International Journal of Membrane Science and Technology, 2023, Vol. 10, No. 1, pp. 64-71

Design and Implementation of IoT Gateway with MQTT and IEC 61850 MMS Protocol

Soo-Yeon Oh[1], Yong-Joon Lee[2], Sung-Ho Hwang[3*]

[1,2,3]*Department of Electronics, Information and Communication Engineering, Kangwon National University, P.O. Box: 25913, Samcheok, South Korea.*

E-mail: shhwang@kangwon.ac.kr

Abstracts: With microgrids deployed in multiple areas, communication technologies that control devices and collect data play an important role. Microgrid environments are similar to the internet of things (IoT) environment, where multiple devices exchange information. Recent studies have attempted to utilize IoT protocols as communication protocols for microgrids. Studies have also begun on interworking IoT protocols with IEC 61850, an existing protocol for power utility. In this study, we designed an IoT gateway that maps message queuing telemetry transport (MQTT), an IoT protocol, to the IEC 61850 data model. We built a testbed and performed a verification using TShark to verify that MQTT and IEC 61850 MMS traffic are properly generated and mapped by the IoT gateway. We also used IEEE 1588 to perform time synchronization between Raspberry Pi units, which are IoT devices. Performance analysis was conducted by measuring the mean transfer time of the MQTT and IEC 61850 MMS traffic for various Ethernet speeds and Wi-Fi protocols.

Keywords: IEC 61850, IEC 61850 MMS, MQTT, IoT, Smart Grid, Micro Grid.

1. INTRODUCTION

Microgrids are increasingly being deployed in several regions and fields. They generate distributed energy resources, adjust the load, and store surplus power in an energy storage system (ESS) or store hydrogen generated through water electrolysis in a hydrogen tank.

IoT communication protocols are widespread and are highly associated with microgrids in that they prefer the Publisher-Subscriber paradigm. Among IoT protocols, the message queuing telemetry transport (MQTT) protocol was developed for embedded systems that use light central processing unit (CPU) load and memory.

IEC 61850 communication is fundamentally provided using Ethernet (Kanabar & Sidhu, 2011). However, the smart grid roadmap of the National Institute of Standards and Technology (NIST) includes Wi-Fi (EPRI, 2009). Recent research proves that Wi-Fi technology can be applied instead of Ethernet (Akmal, Hussain, Akhtar, Jabbar, Anwar, & Ullah, 2020; Cena, Bertolotti, Valenzano & Zunino, 2007).

In this study, we utilize MQTT in an IEC 61850-based grid. IEC 61850 services use the manufacturing message specification (MMS) protocol for client-server-based services, such as reports, logs, and general queries. In addition, the experiment is not conducted using computer-based simulation tools; instead, Cisco IE 4000 and Raspberry Pi, an IoT device, are used. We evaluate performance by measuring the mean transfer time of packets according to the protocol using IEEE 1588 time synchronization when MQTT and IEEE 61850 MMS are interworking.

2. RELATED WORKS

In this study, we selected MQTT among IoT protocols as the protocol for microgrids for the following reasons: One study compared MQTT and the hypertext transfer protocol (HTTP) and determined that MQTT is better for smart grid applications (Ullah et al., 2020). Another study compared MQTT and the extensible messaging and presence protocol (XMPP) in an IEC 61850-based microgrid communication architecture and selected the MQTT (Jun & Yang,2021).

