

UDC: 007.5

COMMUNICATION ARCHITECTURES FOR CLOUD-INTEGRATED SPECTROMETRIC LABORATORIES: ESP-NOW VS MQTT

Andriy Krupych*, Pavlo Levush

Department of Radiophysics and Computer Technologies,
Ivan Franko National University of Lviv
107 Gen. Tarnavsky Str., UA-79017 Lviv, Ukraine

Krupych, A., Levush, P. (2026). Communication Architectures for Cloud-Integrated Spectrometric Laboratories: ESP-NOW vs MQTT. *Electronics and Information Technologies*, 33, 43–56
<https://doi.org/10.30970/eli.33.4>

ABSTRACT

Background. Cloud-integrated spectrometric laboratories face communication challenges in achieving real-time data access and analysis. This study compares two wireless protocols, MQTT (Message Queuing Telemetry Transport) and ESP-NOW (Espressif NOW), for LED control in such environments. MQTT offers lightweight, bandwidth-efficient, publish-subscribe messaging [1–3], while ESP-NOW provides energy-efficient direct communication without a Wi-Fi router. The objective is to evaluate their performance and suitability.

Materials and Methods. An experimental setup involved a StellarNet spectrometer, LED light sources, and ESP32 microcontrollers. Two architectures were tested: 1) direct MQTT Communication, where each ESP32 connected directly via Wi-Fi to an MQTT broker; and 2) an edge device with ESP-NOW relay, using an edge ESP32 for MQTT/Wi-Fi communication, then relaying commands via ESP-NOW to other ESP32s. Response times for LED control were measured over 100 cycles, and data were analyzed using descriptive statistics and an independent samples t-test.

Results and Discussion. Direct MQTT Communication exhibited significantly lower latency (median ~60 ms) and tighter distribution compared to the Edge Device with ESP-NOW Relay (median ~170 ms). A t-test confirmed a statistically significant difference ($t=46.28$), with MQTT demonstrating faster response times. However, the ESP-NOW relay system offers architectural advantages: reduced Wi-Fi dependency for individual nodes, enhanced deployment flexibility in areas with poor Wi-Fi coverage, improved scalability [4], and energy efficiency, making its higher latency a practical trade-off for large-scale laboratory integration.

Conclusion. Direct MQTT Communication provides superior low-latency performance. However, the edge device with ESP-NOW relay, despite higher latency, is a highly acceptable solution due to its flexibility, scalability, and reduced Wi-Fi dependency for distributed spectrometric laboratories. This highlights a critical trade-off between absolute speed and architectural benefits for robust, smart, cloud-enabled analytical laboratories.

Keywords: MQTT, ESP-NOW, spectrometric laboratory, Internet of Things, latency, embedded systems hardware

INTRODUCTION

Spectrometric laboratories are crucial in various scientific and industrial domains, generating vast amounts of data essential for research, quality control, and process



© 2026 Andriy Krupych & Pavlo Levush. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

optimization. The increasing complexity of analytical processes and the demand for real-time data access and analysis have driven the adoption of cloud integration within these laboratory environments [5]. Cloud-based systems offer significant advantages by providing a centralized infrastructure for data storage, archiving, and analysis, thereby supporting knowledge acquisition from analytical data [5, 6]. However, the integration of diverse analytical instruments and systems into a cohesive cloud-integrated framework presents considerable communication challenges [7].

The analytical laboratory often comprises numerous computer-assisted tools and software applications, yet communication constraints frequently impede the seamless sharing of scientific data [7]. Traditional wired data acquisition systems, while reliable, can limit accessibility and mobility, leading to higher installation and troubleshooting costs [8]. The shift towards wireless communication offers benefits such as reduced system size, lower costs, enhanced flexibility, and simplified deployments [8]. However, the diverse requirements of laboratory automation, including real-time data acquisition, scalability, energy efficiency, interoperability, and security, pose significant hurdles for selecting appropriate communication architectures [9].

In the context of the Internet of Things and smart laboratories, various wireless communication protocols have emerged to address these challenges. Among these, Message Queuing Telemetry Transport (MQTT) and ESP-NOW are prominent for their distinct characteristics. MQTT is a lightweight messaging protocol well-suited for IoT applications due to its low power consumption and bandwidth efficiency, often implemented on microcontrollers like the ESP32 [2]. It facilitates communication through a publish-subscribe model, enabling devices to connect and exchange data efficiently [10]. Conversely, ESP-NOW offers an energy-efficient, connectionless communication protocol that allows devices to directly exchange data without requiring a Wi-Fi router, making it ideal for scenarios requiring fast, local data exchange with minimal delay [11, 12]. It utilizes MAC addresses for device addressing and has a maximum message size of 250 bytes, offering an alternative when parallel Wi-Fi communication is not feasible [13].

This paper presents a comparative analysis of ESP-NOW and MQTT as communication architectures for cloud-integrated spectrometric laboratories. By examining their respective strengths and weaknesses in terms of performance, reliability, and suitability for the unique demands of laboratory environments, this study aims to provide insights into optimizing data flow and connectivity. The findings will contribute to the development of more efficient and robust communication strategies for the next generation of smart, cloud-enabled analytical laboratories.

MATERIALS AND METHODS

This section details the experimental design and methodologies employed to compare two distinct communication architectures for remotely controlling LED light sources (CHANZON 3V 5mm 30°: LED#1 – White, 12-14 cd; LED#2 – Warm White, 14-16 cd) and a miniature StellarNet GREEN-Wave VIS-50 fiber-optically coupled spectrometer within a cloud-enabled laboratory environment. The primary objective was to assess the performance, particularly in terms of command response times, and identify the pros and cons of each architecture.

Experimental Setup

The experimental setup was configured to simulate a practical cloud-integrated spectrometric laboratory, where LED light sources are controlled remotely, and their spectral data can be acquired.

