International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 6, December 2024, pp. 7178~7185
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i6.pp7178-7185

A performance evaluation of the internet of things-message queue telemetry transport protocol based water level warning system

Kachane Sonklin, Chanipa Sonklin
Faculty of Science and Technology, Uttaradit Rajabhat University, Uttaradit, Thailand

Article Info

Article history:

Received Mar 24, 2024
Revised Aug 8, 2024
Accepted Aug 14, 2024

Keywords:

Internet of things
Message queue telemetry transport
NodeMCU
Ultrasonic sensor
Water level warning system

ABSTRACT

The internet of things (IoT) and message queue telemetry transport (MQTT) play crucial roles in connecting sensor networks, data exchange among diverse devices, and enabling various smart systems. Several studies have been conducted on IoT-MQTT-based applications because of their ease of implementation and deployment. It also offers real-time and reliable communication between a publisher and a subscriber. However, there is a lack of comprehensive studies covering overall performance metrics. Therefore, this paper aims to develop a water level warning system prototype and evaluate its performance through simulation experiments, focusing on critical metrics, such as latency, throughput, packet loss rate (PLR), packet delivery ratio (PDR), and availability at various data transmission rates. The results demonstrate that the proposed system achieves significantly lower latency, compared to existing solutions and achieves up to 98% availability and reliability with minimal packet loss. The experimental findings also reveal that higher data transmission rates lead to higher throughput and latency performance with lower performance in terms of availability, PDR, and sensor accuracy.

Corresponding Author:

Chanipa Sonklin
Faculty of Science and Technology, Uttaradit Rajabhat University
27 Injaimee Road, Muang District, Uttaradit 53000, Thailand
Email: chanipa.son@uru.ac.th

1. INTRODUCTION

The internet of things (IoT) plays a vital role in the connectivity of sensor networks because the IoT allows sensors, such as actuators, ultrasonic sensors, and humidity sensors to communicate and exchange data with each other [1], [2]. This leads to various smart systems, including smart homes, smart vehicles, and smart farms. With the rapid movement of this device's communication, smart systems require protocols to support them in terms of availability, reliability, low latency, and high throughput. Message queue telemetry transport (MQTT) is one of the most popular protocols in IoT-MQTT-based applications. MQTT is the protocol in IoT based on the publish-subscribe communication to send data from a publisher to a subscriber through a broker (server). It also provides lightweight IoT messaging, simple implementation, open standards, reliability, and system efficiency, including memory, central processing unit (CPU), and network resources [3]. MQTT has been designed to support various communication patterns, such as one-to-one and many-to-many communication [4]. This communication enables diverse IoT applications and allows smart devices to connect and exchange information. Moreover, this protocol has a small overhead, which includes a fixed header (only 2 Bytes) and a small message payload (256 MB). This means it can transmit data efficiently in environments characterized by limited bandwidth, low power consumption, and unreliability networks [5].

