A Comparative Performance Evaluation of Real-Time Communication Architectures for ESP32-Based Chatbot Systems

[1]Ridwan Sanjaya, [2]Christin Wibhowo
[1]Department of Information Systems, Faculty of Computer Science
[2]Department of Psychology, Faculty of Psychology
Soegijapranata Catholic University, Semarang, Indonesia
[1]ridwan@unika.ac.id, [2]christine@unika.ac.id

Abstract— ESP32-based devices are widely used as a low-cost and resource-constrained platform for Internet of Things (IoT) applications that require responsive communication. However, real-time chatbot systems need higher requirements regarding delays that can reduce the perception of responsiveness, conversation continuity, and user trust. This study evaluates the use of WebSocket, Hybrid MQTT-WebSocket, Hybrid MQTT-Server-Sent Events (SSE), and Cloudflare as four real-time communication architectures for an ESP32-based chatbot system. This study used a systematic literature review on ESP32 communication models and experimental performance evaluation using Sovi Lau 2.0 logs. This dataset consists of 400 communication logs from the performance of the four architectures. The four metrics analyzed were latency, throughput, packet loss, and power consumption. One-way ANOVA was used to compare the architectures, followed by Tukey's HSD for significant metrics. The results show that Cloudflare has the lowest average latency and the highest throughput, while Hybrid MQTT-SSE produces the lowest packet loss and the lowest power consumption. The ANOVA test showed that of the variables tested, only latency had a significant difference between the architectures. The Tukey HSD test confirmed that the Cloudflare architecture had the significantly lowest latency.

Keywords—Cloudflare, esp32, mqtt, server-sent events, websocket

I. INTRODUCTION

The rapid development of Internet of Things (IoT) systems has expanded the role of microcontroller-based platforms from simple sensing and actuation to interactive and conversational applications. Among these platforms, ESP32 and ESP32-S3 are widely adopted because they provide integrated Wi-Fi, Bluetooth, sufficient processing capacity, and relatively low power consumption [1]. These characteristics make them suitable for smart devices that need to communicate continuously with servers with energy constraints.

Real-time chatbot systems place higher demands on communication architecture. Unlike conventional IoT, chatbot interactions require fast, stable, and continuous message exchange. In voice-based chatbot scenarios, delayed responses can disrupt the conversation flow and detract from the natural feel of interaction [2], [3]. All of these are critical in systems like Sovi Lau 2.0, a rule-based chatbot system designed to support structured, controlled, and empathetic interactions. The designed scenario places an ESP32/ESP32-S3 smart device as the hardware interface between the user and the Sovi Lau 2.0 rule-based AI server, with communication performance evaluated through latency, throughput, packet loss, and power consumption [4].

Although WebSocket and MQTT are widely discussed in IoT communications research [5], the optimal communication architecture for ESP32-based chatbot systems remains underexplored. Previous studies often evaluate general IoT scenarios such as monitoring, digital twins, or web notifications, but rarely focus on the specific requirements of real-time chatbot interactions on resource-constrained ESP32 devices. Furthermore, hybrid models such as MQTT-

WebSocket and MQTT-SSE are theoretically appealing, but empirical evidence comparing their practical performance is limited [6], [7].

This study addresses this gap by evaluating four real-time communication architectures for ESP32-based chatbot systems: WebSocket, hybrid MQTT-WebSocket, hybrid MQTT-SSE, and Cloudflare. The study employs a systematic literature review (SLR) and experimental performance evaluation using dashboard logs on four devices. The key question concerns which communication architecture provides the best balance between latency, throughput, packet loss, and power consumption for ESP32-based real-time chatbot systems.

The contributions of this paper are threefold: 1) synthesizing recent evidence on ESP32 communication models for real-time interactive applications, 2) providing a comparative empirical evaluation of four communication architectures using the same set of performance metrics, and 3) identifying considerations on architecture selection for ESP32-based chatbot systems, especially when responsiveness and power efficiency are both important.

II. RELATED WORK

A systematic literature review was conducted to understand the current state of communication models for ESP32-based systems. The review used keywords related to ESP32, communication, WebSocket, MQTT, Server-Sent Events, Cloudflare, hybrid architecture, IoT, and real-time chatbot interaction. The review identified 20 papers published between 2020 and 2025. After filtering, 9 papers were included for synthesis. The included studies primarily addressed WebSocket, MQTT, MQTT over WebSocket, Cloudflare integration, and multi-protocol IoT architectures.