The core hardware elements utilized in this study included:

- Spectrometer: A StellarNet spectrometer was used to acquire spectral data from the various LED light sources. This spectrometer served as the data collection point for the light emitted by the remotely controlled LEDs.

- LED Light Sources: Multiple LED light sources were implemented, each connected to an individual ESP32 microcontroller for remote control.
- Microcontroller Units: ESP32 development boards served as the primary control units for each LED light source. These boards are widely used in IoT applications [14–16]. An additional ESP32 board functioned as an edge device for one of the proposed communication architectures.

Two distinct communication architectures were designed and implemented for comparison.

Architecture 1: Direct MQTT Communication

In this architecture, each ESP32 device controlling a LED light source was directly connected to the local Wi-Fi network (**Fig. 1**). These individual ESP32 devices were configured as MQTT clients and subscribed to a specific topic on the HiveMQ MQTT broker. The HiveMQ MQTT broker served as the central message broker, facilitating the remote transmission of commands from a central control system to the individual LED-controlling ESP32s. This architecture required each ESP32 to maintain an active Wi-Fi connection and an MQTT client subscription.

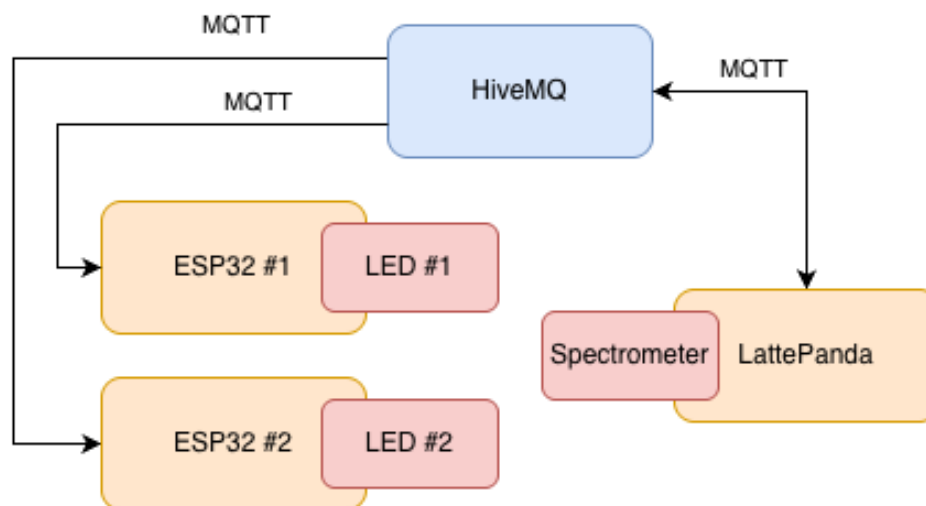


Fig. 1. Architecture 1 with direct MQTT connections

The LattePanda Single-Board Computer serves as the central control unit in this system, integrating both hardware-level control and high-level data processing capabilities. Its hybrid architecture, which combines a Windows-based computing environment with an embedded Arduino Leonardo microcontroller, makes it ideal for IoT applications requiring both GPIO pin access for direct hardware manipulation and Python-based spectral analysis.

Architecture 2: Edge Device with ESP-NOW Relay

This architecture introduced an intermediary "edge" ESP32 device (**Fig. 2**). This edge device was connected to the local Wi-Fi network and functioned as an MQTT client, subscribing to HiveMQ MQTT broker to receive commands. Upon receiving a command from the broker, the edge ESP32 identified the target LED and forwarded the command to the corresponding LED-controlling ESP32 device using the ESP-NOW protocol. Crucially, the individual ESP32 devices connected to the LED light sources in this architecture were not required to have a direct Wi-Fi connection, relying instead on the ESP-NOW communication link with the edge device.

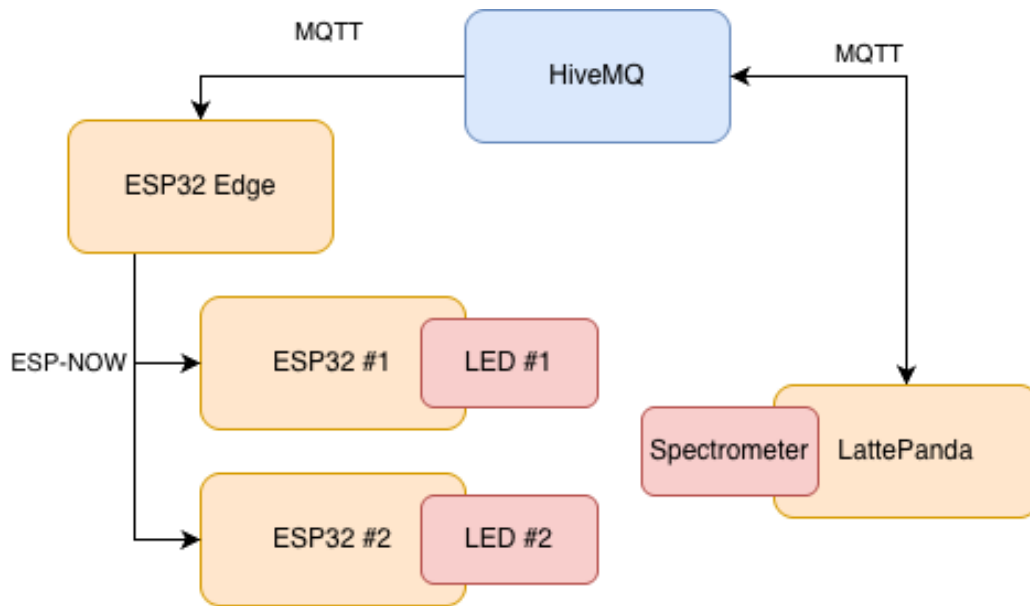


Fig. 2. Architecture 2 with ESP-NOW connection

Architecture 2 distinguishes itself from Architecture 1 by introducing an intermediary ESP32 edge device. This edge device is the sole component requiring a Wi-Fi connection and MQTT subscription, relaying commands via ESP-NOW to the individual LED-controlling ESP32s, thereby freeing them from direct Wi-Fi dependency.

