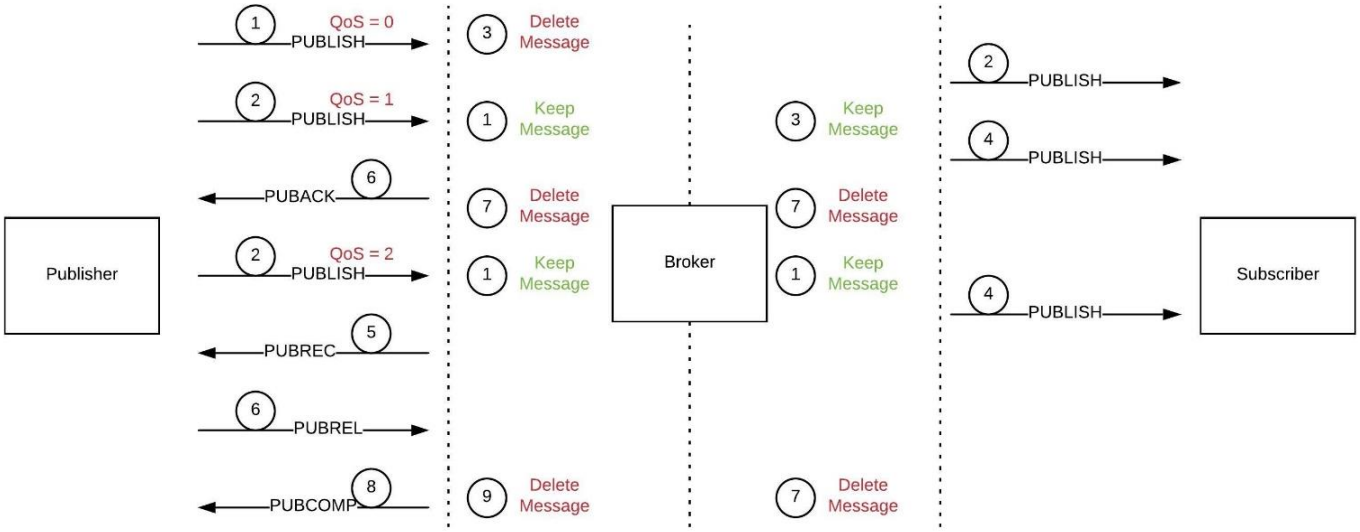

Figure 2. QoS Data Transportation

The MQTT uses QoS levels to transmit messages. QoS is a protocol based on a bilateral agreement of a message with respect to the assurance of data distribution. Although TCP / IP provides guaranteed data presentation, data loss may occur if a TCP connection is broken and transmitted messages are lost. Therefore, MQTT adds 3 QoS levels to the top of TCP:

a) QoS0 (Up to one time): At this QoS level, the message is sent up to one time and delivery of a message is not guaranteed.

b) QoS1 (At least once): At this QoS level, the data is sent at least once. It is possible to send a message more than once by setting the repetition of the message.

c) QoS2 (Exactly once): At this QoS level, the message is sent precisely once using the 4-way handshake system.

3. Developed Application Testbed

In this study, a smart home system test-bed based on IoT system architecture, where communication between things/devices is guaranteed according to MQTT protocols, is realized. The architecture of the system is given in Figure 3.

