Energy Profiling of Secure MQTT over WiFi on ESP32-Based IoT Devices

Enrico Mayo[*], Miguel Gutiérrez Gaitán[*†], Silvia E. Restrepo[‡], Miguel Solís[§],
Felipe Nuñez[*], Francesca Marcello[¶], Virginia Pilloni[¶]
[*] Departamento de Ingeniería Eléctrica, Pontificia Universidad Católica de Chile, Chile
[†] CISTER Research Centre, Portugal
[‡]Universidad Católica de la Santísima Concepción, Chile
[§]Universidad Nacional Andrés Bello, Chile
[¶]University of Cagliary, Italy

***Abstract*—This paper investigates the energy impact of Transport Layer Security (TLS) protocol on an ESP32-based IoT system for ambient intelligence applications. The prototype integrates inertial and electrocardiogram (ECG) sensors and transmits data over WiFi using MQTT, with and without TLS. Current consumption is profiled using an INA219 sensor under varying payload sizes (158 to 4096 bytes) and transmission intervals (1 to 600 seconds). Results show that TLS introduces negligible overhead (below 2%) on average current consumption, even under frequent transmissions and large payloads. Transmission interval remains the dominant factor shaping energy consumption. Deep Sleep mode significantly extends battery life, yielding up to 75% reduction in average current for low-duty-cycle settings. The findings provide guidance for designing secure, energy-efficient IoT systems in constrained environments.**

***Index Terms*—ESP32, MQTT, TLS, IoT, Deep Sleep, WiFi.**

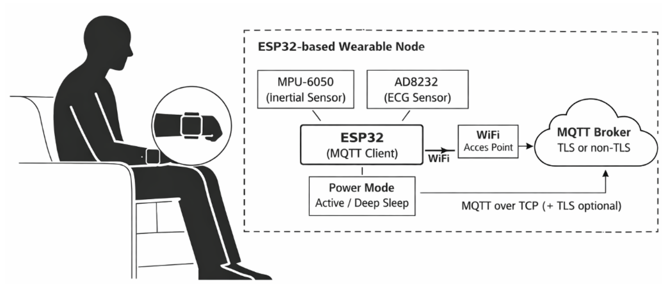


Fig. 1: Application scenario: wearable motion monitoring in a smart home with periodic data transmission to a remote broker.

I. INTRODUCTION

The proliferation of Internet of Things (IoT) technologies has enabled a wide range of applications in healthcare, ambient intelligence, and context-aware monitoring. In these domains, wearable and embedded devices support continuous sensing and remote supervision, particularly in elder care environments, where medical sensing (e.g., ECG) is used for continuous monitoring and fall prevention [1]–[5].

Among wireless technologies, WiFi remains a practical choice in ambient intelligence systems due to its ubiquity, high throughput, and integration with existing infrastructure. Although often perceived as more power-hungry than technologies like Bluetooth Low Energy (BLE), prior studies on WiFi-based microcontrollers such as the ESP8266 show that they can perform efficiently in low-duty-cycle scenarios [6]. The ESP32, its widely adopted successor with improved hardware and power management, has not been thoroughly assessed under secure communication workloads, motivating this TLS energy study in periodic IoT transmissions.

Fig. 1 illustrates a typical application scenario that motivates this work, where an elderly person is monitored through a wearable device. The device collects inertial and physiological data and transmits it via WiFi using the Message Queuing Telemetry Transport (MQTT) protocol. TLS can optionally be enabled to secure the communication. While TLS provides confidentiality and integrity, its combined overhead from computation, session handling, and increased payload size may reduce device autonomy in resource-constrained IoT nodes.

In parallel, low-power strategies such as Deep Sleep are often employed to extend battery life. However, these must be balanced against the overhead of waking up, reconnecting, and transmitting securely. While prior studies have assessed TLS or power-saving modes independently, few offer an integrated evaluation of their combined impact on real IoT workloads.

