



HAL
open science

Inter-Technology Application Data Exchange Based on Advertisement Packets

Mariam Issa, Paul Couderc, Jean-Marie Bonnin

► **To cite this version:**

Mariam Issa, Paul Couderc, Jean-Marie Bonnin. Inter-Technology Application Data Exchange Based on Advertisement Packets. EAI-INTSYS 2025 - 9th EAI International Conference on Intelligent Transport Systems, eai, Dec 2025, Marietta, United States. <hal-05594058>

HAL Id: hal-05594058

<https://hal.science/hal-05594058v1>

Submitted on 16 Apr 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

Inter-Technology Application Data Exchange Based on Advertisement Packets

Mariam Issa¹, Paul Couderc², and Jean-Marie Bonnin³

¹ Université de Rennes, Rennes, France

² Centre INRIA de l'Université de Rennes, Rennes, France

{`mariam.issa,paul.couderc`}@inria.fr

³ IMT Atlantique / IRISA, Rennes, France

`jean-marie.bonnin@irisa.fr`

Abstract. Ephemeral, spontaneous interactions like proximity-based alerts and short-lived data exchanges, are becoming central to mobile and IoT applications. Traditional wireless systems, designed for persistent connections, are often too heavy for these use cases. We present in this paper a unified framework that enables opportunistic, connectionless data exchange across BLE and WiFi advertisements. Our system, deployed on ESP32 hardware, serves as a proof-of-concept and as a performance study for a middleware that abstracts low-level protocol details. It requires no firmware modification, and supports local and context-aware communication. We provide an in-depth experimental study highlighting the power consumption, coexistence constraints, and design trade-offs of concurrent BLE and WiFi beaconing. Our results demonstrate the framework's potential for enabling seamless, standard-compliant inter-technology communication tailored to ephemeral applications.

Keywords: Ephemeral Interaction · Local Communication · Dataspaces · Association-less Communication · WiFi Beacon Stuffing · BLE Advertisements

1 Introduction

Recently, digital interactions have become increasingly ephemeral by nature. Examples could be pop-up data sharing between smartphones in proximity, dynamic alerts from wearables or sensors in a smart city such as Vehicle-to-Pedestrian (V2P) warnings, or temporary collaboration among devices in crowded events. Many of these emerging use cases are characterized by short-lived, spontaneous, and context-dependent exchanges of information. The ephemeral nature [1] of these interactions requires systems that can exploit brief encounters and minimal user involvement. We can imagine such systems as being analogous to a whiteboard or a graffiti board. Anyone could freely read or write. Traditional wireless systems, designed for long-lived, stable sessions, often impose unnecessary overheads and complex setups that are ill-suited for these scenarios. Therefore there is a need for communication paradigms that are tailored to the requirements and constraints of the short-lived nature of the interactions.

In this direction, association-less or connection-less communication paradigms [2] [3] [4] were investigated. They are based on the idea of embedding messages within discovery packets. Such packets are broadcasted whenever a node wants to announce itself. At the receiver side, scanning is done to extract the messages. Such paradigms proved to be efficient and served the purpose of context-related ephemeral interactions. Two popularly existing technologies that implement this concept are based on WIFI and BLE, in particular WiFi Beacon Stuffing [2], and BLE connectionless mode [5]. They have been applied in different application domains due to their abundance and ease of use. Through communicating over advertisements, they showed a high potential in different application domains including Vehicle to Pedestrian (V2P) safety systems [6] [7], sensor networks [8] and proximity based messaging [9]. Bluetooth Low Energy (BLE) and WiFi have become the dominant wireless standards in today’s smart devices, powering a wide range of applications from mobile sensing to ubiquitous computing and IoT deployments. While each technology offers unique advantages such as BLE’s low-power and latency, and WiFi’s higher throughput and coverage; the challenge of enabling direct, connectionless application data exchange between heterogeneous BLE and WiFi devices persists [10]. From a networking perspective, bridging the gap between these protocols has required centralized gateways, protocol conversion layers, or connection-oriented handshakes, often adding latency, complexity, and extra energy cost to the system. Moreover, the adoption of such approaches at an application level has been limited by the lack of simple, high-level software models and APIs that abstract away the low-level details. Existing works often propose systems that are complex to replicate or to use. Many reported complexities come from facing constraints on system parameters due to device specificity, vendor limitations, and developmental complexities.

Given the increasing need for inter-technology communication in diverse, real-world deployments, and motivated by the growing integration of both BLE and WiFi in off-the-shelf devices, we extend our prior work in [4] to develop a unified, standard-compliant, broadcast framework that seamlessly spans both protocols. A unified framework that bridges BLE and WiFi broadcast offers several practical advantages: it enhances reliability through redundancy and enables wider coverage and adaptability. However, practical implementation reveals important trade-offs and technical hurdles.

The proposed framework provides application-level data filtering and secure payload integrity using a lightweight hashing mechanism, supporting ephemeral, local, and battery-efficient scenarios, where classic connection-based approaches are unsuitable or too costly. Furthermore, our work delivers a detailed experimental characterization of radio coexistence, power consumption, and interoperability challenges when concurrently using BLE and WiFi in beacon mode.