The work of [6] evaluated the performance of an MQTT broker named Mosquitto. The results illustrated that Mosquitto broker outperformed the other brokers, including Bevywise, ActiveMQ, HiveMQ, and EMQX in terms of latency and CPU usage. Similarly, Bertrand-Martinez *et al.* [7] proposed a methodology to classify and evaluate IoT brokers with three main contributions, including, framework, middleware for smart environments, and practical implementation. The authors also suggested exploring techniques to optimize the performance of IoT brokers and developed standards or protocols that promote interoperability among different IoT brokers. Azzedin and Alhazmi [8] proposed a secure data distribution architecture using MQTT brokers. This work aimed to improve the architecture's performance, such as delay and security, and also enabled brokers to collaborate and exchange data with other brokers based on the subscribers' interests. Sonklin *et al.* [9] proposed a framework based on publish-subscribe communication using the MQTT protocol. To evaluate the framework's performance, such as latency and reliability, extensive experiments were conducted. They claimed that the reliability of their system was more than 96% while the latency was less than 300 ms. The fundamentals of MQTT and constrained application protocol (CoAP) protocols were proposed in [10] to examine the communication delay. The authors also emphasized that MQTT was particularly well-suited for situations in which there was communication among numerous devices in a many-to-many configuration. Wan *et al.* [11] presented a blockchain-based MQTT IoT sensor forensic system and the aim was to protect the system from digital crime attacks. Yeh *et al.* [12] proposed a smart factory network architecture based on IoT communication protocol. MQTT was used in this work to solve the use of different industrial network protocols for different equipment in factories. The main key performance of MQTT protocol was proposed in [13], including delay and energy consumption. The results of this work also illustrated that MQTT outperforms the QUIC protocol in terms of delay. The two works [14], [15] worked on exploring the IoT protocol that could transmit data between IoT devices in real time. The authors evaluated the performance of the different types of IoT protocols, such as MQTT, real-time publish-subscribe (RTPS), hypertext transfer protocol (HTTP) and advanced message queuing protocol (AMQP). The results showed that MQTT was suitable for transporting short messages when compared to the other protocols. An implementation of MQTT load test tools was presented in study [16] to evaluate the performance of their system using MQTT protocol while transmitting large amounts of IoT data. The authors claimed that their new system was executable on various platforms, including graphical user interface (GUI) and a text-based console version.

Several studies have been done on the MQTT protocol because it is easy to implement and deploy. The MQTT protocol also provides real-time and reliable communication between a publisher and a subscriber. However, few studies have been conducted to cover the overall performance metrics. Therefore, this paper aims to develop the water level warning system prototype using the MQTT protocol and then evaluate the performance of the proposed system with different metrics, including latency, throughput, packet loss rate (PLR), packet delivery ratio (PDR), availability, and accuracy by simulation experiments. The analysis is performed to discuss the experimental result. The main contribution of this paper is a performance evaluation of the proposed system through simulation experiments and their analysis to investigate the optimal parameter values and their correlation with different data transmission rates to support IoT applications. This is essential to provide practical guidance for the future design of IoT applications.

The paper is organized as follows. Section 2 illustrates the proposed system design. Section 3 presents the experimental results and discussion. Finally, this paper is concluded and future work is discussed in section 4.

2. METHOD

2.1. Water level warning system design

The performance of the proposed water level warning system is evaluated through simulation and experimental analyses. Several hardware and software tools are used to develop and evaluate the proposed system. Figure 1 shows the proposed water level warning system prototype. It is implemented using Node-RED as a subscriber, Node MicroController unit (NodeMCU) based on ESP-WROOM-32 module (ESP32 NodeMCU) as a publisher, the MQTT broker as a broker, and an ultrasonic sensor to send data.

According to study [17], the ultrasonic sensor is a low-cost sensor and can be performed well under extensive conditions, such as temperature and pressure. In this study, we use an ultrasonic sensor to measure the water level and send data to ESP32 NodeMCU, which is a system-on-chip (SOC) microcontroller with integrated Wi-Fi and Bluetooth-enabled development board based on the ESP32 microcontroller [18]. It has been widely used in IoT systems due to its affordability, energy efficiency, and powerful computing capabilities [19]. Figure 2 shows hardware schematic design for the ultrasonic sensor and NodeMCU.

Then, NodeMCU needs to be connected to a Wi-Fi modem. Once connected, NodeMCU could send the sensor-generated data to the MQTT broker (or server) over the internet. After the server processes the data, subscribers can access it through a created application. Our system uses Node-RED, a visual programming tool, as a subscriber and creates an application. This is because Node-RED allows users to wire together devices, application programming interfaces (APIs), and online services to develop applications and automate workflows [20]–[22]. Particularly, this tool is popular in IoT implementations, where it is used to connect and control IoT devices, collect and analyze sensor data, and create IoT applications.

