Data Communication System from Attendance Machines to Server Using MQTT Communication

Revino Jantri Putra[1], Rifqi Amalya Fatekha[1], Rifky Afriza[1], Senanjung Prayoga[1]

Jurusan Teknik Elektro, Prodi Teknik Rekayasa Robotika, Politeknik Negeri Batam, Batam, Indonesia

[*]Email: the corresponding author's e-mail address

Received on dd-mm-yyyy | *Revised on dd-mm-yyyy* | *Accepted on dd-mm-yyyy*

***Abstract*— This study describes the design and evaluation of a data communication system on a fingerprint-based attendance device that uses the MQTT protocol. This system integrates Arduino Mega 2560, DY50 fingerprint sensor, ESP32 module as an MQTT publisher, and RS485 as an inter-device communication channel. The results of the fingerprint registration and verification process are sent to the MQTT broker, then forwarded to a web-based server for data storage and monitoring purposes.**

The communication performance of the system is evaluated using four Quality of Service (QoS) parameters. All measurement results are normalized based on the TIPHON standard to determine the appropriate service quality category. The testing was conducted using six different fingerprints with six trials for each activity. The test results showed that these activities did not affect system performance, as the fingerprint data was sent intermittently with a payload size of 768 bytes per template.

The evaluation results show that the system has stable throughput, low latency, minimal jitter, and a near-zero packet loss rate during the communication process. The Final Score (FS) obtained is in the range of 3.21 to 3.54, which indicates that the system is capable of providing communication performance in the "Good" to "Very Good" category based on TIPHON criteria.

Keywords: MQTT, Fingerprint, IoT, QoS TIPHON

I. INTRODUCTION

The development of Internet of Things (IoT) technology in recent years has driven major changes in the patterns of interaction between physical devices. Through IoT, various devices can be integrated to communicate and exchange data in real time, thereby supporting the implementation of automation, remote monitoring, and more efficient and accurate data-driven decision making [3], [4]. As the number of connected devices increases, communication systems are required to be able to work reliably in various network conditions, including environments with limited bandwidth and unstable connections [5]. Therefore, the selection of communication protocols is a crucial aspect because it directly affects service quality, especially in systems that require timing accuracy and data integrity.

One of the communication protocols widely used in the IoT ecosystem is Message Queuing Telemetry Transport (MQTT) [1], [6]. This protocol is designed to support lightweight communication with low resource consumption, making it suitable for use on devices with limited memory and computing power. MQTT adopts a publish–subscribe mechanism, which enables structured data exchange with minimal communication overhead [1]. These characteristics enable MQTT to maintain stable communication performance even under fluctuating network conditions, making it a popular choice for microcontroller-based systems and monitoring applications that require fast and continuous data transmission [2], [7], [11]. Compared to request–response-based protocols, MQTT is considered more effective in supporting real-time communication needs.

Previous studies have implemented MQTT in various IoT applications, such as smart homes, environmental monitoring, smart agriculture, and security systems [8]. However, most of these studies focus more on the functional aspects of the system or the success of data transmission, without a comprehensive evaluation of communication performance [9]. In IoT-based biometric authentication systems, existing studies generally only emphasize the success rate of the user verification process, while technical analysis related to communication quality is still relatively limited [4], [9], [12], [13]. In fact, Quality of Service (QoS) parameters such as throughput, latency, jitter, and packet loss play an important role in assessing communication stability and reliability. These parameters are determining factors in ensuring that biometric data, especially fingerprints, can be sent and processed in a timely manner without packet loss that could potentially interfere with the authentication process [14].

This study develops an IoT-based fingerprint attendance communication system using Arduino Mega 2560, a DY50 fingerprint sensor, ESP32 as an MQTT publisher, and RS485 as the inter-device communication medium [3], [6], [10]. Fingerprint registration and verification data are transmitted to an MQTT broker and forwarded to a web server for storage and visualization [8], [11]. Communication performance is evaluated using four QoS parameters normalized with the TIPHON standard to classify service quality into Very Good, Good, Fair, and Poor categories [12], [15]. The analysis focuses on service quality during the registration and verification processes in terms of communication stability, connection consistency, and data delivery reliability under specific network conditions. The results are expected to serve as a reference for the development of IoT-based biometric systems requiring fast, stable, and reliable communication.