In summary, our contributions can be summarized as follows:

- We design and implement an opportunistic broadcast framework enabling seamless, direct data exchange between BLE and WiFi devices via advertisement packets, with no radio firmware modification required.

- We deploy the system on commercial ESP32 hardware and conduct an in-depth empirical study, characterizing energy consumption, payload constraints, and the interplay between BLE and WiFi advertisement scheduling.

The rest of this paper is organized as follows: Section 2 provides a background and some of the related works. Section 3 details the proposed framework. Then the experimental procedure is explained in section 4. Finally, the paper is concluded in section 5.

2 Related Work

Understanding the basis of connectionless communication, or as we refer to it in our works as "Association-less Communication" is crucial for developing applications tailored to ephemeral interactions. In this section, we first present a brief background on BLE connectionless mode and on WiFi Beacon Stuffing. We then review some works that compare WiFi against BLE advertisements. Having such comparisons is important to identify the strengths and weaknesses of one technology over the other. This is particularly necessary when deciding on which technology to use for a certain application use-case. Finally, we provide some papers that have implemented frameworks that bridge the gap between the two technologies, allowing inter-technology communication. We summarize such frameworks, and discuss their limitations in terms of standard compliance, hardware requirements, and real-world applicability.

2.1 Association-less Communication

A technology that allows devices to communicate without prior configuration, to act in a decentralized manner, to discover and broadcast data to devices in range, while not being affected by node mobility, would be a suitable enabler for association-less communication. WiFi and BLE are two strong candidate technologies to implement association-less communication. This is motivated by the facts that both of the technologies allow to embed data within packets that are already exchanged among communicating peers. The data to be embedded is small, and usually acts as some state description, some identifier, or even could correspond to some sensor measurements. The other fact is the reliance on existing technologies without the need to deploy and install new equipment or apply major changes to the current infrastructure. For the case of WiFi Beacon Stuffing, the WiFi is accessible in a highly ubiquitous and dense manner. For BLE beacons, they come at a low cost and are easy to deploy and move. In what follows we provide a brief background on the operation of the two technologies.

WiFi Beacon Stuffing In WiFi infrastructure mode, when an AP wants to announce its existence, it sends periodically beacon frames. Client devices perform active or passive scans to detect these beacons. In default mode of operation, advertisement beacons are broadcasted by APs every 100ms [11]. WiFi Beacon

Stuffing was first introduced in [2]. This was motivated by the fact that these beacons are exchanged among APs and clients despite of their connection state. Despite the fact that beacon stuffing has been proposed more than 15 years ago (2007), it continues to be a strong candidate for utilization in recent applications. Different implementations of this technique include embedding of the messages within the SSID (Service Set Identifier), BSSID (Basic Service Set Identifier) or the VSIE (Vendor-Specific Information Element) fields of beacon frames, a.k.a management frames.

BLE Connectionless mode BLE has gained popularity for its energy efficient and short range capabilities. It stands as a promising candidate wireless technology to realize association-less communication. The terms used in literature to describe such mode of communication are "Connectionless communications" or "Legacy advertising". BLE was designed with the attempt to support ephemeral interactions, due to the introduction of two modes: connectionless and connection-oriented modes. In the connection-less, the broadcaster (advertiser) periodically sends advertising packets to any device able to receive them. On the other side, the observer (scanner) continuously scans for these advertisements at periodic intervals [12]. Advertisement packets can take four forms: *Connectable undirected Advertising*, *Connectable Directed Advertising*, *Non-connectable Undirected Advertising* or *Scannable Undirected Advertising* [13].

Our previous work in [3] presents a comprehensive experimental comparison between WiFi Beacon Stuffing and BLE Connectionless mode as mechanisms for association-less (connectionless) communication in smart environments. The paper focuses on evaluating both technologies on the ESP32 platform in terms of power consumption, delivery rate, and range, with the goal of guiding technology selection for ephemeral and local interactions.

To illustrate the trade-offs, consider two typical scenarios: (1) An event-driven alarm system in a smart building, where battery-powered sensors must signal emergencies rapidly and efficiently to nearby nodes; here, BLE's low power consumption and rapid advertising intervals are essential. (2) A video surveillance network, where WiFi Access Points and always-powered cameras periodically announce status and metadata; here, WiFi's greater payload and range, combined with data persistence on always-on devices, are more valuable.

Our results show that while WiFi Beacon Stuffing theoretically offers higher payload capacity and longer transmission range, it suffers from higher power consumption compared to BLE. BLE connectionless mode, on the other hand, demonstrates lower power consumption and easier implementation thanks to standardized APIs, making it more suitable for energy-constrained and time-critical applications, despite its smaller payload and shorter range.