According to Figure 3, the block diagram illustrates the complete process of the proposed system performance analysis. In the process of collecting data, ultrasonic sensor is used to collect water level data from water tank. After that, data is transferred from NodeMCU via the MQTT broker to analyze the performance metrics of the proposed system prototype. The details of the performance parameters are described in the next section.

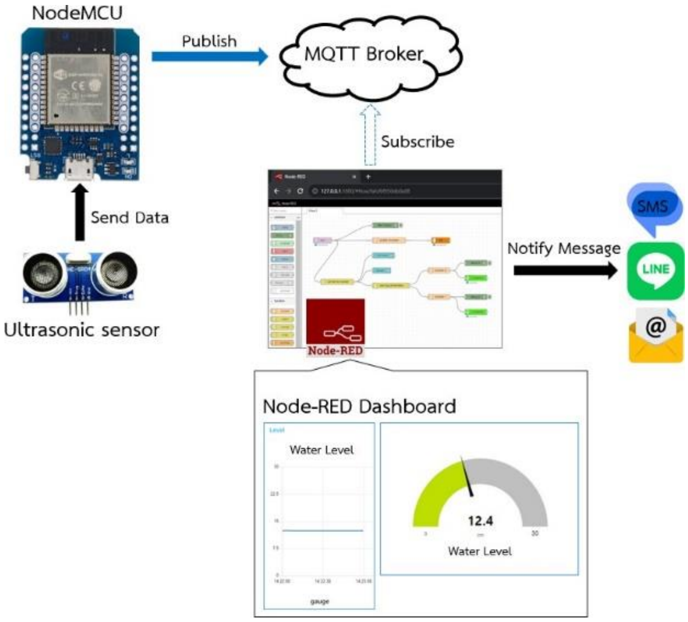

Figure 1. System prototype design

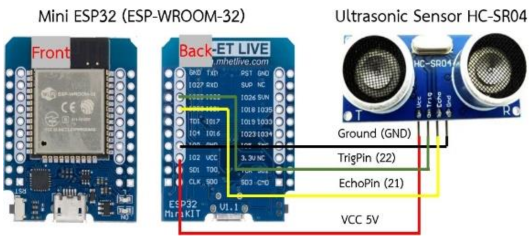

Figure 2. Hardware design

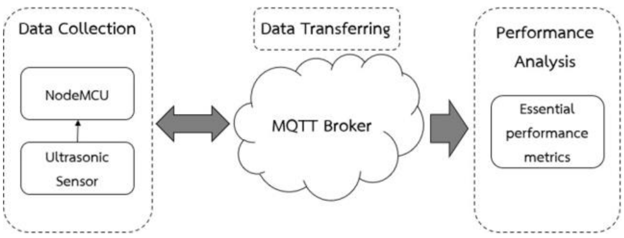

Figure 3. Block diagram of the proposed system prototype

2.2. Performance test parameters

Various parameters are used to evaluate the performance of the proposed system prototype. These parameters are latency, throughput, packet loss rate (PLR), packet delivery ratio (PDR), availability, and sensor accuracy. The following are the details of each parameter.

2.2.1. Latency

Performance testing is conducted to assess a system's responsiveness, with latency being a key metric in this evaluation. The primary focus of latency performance testing is to quantify the duration required for data transmission from the publisher to the subscriber. To calculate the latency of our system, the formula is expressed as in (1) [23].

$$Latency = \frac{Length\ of\ packets\ (bit)}{Bandwith\ usage\ (bit/s)} \tag{1}$$

2.2.2. Throughput

It is essential to ensure that a system can manage the expected workload effectively with high-level performance and a prompt response for the end-users. Throughput is an essential factor to consider as it involves assessing the capacity and efficiency of a system to handle a substantial volume of data or transactions. It measures the amount of data processed per unit of time and then evaluates the system's ability to scale with increasing workloads. Throughput is usually measured in bytes per second and can be calculated using formula (2) [23], [24].

