

Quality of Service (QoS) Analysis of MQTT Protocol for Smart Farming Monitoring System Optimization

Adi Hermansyah¹, Huda Ubaya², Achmad Andriansyah Aljali³, Nurul Afifah⁴ and Septiani Kusuma Ningrum⁵

^{1,2,3,4} Departement Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya

⁵ Faculty of Computer Science, Universitas Sriwijaya

ABSTRACT

So far, plantation conditions that have only been monitored manually can now be monitored automatically, making it easier to monitor the condition of the plantation. This study proposes a system for monitoring the condition of the plantation by implementing the MQTT communication protocol based on smart farming. Message Queue Telemetry Transport (MQTT) is a machine-to-machine connectivity protocol designed as messaging and provides a publish/subscribe architecture. In this study, MQTT is implemented using a mosquitto broker which functions to regulate the sending of messages between Publishers and Subscribers. With the scheme of increasing the number of publishers (1 to 4 publishers) in each test, the MQTT protocol has the largest average throughput during the condition of 4 publishers which is 13093.5070 bps and the smallest in the condition of 1 publisher 4281.7529 bps, the average packet delivery ratio is the largest in the condition of 4 publishers which is 99.8328% and the smallest when 1 publisher is 99.5615%, the average packet loss is the largest when the condition of 1 publisher is 0.4385% and the smallest when 4 publishers is 0.1672%, And the average largest delay during the condition of 1 publisher is 161.5244 ms and the smallest when 4 publishers is 59.0415 ms.

PAPER HISTORY

Received Jan 02, 2025

Revised Feb 12, 2025

Accepted March 7, 2025

Published March 30, 2025

KEYWORDS

MQTT;
Smart Farming;
Quality of Service;
Performance;
IoT

CONTACT:

adihermansyah@unsri.ac.id
huda@unsri.ac.id
nurul@unsri.ac.id
septianikuning@gmail.com

1. INTRODUCTION

The development of technology has had a significant impact on various fields, including the agricultural sector. Rapid technological innovations significantly drive improvements in efficiency and productivity in agricultural activities. Smart Farming has developed as an innovative method that integrates the Internet of Things (IoT) into agricultural systems[1][2]. This approach involves the utilization of various digital technologies, including IoT, cloud computing, robotics, sensors, location systems, and artificial intelligence to optimize agricultural processes[2][3][4]. In various contexts, this concept is also referred to as digital agriculture, data-driven agriculture, Agriculture 4.0, or even Agriculture 5.0. Smart agricultural infrastructure consists of sensors that collect environmental data as well as surveillance cameras. The data collected is then transmitted through a gateway to a service platform accessible by farmers, enabling more accurate decision-making based on real-time information. One important aspect of implementing Smart Agriculture

is the monitoring system, which allows for real-time monitoring of agricultural conditions[5][6][7].

The MQTT (Message Queuing Telemetry Transport) protocol has become one of the most widely used communication protocols in monitoring systems due to its lightweight and efficient nature. Monitoring itself is the process of systematically collecting and analyzing information to support decision-making and corrective actions in agricultural management. By adopting the MQTT Protocol, conditions in the garden can be monitored directly, allowing for a quick response to environmental changes [8][9][10]. The MQTT protocol works by using a broker that acts as an intermediary in the communication between two types of clients, namely the Publisher Client and the Subscriber Client [11][12][13][14]. The broker receives messages from the publisher client and forwards them to the subscriber client who is subscribed to a specific topic. In this study, the broker platform used is Mosquitto, which is an MQTT-based client-server implementation. Mosquitto consists of three main components, namely the main Mosquitto server, as well as two types of clients, mosquitto pub and

mosquito sub. This research aims to measure the quality of service (Quality of Service/QoS) of the MQTT Protocol implemented in the garden condition monitoring system. Several parameters used in the QoS evaluation include throughput, packet delivery ratio (PDR), packet loss, and delay. The development of technology has had a significant impact on various fields, including the agricultural sector. Rapid technological innovations significantly drive improvements in efficiency and productivity in agricultural activities. Smart Farming has developed as an innovative method that integrates the Internet of Things (IoT) into agricultural systems [15]. This approach involves the utilization of various digital technologies, including IoT, cloud computing, robotics, sensors, location systems, and artificial intelligence to optimize agricultural processes [16][13]. In various contexts, this concept is also referred to as digital agriculture, data-driven agriculture, Agriculture 4.0, or even Agriculture 5.0. Smart agricultural infrastructure consists of sensors that collect environmental data as well as surveillance cameras. The data collected is then transmitted through a gateway to a service platform accessible by farmers, enabling more accurate decision-making based on real-time information. One important aspect of implementing Smart Agriculture is the monitoring system, which allows for real-time monitoring of agricultural conditions.