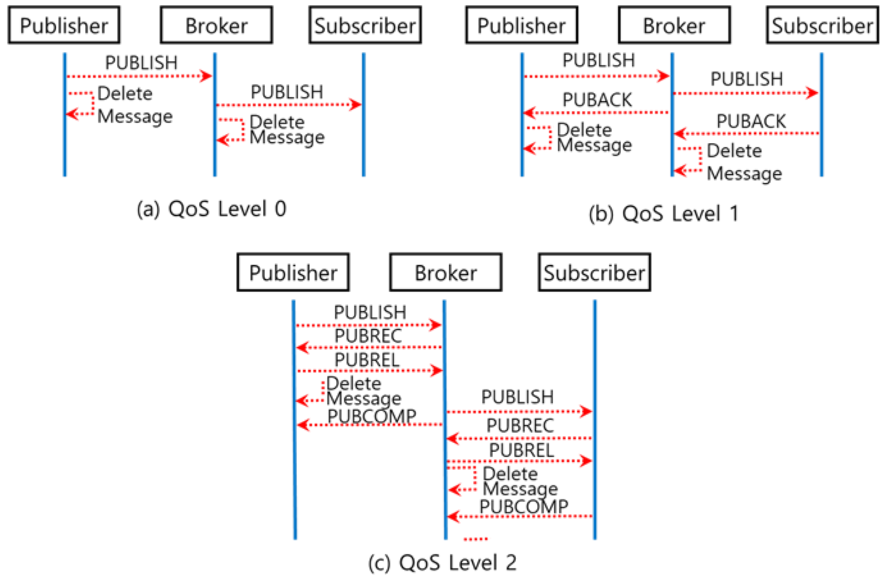

Figure 1. MQTT quality of service levels.

MQTT is a lightweight publisher–subscriber messaging protocol where messages can be published and received as a client through a broker. A client connects to a broker and subscribes to a specific topic. After connecting to the broker, the client publishes messages with a specific topic, with the broker functioning as the interface for connecting clients. Topics handle a hierarchical structure using a forward slash '/' as a delimiter. MQTT provides three levels of quality of service (QoS), as shown in Figure 1. In QoS level 0, the broker or client sends a message at most once and does not verify whether the message has been received. In QoS level 1, messages are delivered at least once, with messages delivered without verifying whether a connection has been established, such as performing a handshake. In QoS level 2, messages are delivered precisely once by performing a four-step handshake.

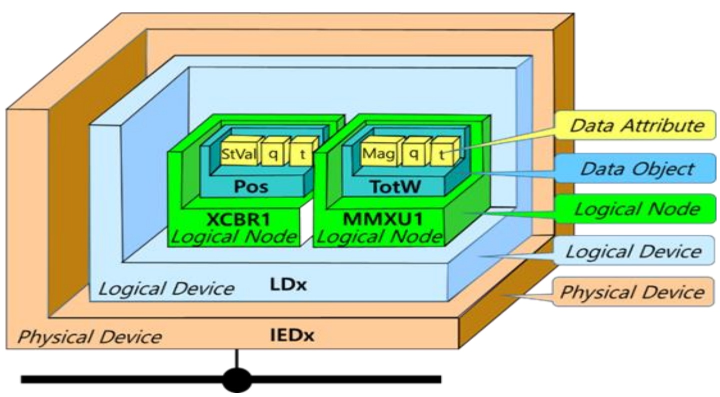

Figure 2. IEC 61850 data modeling.

Figure 2 shows the data model defined in IEC 61850. The class model corresponding to the information model has the physical device, logical device, logical nodes, and data objects. The physical device and logical device classes were defined for applying device implementation, while the logical node class defined all functions related to protection, control, and measurement. The data class was defined to define the data composing these logical nodes. Data class is an abstract class for defining the common data class (CDC) defined in Parts 7-3 of the IEC 61850 standard document. CDC corresponds to the data types comprising specific logical nodes related to functions.

3. SYSTEM DESIGN AND IMPLEMENTATION

In this study, we conducted an experiment in which the MQTT publisher transmits solar panel data, and the

MQTT subscriber of the IoT gateway maps them to an IEC 61850 data model. For example, the MMDC logical node sends the PV module's voltage, current, and watt. MMDC is defined in IEC 61850-7-420 and represents the measurement information of a DC system. The data objects (DO) in MMDC are not required and are all optional. In this study, the current, voltage, and watt of the MMDC logical node are used as the DOs. The CDC of all DOs is used in the measured value (MV), comprising attributes mag, q, and t, denoting magnitude, quality, and timestamp, respectively.

This study uses REUnit as the physical device, PV1 as the logical device, MMDC1 as the logical node, MV as the data object, and mag, q, and t as the data attributes.

In the MQTT publisher, REUnit/PV1/MMDC1 is used as a topic, Amp.mag.f, Amp.q, and Amp.t as the current attributes, Vol.mag.f, Vol.q, Vol.t as the voltage attributes, and Watt.mag.f, Watt.q, and Watt.t as the watt attributes.