$$Throughput = \frac{Amount\ of\ data\ transfer\ (byte)}{Time\ taken\ (s)} \quad (2)$$

2.2.3. Packet loss rate

Packet loss rate (PLR) is a crucial metric in networking that quantifies the percentage of data packets that fail to reach the destination within a communication system. PRL indicates the reliability and robustness of the network as a lower packet loss rate implies more successful data transmission. The PLR formula can be calculated using the formula (3) [24], [25].

$$PLR(\%) = \frac{Total\ packets\ sent - Total\ packets\ received}{Total\ packets\ sent} \times 100 \quad (3)$$

2.2.4. Packet delivery ratio

Packet delivery ratio (PDR) is a metric that measures the ratio of successfully delivered packets to the total number of packets transmitted. A high PDR indicates an efficient and reliable network, reflecting the percentage of packets that reach the destinations without errors or loss. The PDR can be calculated as in (4) [25], [26].

$$PDR(\%) = \frac{Total\ packets\ received}{Total\ packets\ sent} \times 100 \quad (4)$$

2.2.5. Availability

System availability refers to a system that is operational and accessible when needed while optimizing its functionality for end-users. Availability is typically mentioned as a percentage, representing the proportion of time the system is functional over a specific period. The availability can be calculated as in (5) [25], [27].

$$Availability(\%) = \frac{Total\ packets\ sent}{Total\ packets\ sent + Total\ packet\ lost} \times 100 \quad (5)$$

2.2.6. Sensor accuracy

Sensor accuracy is the metric to assess the performance of the ultrasonic sensor employed for the proposed system prototype. Root mean square error (RMSE) is a key statistical metric that has been used to evaluate the accuracy of a model or sensor by quantifying the difference between measured values and true values [28]. The sensor accuracy can be calculated using (6) [28].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \quad (6)$$

where x_i is the actual water level value, which is measured by a measuring instrument, $\hat{x}_i$ is the measured water level value, measured by the ultrasonic sensor, and n is the number of observations.

3. RESULTS AND DISCUSSION

In this section, the experiments are conducted to illustrate the performance of the proposed water level warning system prototype using IoT protocol with various parameters described in the previous section. Then, the results are analyzed and discussed to find the optimal parameter values and their correlation with different data transmission rates to support IoT applications. To evaluate the performance of the proposed system prototype in terms of latency, throughput, packet loss rate, packet delivery ratio, reliability, and availability, four sets of experiments are designed. The MQTT protocol is used to demonstrate a message transfer with different data transmission rates. In the experiment, Node-RED as a subscriber is set up on a personal computer with a wired connection (LAN cable), while NodeMCU as a publisher and the MQTT broker as a broker are set up on the same laptop with a Wi-Fi connection. The MQTT broker is installed to run as middleware of the publish-subscribe communication with quality of service (QoS) level 0. Data transmission rates are 1, 2, 5, and 10 Hz. In this study, ‘Hz’ represents the frequency at which messages are sent per second. For example, the data transmission rates of 5 and 10 Hz mean 5 and 10 messages are sent every second.

Every experiment is run for 60 seconds and repeated 30 times. The Wireshark software is used in the experiments to measure the system’s performance. The details of each set of experiments and the results are discussed in the following subsections.

3.1. Experiment 1: latency of message exchange over MQTT

The measured latency includes the latency between NodeMCU and MQTT broker, MQTT broker and Node-RED, and NodeMCU and Node-RED (one-way latency). Figure 4 demonstrates the average message transmission for three measured latencies at various data transmission rates. It can be seen from the figure that the latency increases when the data transmission rate increases from 1 to 10 Hz. Transmitting data at a high data rate (10 Hz) enables data to be published quickly. Similarly, Zulfikri *et al.* [23] concluded that an increase in data transmission rate resulted in a greater latency. However, our latency results outperform the system prototype [23] in the message transmission rate at 1, 2, and 5 Hz. This clear visualization illustrates the direct impact of data transmission rates on latency, emphasizing the need for higher transmission rates for enhanced efficiency.