Communication Protocol Implementation

The MQTT protocol was implemented using standard client libraries on the ESP32 platforms [17]. The HiveMQ MQTT broker served as the central message broker, facilitating communication between the control system and the ESP32 devices. In Architecture 1, each LED-controlling ESP32 is directly connected to this broker. In Architecture 2, only the edge ESP32 maintained a connection to the HiveMQ broker. Commands were structured as JSON payloads containing the target LED identifier and the desired action (turning on/off, adjusting intensity). Quality of Service levels could be varied during experiments to assess their impact on reliability and latency [18].

In Architecture 2, the edge ESP32 was configured as the sender for ESP-NOW messages, and the LED-controlling ESP32s were configured as receivers. Each LED-controlling ESP32 was registered with the edge device using its unique MAC address. Upon receiving an MQTT command, the edge ESP32 encapsulated the command into an ESP-NOW packet and transmitted it directly to the MAC address of the target LED-controlling ESP32. A callback function was implemented on both the sender and receiver sides to confirm message delivery. The connectionless nature of ESP-NOW was leveraged for fast, local data exchange without the overhead of establishing and maintaining Wi-Fi connections [19].

Command Transmission and Response Time Measurement

Commands for controlling the LED light sources were initiated from a central control application. The primary data collected for analysis was the response time, defined as the duration from when a command was issued by the central control application until the corresponding LED light source executed the command (e.g., changed its state or intensity). For both architectures, timestamps were recorded at the point of command issuance and at the point of command execution by the LED-controlling ESP32.

Performance Metrics and Measurement

The performance evaluation primarily focused on the response time, defined as the time taken for a data packet (command) to travel from the command source to the target LED-controlling ESP32 and be executed. This was measured by embedding timestamps at the origin and recording them upon reception and execution. The ability to measure and compare latency is crucial for evaluating communication systems [20].

Measurement tools included internal timers on the ESP32 devices for precise timing of command reception and execution, as well as timestamps recorded by the central control system for command issuance.

The use of MQTT QoS 2 (Exactly Once delivery) is particularly significant for the reliability discussion in this study. QoS 2 is the highest assurance level in the MQTT protocol, employing a four-step handshake (PUBLISH, PUBREC, PUBREL, PUBCOMP) to guarantee that each message is delivered exactly once to the subscriber, eliminating both message loss and duplication. In a spectrometric laboratory context, this is critical because duplicate LED commands could trigger redundant state changes during a spectral acquisition sequence, while lost commands could leave a light source in an incorrect state, compromising measurement integrity. The fact that QoS 2 was used in this study means the latency figures reported for both architectures already include the overhead of this four-step handshake, which inherently adds processing time compared to QoS 0 or QoS 1. This has two important implications: first, the MQTT leg of both architectures benefits from protocol-level delivery guarantees, providing a strong reliability baseline; second, the latency values presented represent a conservative (i.e., worst-case) scenario for MQTT performance, and lower latencies could be achieved if a less strict QoS level were acceptable for a given application.

In Architecture 2, to compensate for the absence of built-in reliability guarantees comparable to MQTT QoS levels, a programmatic retry mechanism was implemented on the ESP-NOW communication leg based on the protocol's delivery callback functionality. ESP-NOW provides a send callback (`on_data_sent`) that reports whether a transmitted packet was successfully received by the target peer at the MAC layer. When this callback indicated a delivery failure, the edge ESP32 device automatically re-attempted transmission of the command to the target LED-controlling ESP32, up to a predefined number of retries before reporting an error. This approach effectively introduces an application-layer reliability mechanism that mirrors, to a degree, the acknowledgment logic inherent in MQTT QoS levels, ensuring that transient interference or momentary unavailability of a receiver node does not result in a permanently lost command.

Experimental Procedure

Experiments were conducted to compare the response times of Architecture 1 and Architecture 2. Each experimental run involved:

1. *System Initialization*: All ESP32 devices and the StellarNet spectrometer were powered on and initialized. The HiveMQ MQTT broker was confirmed to be operational.
2. *Command Execution*: A series of predefined commands - sequential ON/OFF toggling of LEDs - were issued from the central control application.
3. *Data Logging*: Response times for each command were recorded for both architectures. This involved logging the timestamp of command issuance and the timestamp of command execution on the target ESP32.
4. *Repetition*: Each experimental scenario was repeated 100 times to ensure statistical significance and minimize the impact of transient network fluctuations. Experiments were conducted under consistent environmental conditions to ensure comparability.

Experiments were conducted under consistent environmental conditions to ensure comparability. Wi-Fi channel congestion was monitored; no other devices shared the

network, and tests were run at specific times to avoid interference. A dedicated access point with no competing traffic was used, along with fixed physical distances between devices.

A follow-up study measuring delivery success rates under varying network loads and interference conditions would substantially complement the latency analysis and provide a more complete basis for recommending one architecture over the other.

Data Analysis

Raw response time data, collected from the experimental runs, were processed using custom scripts developed in the Python programming language. For statistical analysis, Python was also utilized, incorporating relevant statistical libraries such as *Pandas* for data manipulation, *Scipy.stats* for inferential statistics, and *Matplotlib.pyplot* and *Seaborn* for data visualization.

Descriptive statistics, including mean, median, and measures of spread, were calculated for the response times of both the Direct MQTT Communication and Edge Device with ESP-NOW Relay architectures.