Figure 3 shows the overall testbed. The MQTT publisher transmits the current, voltage, and watt values to the MQTT broker once per second. The MQTT broker sends these values to the MQTT subscriber of the IoT gateway and stores the values transmitted to the IEC 61850 MMS server. When the IEC 61850 MMS client requests a dataset from the IEC 61850 MMS server, the IEC 61850 MMS server provides the dataset.

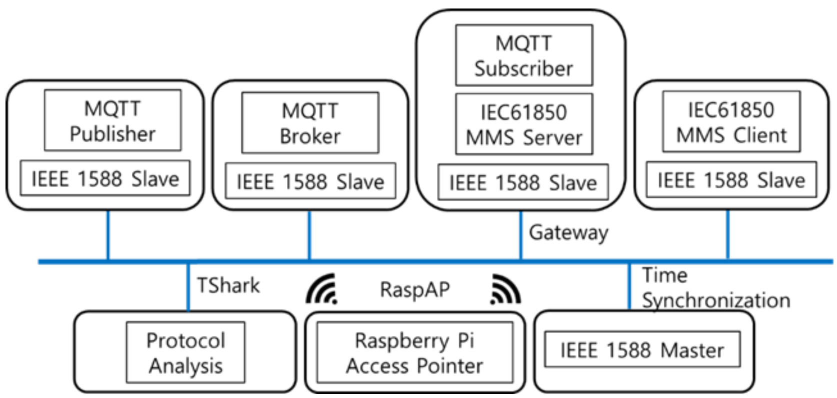

Figure 3. Functional diagram of testbed.

Raspberry pi access pointer (RaspAP) provides the access point function according to the Wi-Fi protocol. RaspAP can simply set up an access point (AP) for Debian-based devices, including Raspberry Pi, and can manage Wi-Fi. RaspAP provides network services, such as DHCP setting, OpenVPN, SSL, and security audit. RaspAP also provides the IEEE 802.11 wireless mode options for the provided hardware. As Raspberry Pi 4B is used in this study, IEEE 802.11b, IEEE 802.11g, and IEEE 802.11n are used as the wireless mode options at 2.4 GHz, and IEEE 802.11a and IEEE 802.11ac are used as the wireless mode options at 5 GHz.

TShark is an open-source network analyzer used for troubleshooting, analysis, software development, and training. TShark is a cross-platform software that runs on Linux, Unix, Raspbian, and Microsoft operating systems. In this study, TShark is used to capture and analyze the traffic generated in the testbed.

The IEEE 1588 master can synchronize all devices in the testbed within an error range of 1 µs or less.

Table 1 shows the specifications of the functions used in the testbed. The hardware of all devices uses Raspberry Pi 4B, with Raspbian used as the operating system.

Table 1. Testbed specification.

	Functions	Specification
MQTT Publisher	MQTT Publisher	Paho v1.3
MQTT Broker	MQTT Broker	Mosquitto v1.5.7
MQTT Subscriber	MQTT Subscriber	Paho v1.3
IEC 61850 MMS Server	IEC 61850 Server Library	LibIEC61850 v1.5.1
IEC 61850 MMS Client	IEC 61850 Client Library	LibIEC61850 v1.5.1
Raspberry Pi Access Pointer	RaspAP	RaspAP Version 2.8.6
Protocol Analysis	TShark	TShark 3.4.10
IEEE 1588 Master	Precision Time Protocol Daemon	Ptpd2 version 2.3.1