The review indicated that WebSocket is preferred for persistent, bidirectional, and low-latency communication. WebSocket maintains an open TCP-based connection, allowing clients and servers to exchange messages without the overhead of repeated request-responses [8], [9]. For chatbot applications, this characteristic is important because it enables continuous interaction between the ESP32 device and the chatbot server.

MQTT is also widely used in ESP32-based IoT systems because it is lightweight and follows a publish-subscribe model [10]. MQTT is well-suited for event-based communications and resource-constrained devices [11]. Existing literature shows that MQTT over TCP can provide low latency, although performance depends on network conditions, quality of service configuration, broker implementation, and payload size [4], [10]. Based on this, a hybrid MQTT-based architecture becomes attractive when command delivery, telemetry, and server push mechanisms need to run simultaneously [12].

Server-Sent Events (SSE) uses a server-to-client push mechanism over HTTP. While SSE is simpler than full-duplex WebSockets, it is useful when the server needs to continuously send chatbot responses or status updates to the device. However, existing literature reveals that Hybrid MQTT-SSE has not been extensively evaluated in ESP32 chatbot settings [13], [14]. This limitation creates a research gap that could be empirically evaluated.

Cloudflare has emerged in the literature as an approach for secure and globally accessible IoT connectivity [15], [16]. In the context of a chatbot system, edge routing can reduce perceived delays by placing communications closer to the user or by improving routing stability. However, Cloudflare can also introduce different considerations related to power and connection management [17], [18]. Therefore, it needs to be evaluated in conjunction with device-level communication protocols.

Table 1. SLR-derived comparison of communication architectures

Architecture	Core mechanism	Potential use in chatbot systems	Expected advantage	Main concern

WebSocket	Persistent full-duplex connection	Interactive chat, real-time UI, continuous exchange	Stable two-way communication	Requires persistent connection management
Hybrid MQTT-WebSocket	MQTT messaging integrated with WebSocket transport/interface	IoT messaging with real-time dashboard/server interaction	Lightweight publish-subscribe plus bidirectional channel	Additional architectural complexity
Hybrid MQTT-SSE	MQTT messaging combined with server push via SSE	Event-driven IoT with server-to-client updates	Simple server push and MQTT-based telemetry	Limited empirical evidence for ESP32 chatbot use
Cloudflare	Edge-mediated communication/access layer	Globally accessible and secure IoT-chatbot interaction	Potential latency and routing benefit	May increase power or connection overhead

III. METHOD

This study uses SLR-based synthesis and experimental performance evaluation. SLR is used to identify the theoretical strengths and weaknesses of the proposed communication architecture. Performance experiments are then used to compare these architectures within the same dashboard, logging, and export environments.

A. System Context

This evaluation uses an ESP32/ESP32-S3 smart device connected to a server-side chatbot system. The chatbot context is based on Sovi Lau 2.0, a rule-based conversational system where responses are controlled through knowledge and structured rules. In this context, the communication architecture is crucial because the device must transmit user interaction events and receive chatbot responses with minimal delay.

B. Communication Architectures

The four communication architectures compared are 1) WebSocket, 2) Hybrid MQTT-WebSocket, 3) Hybrid MQTT-SSE, and 4) Cloudflare. WebSocket represents a persistent bidirectional connection. Hybrid MQTT-WebSocket represents a combined messaging architecture that uses lightweight communications like MQTT while leveraging WebSocket for real-time interactions. Hybrid MQTT-SSE represents a mixed architecture where MQTT handles lightweight messaging while SSE handles server-to-client event streaming. Cloudflare represents an edge-assisted communications layer with the goal of improving accessibility and routing responsiveness.

C. Dataset And Metrics

The empirical dataset is obtained from Sovi Lau 2.0 logs. This dataset contains 400 records, with 100 records in each architecture. Each record contains a timestamp, communication model, latency, throughput, packet loss, and power consumption. Latency is measured in milliseconds (ms), throughput in kilobits per second (kbps), packet loss in percentage (%), and power consumption in milliwatts (mW).

