Analysis of HTTP Communication Quality in Fingerprint Attendance Systems Based on QoS Standard TIPHON

Abdi Wijaya Sasmita[1], Rifqi Amalya Fatekha[2], Senanjung Prayoga[3], Rifky Afriza [4], and Ricard Riovaldo Simatupang[5]

[1]Department of Electrical Engineering, Robotics Engineering Technology, Politeknik Negeri Batam, Batam, Indonesia

[*]Email: abdi.wijaya.sasmita.07@gmail.com

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

***Abstract*— This study analyzes the performance of an Internet of Things-based fingerprint attendance system using an ESP32 and the HTTP communication protocol through a dual-layer measurement approach for quality of service evaluation. Performance analysis was conducted by comparing application layer logs obtained from the ESP32 with network layer measurements captured using Wireshark based on TIPHON standards. Results indicate that system latency ranges from 16.31 ms at the network layer to 158.17 ms at the application layer due to processing overhead, while maintaining tolerable jitter and 0% packet loss across all test scenarios. Although the measured throughput values were relatively low, this condition did not affect system performance because fingerprint data were transmitted intermittently with a payload size of 768 bytes per template. Overall, the results demonstrate that the proposed system is suitable for real-time operation in local network environments, while further studies should address communication security for wider network deployment.**

Keywords- ESP32, Fingerprint, HTTP, IoT, QoS TIPHON

I.INTRODUCTION

Fingerprint-based attendance systems have become an effective solution in attendance management due to their accuracy and ability to prevent data manipulation. Biometric technology, particularly fingerprint technology, enables unique and accurate validation of individual identities, overcoming the limitations of manual attendance systems that are prone to human error [1], [2]. In educational environments, the implementation of this system plays an important role in improving accountability and effectiveness in learning, as well as ensuring that network infrastructure is able to handle the load of academic traffic efficiently [3], [4], [5].

The implementation of a fingerprint attendance system at the Barelang Robotics and Artificial Intelligence Lab (BRAIL) at Batam State Polytechnic integrates hardware and software such as DY50 sensors, Arduino Mega, RS485 modules, and ESP32 to automatically record student attendance [6], [7]. Through IoT connectivity, fingerprint data is sent to the server using the HTTP protocol to be verified and stored in a MySQL database [8], [9]. Previous research on biometric systems has generally focused on optimizing matching algorithms to improve data accuracy and security [10]. Although this aspect is vital, evaluation of real-time data transmission performance is often overlooked.

In scenarios involving mass attendance, such as in schools or factories, network response delays and packet arrival instability can cause queue backlogs [11]. This is in line with findings that real-time IoT-based monitoring is crucial to prevent data transmission failures and maintain system efficiency [12]. The TIPHON standard recommends evaluating network-based system performance using measurable parameters such as latency, jitter, and packet loss [13], [14], [15].

This biometric system utilizes two-way communication based on the HTTP protocol: uplink to send fingerprint templates from ESP32 to the server, and downlink to receive verification results or top-match from the server back to the device [16]. This process creates different traffic patterns between registration, verification, and top-match activities, where repeated packet transmission during the enrollment stage produces a traffic pattern resembling burst traffic.

Current IoT research trends have widely adopted event-driven protocols like MQTT because they use bandwidth more efficiently. However, HTTP remains the dominant standard in Web of Things (WoT) architecture due to its broad compatibility with conventional server infrastructure (such as PHP/MySQL) without requiring additional middleware or brokers [17]. Therefore, this study maintains the use of HTTP to test its feasibility in a simple client-server ecosystem commonly used in educational institutions. The analysis was conducted using a dual-layer measurement approach: ESP32 firmware logging for the application layer and Wireshark for the network layer to obtain comprehensive data [18], [19]. This approach enabled end-to-end analysis of the performance of two-way HTTP communication in biometric attendance systems.

The performance of an Internet of Things-based fingerprint attendance system using ESP32 and the HTTP protocol was analyzed and evaluated through a dual-layer measurement approach based on TIPHON standard Quality of Service parameters.

II. METHOD

A. System Architecture

The architecture of the Internet of Things-based fingerprint attendance system is shown in Figure 1. This fingerprint attendance system combines a DY50 sensor for biometric acquisition, an RS485 module for inter-device connection, and an Arduino Mega 2560 for initial processing via UART. The ESP32 functions as the transmitter of the main fingerprint template data to the server via a Wi-Fi connection. The decision to use the ESP32 was based on the capabilities of its internal Wi-Fi module to handle portable biometric data management [6].