Throughput is the effective data transfer rate, measured in bits per second (bps). This concept includes the total length of packets that successfully arrive at the destination over a certain period of time, which is then divided by the duration of that interval [5]. Packet Delivery Ratio (PDR) is defined as the ratio between the total number of data packets sent and the total number of data packets received [6]. Packet Loss is a parameter that reflects the condition where a number of data packets are lost during the transmission process. This can occur due to various factors, including collisions (collisions between packets) and congestion (slowdown due to load) on the network [7]. Delay refers to the duration required for data to reach its destination from the starting point. Factors affecting delay include distance, characteristics of the physical medium, congestion, and also the duration of the processing time [11].

[17] conducted a comparative study on the performance of the MQTT protocol, focusing on payload size and security level as the main variables. The research results show that increasing the level of security does not significantly impact latency as long as the payload size remains small. Additionally, the implementation of mutual authentication using Transport Layer Security (TLS) does not affect the response time of MQTT for persistent connections, compared to the standard security scheme that only authenticates the server. In addition to specifically examining the performance of MQTT, several studies also compare this protocol with other alternatives. Seoane et al. [11][12][18][19] compared the MQTT protocol and the Constrained Application Protocol (CoAP) to assess the features, capabilities, and feasibility on resource-constrained devices. This study considers aspects of security support as well as various network conditions.

Other research also shows similar analyses, where various protocols are applied to IoT devices.

[20] evaluated the impact of IoT communication protocols on the microgrid environment, focusing on the real-time requirements necessary for smart grid functions, such as protection, control, and monitoring. This study analyzes communication traffic and delays to assess the scalability of the protocols. The three protocols tested in this study are CoAP/DTLS, MQTT/TLS, and XMPP/TLS. The results show that the implementation of security mechanisms causes latency and overhead spikes of up to three times. Although the increase in delay is still within reasonable limits for microgrid monitoring, it does not meet the recommended standards for control operations. This study discusses various aspects of security in the MQTT protocol, including potential threats, attack techniques, and effective mitigation strategies. This study discusses machine learning-based security, replay attacks, man-in-the-middle attacks, anomaly detection, blockchain-based trust mechanisms, DoS attacks, and encryption in data transmission as protective measures [18][19][21].

[22] proposed a QoS controller for MQTT in an electrical IoT system. The controller functions to monitor the electrical network through field gateways and brokers. This controller is designed as an optimization problem, with the main goal of minimizing the packet loss ratio and delay, which are modeled analytically. Based on the test results, the multi-armed bandit (MAB) approach is integrated with the application of reinforcement learning to solve the problem. In addition to the approach in the electrical IoT scenario, several other studies discuss QoS in different contexts. One of them is the study [20][23][24], which examines a distributed scenario with multiple MQTT brokers and a special device called a gateway. This device is responsible for tracking the quality of service (QoS) based on the latency between the client and the broker. This approach aims to minimize communication latency while performing load balancing, thereby improving network efficiency and ensuring optimal service quality.

[25][26] proposed an adaptive QoS controller that can adjust the optimal QoS level for each node based on specific delivery performance constraints. This controller uses a centralized architecture running on the MQTT-SN broker, with a mechanism for periodically collecting network statistics to adjust the QoS level according to end-to-end latency and packet loss rate. To test its effectiveness, this system was implemented in an NS-3-based simulation and tested on an 802.11b-based IoT network with several clients and interfering nodes. The results were compared with traditional scenarios, where QoS selection was done manually by users.