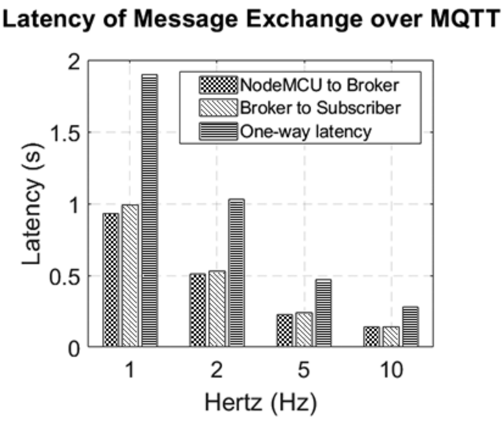

Figure 4. Latency of message exchange over MQTT comparisons

3.2. Experiment 2: throughput of message exchange over MQTT

This experiment aims to investigate how throughput is affected by data transmission rate. Figures 5 and 6 show that throughput performs well when transmitting messages from NodeMCU to the MQTT broker and from the MQTT broker to the subscriber at the 10 Hz data transmission rate. The results also demonstrate that a higher data transmission rate leads to an increase in throughput.

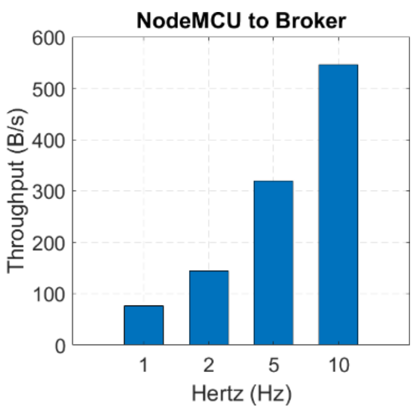

Figure 5. Throughput between NodeMCU and Broker

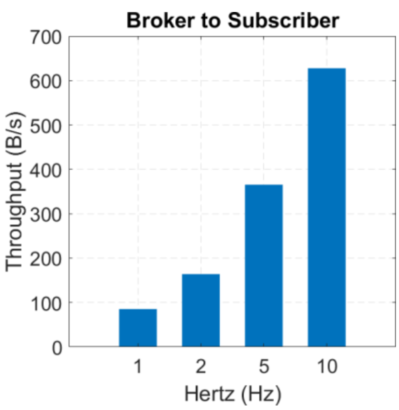

Figure 6. Throughput between Broker and Subscriber

As mentioned previously, data transmission rate refers to the frequency of data sent per second. Higher data rates indicate more data being transmitted. According to (2), the throughput depends on the amount of data transferred and the total time taken. However, for this experiment, the total time taken does not affect the throughput because the total time for sending the complete message (from NodeMCU to subscriber) is the same (60 seconds) for all experiments. Therefore, the data transmission rate plays a crucial role in determining throughput.

3.3. Experiment 3: availability, PDR, and PLR

This experiment investigates how data transmission rate affects the availability, PDR, and PLR of messages transmitted from the publisher (NodeMCU) to the subscriber (Node-RED). Table 1 presents the results of the proposed system performance, including availability, PDR, and PLR for different data transmission rates. It can be seen from the table that when the data transmission rate increases from 1 Hz to 10 Hz, PLR increases whereas availability, and PDR decrease. As the data transmission rate increases, the number of packets transmitted per second increases, leading to more packet loss [23]. Furthermore, more packets are being dropped because of the MQTT QoS Level 0 message transmission method, which does not guarantee that the subscriber will receive the message [29]. Consequently, this can lead to packet loss and reduced availability, and PDR. However, the results demonstrate that availability and PDR have a value of up to 97.8%, with only 2.2% PLR at a 1 Hz data transmission rate. Therefore, the data transmission rate affects the availability, PDR, and PLR.