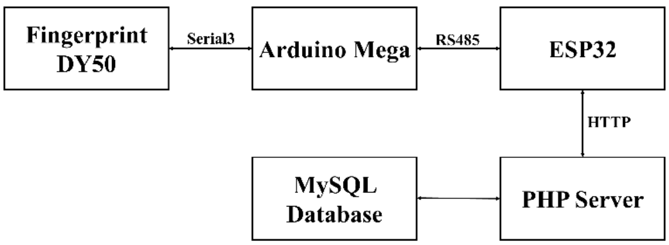

Figure 1 Architecture of the HTTP-Based Fingerprint Attendance System

A PHP-based server receives the fingerprint data in JSON format and stores it in a MySQL database using the binary data type. This method was chosen because the HTTP protocol is the most common in web-based Internet of Things (IoT) systems. In addition, the HTTP protocol has been widely used and proven to be compatible with modern client-server architectures without requiring additional middleware [16]. Based on the system architecture, the operational flow of the fingerprint attendance system is explained in the following section using a general system flowchart.

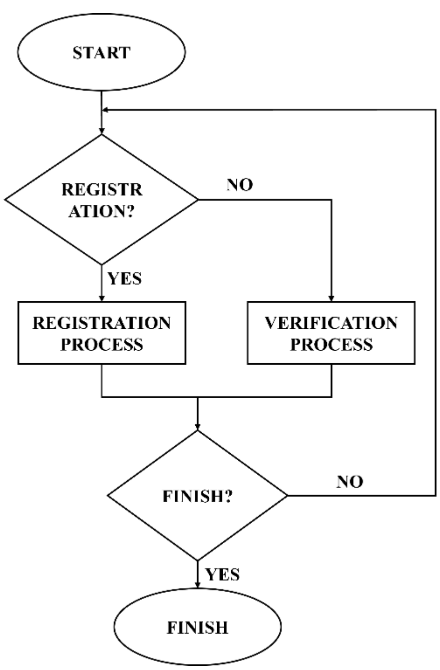

Figure 2 General System Flow

Figure 2 shows the general flow of the fingerprint attendance system. The process begins with system initialization, followed by the user selecting either the registration or verification mode. If the registration mode is selected, the system runs the fingerprint template recording and transmission process. Conversely, in verification mode, the system matches the fingerprint template with the data stored on the server. Once the process is complete, the system provides a final status and the process ends. A detailed explanation of the fingerprint registration and verification mechanisms is discussed further in Subsection B and Subsection C.

B. Fingerprint Registration Process

The registration flowchart is shown in Figure 3. During the registration process, users place their finger six times to obtain a variety of fingerprint templates that represent unique patterns, thereby increasing the accuracy and reliability of the identification process. Arduino Mega reads the template via the Serial3 protocol and then sends the data to ESP32 via RS485 [4], [7]. ESP32 packages the data in JSON format and sends it to the server via HTTP POST. The server stores the template in a MySQL database and sends a “200 OK” response as a sign of successful storage.

