Performance Evaluation of an ESPresense/MQTT Architecture for Real-Time Locating System

Emanuel de Franceschi Vieira[1], Tiago Antônio Rizzetti[1]

[1]Colégio Técnico Industrial de Santa Maria (CTISM)
Universidade Federal de Santa Maria (UFSM)

emanuel.franceschi@acad.ufsm.br, tiago.rizzetti@ufsm.br

Abstract. *This work evaluates the performance of a Real-Time Location System (RTLS) based on low-cost and open-source technologies. The system uses ESP32 devices running the ESPresense firmware to detect Bluetooth Low Energy (BLE) advertisements and publish data through the MQTT protocol. The performance tests were conducted on a single locator device that is part of an existing system deployed in a real environment, where multiple locators are distributed across different rooms. The results indicated an average latency of approximately 85 ms under low traffic and around 44 ms under continuous traffic, both with 0% message loss and stable response times. Furthermore, the throughput increased almost linearly with the number of active beacons, showing that the ESP32 did not become a processing bottleneck within the evaluated limits. These findings demonstrate that the ESPresense/MQTT architecture provides adequate performance, robustness, and scalability for medium-sized RTLS deployments at low cost.*

1. Introduction

Real-Time Location Systems (RTLS) have gained prominence in indoor environments such as industries, warehouses, factories, hospitals, and research centers due to their ability to monitor, in real time, the position of people and objects [Thiede et al. 2021]. However, commercial solutions often present high costs and require specific infrastructure, which hinders their adoption in research projects and low-budget applications [Fernández-Conde 2023]. Thus, there is a demand for low-cost RTLS architectures based on open technologies.

A promising approach is the use of Commercial Off-the-Shelf (COTS) technologies, that is, ready-to-use hardware and software combined with open-source platforms [Liubimov and Liubimov 2023]. This strategy enables cost reduction and increases development flexibility. In the context of this work, solutions such as ESPresense are explored, a firmware for ESP32 capable of detecting Bluetooth Low Energy (BLE) advertisements and transmitting the readings over MQTT, relying exclusively on open and low-cost technologies.

For this study, an existing architecture operating in a real environment was employed, characterized by a simple and modular structure. In this architecture, BLE beacons periodically emit advertisements containing their identifiers; ESP32 sensor nodes running the ESPresense firmware capture these advertisements and extract information such as the beacon identifier, Received Signal Strength Indicator (RSSI), and estimated distance. The resulting data are then published to specific MQTT topics, which are received by the central broker and forwarded to subscribers, in this case, the Home Assistant platform.

The objective of this work is to evaluate the performance of the data collection layer of this architecture. The main contributions are: (1) proposing a methodology to assess the performance of ESPresense/MQTT architectures; and (2) presenting quantitative results regarding latency and scalability in a real production scenario.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the system architecture. Section 4 describes the methodology, detailing the test scenario, performance metrics, and experiments conducted. Section 5 presents the results, and finally, Section 6 concludes the paper.

2. Related Work

This section presents the related work that supports and motivates this study. The selected works address BLE-based localization, the use of ESP32 devices, and MQTT communication, providing context for the performance evaluation carried out in this research.

The work in [Al-Maktary et al. 2025] investigates the use of RSSI on ESP32 modules for distance estimation in indoor environments, targeting asset tracking applications. The authors analyze different scenarios such as line-of-sight, wall obstruction, and mobility, observing that although RSSI is correlated with distance, its variability increases significantly in the presence of obstacles and as distance grows. The results indicate that the estimation is relatively reliable up to approximately 4 meters, but becomes imprecise in more complex environments, highlighting the need for filtering, sensor fusion, or more robust modeling to improve positioning accuracy.

In [Misal et al. 2020], the authors present a system based on BLE, ESP32, and MQTT, using trilateration to estimate the position of a beacon within an indoor environment. In this system, multiple ESP32 nodes act as receivers and send RSSI values to a server via MQTT, which computes distances and determines position on a map. The solution demonstrates practical feasibility with low implementation cost, achieving an average accuracy of approximately 2.3 meters. However, the authors emphasize that RSSI variability remains a major limitation, particularly in the presence of obstacles and environmental factors.