This paper presents a systematic energy evaluation of TLS-secured MQTT communication in an ESP32-based wearable sensing prototype. The system emulates a typical ambient intelligence use case, where inertial and physiological (ECG) data are periodically sent to a cloud-based broker over WiFi. Such systems are relevant for fall detection, patient monitoring, and wearable health tracking in home-based healthcare and assisted living settings. Experimental results under varying payload sizes, transmission intervals, and power modes quantify the trade-offs between communication security and energy efficiency. The study offers practical insights for the design of secure, low-power IoT medical systems.

The rest of the paper is organized as follows. Section II reviews prior work on energy and security in ESP32-based IoT systems. Section III presents the system architecture, experimental setup, and results. Section IV discusses limitations, design guidelines, battery life estimation, and future directions. Section V summarizes the main findings.

II. Related Work

Energy consumption and secure communication remain key challenges in the design of IoT systems based on embedded platforms such as the ESP32 [6], [7]. Prior studies have addressed these aspects individually, including cryptographic benchmarking [8], MQTT protocol evaluation [9], and low-power operation using sleep modes [10]. However, few studies have considered the joint impact of TLS, MQTT, and Deep Sleep in ambient intelligence scenarios.

Fischer et al. [8] profiled the energy cost of various TLS cipher suites on the ESP32, highlighting significant overhead during handshake and encryption phases. Yet, their focus was limited to cryptographic primitives and did not extend to application-layer protocols like MQTT. Baranauskas et al. [11] evaluated MQTT over TLS on ESP8266 and ESP32, noting modest energy increases, but did not consider power-saving modes or realistic sensing workloads.

Other studies implemented TLS-enabled MQTT on ESP32 focusing on connectivity. Dzahir and Chia [9] analyzed real-time behavior including latency and packet loss, while Paris et al. [12] demonstrated a secure MQTT stack for smart homes. Similarly, Li [13] presented an MQTT-based ESP32 system for healthcare, but neither work addressed energy profiling.

Deep Sleep has been evaluated independently. Malarvizhi et al. and Kumar et al. [10] demonstrated its benefits for energy savings in ESP32-based healthcare devices, but without considering TLS. Mesquita et al. [6] optimized modem sleep on ESP8266 for e-health, but excluded security mechanisms.

Several studies in the healthcare domain, such as Al-Shargabi and Abuarqoub [2], proposed ESP32-based monitoring platforms, often without integrating security and energy evaluation. Broader surveys [14], [15] offer taxonomies of energy-aware and secure IoT communication but lack empirical analysis for WiFi-capable microcontrollers.

In contrast, this work presents a combined experimental evaluation of TLS over WiFi/MQTT on ESP32, assessing energy consumption across payload sizes, transmission intervals, and power modes. Table I summarizes the reviewed studies.

III. Evaluation

A. System Architecture

The system architecture follows a common design pattern for low-power IoT nodes in ambient intelligence and healthcare. It supports applications such as fall detection and physiological monitoring in assisted-living environments.

As shown in Fig. 2, the ESP32-WROOM-32 microcontroller acquires data from two sensors: an MPU-6050 inertial sensor (GY-521 module) providing 3-axis acceleration, angular velocity, and temperature measurements, and an AD8232 ECG sensor module for cardiac signal acquisition. The AD8232 is connected to one of the ESP32's analog input pins for voltage sampling. The MPU-6050 communicates digitally via I2C, enabling periodic acquisition of inertial measurements. The ESP32 handles data acquisition, local formatting, and MQTT transmissions over WiFi.

Sensor readings are encoded as JSON and sent to a remote broker, using MQTT over either plain TCP or a TLS-secured connection. The firmware supports configurable payload sizes and transmission intervals, and can operate in either Active or Deep Sleep mode to explore power-performance trade-offs.

Current draw is monitored via an INA219 sensor placed on the 3.3 V line powering the ESP32. The INA219 is connected to an Arduino Uno, which relays measurements to a host PC for logging and analysis.

B. Experimental Setup

Evaluation was performed using the system architecture depicted in Fig. 2, with firmware developed in the Arduino IDE. MQTT version 3.1.1 was used throughout, implemented via the `PubSubClient` library. TLS was implemented using the `WiFiClientSecure.h` library in Arduino, which uses mbedTLS and thus supports TLS 1.2, the most widely supported version in embedded systems. Two power configurations were tested: *Active* mode, where the ESP32 remains continuously powered, and *Deep Sleep* mode with timer-based wake-up and full reconnection on each cycle. In Deep Sleep mode, each wake-up triggers a new TCP/TLS connection, and no TLS session state is preserved across cycles.