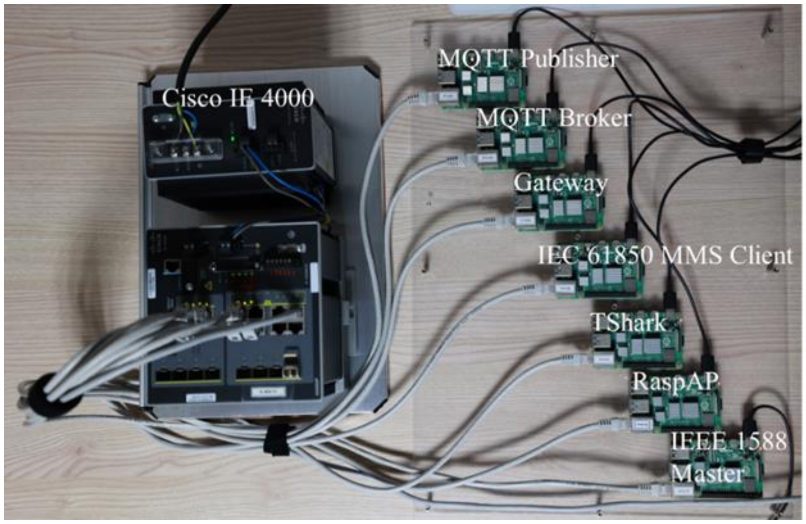

Figure 4. Hardware system.

Figure 4 shows the hardware system used in this study. One Raspberry Pi is used for the MQTT publisher, 1 Raspberry Pi for the MQTT broker, 1 Raspberry Pi for the IoT gateway's MQTT subscriber and IEC 16850 MMS server, and 1 Raspberry Pi for the IEC 16850 MMS client. In addition, 1 Raspberry Pi is used for TShark to analyze the MQTT and IEC 61850 traffic. Moreover, 1 Raspberry Pi is used for RaspAP, which is the Wi-Fi AP, and 1 Raspberry Pi is used for the IEEE 1588 precision time protocol (PTP) master for accurate time synchronization. Finally, 1 Cisco IE 4000 switch is used as the network switch.

Figure 5 shows the result of capturing an MQTT packet using TShark. It shows the packet sent by the MQTT publisher with an IP address of 192.168.1.241 to the broker with an IP address of 192.168.1.210. Figure 6 shows the result of capturing a packet sent by the MQTT Broker with an IP address of 192.168.1.210 to the MQTT subscriber with an IP address of 192.168.1.242.

```
> Frame 1681: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits) on interface eth0, id 0
> Ethernet II, Src: Raspberr_d8:36:01 (dc:a6:32:d8:36:01), Dst: Raspberr_7a:e9:72 (dc:a6:32:7a:e9:72)
> Internet Protocol Version 4, Src: 192.168.1.241, Dst: 192.168.1.210
> Transmission Control Protocol, Src Port: 58218, Dst Port: 1883, Seq: 3387, Ack: 5, Len: 106
v MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 104
    Topic Length: 16
    Topic: REUnit/PV1/MMDC1
    Message: 303033322033332e303030302031363639313137383438383830383336203020333332e30…
```

Figure 5. MQTT publisher capture.

```
> Frame 2570: 169 bytes on wire (1352 bits), 169 bytes captured (1352 bits) on interface eth0, id 0
> Ethernet II, Src: Raspberr_7a:e9:72 (dc:a6:32:7a:e9:72), Dst: Raspberr_d8:35:a4 (dc:a6:32:d8:35:a4)
> Internet Protocol Version 4, Src: 192.168.1.210, Dst: 192.168.1.242
> Transmission Control Protocol, Src Port: 1883, Dst Port: 37248, Seq: 129, Ack: 61, Len: 103
v MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 101
    Topic Length: 16
    Topic: REUnit/PV1/MMDC1
    Message: 3030303120322e3030303020313636393131373831373837343130352030202322e303030…
```

Figure 6. MQTT subscriber capture.

In Figure 7, the IEC 61850 MMS client with an IP address of 192.168.1.243 requests a dataset from the IEC 61850 MMS server with an IP address of 192.168.1.242. In Figure 8, the data received by the MQTT subscriber is mapped onto an IEC 61850 MMS packet, and the IEC 61850 MMS server sends it to the 61850 MMS client as a dataset.