Unlike these works, the present study does not focus on direct position estimation or distance calculation accuracy. Instead, it evaluates the performance of the ESPresense/MQTT architecture by analyzing latency, reliability, and scalability under different quantities of beacons and message traffic. Thus, the contribution lies in providing an operational performance perspective of the infrastructure, allowing future implementations to use these results as guidance for deployment and system sizing.

3. System Architecture

The evaluated architecture, illustrated in Figure 1, is composed of two main layers: the sensor nodes (referred to as locators) and the MQTT broker, implemented using Eclipse Mosquitto on a dedicated virtual machine. The sensors, based on ESP32 running the ESPresense firmware, continuously scan BLE advertisements and publish RSSI and estimated distance data to specific MQTT topics. The broker acts as the central point for message collection and distribution, and all MQTT communication was performed using QoS 0. Although Home Assistant is present in the environment, it does not participate in the performance tests, serving only as a passive subscriber for visualization of location data in the production environment.

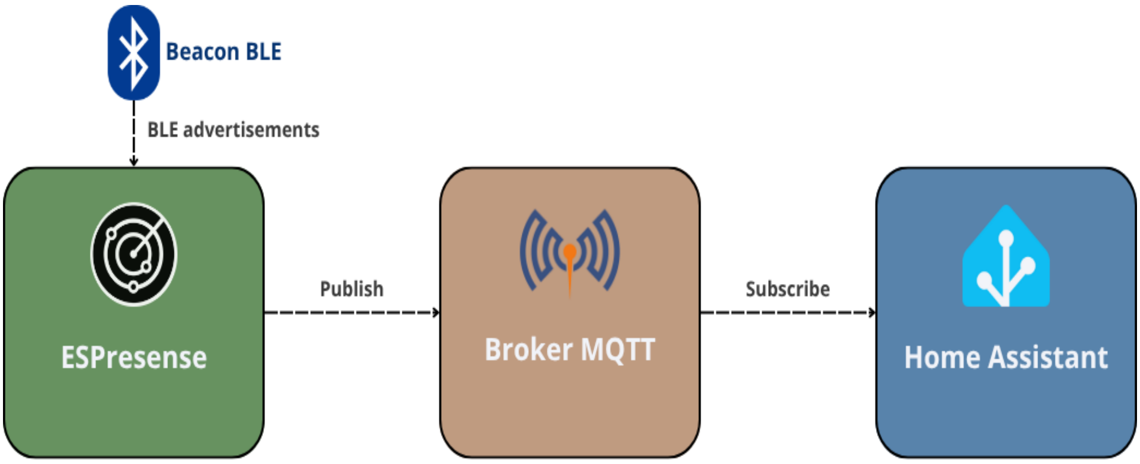

Figure 1. System architecture

4. Methodology

This section describes the methodology used to evaluate the performance of the ESPresense/MQTT data collection architecture. The experiments were conducted in a real production environment composed of 10 ESP32 sensor nodes, each installed at a central position within rooms of approximately 70 m², with all devices operating on the same Wi-Fi network and connected to a single MQTT broker. The device targeted for performance testing was designated as sala_testes.

Three main metrics were defined to evaluate the efficiency of the architecture: latency, reliability, and throughput. Latency was measured as the round-trip time (RTT) of an MQTT packet, using a Python script that published messages to the topic *"espresense/ping"*; the ESP32 running the ESPresense firmware was modified to subscribe to this topic and respond immediately on *"espresense/pong"*, allowing RTT measurement and estimation of one-way latency as RTT/2. Reliability corresponds to the response delivery rate, calculated as the ratio between received and sent messages. Finally, throughput was defined as the number of messages per second received by the broker, measured by a Python script subscribing to the topic *"espresense/devices/+/sala_testes"* for 60 seconds to count all publications from the ESP32 sensors.