To determine if the observed differences in mean response times between the two architectures were statistically significant, an independent samples t-test was performed. This inferential statistical test is specifically designed to compare the means of two independent groups. A significance level of $p < 0.05$ was considered statistically significant for all comparisons. This conventional threshold was initially proposed by R. Fisher and remains a widely accepted standard in many scientific fields [21–23].

Data visualization techniques, including histograms and scatter plots, were employed to illustrate the distribution characteristics, variability, and overall comparison of response times for both architectures. These visualizations complemented the statistical analysis by providing a clear graphical representation of the performance differences.

RESULTS AND DISCUSSION

This section presents empirical results obtained from the comparative analysis of Direct MQTT Communication and Edge Device with ESP-NOW Relay. The measured response times for LED control commands are analyzed, followed by a statistical comparison and a discussion of the practical implications and acceptability of each architecture within a cloud-integrated spectrometric laboratory.

Latency Analysis

The performance of both communication architectures was evaluated based on command response times over 100 experimental cycles. In practice, 100 repeated measurements per ESP device (yielding $n=200$ per architecture per device and $n=400$ overall) provided sufficient statistical power to detect the large effect size observed — the t-statistic of -46.28 reflects an enormous difference relative to within-group variance, meaning even a smaller sample would have reached significance. The central limit theorem also ensures that with $n=400$, sampling distributions of the means are well-approximated as normal, supporting the validity of the t-test.

The Direct MQTT Communication architecture consistently demonstrated lower latency (**Fig. 3**). As illustrated in the Histogram of MQTT Latency (**Fig. 4**), the response times were heavily concentrated at the lower end of the spectrum, indicating quick command execution.

Median response time was approximately 60 ms. The interquartile range for MQTT latency was tight, spanning roughly from 55 ms to 75 ms, suggesting consistent performance for the majority of command cycles. However, a few outliers were observed, with one significant spike approaching 440 ms. These transient peaks could be attributed to network congestion, Wi-Fi interference, or processing delays within the MQTT broker or the individual ESP32 Wi-Fi stack.

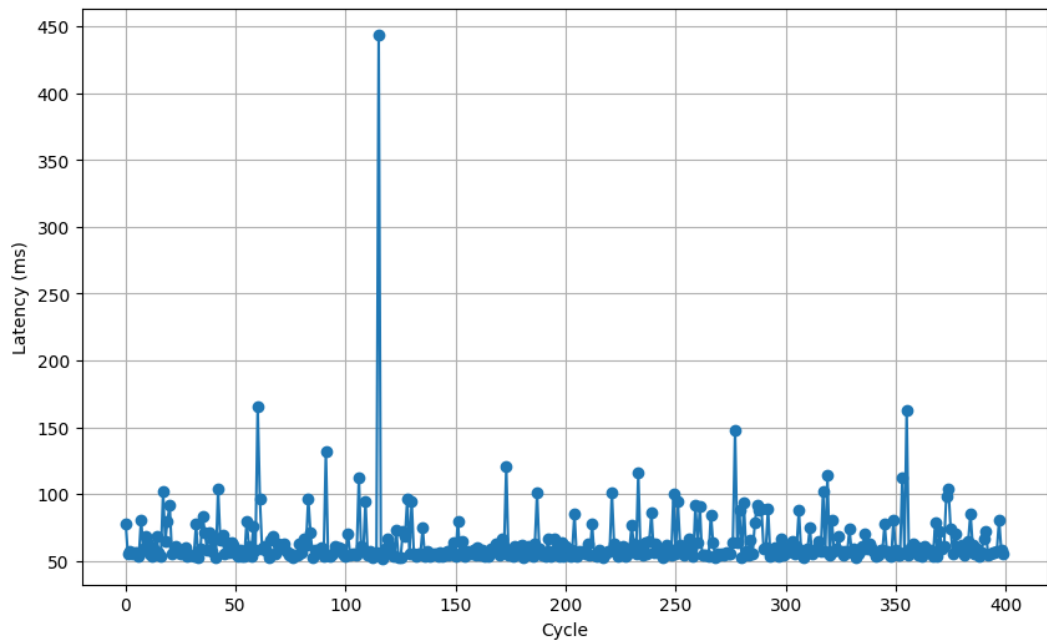


Fig. 3. MQTT architecture latency over 100 cycles plot.

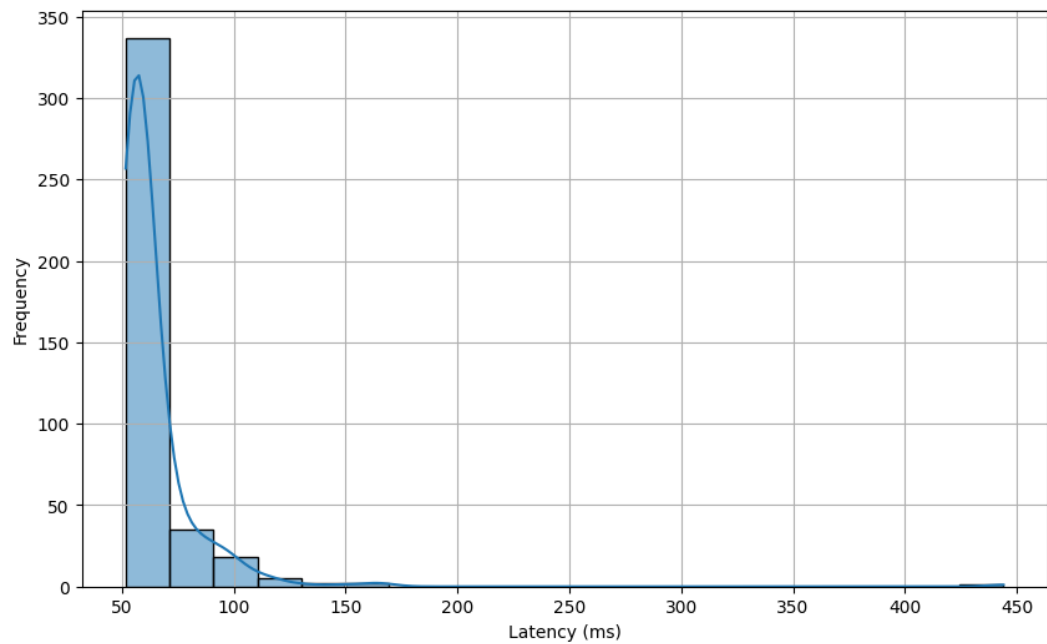


Fig. 4. Histogram of MQTT architecture.