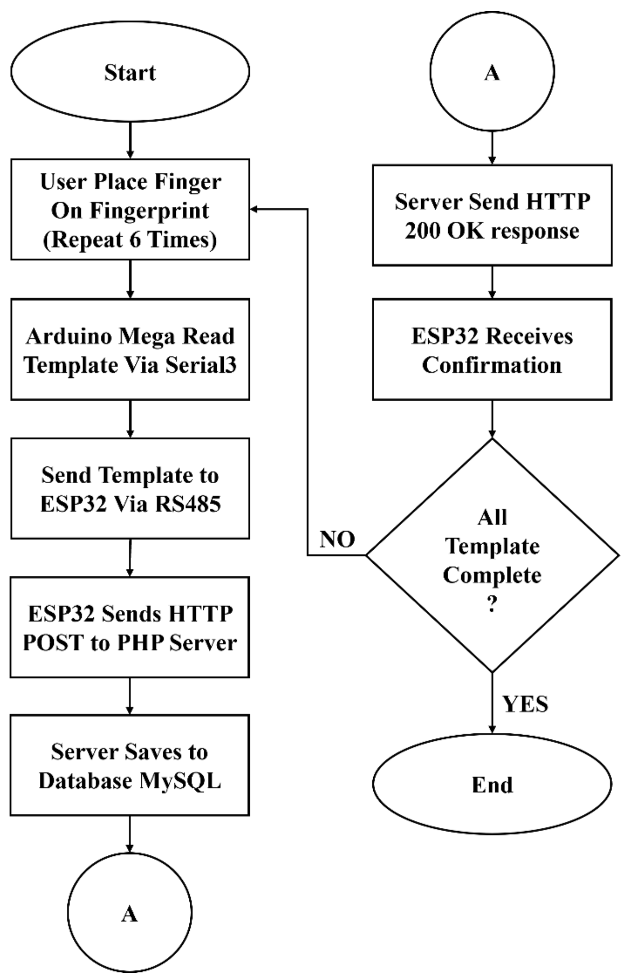

Figure 3 Fingerprint Registration Process Flowchart

C. Fingerprint Verification Flow

The verification stage is shown in Figure 4. The user only needs to place their finger once. The template is sent from Arduino Mega to ESP32 via RS485 and then sent to the server using HTTP POST. The server uses a similarity algorithm to find the three templates with the highest match values (Top-Match). The server then sends the templates to the ESP32, which forwards them to the Arduino Mega for processing using the matchUpload() function in the DY50 module. If the matching results reach the match threshold, a successful authentication status is sent back to the server. This server-side matching approach is implemented to ensure secure academic assessment and reduce the computational load on end devices in line with the principles of designing a secure biometric attendance management system [17].

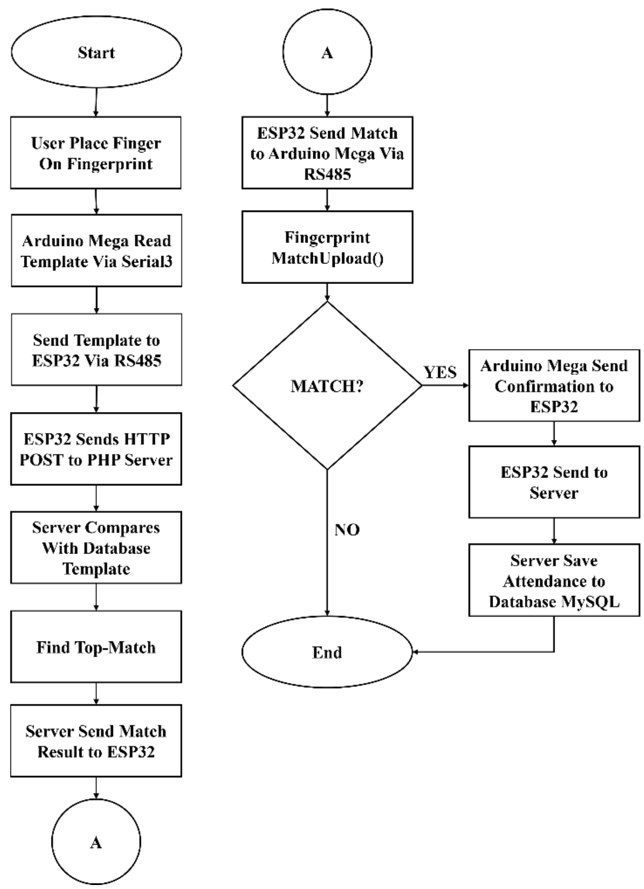

Figure 4 Fingerprint Verification Process Flowchart

The top-match approach is used to reduce database processing time and speed up local authentication on the microcontroller, in line with modern IoT-based biometric studies.

D. Data Communication Mechanism

Data transmission between devices and the server in this system uses the HTTP (Hypertext Transfer Protocol) with the POST method and a JSON-formatted payload. The ESP32 microcontroller functions as a client, sending fingerprint template data read from the Arduino Mega to the PHP-based server, which acts as the application host at the application layer.

The server receives the data, stores it in a MySQL database with the VARBINARY data type, and returns an HTTP response with a 200 OK status code to indicate successful transmission. Thus, this HTTP communication is bidirectional, consisting of an uplink phase (sending data from the ESP32 to the server) and a downlink phase (receiving a response from the server to the ESP32).

There are two important time parts to each HTTP request: the request timestamp (when the data is sent) and the response timestamp (when the response is received). The difference between these two values is used to calculate the latency and throughput parameters on the ESP32 log side.