Two experiments were conducted. The first analyzed latency as a function of MQTT traffic frequency, using 100 packets in two scenarios: low load (5 s interval) and high load (0.1 s interval), to observe how message frequency influences response time. The second experiment evaluated throughput and scalability by varying the number of active BLE beacons in the coverage area (1, 5, 10, 20, and 40), measuring for 60 seconds the total volume of messages published and received at the broker. The limit of 40 devices corresponds to the maximum number of physical beacons available during the experiments. Each test was repeated five times to ensure consistency and reproducibility.

5. Results and Discussion

This section presents and discusses the results obtained, highlighting the impact of latency and throughput metrics on the applicability of the ESPresense/MQTT architecture in RTLS systems.

As described in Section 4, the first experiment evaluated latency under different MQTT traffic loads. Table 1 shows the minimum, maximum, mean, median, and packet loss observed in both scenarios.

Scenario	Min (ms)	Max (ms)	Mean (ms)	Median (ms)	Loss (%)
low load	11.00	223.99	85.95	77.36	0.00
high load	5.69	190.50	44.15	38.84	0.00

Table 1. Latency test results.

Figure 2 shows the RTT distribution for both conditions. The latency values remained stable and within acceptable limits in both scenarios. Under low traffic (5 s interval), response times were slightly higher, as expected, but remained consistent and without significant variation. In both cases, the system maintained full reliability (0% packet loss). These results indicate that the architecture provides predictable latency behavior and robust performance across different message frequencies, reinforcing its suitability for real-time location applications.

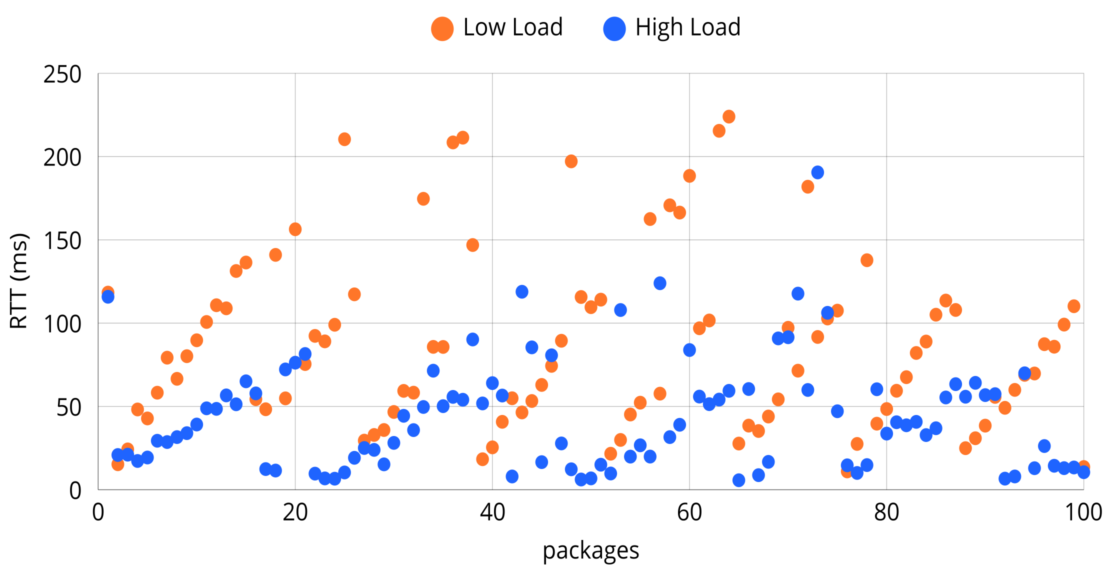

Figure 2. RTT distribution under low and high traffic conditions

The second experiment evaluated throughput and scalability by varying the number of BLE beacons. Table 2 presents the results, including the total number of messages received and the average update rate corresponding to each beacon load.