2. MATERIALS AND METHOD

This research was conducted in 2023 at the Sriwijaya University in Indonesia. Determining the hardware and software components is the initial step taken by the author in the implementation of Message Queue Telemetry Transport (MQTT) in this monitoring system. After determining the components, the next step is to create a

model or initial design of the hardware to be connected to the MQTT (Message Queue Telemetry Transport) protocol. The hardware that has been connected to MQTT will be tested based on testing parameters such as throughput, packet delivery ratio (PDR), packet loss, and delay. In Figure 1, it can be seen that 4 publisher clients and 2 subscriber clients are used with MQTT-Explorer on a laptop and MyMQTT on an Android phone as data display for the sensor readings. The publisher used is the Lilygo T-HiGrow v1.1 equipped with the BH1750, DHT11, and Electrical Conductivity sensor. Data sent via the MQTT protocol will be captured using Wireshark for processing and calculating QoS testing with measurement parameters such as throughput, packet delivery ratio, packet loss, and delay.

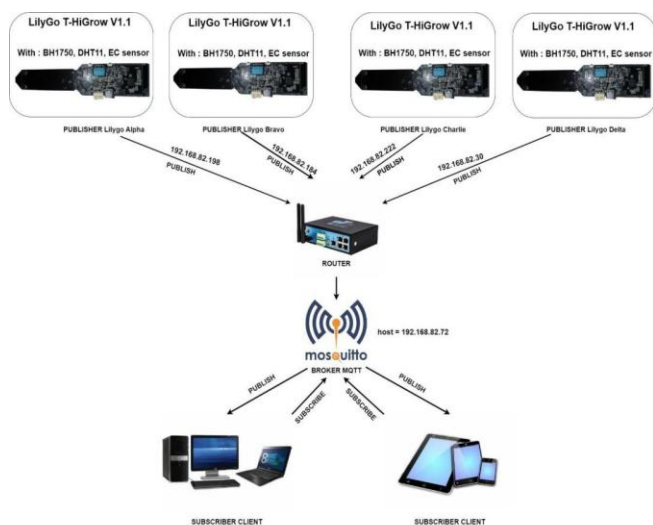


Fig. 1. Monitoring system topology

A. Testing Result Categories

According to [15] Based on [16], the Quality of service category can be defined as listed in Tables 1, 2, and 3 below:

a. Packet Delivery Ratio

Table 1. PDR Category

Very Good	97% until 100%
Good	85% until < 97%
Currently	75% until < 85%
Ugly	<75%

b. Packet Loss

Table 2. Packet Loss Category

Very Good	0% until < 3%
Good	3% until < 15%
Currently	15% until < 25%
Ugly	> 25%

c. Delay

Table 3. Delay Category

Very Good	0% until < 3%
Good	3% until < 15%
Currently	15% until < 25%
Ugly	> 25%

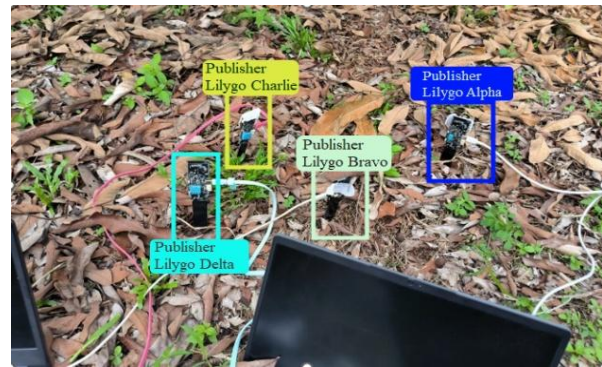


Fig. 2. Implementation of System Design.

In Figure 2, it can be seen that the author used a total of 4 publishers; Lilygo Alpha, Lilygo Bravo, Lilygo Charlie, and Lilygo Delta. Where the QoS testing scheme will use an incremental number of publishers, starting from one publisher, two publishers, three publishers, and four publishers.