Figure 5 illustrates the two-way communication flow in the HTTP protocol. The uplink phase occurs when the ESP32 sends the fingerprint template to the server, while the downlink phase happens when the server sends the verification results or top-match back to the device. These two directions of communication form the basis for calculating the latency, jitter, and throughput parameters in system performance analysis.

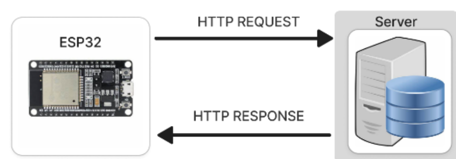

Figure 5 HTTP Request–Response Communication

For network performance evaluation, the Quality of Service (QoS) equation is applied, referencing the TIPHON standard as follows:

- **Latency:** The delay that occurs in data transmission over a network, providing insights into the overall responsiveness and smoothness of the network under different load balancing strategies.
$$Latency = t_{response\ received} - t_{request\ sent} \quad (1)$$
- **Jitter:** Variations in packet arrival time affect the quality of real-time communication.
$$Jitter = \frac{Total\ Delay\ Variation}{Total\ Packets\ Received - 1} \quad (2)$$
- **Throughput:** This is the total number of packets transferred within a specific time period, measuring the capacity and effectiveness of the algorithm in managing network traffic.
$$Throughput = \frac{Total\ Data\ Received\ (bit)}{Transmission\ Time\ (seconds)} \quad (3)$$
- **Packet Loss:** The percentage of packets that fail to reach their destination, affecting network reliability. This metric is measured per second, indicating the capacity and effectiveness of the algorithm in managing network traffic.
$$Packet\ Loss = \frac{Packets\ Sent - Packets\ Received}{Total\ Packets\ Sent}\ x\ 100 \quad (4)$$

It refers to the TIPHON standard and is widely used in IoT network performance research [11], [15]. Evaluation is conducted by analyzing each direction of HTTP uplink and downlink communication to assess network stability and data transfer efficiency.

E. System Testing Activities

Performance testing of the fingerprint attendance system was conducted on three main activities: enrollment, verification, and top-match. These three activities represent different communication patterns within the HTTP architecture used. The enrollment and verification activities are part of the uplink process, in which the ESP32 sends fingerprint data to the server. The top-match activity is part of the downlink process, in which the server sends verification results back to the ESP32.

One fingerprint template in this system is made up of six data packets that are taken one after the other from the fingerprint sensor. Each packet is 128 bytes long, which is what the DY50 sensor's internal specifications say. Therefore, the total raw data size for one fingerprint template is 768 bytes. After all packets are received completely by the ESP32, the six packets are combined and sent to the server in a single HTTP POST transaction. This data structure is used in all system activities, such as registering, verifying, and finding the best match.

Registration process, users perform six fingerprint scans to obtain representative biometric templates. Each completed template is transmitted to the server through a separate HTTP transaction. Although the transmissions are sequential, the short intervals between transactions produce a traffic pattern resembling uplink burst transmission, making enrollment suitable for evaluating throughput and jitter stability.

Verification activity is also included in the uplink process, but it only involves one fingerprint template transmitted in a single HTTP transaction. At this stage, the server performs similarity matching against the database and selects the template with the highest similarity as the top-match. Verification activities are used to measure the network's response time to a single request with a relatively small data payload, as well as to evaluate the stability of HTTP uplink communication under normal conditions.

The top-match activity represents the HTTP downlink phase, which is when the server sends the best-matched template back to the ESP32. The fingerprint data sent from the server to the ESP32 has the same structure: one template with a total size of 768 bytes. This stage simulates the return communication from the server to the device and is an important parameter in evaluating two-way latency in the request-response mechanism. Additionally, the top-match activity is used to assess the reliability of two-way communication and ensure that the verification result payload can be received intact without data loss.

All testing activities were observed using the dual-layer measurement method, which involves simultaneous measurements at the application and network layers.

- Data at the application layer is obtained through an internal logging mechanism implemented in the ESP32 firmware. Each HTTP transaction is logged using the internal timestamp millis(), where the start time is recorded just before the HTTP request is sent and the end time is recorded after the response is received. The difference between these two timestamps is used to calculate end-to-end latency at the application layer. Parameter jitter is calculated as the variation in latency between consecutive transactions, while throughput is calculated based on the actual HTTP payload size transmitted in a single transaction divided by the transmission duration, thus reflecting the real communication load processed by the system.
- Measurements at the network layer were performed using Wireshark to record the timestamps of communication packets between the ESP32 and the server. The captured packet data was then processed manually to calculate latency based on round-trip time, inter-packet jitter, throughput, and packet loss based on the success of packet responses.