Number of Beacons	Messages Received	Rate (msg/s)
1	17	0.28
5	71	1.18
10	132	2.20
20	252	4.20
40	491	8.17

Table 2. Throughput test results.

Figure 3 illustrates the relationship between the number of beacons and the corresponding update rate. The results show an almost linear increase in throughput with the number of beacons, indicating that the ESP32 node processed and published all readings without significant loss or saturation up to 40 devices. This confirms that the ESPresense/MQTT architecture is scalable and suitable for medium-scale RTLS deployments.

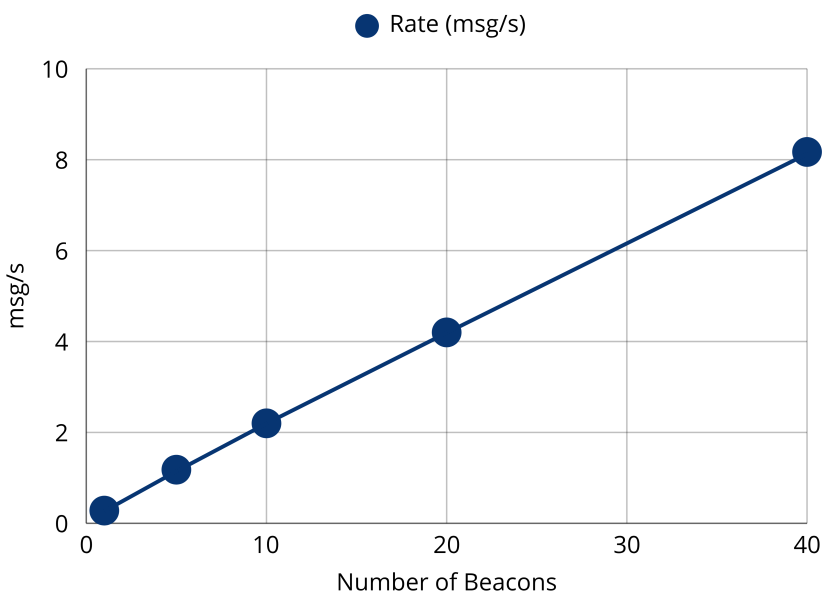

Figure 3. Throughput scalability with increasing number of BLE beacons

6. Conclusion

This work evaluated the performance of the ESPresense/MQTT architecture in a real-time location system. The experimental results demonstrate that the system is highly reliable and scalable, with throughput proportional to the number of monitored beacons and no message loss. The processing capability of the ESP32 was not saturated up to 40 devices per sensor.

The results showed stable and consistent response times in all scenarios, demonstrating that the architecture maintains good performance even under sparse message traffic, with mean latency remaining adequate for real-time tracking applications. Therefore, the ESPresense/MQTT architecture provides sufficient performance, reliability, and scalability for effective RTLS implementations. Future work will evaluate the impact of encrypted MQTT communication on latency and investigate broker saturation limits under multiple simultaneous sensor nodes.

Acknowledgments

This work was carried out with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)

References

Al-Maktary, O., Susanto, M., and Mardiana, M. (2025). Evaluation of rssi-based distance estimation with esp32 ble modules for indoor asset tracking. *ELKHA : Jurnal Teknik Elektro*, 17:172–179.

Fernández-Conde, J. (2023). A low-cost hardware/software platform for lossless real-time data acquisition from imaging spectrometers. *MDPI: Sensors*, 23(9).

Liubimov, O. and Liubimov, M. (2023). Use of open-source cots/mots hardware and software platforms for the build up of the cubesat nanosatellites. *Journal of Rocket-Space Technology*, 31(4):138–147.

Misal, S. R., Prajwal, S. R., Niveditha, H. M., Vinayaka, H. M., and Veena, S. (2020). Indoor positioning system (ips) using esp32, mqtt and bluetooth. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pages 79–82.

Thiede, S., Sullivan, B., Damgrave, R., and Lutters, E. (2021). Real-time locating systems (rtls) in future factories: technology review, morphology and application potentials. *Procedia CIRP*, 104:671–676. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0.