The baseline payload size of 158 B corresponds to a JSON-formatted message containing inertial and ECG data. Larger payloads (500–4096 B) were generated by appending plaintext padding to this base message to assess the impact of message size on energy consumption. Although the padding does not reflect real sensor data, larger messages may occur in practical scenarios where multiple samples are aggregated before transmission. To support large messages, the MQTT client buffer size was extended using the `setBufferSize` PubSubClient library function.

Current was measured using an INA219 sensor with a sampling rate of 5 Hz, and data were logged on a host PC. This sampling rate results in a variable number of measurements per TX interval, depending on its duration. Average current values were computed using all samples collected during each experimental run. MQTT messages were sent to two public brokers: the Eclipse Mosquitto test server[1] (unencrypted) and a TLS-enabled broker on the EMQX cloud[2], accessed via port 8883 using client authentication and CA certificate validation. The INA219 sensor was calibrated using a multimeter during initial tests to ensure better accuracy.

All experiments were conducted in a stable indoor environment with a 5 V USB power supply, line-of-sight WiFi connectivity, and ambient temperature of 22–24,°C. Each configuration was tested for at least 10 minutes, capturing a minimum of three transmission cycles (30 mins for the 600s case). Configurations with shorter intervals included a larger number of idle–transmit cycles. Results were averaged to mitigate transient effects and improve consistency.

The tested configurations are summarized in Table II.

[1] https://test.mosquitto.org

[2] https://www.emqx.com/en/cloud

TABLE I: Comparison of related work on energy profiling, secure communication, and low-power strategies in IoT systems.

Study	Study Type / Platform	Protocol	TLS	Sleep	Use Case or Focus
Mahmoud et al. [14]	Survey	Various	✗	✗	IoT security challenges in constrained devices
Mosenia and Jha [15]	Survey	Various	✗	✗	Energy-aware IoT security; no platform-specific tests
Mesquita et al. [6]	ESP8266	WiFi	✗	✓	e-Health with modem sleep tuning; no TLS
Fischer et al. [8]	ESP32	TLS only	✓	✓	Cipher-level TLS profiling; no MQTT or sensing
Baranauskas et al. [11]	ESP32/8266	WiFi/MQTT	✓	✗	TLS-MQTT profiling; no sleep mode or sensing
Dzahir and Chia [9]	ESP32/8266	WiFi/MQTT	✓	✗	TLS-MQTT in real-time apps; latency only
Paris et al. [12]	ESP32	WiFi/MQTT	✓	✗	TLS feasibility in smart homes; no energy profiling
Li [13]	ESP32	WiFi/MQTT	✗	✗	Health monitoring system; no TLS or profiling
Kumar et al. [10]	ESP32	WiFi	✗	✓	Deep Sleep profiling in healthcare; no MQTT or TLS
This work (2025)	ESP32	WiFi/MQTT	✓	✓	TLS-MQTT energy profiling on ESP32 with Deep Sleep

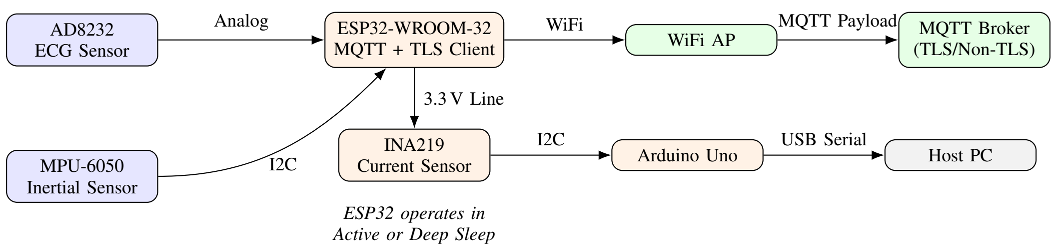


Fig. 2: System architecture. The ESP32 collects inertial and ECG data from an MPU-6050 and an AD8232 module, and transmits them via MQTT over WiFi with optional TLS. Current consumption is measured with an INA219 on the 3.3 V line.