In contrast, the edge device with ESP-NOW relay architecture exhibited generally higher response times (Fig. 5). The histogram of NOW Latency (Fig. 6) reveals a broader distribution of latencies, with the peak shifted towards approximately 170 ms.

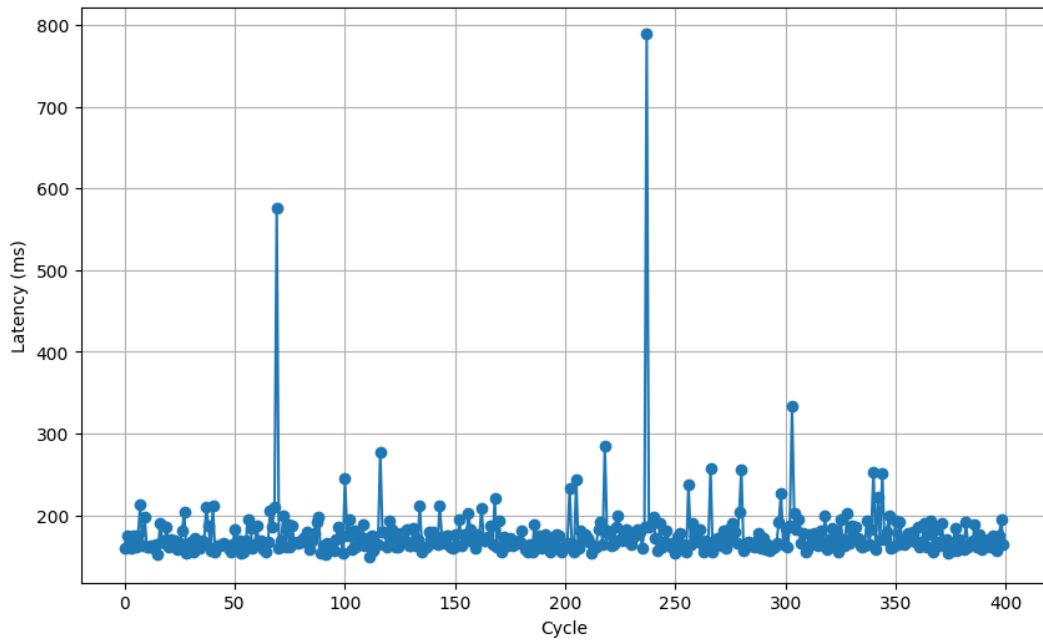


Fig. 5. ESP-NOW architecture latency over 100 cycles plot.

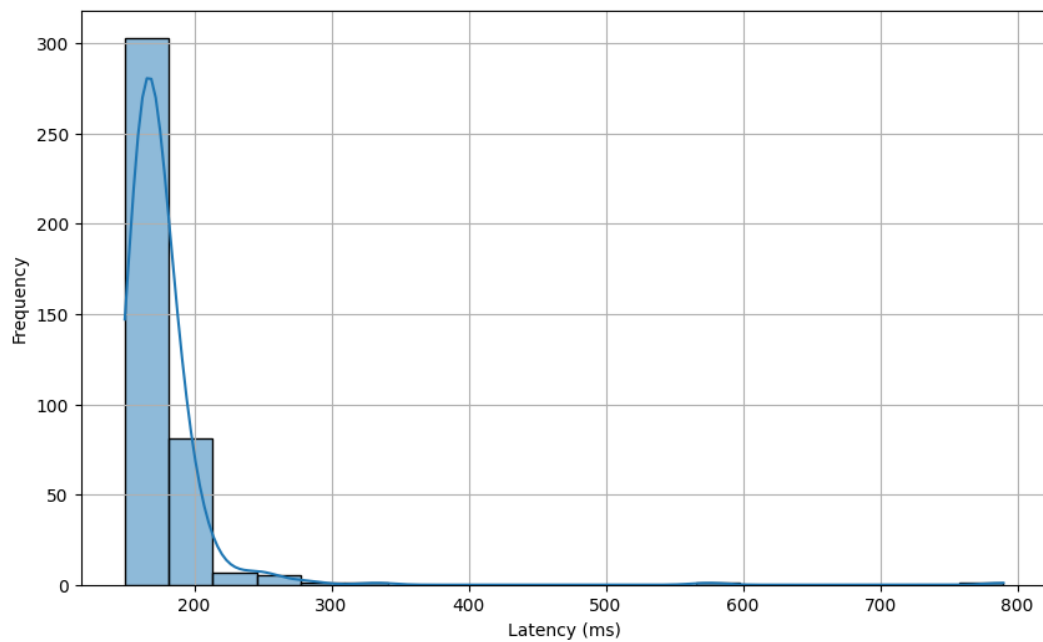


Fig. 6. Histogram of ESP-NOW architecture.

Median response time was around 170 ms. The interquartile range for ESP-NOW was wider than that of MQTT, indicating greater variability in response times. Furthermore, the NOW Latency Over 100 Cycles plot highlighted more frequent and higher magnitude outliers compared to MQTT, with some response times reaching approximately 580 ms and a maximum of about 790 ms. These increased latencies and variability are likely a

consequence of the added processing time at the edge device for translating MQTT commands to ESP-NOW messages, the additional communication hop, and potential overheads in the ESP-NOW protocol itself.