I. METHOD

The system implemented in the fingerprint registration and verification process was designed using the MQTT (Message Queuing Telemetry Transport) communication protocol [1], [6], [11]. Testing was conducted to assess the level of efficiency and reliability of communication between IoT devices with reference to the Quality of Service (QoS) parameters set based on the TIPHON standard [4], [12], [13]. During the testing, four main QoS parameters, namely throughput, latency, jitter, and packet loss, were automatically recorded as communication performance indicators [4], [9], [12]. The measurement data was then sent to the server and stored in a database for averaging [8]. These average values were then compared with the service quality thresholds according to the TIPHON standard, which categorizes network performance into Excellent, Good, Fair, and Poor [12], [15]

A. System Architecture

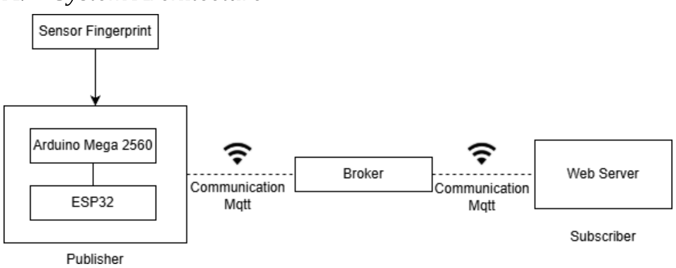

Figure 1. MQTT communication system diagram

The communication system developed in this study implements a publisher and subscriber architecture using the MQTT protocol as a data exchange mechanism [1], [5], [7]. The fingerprint attendance system consists of four main components, namely the DY50 fingerprint sensor, Arduino Mega 2560, RS485 communication module, and ESP32, which acts as an MQTT publisher. The DY50 sensor captures the user's fingerprint image and generates a biometric template, which is then processed by the Arduino Mega as the main controller of the system. The Arduino Mega is responsible for managing all stages of fingerprint registration and verification [3], [14].

Data exchange between Arduino Mega 2560 and ESP32 is carried out via RS485 communication. This communication medium was chosen because it has a high level of resistance to electromagnetic interference and is capable of supporting data transmission over longer distances compared to TTL-based serial communication. Data is sent in the form of frames with a specific format to reduce the possibility of errors during the transmission process. After the data is received, the ESP32 reprocesses the payload into a structured JSON-based format, which contains user identity information, fingerprint verification results, and four Quality of Service (QoS) parameters generated during each registration and verification process.

In MQTT architecture, the broker acts as an intermediary that manages data distribution between publishers and subscribers. The web server acts as a subscriber to receive data from the broker, store it in a database, and display the information through a web-based monitoring dashboard. This dashboard enables real-time monitoring of user activity and QoS parameter test results [1], [6], [11]. The overall system design focuses on reducing latency and jitter values while maintaining throughput stability during the data communication process [4], [9], [12].

B. System Flow Process

The system process flow is used to describe the overall workflow of the system, starting from user interaction to the process of sending data to the server, which shows the two main processes in the system.

Figure 2 illustrates the workflow of the fingerprint registration and verification system on Arduino Mega integrated with MQTT communication. The system flow consists of two main modes, namely enrollment and verification.