Table 1. Availability, PDR and PLR obtained with different data transmission rates

Rate (Hz)	Availability (%)	PDR (%)	PLR (%)
1	97.83	97.78	2.22
2	94.41	94.08	5.92
5	86.67	84.62	15.38
10	78.38	72.42	27.58

3.4. Experiment 4: sensor accuracy

This experiment investigates how accurate the ultrasonic sensor is when transmitting messages from NodeMCU to a subscriber at different data transmission rates. Figure 7 shows the sensor accuracy from 1 Hz, 2 Hz, 5 Hz, and 10 Hz data transmission rates. The results demonstrate that RMSE value increases when the data transmission rate increases. The lower RMSE value implies that the sensor's readings are close to the true values, indicating more accurate performance [30]. The sensor transmitting messages at the 1 Hz data transmission rate has the best accuracy with an RMSE value equal to 0.71 and has a higher accuracy of 82.77% compared to the others. It is concluded that data transmission rate affects the sensor accuracy.

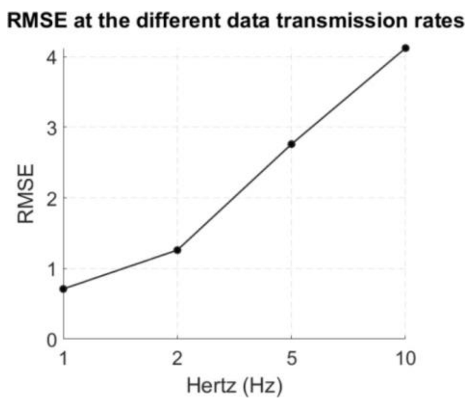

Figure 7. RMSE with the different data transmission rates comparisons

4. CONCLUSION

In this paper, we have designed and assessed an IoT-based water level warning system employing the MQTT protocol, demonstrating marked enhancements in key performance metrics, including latency, throughput, packet loss rate, packet delivery ratio, availability, and accuracy with different data transmission rates. The experimental results have shown that the proposed prototype achieves up to 98% availability, and

PDR, with minimal PLR. The experimental findings also reveal that increasing data transmission rates enhances throughput while decreasing latency, availability, PDR, and sensor accuracy. In other words, higher data transmission rates lead to higher throughput and latency performance with lower performance in terms of availability, PDR, and sensor accuracy. This highlights the critical importance of optimizing these parameters for the efficacy of IoT implementations, especially substantial potential solutions for real-time flood monitoring and response applications. Therefore, these outcomes significantly advance the field of IoT-based environmental monitoring, providing valuable insights for the development of smart city infrastructure and disaster management strategies.

In future works, we will focus on integrating additional sensor types and evaluating the scalability of the system with more extensive and complex environments. Furthermore, assessing the system's performance under diverse environmental conditions and varying data loads will be essential for validating its robustness and applicability across different scenarios.

REFERENCES

[1] K. Sonklin, "Studies of communication and positioning performance of connected vehicles for safety applications," Thesis, Queensland University of Technology, 2020.

[2] Z. Kegenbekov and A. Saparova, "Using the MQTT protocol to transmit vehicle telemetry data," *Transportation Research Procedia*, vol. 61, pp. 410–417, 2022, doi: 10.1016/j.trpro.2022.01.067.

[3] C.-R. Dow, D.-B. Nguyen, S. Cheng, P.-Y. Lai, and S.-F. Hwang, "VIPER: an adaptive guidance and notification service system in internet of vehicles," *World Wide Web*, vol. 22, no. 4, pp. 1669–1697, Jul. 2019, doi: 10.1007/s11280-018-0617-7.

[4] D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia, "A performance analysis of internet of things networking protocols: evaluating MQTT, CoAP, OPC UA," *Applied Sciences*, vol. 11, no. 11, May 2021, doi: 10.3390/app11114879.

[5] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: a survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020, doi: 10.1109/ACCESS.2020.3035849.