The combination of these two measurement approaches provides a comprehensive overview of HTTP-based two-way communication performance from both the application and network perspectives.

Testing was conducted on a local network using a Tenda AC1200 Smart Dual Band router model AC6 with the IEEE 802.11ac standard at a frequency of 2.4 GHz [20]. To minimize interference and traffic load from other devices, the network was configured separately with only the ESP32 and the server laptop connected to the router. All testing was conducted at the BRAIL of the Batam State Polytechnic under controlled network conditions.

F. Network Quality Categories Based on TIPHON

Network quality evaluation is conducted based on the TIPHON standard, which classifies the parameters of latency, jitter, throughput, and packet loss to determine the suitability of IP-based communication services. This standard is widely used to assess the quality of IP-based communication services, including IoT systems and real-time biometrics. TIPHON classifies four main parameters, namely latency, jitter, packet loss, and throughput, into four quality levels: Grade 1-4 [12], [14].

TABLE I
QOS PARAMETER CLASSIFICATION BASED ON TIPHON STANDARD

Parameter	Very Good (Grade 4)	Good (Grade 3)	Moderate (Grade 2)	Poor (Grade 1)
Latency	< 150 ms	150–300 ms	300–450 ms	> 450 ms
Jitter	0 ms	0-75 ms	75-125 ms	125-225 ms
Packet Loss	0 - 2%	3 - 14%	15 - 24%	>25%
Throughput	> 2100 Kbps	1200 - 2100 Kbps	700 - 1200 Kbps	0 - 700 Kbps

This classification is used as a reference to assess the feasibility of an IoT-based fingerprint attendance system in supporting real-time biometric authentication processes. The system is considered suitable when the latency, jitter, and packet loss parameters are at least in the Grade 3 (Good) category, while throughput is evaluated contextually based on the relatively small and non-continuous data service needs of biometric data.

III. RESULT AND DISCUSSION

A. General Results of QoS Testing

TABLE II
AVERAGE QOS PARAMETER VALUES BASED ON ACTIVITY

Aktivitas	Latency Wire-shark (ms)	Late-ncy ESP32 (ms)	Jitter Wire-shark (ms)	Jitter ESP32 (ms)	Through-put Wireshark (kbps)	Through-put ESP32 (kbps)	Pack-et Loss (%)
Pendaftaran	20.28	80.10	8.71	49.39	302.55	218.84	0
Verifikasi	16.68	70.88	12.80	52.00	457.51	259.64	0
Top-Match	83.23	151.97	20.78	62.81	74.69	54.08	0

The results of the quality of service testing on the ESP32-based fingerprint attendance system using the HTTP protocol are shown in Table II, which summarizes the average values for latency, jitter, throughput, and packet loss parameters for the three main activities: registration, verification, and top-match. All data was obtained from 60 tests for each activity. This result was obtained based on the dual-layer measurement method described in the methodology section.

B. Latency Analysis

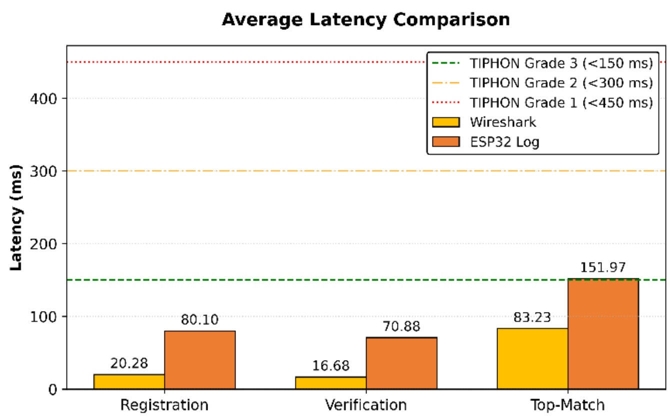

Figure 6 Average Latency Comparison by Activity (Wireshark vs ESP32)

A comparison of average latency values for each activity is shown in Figure 6. Based on the graph and data in Table II, the latency at the Wireshark layer ranges from 16.68 ms to 83.23 ms, while the latency at the ESP32 layer ranges from 70.88 ms to 151.97 ms.