Statistical Comparison

A qualitative statistical comparison of the two architectures reveals distinct characteristics (**Table 1**). MQTT consistently provided lower median latency and a tighter distribution of response times, suggesting superior real-time performance and greater consistency for command execution. The spread of data, as represented by the IQR, was considerably smaller for MQTT, indicating less variability. While both protocols experienced latency spikes, those observed in the ESP-NOW architecture were generally of higher magnitude and occurred with greater frequency.

The results of the independent samples t-test, comparing the two datasets, yielded a t-statistic of approximately -46.28 . This t-statistic, with its large absolute value, indicates a substantial difference between the mean latencies of the two communication architectures. The negative value suggests that the mean latency for Architecture 1 was significantly lower than that for Architecture 2.

This statistically significant result allows us to confidently reject the null hypothesis, which posits that there is no difference in the mean response times between the two architectures.

Therefore, the statistical analysis confirms that there is a highly significant difference in the command response times, with architecture 1 demonstrating a statistically significantly lower latency compared to architecture 2. This quantitative finding supports the qualitative observations made during the initial latency analysis, solidifying the conclusion that MQTT provides faster and more consistent command execution in this experimental setup.

Table 1. Summary of latency statistics

Statistics		Latency, ms	
		Architecture 1 (MQTT → ESP)	Architecture 2 (MQTT → EDGE → ESP-NOW → ESP)
ESP1 (n = 200)	Min	51.76	153.22
	Max	165.91	789.92
	Average	62.97	179.52
ESP2 (n = 200)	Min	52.96	149.39
	Max	444.03	332.91
	Average	64.91	173.44
OVERALL (n = 400)	Min	51.76	149.39
	Max	444.03	789.92
	Average	63.94	176.48

Discussion of Architectural Implications and Acceptability

The choice between architectures involves a trade-off between absolute latency performance and architectural flexibility, scalability, and deployment ease.

While direct MQTT communication clearly demonstrates lower latency, making it ideal for applications requiring almost real-time, highly consistent communication, its deployment necessitates that every single ESP32 device controlling an LED light source maintains an

active Wi-Fi connection and an MQTT client subscription. This approach can lead to several challenges in a large-scale spectrometric laboratory or institutional setting:

- *Wi-Fi Network Load*: Many Wi-Fi-connected devices can strain the access point's capacity and bandwidth.
- *Configuration Overhead*: Managing Wi-Fi credentials and MQTT subscriptions for numerous individual devices can be complex.
- *Coverage Limitations*: Devices in areas with poor Wi-Fi coverage would struggle to maintain connectivity.

Despite its higher latency, architecture 2 offers compelling advantages that make it highly acceptable, and in many scenarios, a more practical solution:

- *Reduced Wi-Fi Dependency*: Only the central edge ESP32 device requires a Wi-Fi connection to the MQTT broker. The individual LED-controlling ESP32s communicate directly with the edge device via ESP-NOW, eliminating their need for a dedicated Wi-Fi connection.
- *Enhanced Deployment Flexibility*: This architecture allows for the "spreading" of ESP-NOW nodes throughout an institution or laboratory space, even in areas with limited or no Wi-Fi connectivity. This is particularly beneficial for experimental setups that are physically dispersed or located in difficult-to-reach areas.
- *Scalability*: ESP-NOW can support numerous devices without each consuming a Wi-Fi connection resource, simplifying network management and potentially scaling better in terms of Wi-Fi access point load. The ability to deal with resource-constrained devices and support frequent connection/disconnection also contributes to scalability [4].
- *Energy Efficiency*: ESP-NOW is designed as a low-power, connection-less communication protocol, making it suitable for sensor nodes where energy conservation is critical, as they do not need to maintain a constant Wi-Fi connection.
- *Acceptable Latency for Application*: For the specific application of remotely controlling LED light sources in a spectrometric laboratory, the observed median latency of approximately 170 ms, even with occasional spikes, is generally well within acceptable limits. Commands such as turning LEDs ON/OFF or adjusting intensity are typically not time-critical to the sub-millisecond level. Human operators or automated scripts can easily tolerate response times within a few hundreds of milliseconds, or even up to a second, without significant impact on the experimental workflow or data integrity. The primary requirement is reliable execution, which ESP-NOW provides through its packet delivery mechanisms [19].

In summary, it should be noted that while architecture 1 offers superior raw latency performance, architecture 2 presents a highly acceptable and often preferred solution for building a cloud-integrated spectrometric laboratory due to its significant architectural advantages. The flexibility to deploy numerous devices without individual Wi-Fi connections, coupled with its scalability and energy efficiency benefits, outweighs the higher, yet application-acceptable, latency for many practical laboratory automation scenarios.

CONCLUSION

This study meticulously compared two distinct communication architectures – direct MQTT communication and an edge device with ESP-NOW relay – for their suitability in cloud-integrated spectrometric laboratories, focusing on command response times for LED light source control. The increasing complexity of analytical processes and the demand for real-time data access and analysis have driven the adoption of cloud integration within these laboratory environments. This research aimed to evaluate their performance, reliability, and practical implications within this specialized environment.

The empirical analysis, reinforced by a rigorous independent samples t-test, conclusively demonstrated a statistically significant difference in latency between the two architectures. Direct MQTT Communication exhibited a significantly lower mean latency, indicating superior speed and consistency in command execution when each ESP32 device maintained a direct Wi-Fi and MQTT connection. MQTT's lightweight nature, low power consumption, and bandwidth efficiency make it well-suited for IoT applications [1–3]. The t-test results, with a t-statistic of approximately -46.28 , unequivocally support this finding.