D. Statistical Analysis

Descriptive statistics were calculated for each architecture and metric. A one-way analysis of variance (ANOVA) was then used to test whether the mean performance differed significantly across communication architectures. When the ANOVA result was significant, Tukey HSD post-hoc analysis was used to identify which pairs of architectures differed. The statistical significance threshold was set at alpha = 0.05. For interpretation, lower latency, lower packet loss, and lower power consumption were considered better, whereas higher throughput was considered better.

IV. RESULTS

The testing-data export consisted of four balanced groups: WebSocket, Hybrid MQTT-WebSocket, Hybrid MQTT-SSE, and Cloudflare, each containing 100 observations. The descriptive results are shown in Table 2.

Table 2. Descriptive statistics by communication architecture

Model	Latency ms, mean ± SD	Throughput kbps, mean ± SD	Loss %, mean ± SD	Power mW, mean ± SD
WebSocket	301.11 ± 321.98	317.52 ± 480.40	3.79 ± 6.21	166.53 ± 54.68
Hybrid MQTT & WS	335.21 ± 360.25	314.82 ± 467.69	3.79 ± 6.03	170.01 ± 41.27
Hybrid MQTT & SSE	394.47 ± 352.67	310.93 ± 466.07	3.76 ± 5.91	164.28 ± 23.29
Cloudflare	185.61 ± 202.50	348.85 ± 527.10	3.90 ± 6.31	183.04 ± 85.26

Cloudflare achieved the lowest average latency (185.61 ms), followed by WebSocket (301.11 ms), Hybrid MQTT-WebSocket (335.21 ms), and Hybrid MQTT-SSE (394.47 ms). Cloudflare also produced the highest average throughput (348.85 kbps). However, Cloudflare showed the highest average power consumption (183.04 mW), indicating that lower latency may come with higher energy cost. Hybrid MQTT-SSE showed the lowest average power consumption (164.28 mW) and the lowest packet loss (3.76%), but it also produced the highest latency among the four models.

Table 3. One-way ANOVA results

Metric	F-statistic	p-value	Decision
Latency	7.751	< 0.001	Significant
Throughput	0.129	0.943	Not significant
Loss	0.011	0.998	Not significant
Power	2.251	0.082	Not significant

The ANOVA results show that only latency differed significantly across communication architectures ($F = 7.751$, $p < 0.001$). Throughput, packet loss, and power consumption did not show statistically significant differences at alpha = 0.05. Therefore, post-hoc analysis was conducted only for latency.

Table 4. Tukey HSD post-hoc comparison for latency

Group 1	Group 2	Mean diff.	p-adj	Lower	Upper	Significant
Cloudflare	Hybrid MQTT & SSE	208.86	0.0000	93.65	324.07	Yes
Cloudflare	Hybrid MQTT & WS	149.60	0.0049	34.39	264.81	Yes
Cloudflare	WebSocket	115.50	0.0492	0.29	230.71	Yes
Hybrid MQTT & SSE	Hybrid MQTT & WS	-59.26	0.5462	-174.47	55.95	No
Hybrid MQTT & SSE	WebSocket	-93.37	0.1579	-208.58	21.84	No
Hybrid MQTT & WS	WebSocket	-34.11	0.8707	-149.32	81.10	No

The Tukey HSD result indicates that Cloudflare had significantly lower latency than Hybrid MQTT-SSE, Hybrid MQTT-WebSocket, and WebSocket. In contrast, the latency differences among WebSocket, Hybrid MQTT-WebSocket, and Hybrid MQTT-SSE were not statistically significant. This suggests that Cloudflare is the only architecture that consistently separated itself from the others in terms of latency performance in the testing environment.

V. DISCUSSION

These findings provide a more varied portrait of communication architecture selection for ESP32-based chatbot systems. Cloudflare emerged as the strongest candidate when latency was the primary design criterion. Its latency advantage was not only descriptive but also statistically supported. This finding aligns with the assumption that edge-assisted communication can improve routing responsiveness and accessibility. However, Cloudflare also consumed the highest average power consumption, making it less of a universally optimal choice for battery-powered deployments.

WebSocket exhibited a more balanced performance profile. While its average latency was higher than Cloudflare, it outperformed both hybrid MQTT variants in terms of latency and remained conceptually appealing due to its persistent, bidirectional communication model. For chatbot systems that require continuous interaction but do not require global edge routing, WebSocket may remain a practical and simpler architecture.