Registration and verification activities have the lowest latency, with values below 100 ms on the ESP32, placing them in the Very Good Grade 4 category according to TIPHON standards. The top-match activity shows the highest latency, 151.97 ms on the ESP32, due to the process of sending matching results data from the server to the device. Nevertheless, this value still falls into the Good Grade 3 category and continues to meet the criteria for real-time communication.

C. Jitter Analysis

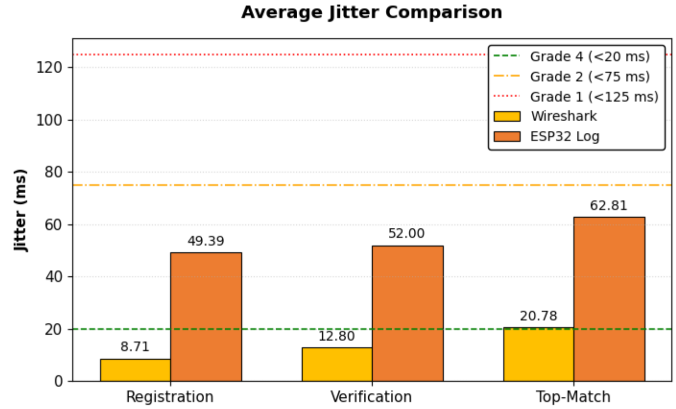

Figure 7 Average Jitter Comparison by Activity (Wireshark vs ESP32)

The average jitter values are shown in Figure 7. Based on the graph and data in Table II, the jitter at the Wireshark layer ranges from 8.71 ms to 20.78 ms, while at the ESP32 layer it ranges from 49.39 ms to 62.81 ms.

The increase in jitter at the ESP32 layer primarily occurs during verification and top-match activities. This is due to variations in server processing time when matching fingerprint templates and sending verification results data. Despite fluctuations in time between transactions, the overall jitter value remains in the Good Grade 3 category and does not cause disruption to the authentication process.

D. Throughput Analysis

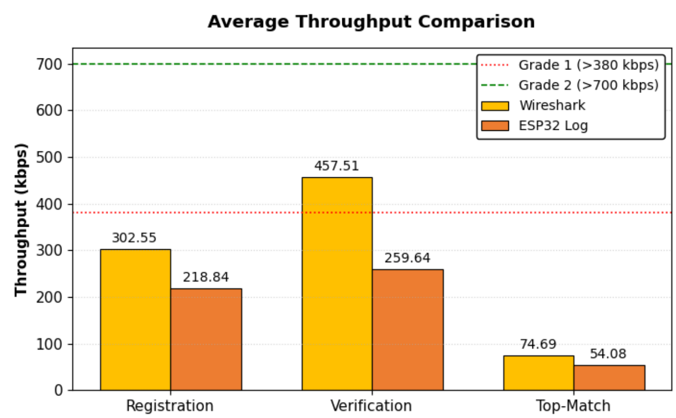

Figure 8 Average Throughput Comparison by Activity (Wireshark vs ESP32)

Based on the graph in Figure 8 and the data in Table II, the highest throughput occurred during the verification activity, which was 457.51 kbps on Wireshark and 259.64 kbps on ESP32, while the lowest throughput occurred during the top-match activity, which was 74.69 kbps on Wireshark and 54.08 kbps on ESP32.

Based on the TIPHON classification, this throughput value falls into the low category. However, this condition is influenced by the relatively small size of the fingerprint data payload, which is a single 768-byte fingerprint template sent in a single HTTP transaction. Therefore, the low throughput does not indicate a network limitation but rather that the system only uses bandwidth as needed.

E. Evaluation Based on the TIPHON Classification

TABLE III
PERFORMANCE EVALUATION BASED ON TIPHON CLASSIFICATION PER ACTIVITY AND LAYER

Activity	Parameter	Average Value	TIPHON Category	Grade
Registration	Wireshark	Latency: 20.28 ms	Very Good	4
		Jitter: 8.71 ms	Good	3
		Throughput: 302.55 kbps	Poor	1
		Packet Loss: 0%	Very Good	4
	ESP32 Log	Latency: 80.10 ms	Very Good	4
		Jitter: 49.39 ms	Good	3
		Throughput: 218.84 kbps	Poor	1
		Packet Loss: 0%	Very Good	4
Verification	Wireshark	Latency: 16.68 ms	Very Good	4
		Jitter: 12.80 ms	Good	3
		Throughput: 457.51 kbps	Poor	1
		Packet Loss: 0%	Very Good	4
	ESP32 Log	Latency: 70.88 ms	Very Good	4
		Jitter: 52.00 ms	Good	3
		Throughput: 259.64 kbps	Poor	1
		Packet Loss: 0%	Very Good	4
Top-Match	Wireshark	Latency: 83.23 ms	Good	3
		Jitter: 20.78 ms	Good	3
		Throughput: 74.69 kbps	Poor	1
		Packet Loss: 0%	Very Good	4
	ESP32 Log	Latency: 151.97 ms	Good	3
		Jitter: 62.81 ms	Good	3
		Throughput: 54.08 kbps	Poor	1
		Packet Loss: 0%	Very Good	4