B. Sensor Data Transmission

Sensor data will be sent by the publisher to the subscriber, in this study, the MQTT Explorer application is used on the laptop and the MyMQTT application on the mobile phone.

a. MQTT Explorer

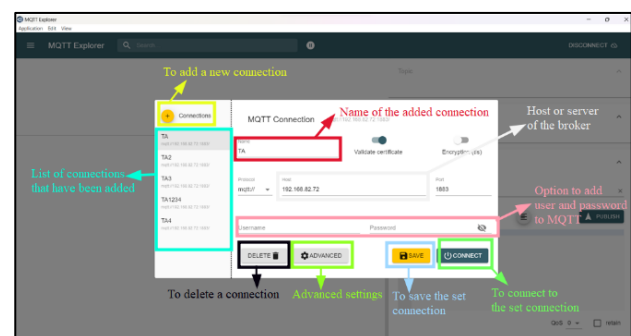


Fig. 3 Initial view of MQTT Explorer

In Figure 3, the initial view of MQTT-Explorer is displayed, which includes a section for naming connections, a section for adding hosts or servers from the broker, and several buttons such as the button to add a new MQTT connection, the button to delete an added connection, the button to save the configured connection, the button to connect to the connection, and the button for advanced settings to add MQTT topics.

b. MyMQTT

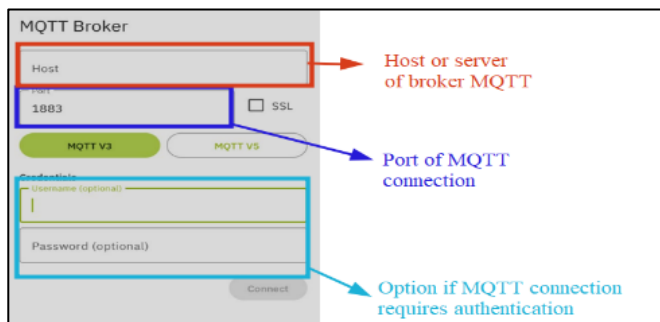


Fig. 4. Initial display MyMQTT

In Figure 4, it can be seen that there are several sections on the initial MyMQTT interface, such as the section for adding a host or broker server, then there is a section for adding the MQTT connection port, and there is also an option to add a username and password if the MQTT network requires authentication.

C. Data Collection for QoS Testing

QoS testing data was obtained from monitoring sensor data transmission monitored through Wireshark in pcap format. Data collection was conducted over durations of 5, 10, 15, 20, 25, and 30 minutes, with the number of publishers increasing according to the testing scheme, from 1 publisher to 4 publisher clients, while each testing scheme consistently used 2 subscriber clients.

3. RESULTS AND DISCUSSION

A. Subscriber Client

In Figure 5 and 6 Subscriber Client is a client that subscribes to an MQTT topic where sensor data will be sent. The subscriber client will display the sensor data that has been successfully sent via the MQTT protocol.

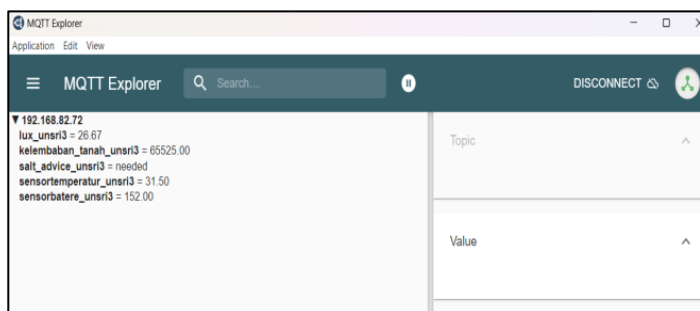


Fig. 5. Sensor Data on MQTT Explorer



Fig. 6. Sensor Data on MyMQTT

B. Capturing Wireshark

The transmission of sensor data will be captured in the Wireshark application as shown in Figure 7 below. Then, QoS parameters will be calculated using equations according to the testing parameters.