TABLE II: Summary of Experimental Setup.

Component	Configuration / Value
Measurement Duration	$\geq$ 10 minutes per configuration
Number of Runs	$\geq$ 3 idle–TX cycles (averaged)
Microcontroller	ESP32-WROOM-32
WiFi Standard	IEEE 802.11 b/g/n
Inertial Sensor	MPU-6050 (GY-521 module)
ECG Sensor	AD8232 module
Protocols	MQTT over TCP and TLS 1.2
MQTT Version	3.1.1 (PubSubClient)
TLS Library	`WiFiClientSecure.h` (Arduino)
Power Modes	Active, Deep Sleep (timer wake-up)
Power Supply	5 V USB (measured at 3.3 V line)
Payload Sizes	158, 500, 1024, 2048, 4096 bytes
Transmission Intervals	1, 6, 60, 600 seconds
Current Sensor	INA219 (5 Hz sampling rate)
MQTT Brokers	EMQX Cloud (TLS) / Mosquitto (non-TLS)

C. Results

This section evaluates current consumption under varying transmission settings, focusing on the impact of TLS, packet size, TX interval, and transceiver behavior. Results are presented using statistical boxplots and empirical histograms.

Figs. 3 and 4 correspond to the active mode configuration, where the ESP32 remains powered between transmissions. Deep Sleep results are discussed in Section III-C4 and illustrated in Fig. 5.

*1) **Impact of TLS**:* TLS introduces negligible differences in median and mean current consumption across all transmission intervals and payload sizes, as shown in Fig. 3. Median current remains consistently around 71–72mA regardless of TLS activation. Although slight differences in mean current were observed, no statistical test was performed.

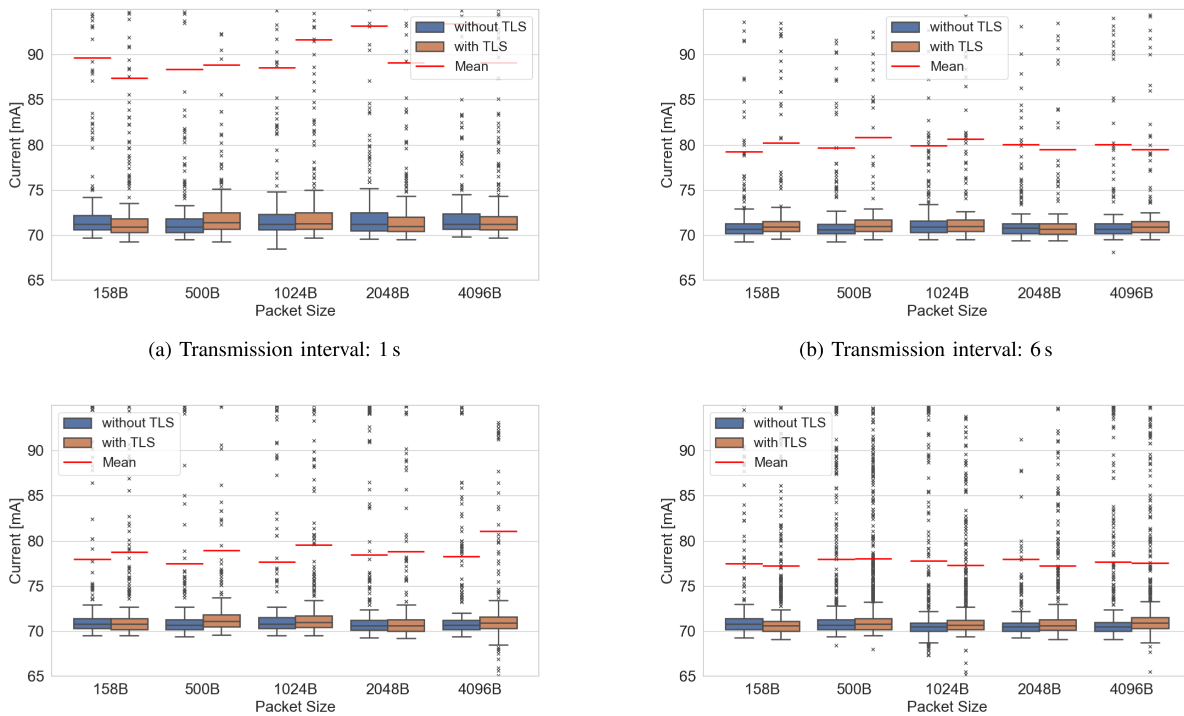
Histograms in Fig.4 highlight occasional instantaneous peaks (160–170mA) at short intervals (1s), visible across payloads and TLS configurations. These peaks occur during transmission bursts and reflect transient transceiver activity. At longer intervals (600s), such effects largely vanish, and current distributions become tightly centered and nearly identical across TLS settings.