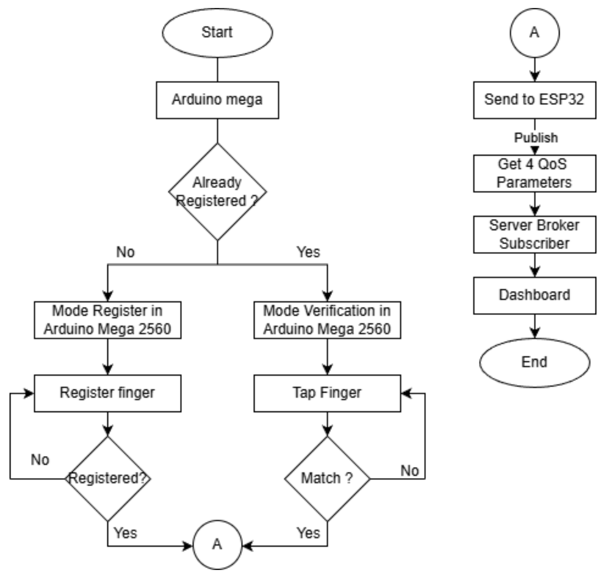

Figure 2. System Flow Process

In registration mode, users first enter their ID, then scan their fingerprints until the system confirms that the registration process has been successful. If the scan results do not meet the storage criteria, the system will automatically ask the user to repeat the scanning process until the fingerprint template is successfully stored. Meanwhile, in verification mode, users only need to scan their fingerprints once to match the data read with the previously stored template. If the match is unsuccessful, the system will ask for a rescan until the verification process is successful. Once both registration and verification are complete, the processed data is sent from the Arduino Mega to the ESP32 via the RS485 communication channel. Next, the ESP32 functions as a publisher that forwards the data to the MQTT broker. The server acts as a subscriber to receive data from the broker, process and evaluate Quality of Service (QoS) parameters, including throughput, latency, jitter, and packet loss, and display test results and system activity through the monitoring dashboard.

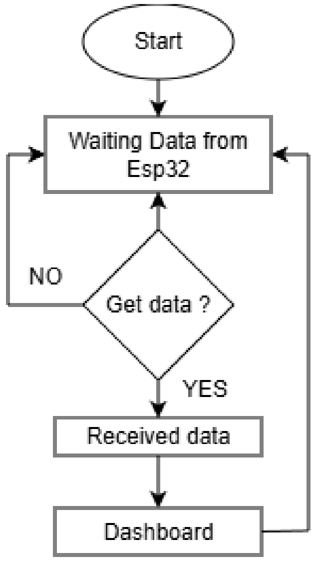

Figure 3. Data Processing Flow

Flowchart in Figure 3 shows the process flow for retrieving Quality of Service (QoS) parameters in Figure 2 of the system flow process. The process begins when the server is waiting for data sent by ESP32 via the MQTT protocol. Once the data is received, the system processes four QoS parameters, namely latency, jitter, throughput, and packet loss. Next, the results of this data processing are displayed on the dashboard for the purpose of monitoring the system's communication performance in real time.

C. *Quality of Service (QoS)*

Quality of Service (QoS) is used as an approach to evaluate network service quality by referring to a number of key parameters, namely throughput, latency, jitter, and packet loss [4], [9], [12]. These four parameters represent different aspects of network performance and are used to assess the stability and reliability of data communication [11].

a. Throughput

Throughput describes a network's ability to transmit data effectively within a certain time frame. A higher throughput value indicates that the network is capable of transmitting data better and more efficiently [11]. Throughput is calculated using Equation (1), where the total data received represents the amount of data successfully delivered to the destination measured in bits, and the transmission time denotes the total duration required to transmit the data measured in seconds.

$$Throughput = \frac{Total\ data\ received\ (Bit)}{Transmission\ time(Second)} \tag{1}$$

b. Latency

Latency or delay is defined as the delay time required for data to move from the sender to the receiver. The smaller the latency value, the faster the communication system response [12]. The latency value is calculated using Equation (2), where $t_{response\ received}$ the time at which the response data is successfully received by the receiver, and $t_{request\ sent}$ denotes the time at which the request data is sent by the sender. The difference between these two timestamps indicates the total delay experienced during the data transmission process.

$$Latency = t_{response\ received-}\ t_{request\ sent} \tag{2}$$

c. Jitter

Jitter is the variation in packet arrival time that occurs due to differences in delay between data packets during the transmission process. A low jitter value indicates a better level of network stability [4], [11]. Jitter is calculated using Equation (3), where the total delay variation represents the accumulated differences in delay between consecutive data packets during transmission, and the total packets received − 1 denotes the number of delay intervals considered in the measurement. The resulting value reflects the average variation in packet delay within the communication network.