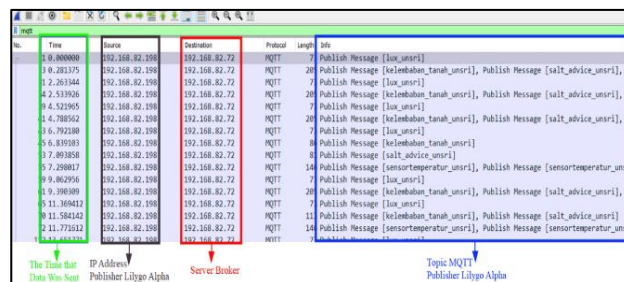


Fig. 7. Capturing Wireshark

C. QoS Testing Results

In capturing data packets through Wireshark, the following results were obtained in the Tables 4, 5, 6 and 7:

a. Length of Data Packet (Bytes)

Table 4. Length of Data Package

Testing	Length of Data Package (Bytes)			
	1 Pub	2 Pub	3 Pub	5 Pub
1	166198	274884	393082	496424
2	320040	550583	803243	983666
3	471595	819337	1328750	1468512
4	634869	1085486	1682506	1959203
5	802823	1354665	2140485	2446324
6	963987	1631609	2595536	2936166

b. Duration of observation (s)

Table 5. Duration of observation (s)

Testing	Duration of observation (s)			
	1 Pub	2 Pub	3 Pub	5 Pub
1	300,110	300,051	299,939	300,054
2	600,018	599,917	599,940	599,993
3	899,793	899,543	899,941	900,072
4	1199,37	1199,37	1199,99	1199,77
5	1499,25	1500,02	1499,80	1499,80
6	1801,12	1802,60	1801,67	1800,66

c. Total data packets sent

Table 6. Total data packets sent

Testing	Total data packets sent			
	1 Pub	2 Pub	3 Pub	5 Pub
1	1963	2958	4038	5180
2	3714	5937	8247	10217
3	5427	8797	12547	15197
4	7334	11613	17271	20287
5	9341	14485	21980	25303
6	11211	17466	26657	30332

d. Total data packets received

Table 7. Total data packets received

Testing	Total data packets received			
	1 Pub	2 Pub	3 Pub	5 Pub
1	1947	2949	4028	5171
2	3698	5925	8231	10202
3	5410	8780	12526	15175
4	7317	11591	17241	20252
5	9300	14452	21934	25259
6	11166	17429	26597	30274

D. Results of QoS Category Calculation

Some data obtained through Wireshark will be used to find the values of throughput, PDR, Packet Loss, and Delay as follows:

a. Throughput

The increase in throughput indicates that the system is capable of handling more data as the number of publishers increases. This indicates good scalability of the MQTT protocol in the smart farming monitoring system in the [Table 8](#).

Table 8. Throughput Calculation

Testing	Throughput (bps)			
	1 Pub	2 Pub	3 Pub	5 Pub
1	4430,3	7328,9	10484,3	13235,5
2	4267	7342,1	10710,97	13115,6
3	4192,9	7286,6	11811,8	13052,3
4	4234,6	7240,3	11216,7	13063,7
5	4283,8	7224,7	11417,3	13048,7
6	4281,7	7241	11524,9	13044,8
Average	4281,7	7277,3	11194,3	13093,5

b. Packet Delivery Ratio

The high PDR value indicates that almost all data packets were successfully transmitted without loss. This demonstrates the reliability of the MQTT protocol in transmitting sensor data in the [Table 9](#).

Table 9. Packet Delivery Ratio Calculation

Testing	PDR (%)			
	1 Pub	2 Pub	3 Pub	5 Pub
1	99,1850	99,6957	99,7524	99,8263
2	99,5690	99,7979	99,8060	99,8532
3	99,6870	99,8068	99,8326	99,8552
4	99,7680	99,8106	99,8263	99,8275
5	99,5610	99,7722	99,7907	99,8261
6	99,5990	99,7882	99,7749	99,8088
Average	99,5615	99,7785	99,7972	99,8328

c. Packet Loss

The decrease in packet loss indicates that the system becomes more efficient in sending data as the number of publishers increases. This also shows [Table 10](#) that the MQTT protocol is capable of reducing packet loss even as network load increases.

Table 10. Packet Loss Calculation

Testing	Packet Loss (%)			
	1 Pub	2 Pub	3 Pub	5 Pub
1	0,8150	0,3043	0,2476	0,1737
2	0,4310	0,2021	0,1940	0,1468
3	0,3130	0,1932	0,1674	0,1448
4	0,2320	0,1894	0,1737	0,1725
5	0,4390	0,2278	0,2093	0,1739
6	0,4010	0,2118	0,2251	0,1912
Average	0,4385	0,2215	0,2028	0,1672

d. Delay

The decrease in delay indicates that the system becomes more responsive as the number of publishers increases. This shows [Table 11](#) that the MQTT protocol is capable of reducing latency in data transmission, which is very important for real-time monitoring systems.