*2) **Impact of Packet Size**:* Packet size has limited impact on current consumption. As shown in Fig. 4, at short intervals (1 s), larger payloads (4096 B) slightly increase the occurrence of higher instantaneous currents (above 160 mA), raising mean current from approximately 89 mA (158 B) to 93 mA (4096 B) without TLS. However, median current remains stable.

At longer intervals (600s), histogram differences across payload sizes in Fig.4 become negligible. Since only three transmissions occurred during the 30-minute test, the limited number of high-current events reduces the impact of packet size on the overall current profile.

*3) **Impact of TX/RX Modes**:* Histograms in Fig. 4 illustrate how TX interval and payload size influence current draw. At short intervals with large payloads (e.g., 1 s, 4096 B in 4b), distributions exhibit a secondary cluster around 160 to 165 mA, reflecting extended transceiver activity. A similar but less marked pattern is seen with smaller payloads (Fig. 4a). At long intervals (e.g., 600s in Fig. 4d and Fig. 4c), distributions are tightly centered around 70 to 72 mA, indicating that idle time dominates energy consumption. These results confirm that transmission timing and duration are the primary drivers of current variability.

*4) **Impact of Deep Sleep Mode**:* To assess energy savings under low-duty-cycle conditions, Deep Sleep mode was en-



(a) Transmission interval: 1 s

(b) Transmission interval: 6 s

(c) Transmission interval: 60 s

(d) Transmission interval: 600 s

Fig. 3: Boxplots of current consumption for different payload sizes and TX intervals, with and without TLS. Boxes show the interquartile range (Q1–Q3), lines the median, and whiskers extend to 1.5×IQR. Outliers above 95 mA are omitted for clarity. Variations reflect both per-packet energy and TX frequency; shorter intervals include more high-current events, raising average.

abled with timer-based wake-up, requiring full reinitialization of WiFi and MQTT after each cycle. Fig. 5 presents the average current across payload sizes and TX intervals, comparing active mode (TLS only) against TLS with Deep Sleep.

Results confirm that Deep Sleep offers substantial benefits at long transmission intervals. At 600s, average current dropped from 76–78mA (active) to 18–19mA across all payloads, yielding savings above 75%. At 60s, current decreased from 78mA to 22mA (about 72%). These reductions were consistent across packet sizes, indicating that idle duration, rather than payload volume, primarily drives energy savings.

At short intervals (e.g., 1s), Deep Sleep can become counterproductive. The cost of frequent wake-up and reconnection may offset the benefits of low-power sleep. For example, with 2048B, average current rose from 89.1mA (TLS only) to 96.9mA with Deep Sleep. However, at 1024 B, Deep Sleep slightly reduced the average current.

IV. Discussion

This section reflects on the main limitations and design insights from the experimental results, offering guidance for practical deployments and identifying future directions.

A. Limitations

The experiments were conducted in a controlled indoor environment with stable WiFi connectivity and no mobility. While this ensured reproducibility and reduced noise in energy measurements, real-world deployments may experience variability due to interference, signal fading, and environmental dynamics. The analysis focused on TLS and Deep Sleep configurations using a single node and broker setup; multi-node contention or gateway bottlenecks were not evaluated.