We found out using TShark that packets are properly sent and received between the MQTT publisher and the IoT gateway's MQTT subscriber. In addition, the datasets were also properly sent and received between the IoT gateway's IEC 61850 MMS server and the IEC 61850 MMS client.

```
> Frame 2567: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface eth0, id 0
> Ethernet II, Src: Raspberr_d8:35:6b (dc:a6:32:d8:35:6b), Dst: Raspberr_d8:35:a4 (dc:a6:32:d8:35:a4)
> Internet Protocol Version 4, Src: 192.168.1.243, Dst: 192.168.1.242
> Transmission Control Protocol, Src Port: 45094, Dst Port: 102, Seq: 15351, Ack: 32146, Len: 69
> TPKT, Version: 3, Length: 69
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8823 OSI Presentation Protocol
v MMS
  v confirmed-RequestPDU
      invokeID: 221
    v confirmedServiceRequest: read (4)
      v read
          specificationWithResult: True
        v variableAccessSpecificatn: variableListName (1)
          v variableListName: domain-specific (1)
            v domain-specific
                domainId: REUnitPV
                itemId: MMDC1$AnalogueValues
```

Figure 7. IEC 61850 MMS client capture.

```
> Frame 2568: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface eth0, id 0
> Ethernet II, Src: Raspberr_d8:35:a4 (dc:a6:32:d8:35:a4), Dst: Raspberr_d8:35:6b (dc:a6:32:d8:35:6b)
> Internet Protocol Version 4, Src: 192.168.1.242, Dst: 192.168.1.243
> Transmission Control Protocol, Src Port: 102, Dst Port: 45094, Seq: 32146, Ack: 15420, Len: 148
> TPKT, Version: 3, Length: 148
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8823 OSI Presentation Protocol
v MMS
  v confirmed-ResponsePDU
      invokeID: 221
    v confirmedServiceResponse: read (4)
      v read
        v variableAccessSpecificatn: variableListName (1)
          v variableListName: domain-specific (1)
            v domain-specific
                domainId: REUnitPV
                itemId: MMDC1$AnalogueValues
        v listOfAccessResult: 3 items
          > AccessResult: success (1)
          > AccessResult: success (1)
          > AccessResult: success (1)
```

Figure 8. IEC 61850 MMS server capture.

4. PERFORMANCE EVALUATION

We used the IEEE 1588 protocol in the Raspberry Pi to evaluate the performance of the transmission of MQTT and IEC 61850 MMS packets. Figure 9 shows that the clock stabilizes for synchronization 300 seconds after a cold start. Therefore, in this study, MQTT and IEC 61850 MMS packets were generated 300 seconds after a cold boot.

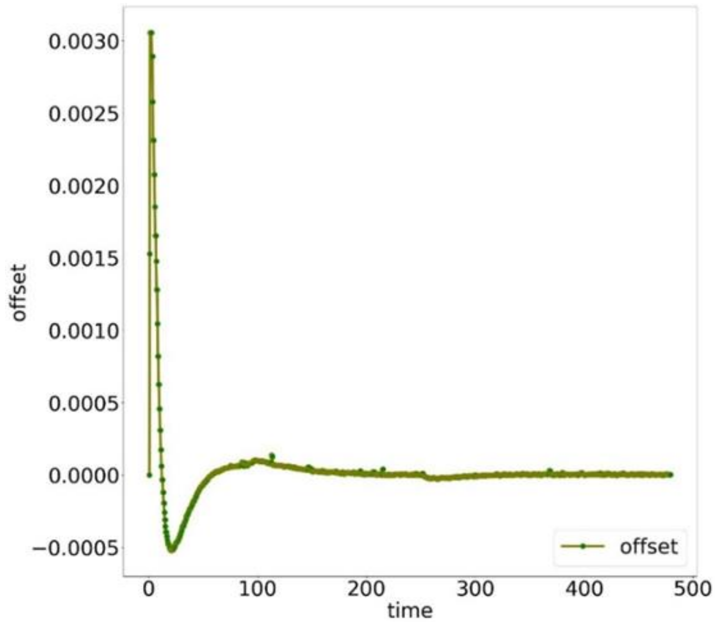

Figure 9. PTP clock stabilization from cold boot.

In this study, the MQTT publisher generates data once every second. The MQTT publisher generated 10,000 packets, with all packets verifiably sent without loss to the IEC 61850 MMS client through the gateway. Figure 10 shows the mean transfer time measurements of MQTT QoS according to Ethernet speed. The 10 Mbps Ethernet experiences the most delay. Moreover, it was found that the faster the Ethernet speed, the smaller the delay. In addition, MQTT QoS level 0 experienced the least delay, while MQTT QoS level 2 experienced the most delay.