[6] B. Mishra, B. Mishra, and A. Kertesz, "Stress-testing MQTT brokers: a comparative analysis of performance measurements," *Energies*, vol. 14, no. 18, Sep. 2021, doi: 10.3390/en14185817.

[7] E. Bertrand-Martinez, P. Dias Feio, V. de Brito Nascimento, F. Kon, and A. Abelém, "Classification and evaluation of IoT brokers: a methodology," *International Journal of Network Management*, vol. 31, no. 3, May 2021, doi: 10.1002/nem.2115.

[8] F. Azzedin and T. Alhazmi, "Secure data distribution architecture in IoT using MQTT," *Applied Sciences*, vol. 13, no. 4, Feb. 2023, doi: 10.3390/app13042515.

[9] K. Sonklin, C. Wang, D. Jayalath, and Y. Feng, "Connected-vehicle data exchanges and positioning computing based on the publish-subscribe paradigm," *Journal of Computer and Communications*, vol. 07, no. 10, pp. 82–93, 2019, doi: 10.4236/jcc.2019.710008.

[10] S. Quincozes, T. Emilio, and J. Kazienko, "MQTT protocol: fundamentals, tools and future directions," *IEEE Latin America Transactions*, vol. 17, no. 09, pp. 1439–1448, Sep. 2019, doi: 10.1109/TLA.2019.8931137.

[11] C. Wan, A. Mehmood, M. Carsten, G. Epiphaniou, and J. Lloret, "A blockchain based forensic system for IoT sensors using MQTT protocol," in *2022 9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Nov. 2022, pp. 1–8, doi: 10.1109/IOTSMS58070.2022.10062190.

[12] C. S. Yeh, S. L. Chen, and I. C. Li, "Implementation of MQTT protocol based network architecture for smart factory," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Nov. 2021, vol. 235, no. 13, pp. 2132–2142, doi: 10.1177/09544054211014488.

[13] S. Jeddou, F. Fernández, L. Diez, A. Baina, N. Abdallah, and R. Agüero, "Delay and energy consumption of MQTT over QUIC: an empirical characterization using commercial-off-the-shelf devices," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103694.

[14] A. Gavrilov, M. Bergaliyev, S. Tinyakov, K. Krinkin, and P. Popov, "Using IoT protocols in real-time systems: protocol analysis and evaluation of data transmission characteristics," *Journal of Computer Networks and Communications*, vol. 2022, pp. 1–18, Aug. 2022, doi: 10.1155/2022/7368691.

[15] K. T. M. Tran, A. X. Pham, N. P. Nguyen, and P. T. Dang, "Analysis and Performance comparison of IoT message transfer protocols applying in real photovoltaic system," *International Journal of Networked and Distributed Computing*, vol. 12, no. 1, pp. 131–143, Jun. 2024, doi: 10.1007/s44227-024-00021-4.

[16] H. Y. Chien *et al.*, "A MQTT-API-compatible IoT security-enhanced platform," *International Journal of Sensor Networks*, vol. 32, no. 1, 2020, doi: 10.1504/IJSNET.2020.104463.

[17] S. L. Mohammed, A. Al-Naji, M. M. Farjo, and J. Chahl, "Highly accurate water level measurement system using a microcontroller and an ultrasonic sensor," *IOP Conference Series: Materials Science and Engineering*, vol. 518, no. 4, May 2019, doi: 10.1088/1757-899X/518/4/042025.

[18] A. Anggrawan, S. Hadi, and C. Satria, "IoT-based garbage container system using NodeMCU ESP32 microcontroller," *Journal of Advances in Information Technology*, vol. 13, no. 6, 2022, doi: 10.12720/jait.13.6.569-577.

[19] N. Kurniawan, "Electrical energy monitoring system and automatic transfer switch (ATS) controller with the internet of things for solar power plants," *Journal of Soft Computing Exploration*, vol. 1, no. 1, Sep. 2020, doi: 10.52465/joscex.v1i1.2.