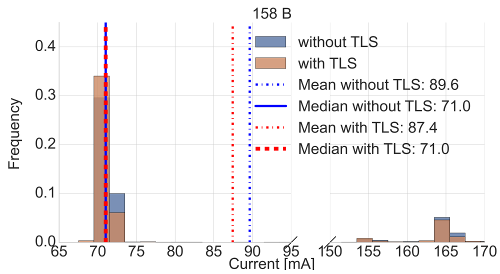
Intermediate power modes (e.g., light sleep) and alternative protocols (e.g., CoAP, HTTP) were also excluded, which may influence the trade-offs in different application scenarios.

B. Design Insights and Guidelines

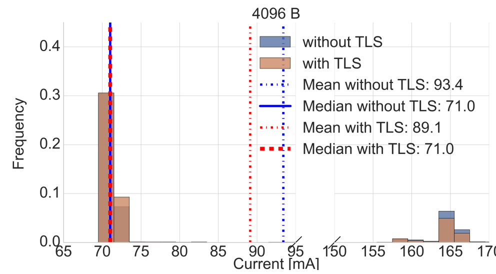
The results support several practical recommendations:

- TLS introduces a modest overhead primarily during secure session setup, resulting in brief current peaks. These have negligible impact on average consumption.
- Deep Sleep was only beneficial when the transmission period exceeds 6s; otherwise, the cost of repeated wake-ups and reconnection outweighs savings.
- Larger packets slightly raise TX time and current peaks, but average consumption remains stable, with median current consistently around 71–72mA across all intervals.
- These insights are particularly relevant for wearable medical devices in smart home or outpatient care settings, where trade-offs between security, energy efficiency, and response time are critical.

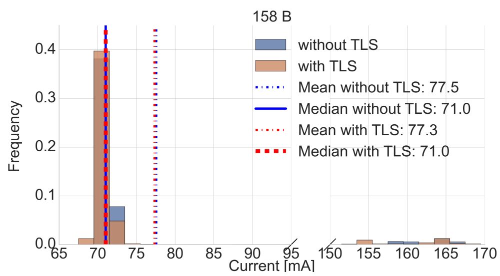
These findings can guide designers in selecting protocol configurations, payload sizes, and sleep strategies for secure and efficient IoT nodes with focus in ambient intelligence.



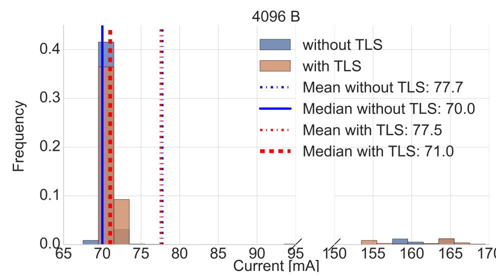
(a) TX interval 1 s and packet size of 158 B



(b) TX interval 1 s and packet size of 4096 B



(c) TX interval 600 s and packet size of 158 B



(d) TX interval 600 s and packet size of 4096 B

Fig. 4: Current consumption histograms for payloads of 158 B and 4096 B at transmission intervals of 1 s and 600 s, with and without TLS. Means and medians are indicated.

TABLE III: Estimated battery life (1000 mAh).

Interval [s]	Payload [B]	Mode	Avg. Current [mA]	Life [h]
600	158	Active (TLS)	76.46	13.1
		Deep Sleep	18.83	53.1
600	4096	Active (TLS)	78.28	12.8
		Deep Sleep	19.38	51.6
60	158	Active (TLS)	78.85	12.7
		Deep Sleep	22.05	45.3
1	2048	Active (TLS)	87.40	11.4
		Deep Sleep	97.40	10.3

C. Battery Life Estimation

Table III estimates the device operating time assuming a 1000 mAh battery and a constant current draw per configuration. Battery life was obtained using the simple relation Life $= 1000\,\text{mAh}/\text{Avg. Current (mA)}$. The selected configurations represent a diverse yet concise set of TX intervals and payload sizes, chosen to illustrate typical usage scenarios and energy trade-offs at both extremes of data frequency and size.