$$Jitter = \frac{Total\ \ delay\ \ variation}{Total\ packets\ receiverd-1} \tag{3}$$

d. Packet Loss

Packet loss indicates the percentage of data packets that fail to be received by the system during transmission. The higher the packet loss value, the lower the quality of the resulting network service [2], [5], [15]. The packet loss value is calculated using Equation (4), where the difference between the total packets sent and the total packets received represents the number of lost packets during transmission, while the total packets sent denotes the overall number of packets transmitted. The result is expressed as a percentage to indicate the proportion of packet loss within the communication network.

$$Packet\ Loss = \frac{packet\ sent -\ packet\ received}{Total\ packets\ sent}\ X\ 100\% \tag{4}$$

D. *Standard TIPHON*

The data collection process was carried out through seven tests involving seven fingers from different individuals. In each test session, the system generated four Quality of Service (QoS) parameter values, namely throughput, latency, jitter, and packet loss [4], [9], [12]. All measurement data from each experiment was then collected and analyzed to evaluate the system performance on each finger tested. To simplify the calculation process and grouping of values for each QoS parameter, the test data was processed using Microsoft Excel [8].

TABLE I
QOS PARAMETER CLASSIFICATION TIPHON STANDARD

Parameter	Very Good (Grade 4)	Good (Grade 3)	Fair (Grade 2)	Poor (Grade 1)
Throughput	>2048 Kbps	1024 – 2047 Kbps	512 - 1023 Kbps	0 - 512 Kbps
Latency	0 – 150 ms	151 – 300 ms	301 – 450 ms	>450 ms
Jitter	0 ms	1 – 50 ms	51 – 75 ms	>75 ms
Packet Loss	0 – 1 %	2 – 2,5 %	2,5 – 5 %	>5 %

The TIPHON standard uses a rating scale with an index of 1 to 4, where each index represents a specific service quality category.

- **Index 4 (Very Good)** indicates optimal and stable service quality.
- **Index 3 (Good)** indicates that service quality is still within normal and acceptable limits.
- **Index 2 (Fair)** indicates minor disruptions, but the service is still usable.
- **Index 1 (Poor)** indicates unacceptable service quality due to frequent disruptions or data loss.

This index classification is then used to interpret the average values of the four QoS parameters during the testing process [11], [12].

After each Quality of Service (QoS) parameter is classified into an index according to the TIPHON standard category, the next step is to calculate the average index value of the four parameters in each experiment. The average index value is calculated using Equation (5), by combining the index throughput (I_T), latency (I_L), jitter (I_J), and packet loss (I_P).

$$\bar{X} = \frac{I_T + I_L + I_J + I_P}{4} \quad (5)$$

Next, the Final Score (FS) is calculated to represent the overall service quality. The Final Score is obtained by calculating the average of all index scores in each experiment using equation (6).

$$FS = \frac{\sum_{i}^{n} = 1\, \mathcal{X}_i}{n} \quad (6)$$

The Final Score (FS) obtained is then classified into service quality categories based on the TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) standard, as shown in Table II.

TABLE II
TIPHON STANDARD QUALITY CATEGORY

Quality Category	Final Score (FS)	Index
Very Good	3,30 – 4,00	4
Good	2,50 – 4,29	3
Fair	1,75 – 2,49	2
Poor	0 – 1,74	1

The TIPHON standard is not only used as a reference for service quality classification, but also plays an important role in ensuring that network performance evaluation processes are carried out objectively and consistently.

This standard was developed by ETSI as an international guideline for assessing IP-based network performance, including IoT communication systems. By setting numerical limits for each QoS parameter, measurement results can be grouped into clear and measurable service quality categories. The application of the TIPHON standard in this study allows the evaluation process to be conducted transparently, structurally, and easily compared with other studies using similar indicators. In addition, the use of TIPHON helps ensure that communication system performance is accountable, especially in critical processes such as biometric authentication, which requires high precision and reliability.

II. RESULT AND DISCUSSION

This chapter discusses the results of data communication testing on an MQTT-based fingerprint attendance system and network performance analysis based on four Quality of Service (QoS) parameters. Service quality assessment was carried out using the TIPHON standard to classify the level of communication performance produced.