[20] R. Lakshmanan, M. Djama, S. Perumal, and R. Abdulla, "Automated smart hydroponics system using internet of things," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6389–6398, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6389-6398.

[21] D. Torres, J. P. Dias, A. Restivo, and H. S. Ferreira, "Real-time feedback in Node-RED for IoT development: an empirical study," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Sep. 2020, pp. 1–8, doi: 10.1109/DS-RT50469.2020.9213544.

[22] A. D. Salman, "IoT monitoring system based on MQTT publisher/subscriber protocol," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 75–83, Jul. 2020, doi: 10.33103/uot.ijccce.20.3.7.

[23] A. A. Zulfikri, D. Perdana, and G. Bisono, "Design and analysis of trash monitoring system prototype based on internet of things (IoT) using MQTT protocol," *Jurnal Infotel*, vol. 10, no. 3, Aug. 2018, doi: 10.20895/infotel.v10i3.381.

[24] A. Mamane, M. Fattah, M. El Ghazi, M. El Bekkali, Y. Balboul, and S. Mazer, "Scheduling algorithms for 5G networks and beyond: classification and survey," *IEEE Access*, vol. 10, pp. 51643–51661, 2022, doi: 10.1109/ACCESS.2022.3174579.
[25] B. Mbarek, M. Ge, and T. Pitner, "An efficient mutual authentication scheme for internet of things," *Internet of Things*, vol. 9, Mar. 2020, doi: 10.1016/j.iot.2020.100160.
[26] R. Ratnasih, D. Perdana, and Y. G. Bisono, "Performance analysis and automatic prototype aquaponic of system design based on internet of things (IoT) using MQTT protocol," *Jurnal Infotel*, vol. 10, no. 3, 2018, doi: 10.20895/infotel.v10i3.388.
[27] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, "Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs)," in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, Dec. 2013, pp. 324–329, doi: 10.1109/MSN.2013.74.
[28] A. R. Abdellah and A. Koucheryavy, "Deep learning with long short-term memory for IoT traffic prediction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12525 LNCS, pp. 267–280, 2020, doi: 10.1007/978-3-030-65726-0_24.
[29] H. H. Alshammari, "The internet of things healthcare monitoring system based on MQTT protocol," *Alexandria Engineering Journal*, vol. 69, pp. 275–287, Apr. 2023, doi: 10.1016/j.aej.2023.01.065.
[30] A. Barthwal and D. Acharya, "An IoT based sensing system for modeling and forecasting urban air quality," *Wireless Personal Communications*, vol. 116, no. 4, pp. 3503–3526, Feb. 2021, doi: 10.1007/s11277-020-07862-6.

BIOGRAPHIES OF AUTHORS

Kachane Sonklin received the B.Sc. degree in computer science and the M.Sc. degree in information technology from Naresuan University, Thailand, and the Ph.D. degree in computer science from Queensland University of Technology, Brisbane QLD, Australia. He is currently a lecturer at Faculty of Science and Technology, Uttaradit Rajabhat University, Thailand. His current research interests include device-to-device (D2D) communications, vehicle networks and safety, connected vehicles, global navigation satellite systems (GNSS) algorithms, internet of things (IoT), publish-subscribe communication paradigm, artificial intelligence, and machine learning. He can be contacted at email: kachane.son@uru.ac.th.

Chanipa Sonklin received the B.Sc. degree in computer science and the M.Sc. degree in information technology from Naresuan University, Thailand, and the Ph.D. degree in computer science from Queensland University of Technology, Brisbane QLD, Australia. She is currently a lecturer at Faculty of Science and Technology, Uttaradit Rajabhat University, Thailand. Her current research interests include evolutionary computation, artificial intelligence, machine learning, optimization algorithms, cloud computing, internet of things (IoT) and their applications on real-world problems. She can be contacted at email: chanipa.son@uru.ac.th.