The data confirms that Deep Sleep significantly extends autonomy at long transmission intervals. For example, at 600 s, average current dropped from 76–78mA (active) to 18–19mA (Deep Sleep), resulting in a 4x increase in battery life (from $\sim$13 to $\sim$52 hours). Similarly, at 60s, operating time improves from 12.7 to 45.3 hours. However, for short intervals (e.g., 1s), the overhead of frequent reconnection outweighs the benefits, with Deep Sleep yielding even lower autonomy than active mode. These results highlight the importance of duty-cycle-aware power management in low-power IoT design.

D. Challenges and Future Opportunities

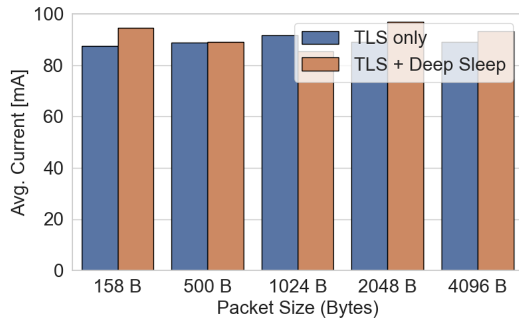
Future work will address scalability in multi-node deployments sharing a single access point, evaluate performance under mobility and dynamic signal conditions, and explore lightweight session resumption mechanisms in TLS to reduce connection overhead. Additional study of access point parameters (e.g., beacon interval, DTIM settings) and sleep transition costs may further enhance secure and low-power design. It would also be relevant to investigate how the observed trends may differ with alternative security protocols such as DTLS, and with other communication schemes like CoAP or HTTP, which have distinct handshake processes, payload structures, and transport characteristics.

V. Conclusion

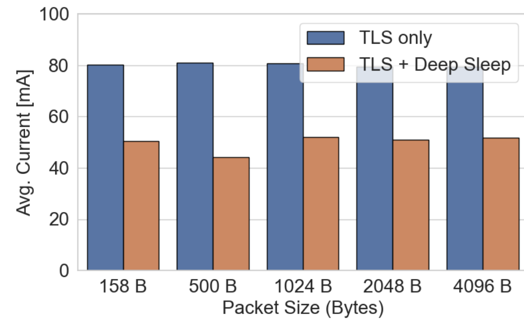
This paper presented a systematic evaluation of current consumption in an ESP32-based IoT system for ambient intelligence, analyzing the impact of secure MQTT communication over WiFi using TLS. Experiments covered different payload sizes, TX intervals, and Active and Deep Sleep power modes under controlled conditions.

Results show that TLS introduces a negligible to modest overhead (typically below 2%), more apparent with large payloads and short intervals. Packet size and TX frequency shape the energy profile, though their impact decreases at longer duty cycles. Deep Sleep mode was highly effective at intervals of 6s and above, reducing average current by up to 75%, though it incurred added cost under frequent wake-ups.

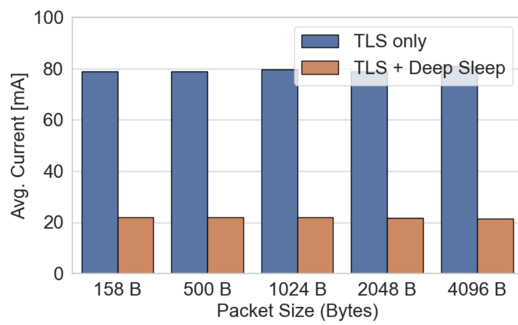
These findings offer quantitative insight into trade-offs in secure, low-power IoT communication and provide a baseline for protocol selection and duty-cycle optimization.



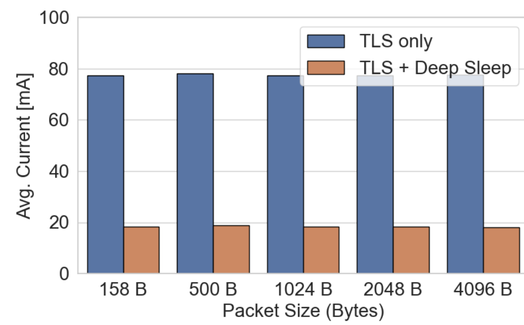
(a) Transmission interval: 1 s



(b) Transmission interval: 6 s



(c) Transmission interval: 60 s



(d) Transmission interval: 600 s

Fig. 5: Average current consumption with and without Deep Sleep across payload sizes and transmission intervals. Energy savings are most pronounced at long intervals due to extended idle periods in low-power mode.