Table 11. Delay Calculation

Testing	Delay			
	1 Pub	2 Pub	3 Pub	5 Pub
1	154,1400	101,7469	74,4635	58,0264
2	162,2550	101,2519	72,8879	58,8113
3	166,3205	102,4537	71,8459	59,3128
4	163,9161	103,4748	69,6015	59,2424
5	161,2106	103,7934	68,3782	59,3769
6	161,3043	103,4259	67,7397	59,4788
Average	161,5244	102,6911	70,8195	59,0415

These results indicate that the MQTT protocol is very suitable for use in smart farming monitoring systems, as it can provide excellent performance in terms of speed, reliability, and responsiveness. With the results of the QoS category calculations above, based on the QoS parameter value categories, the following categories were obtained:

Table 12. Throughput Category

Publisher	Throughput	Category
1 Publisher	4281,7 bps	Very Good
2 Publisher	7277,3 bps	Very Good
3 Publisher	11194,3 bps	Very Good
4 Publisher	13093,5 bps	Very Good

Table 13. PDR Category

Publisher	PDR	Category
1 Publisher	99,5615 %	Very Good
2 Publisher	99,7785 %	Very Good
3 Publisher	99,7972 %	Very Good
4 Publisher	99,8328 %	Very Good

Table 14. Packet Loss Category

Publisher	Packet Loss	Category
1 Publisher	0,4385 %	Very Good
2 Publisher	0,2215 %	Very Good
3 Publisher	0,2028 %	Very Good
4 Publisher	0,1672 %	Very Good

Table 15. Packet Delay Category

Publisher	Delay	Category
1 Publisher	161,5244	Very Good
2 Publisher	102,6911	Very Good
3 Publisher	70,8195	Very Good
4 Publisher	59,0415	Very Good

Based on the calculation results of the QoS parameters, the more publishers there are, the greater the average throughput (bps) produced (Table 12,13,14 and 15). The

more publishers there are, the smaller the average delay will be. The packet delivery ratio and packet loss values are closely related; if the packet delivery ratio value is low, the packet loss value will be high, whereas if the packet loss value is low, the packet delivery ratio value will be high. The more publishers there are, the higher the average PDR, and the lower the packet loss value. The highest average throughput occurs under the condition of 4 publishers, which is 13093.5 bps, and the lowest under the condition of 1 publisher, which is 4281.7 bps. The highest average packet delivery ratio occurs under the condition of 4 publishers, which is 99.8328%, and the lowest under the condition of 1 publisher, which is 99.5615%. The highest average packet loss occurs under the condition of 1 publisher, which is 0.4385%, and the lowest under the condition of 4 publishers, which is 0.1672%. The highest average delay occurs under the condition of 1 publisher, which is 161.5244 ms, and the lowest under the condition of 4 publishers, which is 59.0415 ms.

4. CONCLUSION

The real-time monitoring system for garden conditions can be implemented by applying the MQTT protocol, with sensor data displayed using the MyMQTT application on mobile phones and the MQTT-Explorer application on laptops. QoS testing was conducted with parameters measuring throughput, packet delivery ratio, packet loss, and delay. Based on the QoS parameter categories from TIPHON, both the 1 publisher, 2 publisher, 3 publisher, and 4 publisher conditions fall into the very good category. Overall, the test results show that the MQTT protocol is very effective in smart farming monitoring systems. The more publishers used, the better the system's performance in terms of throughput, PDR, packet loss, and delay. This shows that MQTT is a scalable and reliable protocol for IoT applications in smart agriculture.