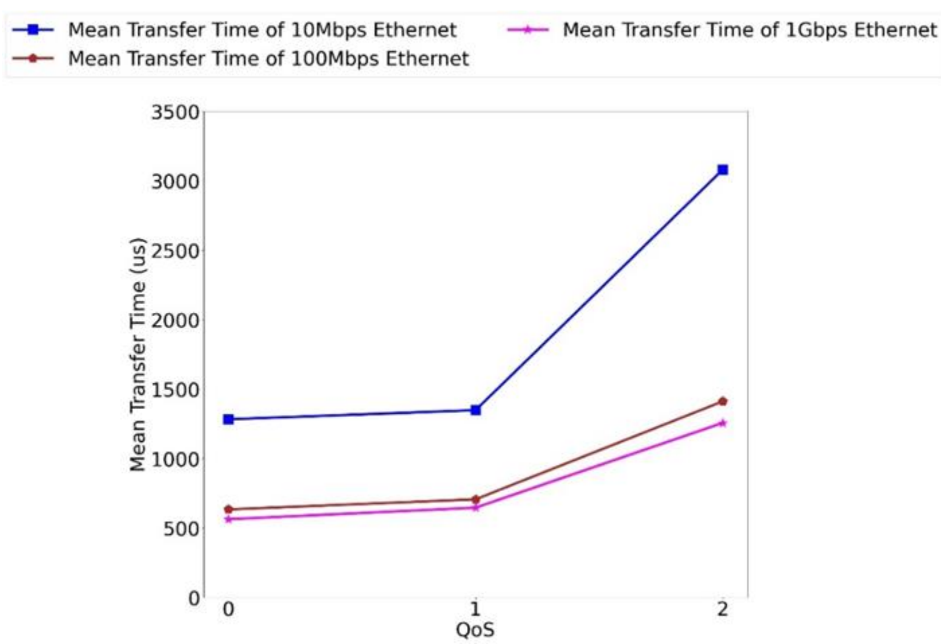

Figure 10. Mean transfer time of MQTT according to Ethernet speed.

Figure 11 shows the mean transfer time measurements for various Wi-Fi protocols. Countless delays were confirmed in IEEE 802.11b, IEEE 802.11g, and IEEE 802.11n of 2.4 GHz, while a relatively small delay occurred in IEEE 802.11a and IEEE 802.11ac of 5 GHz. Figure 12 shows the mean transfer time measurements of IEC 61850 MMS according to Ethernet speed, which verified that 10 Mbps Ethernet experiences the most delay, and the delay decreases as the Ethernet speed increases. Figs. 11 and 12 show that MQTT QoS level 0 experiences the least delay and MQTT QoS level 2 experiences the most delay.

Figure 13 shows the result of adding up the mean transfer time of the MQTT packets and the mean transfer time of sending and receiving datasets between the client and server in IEC 61850 MMS. When the Wi-Fi protocol was used, the delay varied from 11 ms to a maximum of 18 ms. When using Ethernet, the delay varied from a minimum of 7.7 ms to a maximum of 10 ms. The performance requirements of IEC 61850 MMS messages are classified according to the message functions. For example, medium-speed messages are classified into type 2 (< 100ms), low-speed messages are classified into type 3 (< 500ms), and file transfer messages are classified into type 5 (< 1000ms). Figure 13 shows that the mean transfer times satisfy the IEC 61850 MMS performance requirements. Moreover, it was verified that transmission via Ethernet experiences less delay than transmission via Wi-Fi. When MQTT and IEC 61850 interwork, the protocol to be used according to the transfer time requirement can be determined based on the results shown in Figure 13.