Future work includes evaluating multi-node scalability, mobility and fading effects, and TLS session resumption to reduce overhead while maintaining reliability.

Acknowledgments

The authors thank Benjamín Yaupe for his experimental work and the preliminary version of the plots. This work was funded by ANID Fondecyt 11241221, ANID Fondequip EQM240128 and ANID CPS-RTC CIA250016; and partially by the CISTER Research Unit (UIDP/UIDB/04234/2020) through FCT/MCTES (Portuguese Foundation for Science and Technology) National Funds.

References

[1] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[2] B. Al-Shargabi and S. Abuarqoub, "IoT-enabled healthcare: Benefits, issues and challenges," in *Proc. of the 4th International Conference on Future Networks and Distributed Systems (ICFNDS '20)*, 2020, pp. 1–5.

[3] W. B. Qaim, A. Ometov, A. Molinaro, I. Lener, C. Campolo, E. S. Lohan, and J. Nurmi, "Towards energy efficiency in the Internet of wearable things: A systematic review," *IEEE Access*, vol. 8, pp. 175 412–175 435, 2020.

[4] G. Araniti, C. Garibotto, A. Jose, F. Marcello, V. Pilloni, A. Sciarrone, C. Suraci, P. Zema, and M. Zerbino, "Medical digital twin and IoT devices: Monitoring patients through human-centered technologies," *IEEE Internet of Things Magazine*, 2025.

[5] L. A. Durán-Vega, P. C. Santana-Mancilla, R. Buenrostro-Mariscal, J. Contreras-Castillo, L. E. Anido-Rifón, M. A. García-Ruiz, O. A. Montesinos-López, and F. Estrada-González, "An IoT system for remote health monitoring in elderly adults through a wearable device and mobile application," *Geriatrics*, vol. 4, no. 2, p. 34, 2019.

[6] J. Mesquita, D. Guimarães, C. Pereira, F. Santos, and L. Almeida, "Assessing the ESP8266 WiFi module for the Internet of Things," in *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, vol. 1. IEEE, 2018, pp. 784–791.

[7] A. O. Akmandor, Y. Hongxu, and N. K. Jha, "Smart, secure, yet energy-efficient, Internet-of-Things sensors," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 914–930, 2018.

[8] T. Fischer, H. Linka, M. Rademacher, K. Jonas, and D. Loebenberger, "Analyzing power consumption of TLS ciphers on an ESP32," in *Crypto Day Matters 30*. Gesellschaft für Informatik eV/FG KRYPTO, 2019, pp. 10–18.

[9] M. A. S. M. Dzahir and K. S. Chia, "Evaluating the energy consumption of ESP32 microcontroller for real-time MQTT IoT-based monitoring system," in *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*. IEEE, 2023, pp. 255–261.

[10] M. Kumar, C. Choudhary, and M. ElBarachi, "Sustainability of energy saving in healthcare IoT devices," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 6494–6501.

[11] E. Baranauskas, J. Toldinas, and B. Lozinskis, "Evaluation of the impact on energy consumption of MQTT protocol over TLS," in *Proc. IVUS 2019 Int. Conf. on Information Technologies, CEUR Workshop Proc., vol. 2470, 2019, pp. 56–60.*, vol. 2470, 2019, pp. 56–60.

[12] I. L. B. M. Paris, M. H. Habaebi, and A. M. Zyoud, "Implementation of SSL/TLS security with MQTT protocol in IoT environment," *Wireless Personal Communications*, vol. 132, no. 1, pp. 163–182, 2023.

[13] S. Li, "IoT healthcare system based on ESP32 for smart home," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2023, pp. 1768–1772.

[14] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *2015 10th international conference for internet technology and secured transactions (ICITST)*. IEEE, 2015, pp. 336–341.

[15] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2016.