TABLE III
RESULTS VALUES OF ACTIVITY

Id	Activity	Throughput (Kbps)	Latency (ms)	Jitter (ms)	Packet Loss (%)
1	Enroll	387,15	33,00	30,03	1,69
2	Enroll	632,41	20,00	14,56	0,87
3	Enroll	442,90	30,00	9,80	0,36
4	Enroll	834,70	15,00	17,25	2,49
5	Enroll	1,354,26	10,00	5,55	1,91
6	Enroll	1,213,72	10,00	0	2,40
7	Verify	399,75	32,00	25,52	0,12

One of the results of service quality testing on the ESP32-based fingerprint attendance system using MQTT communication is shown in Table III, where the values for the latency, jitter, throughput, and packet loss parameters have not been processed for the two main activities: registration and verification. All data was obtained by sending 768 bytes in each of 7 fingerprint trials with 7 activities to determine the parameter values for each activity.

A. QoS Parameter test results

The initial stage of data processing is carried out by calculating the average value of each Quality of Service (QoS) parameter for each fingerprint registration and verification activity. Next, each parameter is classified into a service quality category based on the TIPHON standard, which uses a rating index with a value range of 1 to 4. After the indices for the throughput, latency, jitter, and packet loss parameters are obtained, the next step is to calculate the average index value for each activity using Equation (5). This average index value represents the communication service quality for a single registration or verification process that has been normalized to the TIPHON scale [12].

The results of the average index value calculations are then presented in Table IV, which shows the performance of each User ID in all testing activities. All calculation stages were carried out using the same procedure so that the average values of the four QoS parameters for each activity tested were obtained.

TABLE IV
AVERAGE SCORE FOR EACH ACTIVITY

Id	Enroll 1	Enroll 2	Enroll 3	Enroll 4	Enroll 5	Enroll 6	**Verify**
1	2,75	3,50	3,25	3,25	3,50	3,50	3,00
2	3,00	3,00	3,50	3,75	3,50	3,25	3,50
3	3,50	3,50	2,75	3,00	3,50	3,50	3,50
4	3,75	2,75	3,25	3,25	3,75	3,50	3,75
5	3,25	3,75	3,25	3,75	3,50	3,75	3,50
6	2,75	3,25	3,50	3,75	3,00	3,25	3,25
7	3,75	3,50	3,25	3,25	3,75	2,50	2,50

Based on the calculation results obtained, the throughput value shows stability in all testing activities, while the latency and jitter parameters are at relatively low levels. These conditions indicate that the data transmission process between the Arduino Mega 2560, DY50 fingerprint sensor, and ESP32 can run smoothly without significant delays. In addition, the very small packet loss value indicates that data packet loss during the communication process using the MQTT protocol can be minimized effectively.

The stability of the throughput value and low latency are also influenced by the relatively small size of the MQTT payload, as the data sent only includes user identity, verification results, and four Quality of Service (QoS) parameters. This light transmission load allows the data transmission process to run faster and more efficiently. In addition, the use of RS485 as a communication medium between devices provides a high level of reliability against signal interference, so that variations in delay or jitter can be consistently suppressed throughout the entire test series.

B. Calculation Final Score (FS)

After the average value of each Quality of Service (QoS) parameter is obtained for each activity, the next step is to calculate the Final Value (FV) for each user. The final value is calculated by taking the average of all registration and verification activities performed on one user's finger, in accordance with Equation (6) described in Chapter II.

The calculation of the Final Score aims to provide a comprehensive overview of the quality of communication services received by each user during the registration and verification process. The results of the FS calculation are presented in Table V.

TABLE V
CLASSIFICATION FINAL SCORE

Id User	Final Score	NOTE
1	3,25	Good
2	3,36	Very Good
3	3,32	Very Good
4	3,43	Very Good
5	3,54	Very Good
6	3,25	Good
7	3,21	Good

To clarify the comparison of service quality between users, the final scores will be visualized in graph form. This visualization makes it easier to identify service quality categories based on TIPHON standards. The FS graph is shown in Figure 4.