System performance evaluation is conducted with reference to the TIPHON standards in Table I, which classifies quality of service parameters into several service quality categories. The evaluation results in Table III show that the latency and packet loss parameters for all activities are in the Good to Very Good category. The latency values for registration and verification activities consistently remained below 150 ms, both at the Wireshark and ESP32 layers, thus being classified as Very Good Grade 4. Meanwhile, the top-match activity showed slightly higher latency at the ESP32 layer, at 151.97 ms, but it still falls into the Good Grade 3 category and is within the real-time communication tolerance limits.

Parameter jitter indicates a greater variation in inter-transaction time at the application layer compared to the network layer. During the registration and verification activity, the jitter value on the ESP32 ranged from 49.39 ms to 52.00 ms, while it increased to 62.81 ms during the top-match activity. Based on the TIPHON classification, all of these jitter values still fall into the Good Grade 3 category. This increase in jitter is caused by variations in server processing time, particularly in the fingerprint template matching process and the transmission of verification results to the device.

The parameter throughput for all activities falls into the Poor Grade 1 category according to TIPHON, with values ranging from 54.08 kbps to 457.51 kbps. However, this value needs to be interpreted contextually. Biometric attendance systems do not require a continuous data stream with large capacity, as the fingerprint payload size is relatively small and only sent at specific times. Thus, low throughput does not reflect poor network performance but rather indicates the efficiency of bandwidth usage according to application needs. Additionally, throughput calculations at the application layer include the entire actual HTTP payload, including fingerprint data and POST parameter overhead, so the resulting throughput value represents the system's real communication load.

IV. CONCLUSION

The ESP32-based fingerprint attendance system with HTTP protocol meets real-time communication feasibility based on the TIPHON standard, with latency at the ESP32 application layer of 70.88 ms to 151.97 ms and at the Wireshark network layer of 16.68 ms to 83.23 ms, which falls into the Very Good to Good category (Grade 4 to Grade 3), jitter in both layers falls into the Good category (Grade 3), and packet loss is 0 percent, which falls into the Very Good category (Grade 4). Although the throughput was only in the range of 54.08 kbps to 457.51 kbps and was classified as Poor (Grade 1), this was influenced by the fingerprint payload size of 768 bytes and did not interfere with authentication performance. The limitations of this study are that testing is still limited to local networks without HTTPS and without traffic load, so further research is recommended to test the system on public networks, implement SSL or TLS, and compare HTTP with other protocols such as MQTT or WebSocket.

REFERENCES

[1] H. Dhien Chandra, "Sistem Informasi Absensi RFID Berbasis Web Menggunakan ESP32 di PT Dharma Sentosa Marindo." doi: https://doi.org/10.47324/ilkominfo.v6i1.168.

[2] U. Sumalatha, K. K. Prakasha, S. Prabhu, and V. C. Nayak, "A Comprehensive Review of Unimodal and Multimodal Fingerprint Biometric Authentication Systems: Fusion, Attacks, and Template Protection," *IEEE Access*, vol. 12, pp. 64300–64334, 2024, doi: 10.1109/ACCESS.2024.3395417.

[3] E. G. Abdulkadhim, "Design and Develop an Attendance System Based on Fingerprint and Arduino Board," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1804/1/012011.

[4] J. I. C. Celerez, W. E. Antipuesto, D. R. A. Aratea, I. C. L. Salvador, and J. N. B. Rosello, "Wired Fingerprint-Based Classroom Attendance System for Secured Student Attendance Archiving Using Arduino UNO Microcontroller," *Journal of Image Processing and Intelligent Remote Sensing*, no. 43, pp. 1–13, Apr. 2024, doi: 10.55529/jipirs.43.1.13.