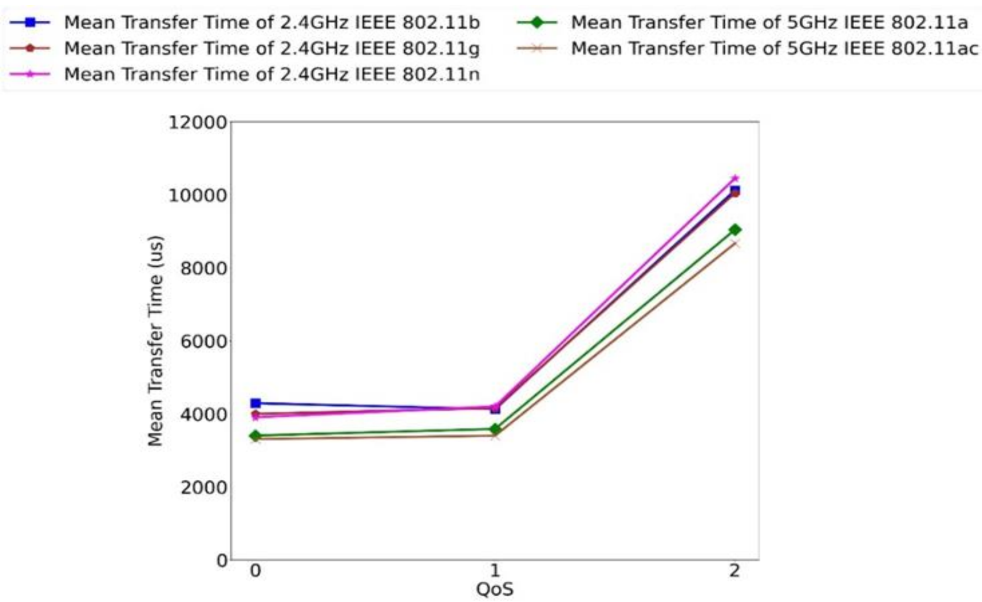

Figure 11. Mean transfer time of MQTT according to Wi-Fi Protocol.

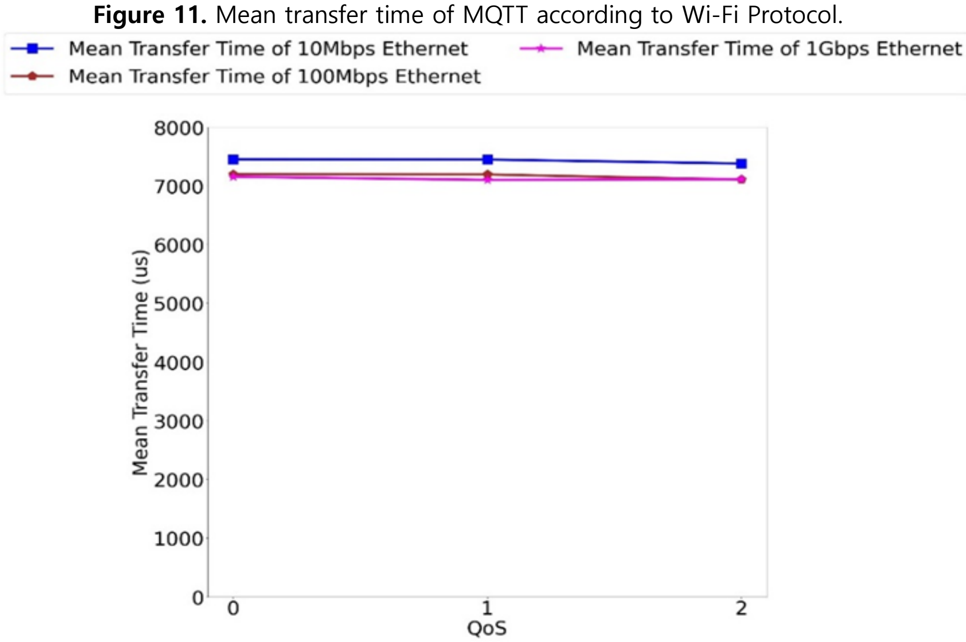

Figure 12. Mean transfer time of IEC 61850 MMS according to Ethernet speed.