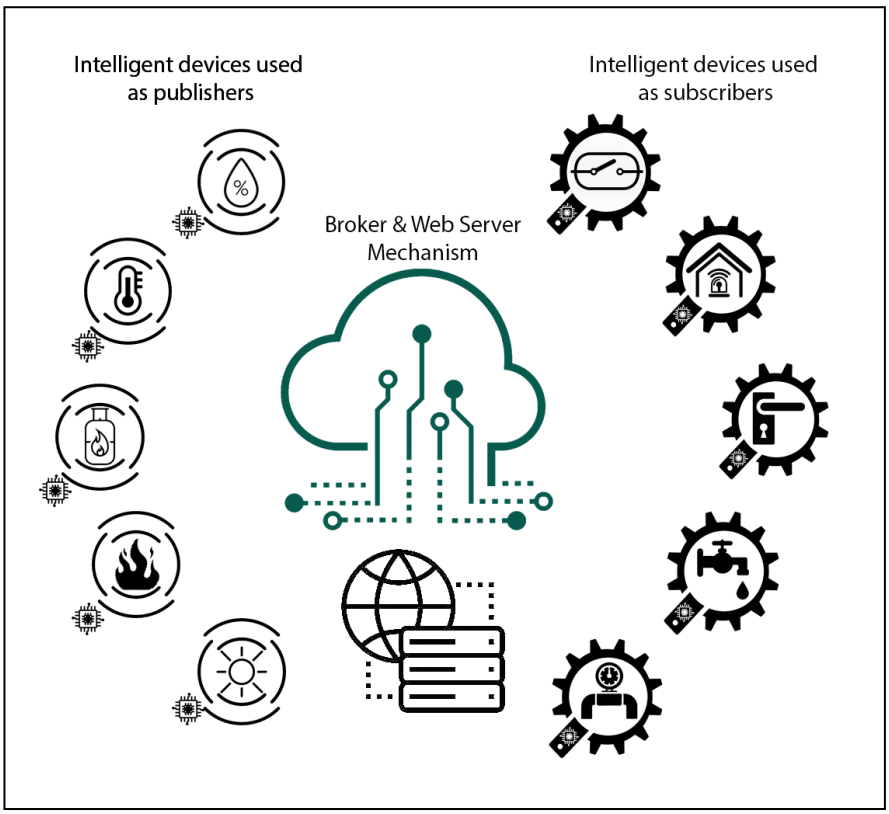

Figure 3. The Architecture of the System

The developed smart home system consists of sensors and control elements (Subscriber / Publisher) and the main central processing unit (Broker). In the developed system, PIR sensor is used for motion detection, MQ4 sensor for gas leak detection, DHT11 sensor for measuring heat and humidity, IR flame sensor for fire detection, photoresistor (LDR) for measuring light intensity and ACS712 for measuring current. Relay, alarm, door lock, water valve and gas valve are used as control elements.

3.1. Establishment of Broker

In the application environment, Raspberry Pi 3 single board computer with ARM microprocessor is used as the main processing unit. This card has been chosen due to its low power consumption, physical dimensions and ease of use. Debian based Raspbian operating system was installed as the operating system on the card. Apache web server is installed for Web Service and database. Following the general configuration of the operating system, the open source Mosquitto MQTT broker, developed by the Eclipse Foundation, was used to manage the MQTT message contents. Mosquitto has been preferred because of its open source structure, it allows for various modifications and add-ons as well as rich documentation. MQTT Server and Apache Web Server integration is provided by means of a developed program to ensure that each data from IoT environment is kept in the database with time tag, message tag and content. In addition, the developed program filters the data collected from IoT environment and sends the data that are determined to be significant at the QoS 2 level.

3.2. Establishment of Publisher

The data collected from the IoT environment via motion sensor, gas sensor, flame sensor, humidity sensor, temperature sensor, LDR and current sensor, which are called as publisher elements, are read with the help of NodeMCU microcontroller and converted to meaningful data and sent to the broker. The general working flow diagram of the publisher elements is given in Figure 4.

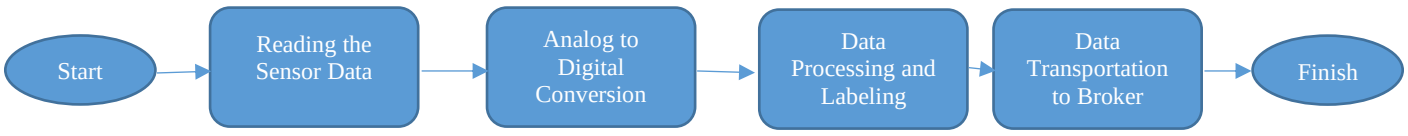

Figure 4. Publisher Systems Flowchart

3.3. Establishment of Subscriber

Relay, alarm, door lock, water valve and gas valve were used as Subscriber elements in the developed system. The NodeMCU microcontroller was used to analyze the data collected from the IoT ecosystem and to transmit the appropriate command to the relevant control elements. The general working flow diagram of the subscriber elements is given in Figure 5.