[5] V. W. Tiarno, E. B. Cahyono, and I. Nuryasin, "Implementasi Sistem Presensi Menggunakan Biometrik Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Malang," *REPOSITOR*, vol. 2, no. 1, pp. 27–34, 2020, doi: 10.22219/repositor.v2i1.30419.

[6] Aadil S, "Portable Biometric Attendance Management System Using ESP32," *International Journal of Scientific Research and Engineering Development*, vol. 8, doi: 10.5281/zenodo.15823749.

[7] A. H. Kale and D. D. Ahire, "Classroom Attendance System by Using IOT with Fingerprint." [Online]. Available: www.ijres.org

[8] H. O. Lasisi, B. R. Ajibade, O. C. Ajayi, O. O. Obiyemi, S. C. Thakur, and E. Adetiba, "Implementation of cloud-based biometric attendance system for educators in a developing country," in

Journal of Physics: Conference Series, IOP Publishing Ltd, Oct. 2021. doi: 10.1088/1742-6596/2034/1/012018.

[9] P. Ahmad Raihan, M. Ramaddan Julianti Candidate, S. Ramdhan, and S. Fazilla Candidate, "International Journal of Communication Networks and Information Security IoT-Based Biometric Attendance System Using Arduino and ThingsBoard".

[10] F. Zhou and T. Zhao, "A Survey on Biometrics Authentication," p. 6, Dec. 2022, doi: 10.48550/arXiv.2212.08224.

[11] W. A. Kuncoro and L. Santoso, "Kemacetan Jaringan Akibat Media Sosial: Menganalisis Dampaknya terhadap Efisiensi Transmisi Data Nirkabel," *Jurnal Ilmiah Sistem Informasi*, vol. 4, no. 2, pp. 130–141, May 2025, doi: 10.51903/jt9z2f09.

[12] D. Bayu Pratama *et al.*, "Analisis Quality Of Service Dan Implementasi Sistem Monitoring Menggunakan Internet Of Things (IoT) Pada Pembangkit Listrik Tenaga Angin," *Jurnal POLEKTRO: Jurnal Power Elektronik*, vol. 12, no. 1, p. 2023, doi: 10.30591/polektro.v12i1.4309.

[13] B. Dwinanto and A. S. Arifin, "Integrated Strategy Framework to Improve Quality of Network on the BMKG Communication Network System," in *10th IEEE International Conference on Communication, Networks and Satellite, Comnetsat 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 244–251. doi: 10.1109/COMNETSAT53002.2021.9530814.

[14] H. Ruli Oktaseli and A. A. Slameto, "Evaluation of Wireless LAN Quality of Service (QoS) in Primary Education Using TIPHON Standards," 2025. doi: https://doi.org/10.30871/jaic.v9i2.8979.

[15] R. Qori, Y. Putri, A. Setiawan, D. A. Kholik, A. Al Nazen, and H. F. Lumbantobing, "Analysis of LTE Network Quality of Service on Streaming Application," 2023. doi: 10.31258/ijeepse.6.2.151-155.

[16] M. A. Adedoyin, O. O. Shoewu, A. I. O. Yussuff, A. A. Adenowo, A. Shitta, and O. Okedokun, "Development of an IoT-Based Biometric Attendance Management System," *FUOYE Journal of Engineering and Technology*, vol. 9, no. 3, pp. 397–405, Dec. 2024, doi: 10.4314/fuoyejet.v9i3.5.

[17] S. Vatchala *et al.*, "Multi-Modal Biometric Authentication: Leveraging Shared Layer Architectures for Enhanced Security," *IEEE Access*, vol. 13, pp. 28029–28041, 2025, doi: 10.1109/ACCESS.2025.3534223.

[18] F. P. E. Putra, M. Surur, M. Mahendra, and G. Arifin, "Internet Network QOS Analysis at Yala Kopitiam pamekasan Using Wireshak," *Brilliance: Research of Artificial Intelligence*, vol. 5, no. 1, pp. 190–200, Jun. 2025, doi: 10.47709/brilliance.v5i1.5940.

[19] Fauzan Prasetyo Eka Putra, Ekawati Ekawati, Akhmad Vidyan, and Moh. Ali, "Evaluasi Kualitas Layanan (QoS) pada Jaringan Wi-Fi 6 Dibandingkan dengan Wi-Fi 5," *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, vol. 5, no. 1, pp. 102–108, Mar. 2025, doi: 10.55606/jitek.v5i1.6010.

[20] Tenda, "AC1200 Dual-band Router User Guide," 2019.