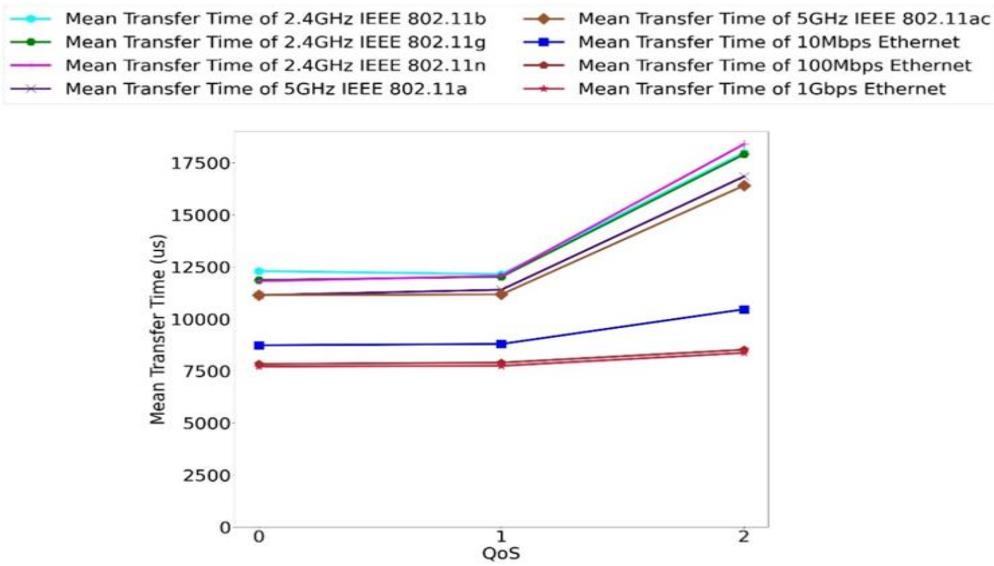

Figure 13. Sum of the mean transfer time of MQTT and IEC 61850 MMS.

5. CONCLUSION

In this study, we implemented an IoT gateway that uses Raspberry Pi 4B and maps MQTT, an IoT protocol used in microgrids, to IEC 61850 MMS. We also used TShark to capture and analyze packets to verify mappings were performed correctly, which was confirmed by the testbed analysis results. In addition, we analyzed performance according to Ethernet speed and Wi-Fi protocol, verifying that the 5 GHz Wi-Fi protocols experience less delay than the 2.4 GHz Wi-Fi protocols. Furthermore, the delay was smaller when Ethernet was used than when Wi-Fi was used. The sum of the mean transfer time of the MQTT packets and the mean transfer time for sending and receiving datasets between the client and the server in IEC 61850 MMS confirms that the IEC 61850 MMS performance requirements are satisfied. When MQTT and IEC 61850 MMS interwork, the protocol to be used according to the mean transfer time constraint can be determined based on the results of this experiment.The results of this study can be helpful for experiments in which the IoT protocol MQTT and IEC 61850 MMS interwork in a microgrid environment.

6. ACKNOWLEDGMENTS

This research was supported by “Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2022RIS-005)

REFERENCES

[1] Akmal, N., Hussain, A., Akhtar, W., Jabbar, A., Anwar, M. Z., & Ullah, S. (2020). Assessment of Economic Impact of Land Laser Leveling in Rice-Wheat and Mixed Cropping Zones of Punjab. Pakistan Journal of Life & Social Sciences, 18(2), 58-64.

[2] Cena, G., Bertolotti, I. C., Valenzano, A. & Zunino, C.(2007). Evaluation of response times in industrial WLANs. IEEE Trans. Ind. Inf., August, 3(3), 191–201.

[3] Jun, H. J. & Yang, H. S.(2021). Performance of the XMPP and the MQTT Protocols on IEC 61850-Based Microgrid Communication Architecture. Energies, 14(16), 1-13.

[4] Kanabar, M. & Sidhu, T. S.(2011). Performance of IEC 61850-9-2 process bus and corrective measure for digital relaying. IEEE Trans. Power Del, April, 26(2), 725–735.

[5] The Smart Grid Interoperability Standards Roadmap Electric Power Research Institute (EPRI) Tech. Rep.(2009), August.

[6] Ullah, M. et al.(2020) IoT Protocol Selection for Smart Grid Applications-Merging Qualitative and Quantitative Metrics.” 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), 993-998.

DOI: https://doi.org/10.15379/ijmst.v10i1.1430