However, the research also highlighted that absolute latency is not the sole determinant of an architecture's practical utility. While the edge device with ESP-NOW relay presented a higher, though application-acceptable, latency, it offered substantial advantages in architectural flexibility, scalability, and reduced Wi-Fi dependency. ESP-NOW is an energy-efficient, connectionless protocol that allows devices to directly exchange data without requiring a Wi-Fi router and utilizes MAC addresses for device addressing. This architecture is particularly beneficial for large or geographically dispersed laboratory setups where direct Wi-Fi connectivity for every node is impractical or creates excessive network load. By centralizing the Wi-Fi connection through an edge device, the ESP-NOW relay system enables the deployment of numerous sensor nodes in areas with limited or no Wi-Fi infrastructure, simplifying network management and enhancing energy efficiency for individual devices. The shift towards wireless communication, in general, offers benefits such as reduced system size, lower costs, enhanced flexibility, and simplified deployments compared to traditional wired systems [8, 24].

In conclusion, the choice between these two architectures necessitates careful consideration of the specific demands of the spectrometric laboratory. For applications demanding the absolute lowest latency and where pervasive Wi-Fi coverage is guaranteed, direct MQTT communication is the preferred choice. Conversely, for scenarios prioritizing architectural flexibility, broad deployment across an institution, scalability (due to less strain on Wi-Fi resources), and reduced individual device Wi-Fi dependency – even with slightly elevated but still acceptable latencies – the edge device with ESP-NOW relay stands out as a highly effective and practical solution. This comparative analysis provides valuable insights for optimizing communication strategies in the development of more efficient and robust smart, cloud-enabled analytical laboratories.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [A.K.]; methodology, [A.K.]; validation, [P.L.]; formal analysis, [A.K.]; investigation, [A.K., P.L.]; resources, [P.L.]; data curation, [P.L.]; writing – original draft preparation, [A.K.]; writing – review and editing, [P.L.]; visualization, [A.K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

- [1] Domingues, M., Faria, J. N., & Portugal, D. (2024). Dimensioning payload size for fast retransmission of MQTT packets in the wake of network disconnections. *EURASIP Journal on Wireless Communications and Networking*, 2024(1). <https://doi.org/10.1186/s13638-023-02327-3>.

- [2] Kamil, A. A., Mousa, A. J., & Abdul-Rahaim, L. A. (2024). Smart Cloud Computing System for Environment Based on MQTT Protocol and Node MCU (ESP8266). *Journal Européen Des Systèmes Automatisés*, 57(5), 1503. <https://doi.org/10.18280/jesa.570526>.
- [3] Shah, P. H. (2024). MQTT Systems: A Survey. *International Journal for Research in Applied Science and Engineering Technology*, 12(3), 1063. <https://doi.org/10.22214/ijraset.2024.59000>.
- [4] Puthiyidam, J.J., & Joseph, S. (2017). IoT Smart Home: Protocols and Architectures. *International Journal for Research in Applied Science and Engineering Technology*, 5, 2031. <https://doi.org/10.22214/ijraset.2017.11293>.
- [5] Eisen, K., Eifert, T., Herwig, C., & Maiwald, M. (2020). Current and future requirements to industrial analytical infrastructure—part 1: process analytical laboratories. *Analytical and Bioanalytical Chemistry*, 412(9), 2027. <https://doi.org/10.1007/s00216-020-02420-2>.
- [6] Arco, E., Boccardo, P., Gandino, F., Lingua, A. M., Noardo, F., & Rebaudengo, M. (2016). An Integrated Approach for Pollution Monitoring: Smart Acquisition and Smart Information. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 67. <https://doi.org/10.5194/isprs-annals-iv-4-w1-67-2016>.
- [7] Gardiner, S., Haynie, C., & Corte, D. D. (2024). Rise of the Allotrope Simple Model: Update from 2023 Fall Allotrope Connect. *Drug Discovery Today*, 29(4), 103944. <https://doi.org/10.1016/j.drudis.2024.103944>.
- [8] Dohare, R. K., Mainuddin, M., Verma, A., & Singhal, G. (2022). Uncertainty evaluation of data acquisition and analysis system relevant to infrared flowing medium laser. *Scientific Reports*, 12(1). <https://doi.org/10.1038/s41598-022-22667-2>.
- [9] Munir, T., Akbar, M. S., Ahmed, S., Sarfraz, A., Sarfraz, Z., Sarfraz, M., Félix, M., & Chérrez-Ojeda, I. (2022). A Systematic Review of Internet of Things in Clinical Laboratories: Opportunities, Advantages, and Challenges. *Sensors*, 22(20), 8051. <https://doi.org/10.3390/s22208051>.
- [10] Knight, N., Kanza, S., Cruickshank, D., Brocklesby, W. S., & Frey, J. G. (2020). Talk2Lab: The Smart Lab of the Future. *IEEE Internet of Things Journal*, 7(9), 8631. <https://doi.org/10.1109/jiot.2020.2995323>.
- [11] Nguyen, H., Nguyen, L. V., & Ha, Q. P. (2020). IoT-enabled Dependable Co-located Low-cost Sensing for Construction Site Monitoring. *Proceedings of the 37th ISARC*, 616-624. <https://doi.org/10.22260/isarc2020/0086>.
- [12] Wicaksono, M. F., & Rahmatya, M. D. (2022). IoT for Residential Monitoring Using ESP8266 and ESP-NOW Protocol. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 8(1), 93. <https://doi.org/10.26555/jiteki.v8i1.23616>.
- [13] Rákay, R., Galajdová, A., Šeminský, J., & Cvitić, I. (2019). Selected Wireless Communication Protocols and their Properties for Use in IoT Systems. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 27(45), 26. <https://doi.org/10.2478/rput-2019-0022>.
- [14] Alangcas, A., Marjhon, K., Daligdig, C., & Encarnacion, P. (2024). Embedded IoT Data Collection for Snore Analysis. *International Journal of Advanced Trends in Computer Science and Engineering*, 13(3), 119. <https://doi.org/10.30534/ijatcse/2024/041332024>.
- [15] Babiuch, M., & Postulka, J. (2020). Smart Home Monitoring System Using ESP32 Microcontrollers. In *Internet of things*. Springer Science+Business Media. <https://doi.org/10.5772/intechopen.94589>.
- [16] Le, L. H., Nguyen, D. T., Bui, H. P., Nguyen, C. C., & Ngo, T. Q. (2024). Integration of Modbus Ethernet Communication for Real-Time Electrical Power Consumption,