REFERENCES

- [1] C. Patel and N. Doshi, "a Novel MQTT Security framework in Generic IoT Model," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1399–1408, 2020, doi: 10.1016/j.procs.2020.04.150.
- [2] M. El Basioni, B. M. Mohammad, and E. Basioni, "A conceptual modeling approach of MQTT for IoT - based systems," *J. Electr. Syst. Inf. Technol.*, 2024, doi: 10.1186/s43067-024-00181-x.
- [3] M. A. Khan *et al.*, "A deep learning-based intrusion detection system for mqtt enabled iot," *Sensors*, vol. 21, no. 21, pp. 1–25, 2021, doi: 10.3390/s21217016.
- [4] A. K. Pandey and A. Mukherjee, *A Review on Advances in IoT-Based Technologies for Smart Agricultural System*, vol. 99. 2021. doi:

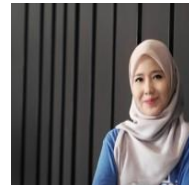
- 10.1007/978-981-16-6210-2_2.
- [5] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A Review on the Study on MQTT Security Challenge," *Proc. - 2020 IEEE Int. Conf. Smart Cloud, SmartCloud 2020*, pp. 128–133, 2020, doi: 10.1109/SmartCloud49737.2020.00032.
- [6] S. De Alwis, Z. Hou, Y. Zhang, M. H. Na, B. Ofoghi, and A. Sajjanhar, "A survey on smart farming data, applications and techniques," *Comput. Ind.*, vol. 138, p. 103624, 2022, doi: 10.1016/j.compind.2022.103624.
- [7] F. Palmese, A. E. C. Redondi, and M. Cesana, "Adaptive Quality of Service Control for MQTT-SN," *Sensors*, vol. 22, no. 22, 2022, doi: 10.3390/s22228852.
- [8] M. A. Spohn and W. B. Genero, "Análise experimental dos protocolos MQTT e MQTT-SN," *Rev. Bras. Comput. Apl.*, vol. 15, no. 1, pp. 22–33, 2023, doi: 10.5335/rbca.v15i1.13510.
- [9] H. J. Jara Ochoa, R. Peña, Y. Ledo Mezquita, E. Gonzalez, and S. Camacho-Leon, "Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Implemented for Remote Real-Time Monitoring of Long-Term Cold Chain Transport Operations," *Sensors*, vol. 23, no. 10, 2023, doi: 10.3390/s23104896.
- [10] A. Alatram, L. F. Sikos, M. Johnstone, P. Szweczyk, and J. J. Kang, "DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol," *Comput. Networks*, vol. 231, no. March, p. 109809, 2023, doi: 10.1016/j.comnet.2023.109809.
- [11] M. Has, D. Kreković, M. Kušek, and I. Podnar Žarko, "Efficient Data Management in Agricultural IoT: Compression, Security, and MQTT Protocol Analysis," *Sensors*, vol. 24, no. 11, 2024, doi: 10.3390/s24113517.
- [12] M. Kashyap, A. K. Dev, and V. Sharma, "Implementation and analysis of EMQX broker for MQTT protocol in the Internet of Things," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 10, no. May, p. 100846, 2024, doi: 10.1016/j.prime.2024.100846.
- [13] E. Nwankwo, M. David, and E. N. Onwuka, "Integration of MQTT-SN and CoAP protocol for enhanced data communications and resource management in WSNs," *Bull. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 1613–1620, 2024, doi: 10.11591/eei.v13i3.5158.
- [14] F. YALÇINKAYA, H. AYDİLEK, M. Y. ERTEN, and N. İNANÇ, "IoT based Smart Home Testbed using MQTT Communication Protocol," *Uluslararası Muhendis. Arastırma ve Gelistirme Derg.*, p. 317, 2020, doi: 10.29137/umagd.654056.
- [15] S. J. Suneetaasatpathy, editors, *EAI/Springer Innovations in Communication and Computing Internet of Things and Its Applications*. [Online]. Available: <http://www.springer.com/series/15427>
- [16] A. D. Boursianis *et al.*, "Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review," *Internet of Things (Netherlands)*, vol. 18, p. 100187, 2022, doi: 10.1016/j.iot.2020.100187.
- [17] A. Piancharoenwong and Y. F. Badir, "IoT smart farming adoption intention under climate change: The gain and loss perspective," *Technol. Forecast. Soc. Change*, vol. 200, no. June 2023, p. 123192, 2024, doi: 10.1016/j.techfore.2023.123192.
- [18] C. D'ortona, D. Tarchi, and C. Raffaelli, "Open-Source MQTT-Based End-to-End IoT System for Smart City Scenarios," *Futur. Internet*, vol. 14, no. 2, 2022, doi: 10.3390/fi14020057.
- [19] R. Alasmari and A. A. Alhogail, "Protecting Smart-Home IoT Devices From MQTT Attacks: An Empirical Study of ML-Based IDS," *IEEE Access*, vol. 12, no. February, pp. 25993–26004, 2024, doi: 10.1109/ACCESS.2024.3367113.
- [20] F. Azzedin and T. Alhazmi, "Secure Data Distribution Architecture in IoT Using MQTT," *Appl. Sci.*, vol. 13, no. 4, pp. 1–13, 2023, doi: 10.3390/app13042515.
- [21] G. Alfian, M. Syafrudin, and J. Rhee, "Real-time monitoring system using smartphone-based sensors and NoSQL database for perishable supply chain," *Sustain.*, vol. 9, no. 11, 2017, doi: 10.3390/su9112073.
- [22] R. Herrero, "RTP transport in IoT MQTT topologies," *Internet Things Cyber-Physical Syst.*, vol. 3, no. January, pp. 37–44, 2023, doi: 10.1016/j.iotcps.2023.02.001.
- [23] S. ul A. Laghari, W. Li, S. Manickam, P. Nanda, A. K. Al-Ani, and S. Karuppayah, "Securing MQTT Ecosystem: Exploring Vulnerabilities, Mitigations, and Future Trajectories," *IEEE Access*, no. May 2024, pp. 139273–139289, 2024, doi: 10.1109/ACCESS.2024.3412030.
- [24] Y. Akkem, S. K. Biswas, and A. Varanasi, "Smart farming using artificial intelligence: A review," *Eng. Appl. Artif. Intell.*, vol. 120, no. September 2022, p. 105899, 2023, doi: 10.1016/j.engappai.2023.105899.
- [25] H. H. Alshammari, "The internet of things healthcare monitoring system based on MQTT protocol," *Alexandria Eng. J.*, vol. 69, pp. 275–287, 2023, doi: 10.1016/j.aej.2023.01.065.
- [26] M. McCaig, R. Dara, and D. Rezania, "Farmer-