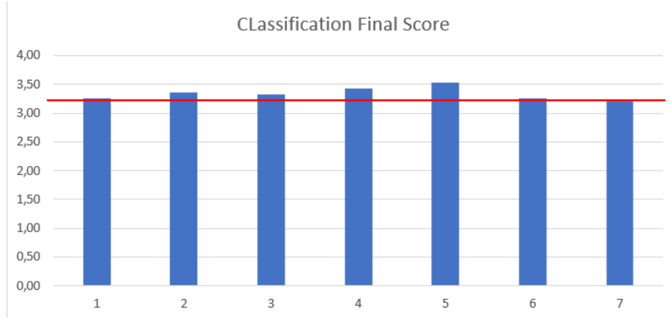

Figure 4. *Final Classification Score of Chart*

The graph in Figure 3 shows the Final Score (FS) for each User ID after four Quality of Service (QoS) parameters, namely throughput, latency, jitter, and packet loss, are normalized using the TIPHON index. A threshold line of 3.30 has been added to the graph, representing the minimum limit for the Excellent category based on the TIPHON standard.

In general, the FS obtained by all users ranged from 3.32 to 3.54. Based on these results, several key findings can be identified as follows:

- ID Users 2 to 5 have FS values above the threshold (>3.30), indicating that their communication performance is in the excellent category. This condition reflects the stability and efficiency of MQTT-based data communication during the registration and fingerprint verification processes.
- User IDs 1, 6, and 7 have values that are slightly below the threshold line, but still fall within the good category with a range of 3.21 to 3.25. This difference in values is generally caused by small fluctuations in throughput or jitter parameters. However, these values are still within normal service limits and do not have a significant impact on the overall reliability of the system.

Differences in FS values between users can be influenced by several external factors, such as the condition of the fingerprint during the scanning process (dry or oily), the pressure of the finger when placed on the sensor, and micro variations in device response time. These variations are normal and do not indicate a decline in overall system quality.

Based on the final value calculations and graphical visualizations obtained, it can be concluded that the MQTT-based communication system on fingerprint attendance devices is capable of providing stable, efficient, and reliable service quality.

C. Analysis System Quality based on Final Score (FS)

The final score (FS) obtained is then compared with the TIPHON standard classification to determine the service quality level. The TIPHON categories used are shown in Table VI.

TABLE VI
STANDARD FINAL CLASSIFICATION

Final Score	Category
3,30 – 4,00	Very Good
2,50 – 3,29	Good
1,75 – 2,49	Fair
0 – 1,74	Poor

Based on the calculations in Table 13, several important findings were obtained, as follows:

- Most User IDs have FS values above 3.30, which falls into the excellent category. This shows that the MQTT protocol is capable of providing stable and efficient communication performance during the registration and fingerprint verification processes.
- Some users have FS values in the range of 3.21–3.25, which still falls within the good category. This variation is generally caused by slight changes in throughput or packet loss, but does not have a significant impact on system reliability.
- No FS values were found in the fair or poor categories, so it can be concluded that the system's performance is of very good quality overall.

These values show that the MQTT communication system has excellent data transmission quality. Low latency, jitter, and packet loss, as well as stable throughput, are the main factors that contribute to the high final score. This reinforces that the publisher-subscriber architecture in MQTT is very suitable for IoT, which requires light and stable communication.

This research still has room for further development, especially in terms of improving data security. Implementing TLS-based MQTT encryption could be the next step in enhancing protection against network attacks. In addition, testing the system on different networks such as 4G, 5G, or public Wi-Fi can provide a broader picture of the system's resilience under various network conditions. System development can also be directed towards cloud database integration, increasing the number of users for load testing, and adding real-time usage analytics features to the monitoring dashboard.

III. CONCLUSION

Based on the research conducted, it has been proven that the MQTT-based fingerprint attendance machine data communication system can operate to obtain four QoS parameters, namely throughput, latency, jitter, and packet loss, where the overall user value ranges from 3.21 to 3.54. The average system value is 3.30, which is stable and efficient. The publisher-subscriber architecture used by MQTT is very suitable for IoT applications that require lightweight and real-time communication. All data was obtained by sending 768 bytes in each of 7 fingerprint trials with 7 activities to determine the parameter values for each activity. The limitation of this study is that testing was only conducted in one network environment.