The MQTT-WebSocket hybrid did not outperform WebSocket in experimental data. This result is important because hybrid architectures are often assumed to improve performance by combining the strengths of multiple protocols. The results show that hybridization can increase architectural complexity without automatically increasing latency or throughput. In practice, Hybrid MQTT-WebSocket can still be an option when a system requires MQTT-style message routing or integration with existing IoT brokers, but not solely for latency improvements.

Hybrid MQTT-SSE produced the lowest average power consumption and packet loss, but its average latency was the highest. This makes it less suitable for highly latency-sensitive chatbot interaction if responsiveness is prioritized. Nevertheless, it may be useful in scenarios where server-to-device updates are less frequent, battery efficiency is more important, or the communication pattern is more event-driven than fully conversational.

A critical issue is that none of the architectures achieved an average latency below 100 ms. The original system context emphasizes the importance of very low latency for natural conversational interaction. Therefore, the present findings should not be interpreted as evidence that the system is already optimized for highly sensitive real-time interaction. Rather, they show that Cloudflare is the most promising among the evaluated architectures, while further optimization is needed to approach stricter real-time thresholds.

The results also support the SLR finding that there is no universally superior protocol. Communication performance depends on the interaction pattern, network condition, server deployment, payload size, and device constraints. For ESP32-based chatbot systems, architectural selection should therefore be based on trade-off analysis rather than a single metric. Latency-sensitive systems may prefer Cloudflare, balanced local deployments may prefer WebSocket, and energy-sensitive event-driven systems may consider Hybrid MQTT-SSE.

VI. CONCLUSION

This paper evaluated four real-time communication architectures for ESP32-based chatbot systems: WebSocket, Hybrid MQTT-WebSocket, Hybrid MQTT-SSE, and Cloudflare. By combining SLR-based synthesis with experimental performance evaluation, this study provides empirical evidence for selecting communication architectures in resource-constrained conversational IoT systems. The results show that Cloudflare achieved the lowest latency and highest throughput, while Hybrid MQTT-SSE showed descriptively lower packet loss and power consumption. However, statistical testing indicated that only latency differed significantly across architectures. Therefore, architectural selection should not rely solely on average performance values, but should also consider the trade-off among responsiveness, power efficiency, deployment complexity, and integration requirements.

Based on these findings, Cloudflare is recommended as the primary architecture for latency-sensitive ESP32-based chatbot systems, especially when responsiveness and real-time interaction quality are top priorities. However, this recommendation is conditional, as Cloudflare also exhibits higher power consumption. For battery-powered or energy-constrained deployments, Hybrid MQTT-SSE can be considered as an alternative, especially when lower power consumption and packet loss are more important than minimum latency. WebSocket is a practical baseline choice for simple and balanced implementations, while hybrid MQTT-based architectures are more suitable for systems that prioritize message routing, integration flexibility, and distributed IoT communications. Future research could validate these findings for varying network conditions, different payload sizes, longer test durations, and end-to-end conversation quality metrics.

ACKNOWLEDGMENT

This research was funded by Soegijapranata Catholic University under the Internal Research Funding scheme in 2025/2026 titled Development of Real-time Voice-Based Smart Device for Virtual Assistant Personality Threshold Sovi Lau.

REFERENCES

[1] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, p. 6739, Jul. 2023, doi: 10.3390/s23156739.

[2] S. Bhaganagare, S. Chavan, S. Gavali, and V. Godase, "Voice Controlled Home Automation Using ESP32 and IoT-Based Cloud Integration," *Int. J. Adv. Res. Sci. Commun. Technol.*, 2025, doi: 10.48175/ijarsct-29478.

[3] M. Gladwin, R. Preeda, S. Dayana, G. Aarthi, and V. Fathima, "Voice Controlled Application using ESP32," *J. Electr. Eng. Autom.*, 2025, doi: 10.36548/jeea.2025.3.002.

[4] B. A, T. Natuva, M. Shaik, M. K. Thippirisetti, V. K. Atte, and S. M, "Latency and Throughput Benchmarking of IoT Protocols on ESP32 Microcontrollers," in *2025 IEEE 6th Global Conference for Advancement in Technology (GCAT)*, 2025, pp. 1–6. doi: 10.1109/gcat66372.2025.11368454.