centric design thinking principles for smart farming technologies,” *Internet of Things (Netherlands)*, vol. 23, no. August, p. 100898, 2023, doi: 10.1016/j.iot.2023.100898.

Engineering, Faculty of Computer Science, Sriwijaya University, Indonesia.



Achmad Andriansyah Aljali Computer Engineering Study Program at the Faculty of Computer Science, Sriwijaya University



Nurul Afifah received the Bachelor Degree in Computer Engineering and master's degrees in Computer Science at 2014 and 2019. She joined Universitas Sriwijaya as a Lecturer in January 2020. She is currently a Researcher at COMNETS RG. Her research interests include Information Security, IoT system & security, Blockchain and Machine learning.



Septiani K Ningrum graduated from the Faculty of Computer Science, Universitas Sriwijaya. Now, she is researcher at Comnets RG Unsri concentration of infosec, especially in the field of lateral movement.

AUTHOR BIOGRAPHY



Adi Hermansyah Completed a Bachelor's degree in Computer Systems at the Faculty of Computer Science, Sriwijaya University in 2014, and a Master's degree in Engineering with a specialization in Intelligent Multimedia Networks, Electrical Engineering, at the Sepuluh November Institute of Technology in 2019. Since 2019, he has been a permanent lecturer in the Computer Engineering Study Program at the Faculty of Computer Science, Sriwijaya University. Researcher at COMNETS RG Since 2025, he is an IEEE member and ACM Student Member.



Huda Ubaya, was born in Yogyakarta on June 16, 1981. He obtained his M.Sc. and B.Sc. in Computer Engineering from the Institut Teknologi Bandung, School of Electrical Engineering and Informatics. Currently, he is in the process of completing his doctorate at the Faculty of Engineering, Sriwijaya University. His research interests include Mobile and Pervasive Computing, Internet of Everything, Embedded Systems, Distributed Ledger Technology, and Blockchains. Currently, he is a lecturer in Department of Computer