Figure 5. Subscriber Systems Flowchart

3.4. Development of Mobile Application

Today's technological infrastructure makes it easy to access and control many devices remotely. For a smart home system, remote monitoring and control is an important requirement. A mobile application has been developed to remotely monitor the data in the test environment and to control the control elements remotely. The developed mobile application is designed for two-way operation. If the mobile device is connected to the same Wi-Fi network as the smart home, the mobile application acts as a part of the IoT ecosystem and acts as both a publisher and a subscriber. If the mobile device accesses to the Internet through a different network, the device can control the ecosystem and examine the data via the web service. The mobile application was developed in accordance with Android mobile devices. The screenshots of the application are given in Figure 6.

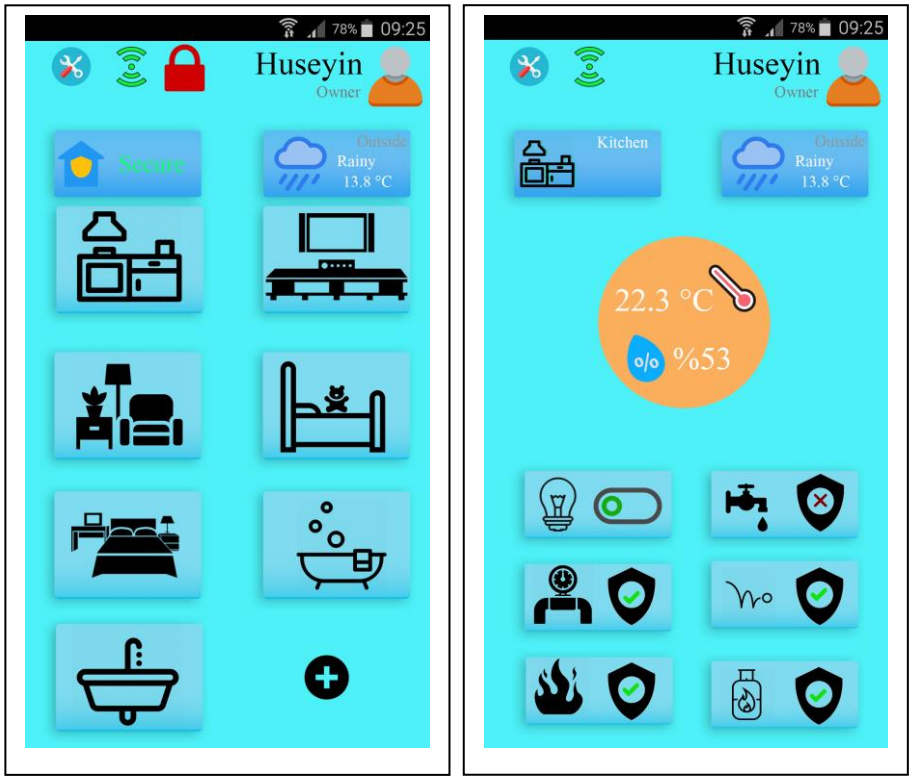

Figure 6. Mobile Application Interface

4. Results

The test environment was tested and run under various scenarios. The scenarios created are summarized below.

Scenario 1: Single Publisher & Single Subscriber are Active

In this scenario, the publisher and the subscriber first checked for the system work and performance for their own case. The system was then tested in cases where both the publisher and the subscriber were working simultaneously. In addition to this scenario, the mobile device was first tested alone by connecting to the same Wi-Fi network and connecting to a different network. The mobile device, which passed the tests without any problems, was tested by connecting to the system even in the case of both the publisher and the subscriber. In cases where there is only one publisher and one subscriber, the system works without any problems.

Scenario 2: Two Publishers & Two Subscribers are Active
In this scenario, the system was tested for two publishers and two participants actively working. The mobile application was tested in this scenario by connecting to both the same Wi-Fi network and different network. In cases where there were two publishers and two participants, the system worked without any problems.

Scenario 3: All Publishers & All Subscribers are Active
In this scenario, the status of all (seven) publishers and all (five) participants in the system is tested. In addition to this scenario, the mobile application is integrated into the system both by connecting to the same Wi-Fi network and over different Wi-Fi networks. While the mobile application was connected to the same Wi-Fi network, there was no problem in communication and control. However, when the mobile application is connected to a different Wi-Fi network, it is observed that there are no problems in the application monitoring functions but there are delays in the control functions.