[5] P. Chouragadey, L. Kawadkar, and B. Rathore, "Comparative Analysis of Communication Protocols: MQTT, HTTP, TCP, UDP, and Websocket," in *2025 International Conference on Electrical, Communication, and Computing Technologies (iCONECCT)*, 2025, pp. 1–6. doi: 10.1109/iconecct67014.2025.11469821.

[6] D. N. Bestari and A. Wibowo, "IoT Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform," *Int. J. Interact. Mob. Technol.*, vol. 17, pp. 4–19, 2023, doi: 10.3991/ijim.v17i06.34129.

[7] D. R. Tisna, T. Maharani, and K. Nugroho, "Pemanfaatan Chatbot Telegram untuk Monitoring dan Kontrol Kualitas Air Menggunakan Esp32," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, 2024, doi: 10.29100/jipi.v9i3.5329.

[8] N. Mitrovic, M. Djordjevic, S. Veljkovic, and D. Dankovic, "Implementation and testing of WebSocket protocol in ESP32 based IoT systems," *Facta Univ. - Ser. Electron. Energ.*, vol. 36, no. 2, pp. 267–284, 2023, doi: 10.2298/FUEE2302267M.

[9] G. Violettas and L. Mamatas, "TriCloudEdge: A multi-layer Cloud Continuum," *ArXiv*, vol. abs/2602.02121, 2026, doi: 10.48550/arxiv.2602.02121.

[10] V. Anooja, M. Khrisnakumar, S.

Sangeetha, and W. Rajaian, "Enhancing Intelligent Connectivity Through Embedded IoT Systems for Real-Time Applications," *ICTACT J. Microelectron.,* 2025, doi: 10.21917/ijme.2025.0348.

[11] T. S. Kartha *et al.*, "Evaluating the Suitability of MQTT and HTTP for ESP32-Based IoT Architectures," in *2025 International Conference on Robotics and Mechatronics (ICRM),* 2025, pp. 1–7. doi: 10.1109/icrm66809.2025.11349088.

[12] E. De Franceschi Vieira and T. A. Rizzetti, "Performance Evaluation of an ESPresense/MQTT Architecture for Real-Time Locating System," *An. da XXII Esc. Reg. Redes Comput. (ERRC 2025),* 2025, doi: 10.5753/errc.2025.17688.

[13] R. Sanjaya, A. K. Pamudji, and E. Vania, "Comparative Analysis and Practical Implementation of loT Communication Using HTTP/2 and MQTT," in *2024 7th International Conference on Green Technology and Sustainable Development (GTSD),* Ho Chi Minh City, Vietnam: IEEE, Jul. 2024, pp. 221–226. doi: 10.1109/GTSD62346.2024.10674902.

[14] R. Sanjaya, A. K. Pamudji, and C. Wibhowo, "Comparative Analysis of ESP8266, ESP32, and ESP32-S3 for Smart Speaker-Based Support in Borderline Personality Disorder," in *2025 IEEE 9th International Conference on Software Engineering & Computer Systems (ICSECS),* 2025, pp. 132–137. doi: 10.1109/icsecs65227.2025.11278945.

[15] M. Tauseef, C. Krishna, C. M, K. Bhagyashree, and K. S, "MediBot: Edge-Enabled AI Healthcare Chatbot with IoT-Based Monitoring and Smart Assistance," in *2026 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE),* 2026, pp. 1–8. doi: 10.1109/iitcee67948.2026.11394057.

[16] W. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017, doi: 10.1109/access.2017.2778504.

[17] D. A. Noordin, N. Z. Nadzri, and R. F. R. Suleiman, "Integration of IoT Platform Environment for Web Server and Global Connectivity Based on ESP32," in *2025 IEEE 6th International Conference in Robotics and Manufacturing Automation (ROMA),* 2025, pp. 339–344. doi: 10.1109/roma66616.2025.11155577.

[18] J. J. L. Escobar, R. Díaz-Redondo, and F. Gil-Castiñeira, "Unleashing the power of decentralized serverless IoT dataflow architecture for the Cloud-to-Edge Continuum: a performance comparison," *Ann. Telecommun.*, pp. 1–14, 2024, doi: 10.1007/s12243-023-01009-x.