Overall, the test results show that communication performance remains consistent throughout each fingerprint registration and verification process. This confirms that the combination of RS485, Arduino Mega 2560, ESP32, and MQTT technologies is capable of reliably supporting the transmission of biometric data. The stability of throughput and low jitter, latency, and packet loss reinforce that the MQTT protocol is the right choice for IoT systems that require high availability and fast response times.

Further research may include various network conditions, increased data traffic, and testing with a larger number of users to assess the system's resilience on a broader scale. Additionally, implementing additional security measures such as TLS on MQTT communication, using cloud-based servers, and integrating load testing can enrich the analysis results. System development can also be directed towards optimizing the monitoring dashboard and utilizing real-time analytics to support implementation in industrial and academic scenarios.

REFERENCES

[1] A. F. Gentile et al., "A Network Performance Analysis of MQTT Security Protocols with Constrained Hardware in the Dark Net for DMS," *Applied Sciences*, vol. 14, no. 18, pp. 9345–9358, 2024

[2] N. Afiyah, R. H. Navilla, and M. Hariyadi, "Sistem Monitoring Cairan Infus Berbasis IoT Menggunakan Protokol MQTT," *JNTETI*, vol. 12, no. 1, 2023, pp. 44–52..

[3] L. Amalia and M. Yuwono, "IoT-Based Fingerprint Authentication System Using ESP32," *Indonesian Journal of Electronics and Instrumentation Systems*, vol. 13, no. 3, 2023, pp. 155–164.

[4] S. Palani, et al., "Comprehensive QoS Measurement for Real-Time IoT Communication," *Sensors*, vol. 22, 2022, pp. 1–15.

[5] J. P. Santos and R. Oliveira, "Performance Evaluation of MQTT Brokers in IoT Networks," *IEEE Access*, vol. 10, 2022, pp. 11599–11610.

[6] R. Li, et al., "Lightweight IoT Authentication Using Fingerprint and Edge Computing," *Sensors*, vol. 22, no. 14, 2022, pp. 5401–5416.

[7] M. L. Putra, D. Arifianto, and R. Rasyid, "Performance Evaluation of MQTT under Various Network Conditions," *Journal of Network and Computer Applications*, vol. 192, 2021, pp. 102–109.

[8] R. K. Desai and A. Vala, "IoT-Based Biometric Attendance System Using ESP32 and Cloud Services," *International Journal of Engineering Research*, vol. 10, no. 5, 2021, pp. 451–455.

[9] A. Singh and S. Tyagi, "Analysis of QoS Parameters in MQTT Protocol for IoT Applications," *International Journal of Advanced Networking and Applications*, vol. 12, no. 3, 2021, pp. 467–474.

[10] M. S. Ibrahim and H. Zaini, "Enhancing Industrial Data Transmission Using RS485 in Distributed IoT Systems," *IEEE Access*, vol. 8, 2020, pp. 124223–124231.

[11] R. García, et al., "Optimized MQTT Communication for Real-Time IoT Applications," *Sensors*, vol. 20, no. 9, 2020, pp. 1–18.

[12] R. Sharma and Y. Pandey, "Delay and Packet Loss Analysis in MQTT Protocol for IoT," *International Journal of Electronics and Telecommunications*, vol. 66, no. 2, 2020, pp. 321–328.

[13] T. Mahmood, et al., "Efficiency of MQTT for Resource-Constrained IoT Devices," *Journal of Communication Systems*, vol. 32, no. 10, 2019, pp. 1881–1894.

[14] H. Abdullah and M. Fahmi, "Design of Fingerprint-Based Access Control with ESP-Series Microcontroller," *Journal of Electrical and Electronics Engineering*, vol. 14, no. 2, 2019, pp. 87–94.

[15] M. M. Rahman, et al., "Real-Time IoT Communication Analysis Using MQTT and Web Server," *Sensors and Systems*, vol. 8, 2019, pp. 221–230.