5. Discussions

In this study, an IoT based smart home test environment was realized by using MQTT communication protocol. Various sensors and control elements were added to the IoT ecosystem and the performance of the MQTT communication protocol on different scenarios was tested. The developed mechanism has been tested in all scenarios in which all publishers and participants are connected and it has been observed that it works without any problem. The developed test environment experienced delay problems in control functions in the scenario where the mobile application was connected to a network other than the multiple subscriber and publisher. It has worked successfully in all other test scenarios and passed the tests successfully. As a result of the test application, it was observed that the MQTT communication protocol is suitable for an IoT smart home system.

It is thought that the mentioned problems can be overcome with the improvements to be made on the mobile application. In future studies, it is considered that studies on comparing MQTT and CoAP communication protocols and system scaling can be performed.

References

Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. IEEE Internet Computing, 21(2), 34-42.

Aminikhanghahi, S., Wang, T., & Cook, D. J. (2018). Real-time change point detection with application to smart home time series data. IEEE Transactions on Knowledge and Data Engineering, 31(5), 1010-1023.

Anvari-Moghaddam, A., Monsef, H., & Rahimi-Kian, A. (2014). Optimal smart home energy management considering energy saving and a comfortable lifestyle. IEEE Transactions on Smart Grid, 6(1), 324-332.

Chen, Y., & Kunz, T. (2016, April). Performance evaluation of IoT protocols under a constrained wireless access network. In 2016 IEEE International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT) (pp. 1-7).

Dabbagh, M., & Rayes, A. (2019). Internet of things security and privacy. In Internet of Things From Hype to Reality (pp. 211-238). Springer, Cham.

Han, J., Choi, C. S., Park, W. K., Lee, I., & Kim, S. H. (2014, January). Smart home energy management system including renewable energy based on ZigBee and PLC. In 2014 IEEE International Conference on Consumer Electronics (ICCE) (pp. 544-545).

Jacobsson, A., Boldt, M., & Carlsson, B. (2016). A risk analysis of a smart home automation system. Future Generation Computer Systems, 56, 719-733.

Johnsen, F. T., Bloebaum, T. H., Avlesen, M., Spjelkavik, S., & Vik, B. (2013, October). Evaluation of transport protocols for web services. In IEEE 2013 Military Communications and Information Systems Conference (pp. 1-6).

Kayal, P., & Perros, H. (2017, March). A comparison of IoT application layer protocols through a smart parking implementation. 2017, IEEE 20th Conference on Innovations in Clouds, Internet and Networks (ICIN) (pp. 331-336).

Komninos, N., Philippou, E., & Pitsillides, A. (2014). Survey in smart grid and smart home security: Issues, challenges and countermeasures. IEEE Communications Surveys & Tutorials, 16(4), 1933-1954.

Kumar, S. (2014). Ubiquitous smart home system using android application. arXiv preprint arXiv:1402.2114.

Lee, S., Kim, H., Hong, D. K., & Ju, H. (2013, January). Correlation analysis of MQTT loss and delay according to QoS level. In 2013 IEEE The International Conference on Information Networking (ICOIN) (pp. 714-717).

Liu, L., Stroulia, E., Nikolaidis, I., Miguel-Cruz, A., & Rincon, A. R. (2016). Smart homes and home health monitoring technologies for older adults: A systematic review. International journal of medical informatics, 91, 44-59.

Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2015, January). A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks. In 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC) (pp. 931-936).

Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2014, September). Testing AMQP protocol on unstable and mobile networks. In International Conference on Internet and Distributed Computing Systems (pp. 250-260). Springer, Cham.

Mun, D. H., Le Dinh, M., & Kwon, Y. W. (2016, June). An assessment of internet of things protocols for resource-constrained applications. In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 555-560).

Nurse, J. R., Creese, S., & De Roure, D. (2017). Security risk assessment in Internet of Things systems. IT Professional, 19(5), 20-26.

Railton, A. R. (1950). Push-Button Manor. Popular Mechanics, 6(94), 84-87.

Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014, April). Performance evaluation of MQTT and CoAP via a common middleware. In 2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP) (pp. 1-6).

Vanus, J., Belesova, J., Martinek, R., Nedoma, J., Fajkus, M., Bilik, P., & Zidek, J. (2017). Monitoring of the daily living activities in smart home care. Human-centric Computing and Information Sciences, 7(1), 30.

Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., & Wang, X. (2016). Smart home energy management systems: Concept, configurations, and scheduling strategies. Renewable and Sustainable Energy Reviews, 61, 30-40.