- Temperature, and Humidity Monitoring System. *Research Square*. <https://doi.org/10.21203/rs.3.rs-4708973/v1>.
- [17] Van Anh, N. T., Anh, N. X. D., Son, L. H., & Khanh, D. Q. (2024, December). A New Architecture for Controlling IoT Devices of Smart Room Using ESP32 Microcontroller. In *International Conference on Smart Objects and Technologies for Social Good* (pp. 106-117). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-01472-6_9.
- [18] Abid, A., Jazib, M., & Riaz, M. (2025). RCD-IoT: Enabling Industrial Monitoring and Control with Resource-Constrained Devices Under High Packet Transmission Rates. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2501.07895>.
- [19] Urazayev, D., Eduard, A., Ahsan, M., & Zorbas, D. (2023, May). Indoor performance evaluation of esp-now. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)* (pp. 1-6). IEEE. <https://doi.org/10.1109/sist58284.2023.10223585>.
- [20] Escobar, J. J. L., Redondo, R. P. D., & Gil-Castiñeira, F. (2024). Unleashing the power of decentralized serverless IoT dataflow architecture for the Cloud-to-Edge Continuum: a performance comparison. *Annals of Telecommunications*, 79, 135–148. <https://doi.org/10.1007/s12243-023-01009-x>.
- [21] Grawe, N. D., & Karaali, G. (2022). Talking about Statistical Significance in Numeracy. *Numeracy*, 15(2), 8. <https://doi.org/10.5038/1936-4660.15.2.1424>.
- [22] Lalchand, V. (2020). A meta-algorithm for classification using random recursive tree ensembles: A high energy physics application. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2001.06880>.
- [23] Maier, M., & Lakens, D. (2022). Justify Your Alpha: A Primer on Two Practical Approaches. *Advances in Methods and Practices in Psychological Science*, 5(2), 25152459221080396. <https://doi.org/10.1177/25152459221080396>.
- [24] Araujo, A., García-Palacios, J., Blesa, J., Tirado, F., Romero, E., Samartín, A., & Nieto-Taladriz, O. (2011). Wireless Measurement System for Structural Health Monitoring with High Time-Synchronization Accuracy. *IEEE Transactions on Instrumentation and Measurement*, 61(3), 801-810. <https://doi.org/10.1109/tim.2011.2170889>.

КОМУНІКАЦІЙНІ АРХІТЕКТУРИ ДЛЯ ХМАРНО-ІНТЕГРОВАНИХ СПЕКТРОМЕТРИЧНИХ ЛАБОРАТОРІЙ: ПОРІВНЯННЯ ESP-NOW ТА MQTT

Андрій Крупич*^{ORCID}, Павло Левуш^{ORCID}

Кафедра радіофізики та комп'ютерних технологій,
Львівський національний університет імені Івана Франка
вул. Тарнавського, 107, 79017 Львів, Україна

АНОТАЦІЯ

Вступ. Хмарно-інтегровані спектрометричні лабораторії стикаються з проблемами зв'язку в забезпеченні доступу до даних та їх аналізу в режимі реального часу. У цьому дослідженні порівнюються два бездротові протоколи, MQTT та ESP-NOW, для керування світлодіодами в таких середовищах. MQTT пропонує легкий, ефективний з точки зору пропускної здатності обмін повідомленнями публікації та підписки [1–3], тоді як ESP-NOW забезпечує бездротовий енергоефективний прямий зв'язок без маршрутизатора Wi-Fi. Метою є оцінка їхньої продуктивності та придатності.

Матеріали та методи. Експериментальна установка включала спектрометр StellarNet, світлодіодні джерела світла та мікроконтролери ESP32. Було протестовано

дві архітектури: 1) Прямий зв'язок MQTT, де кожен ESP32 підключався безпосередньо через Wi-Fi до брокера MQTT; та 2) Керуючий пристрій з ESP-NOW, використовуючи керуючий ESP32 для зв'язку по MQTT/Wi-Fi, а потім передаючи команди через ESP-NOW іншим ESP32. Час відгуку для керування світлодіодами вимірювався протягом 100 циклів, а дані аналізувалися за допомогою описової статистики та t-тесту незалежних вибірок.

Результати. Прямий зв'язок MQTT продемонстрував меншу затримку (медіана ~60 мс) та щільніший розподіл порівняно з ESP-NOW (медіана ~170 мс). T-тест підтвердив статистично значущу різницю ($t = -46,28$), при цьому MQTT продемонстрував швидший час відгуку. Однак ESP-NOW пропонує архітектурні переваги: меншу залежність окремих вузлів від Wi-Fi, гнучкість розгортання з поганим покриттям Wi-Fi, кращу масштабованість [4] та енергоефективність, що робить її вищу затримку практичним компромісом для інтеграції масштабної лабораторії.

Висновки. Прямий MQTT забезпечує кращу продуктивність з низькою затримкою. Однак, керуючий пристрій з ESP-NOW, незважаючи на вищу затримку, є прийнятним рішенням завдяки своїй гнучкості, масштабованості та меншій залежності від Wi-Fi для розподілених лабораторій. Це підкреслює компроміс між абсолютною швидкістю та архітектурними перевагами для надійних аналітичних лабораторій, що працюють у хмарі.

Ключові слова: MQTT, ESP-NOW, спектрометрична лабораторія, Інтернет речей, бездротовий зв'язок, затримка.