In [14], the authors conducted a comparative study of Wi-Fi and BLE advertisements for their application in vehicle-pedestrian collision warning systems. The authors evaluated these technologies in various real-world scenarios, focusing on their reliability and responsiveness for proximity detection. Their findings showed that BLE outperformed Wi-Fi in terms of signal stability, responsiveness, and robustness under different environmental conditions. As a result, the

study concludes that BLE is the most suitable technology for vehicle–pedestrian safety applications due to its consistent performance and operational resilience.

In [15] the authors establish an experimental comparison between BLE and WiFi. They compared the power consumed when advertising BLE beacons, to that by WiFi when advertising as an access point and when it is listening in scanning modes. They demonstrate that BLE advertisements indeed consume significantly less power than WiFi beacons. However, they do not establish a comprehensive study that takes into consideration the different parameters such as advertising and scanning intervals.

Although that the mentioned works compared WiFi to BLE advertisements while providing important design insights, however they did so for each technology independently. To our current knowledge, there is no work that attempts to evaluate the performance of having both technologies operating concurrently on the same hardware for the same application.

We now provide some of the works that have proposed middlewares for seamless communication among the technologies.

The authors in Hyrax [16] propose a inter-technology platform that enables mobile devices to form self-organizing mesh networks, automatically discovering and connecting over WiFi-Direct, Bluetooth, or similar radios to relay data without central infrastructure.

The design and implementation of the Multiuser Interaction Middleware (MIM) presented in [17], enables direct interaction among mobile devices in close proximity using D2D communication technologies, such as WiFi Direct and Bluetooth.

WARBLE [18] introduces a middleware that abstracts away the complexity of discovering, selecting, and interacting with heterogeneous IoT devices by providing developers with high-level programming constructs called selectors and bindings. Developers use selectors to specify the functional and contextual requirements for devices (such as device type, user proximity, or location), and WARBLE automatically discovers and manages connections to matching devices, regardless of their underlying protocol (e.g., BLE, mDNS).

Finally, the Venice framework in [10] introduces a device-to-device (D2D) abstraction that leverages established communication channels such as Wi-Fi and Bluetooth to enable seamless communication between heterogeneous devices, including Android smartphones and computers. The authors address the lack of a unified data link layer for direct mobile interactions, proposing Venice as a middleware that abstracts the heterogeneity of underlying physical channels and platforms. Venice’s main contribution is its channel abstraction, which enables developers to implement D2D communication without concerning themselves with the specifics of each hardware stack. The framework supports multiple communication technologies and is implemented using Flutter, allowing cross-platform compatibility.

These works highlight the need for a middleware-based approach. A middleware can act as a mediator between applications and underlying network technologies, offering several advantages. First, it enables arbitration: the mid-

middleware can observe requests from multiple applications, group events where appropriate, and schedule advertisements to minimize channel contention and optimize energy use—capabilities not possible with direct, application-level use of network APIs. Moreover, a middleware can adapt to performance goals such as low latency (for urgent events), data persistence (for reliable status propagation), and spatial coverage, depending on the interaction pattern and node capabilities.

As advertisement-based communication can lead to channel saturation, especially in dense environments, middleware-driven arbitration and scheduling are crucial to ensuring robust, efficient, and scalable ephemeral communication. For example, a middleware might aggregate several sensor alerts into a single BLE advertisement, or prioritize urgent events for WiFi beacon stuffing, balancing latency, coverage, and power.

While existing middlewares such as Hyrax, MIM, WARBLE, and Venice have demonstrated the feasibility of inter-technology communication by abstracting lower-layer details and providing unified interfaces for heterogeneous radios, they primarily focus on high-level device discovery, dynamic mesh formation, or generalized data exchange across platforms, using established connection-oriented protocols (e.g., WiFi-Direct, Bluetooth Classic, mDNS). In contrast, our work targets seamless inter-technology data exchange in the connectionless (association-free) domain, specifically leveraging BLE advertisements and WiFi Beacon Stuffing on resource-constrained hardware such as the ESP32. This work is a preliminary step for developing a middleware that achieves application level seamless communication between WiFi and BLE devices, using advertising. Nevertheless, a performance study is necessary to decide on the tradeoffs that will result from having such a system.

Our contributions bridge the gap between theoretical middleware abstractions and practical, energy-aware deployments on real IoT platforms, enabling reliable and interoperable inter-technology broadcast without requiring connections or platform-specific drivers. Thus, the motivation and novelty of our work. In the next section, we provide and discuss the proposed framework.

3 inter-Technology Framework for Ephemeral Dataspaces over BLE and WiFi

In [4], we proposed a BLE-only framework for connectionless application data exchange, supporting hash-based filtering and lightweight encoding, and demonstrated that for the same application, it offered energy benefits over GATT-based approaches. This work has emphasized the necessity of elevating data exchange to the application layer, decoupling it from the constraints of connection-oriented or packet-level operations. By implementing a shared ephemeral dataspace, the proposed framework allows applications to both publish and retrieve relevant information in a manner that is directly addressable at the application level. This enables dynamic filtering based on the actual data content—such as application codes or service types—instead of relying on lower-layer identifiers.

Another critical aspect of the framework is its independence from MAC address-based identification and filtering. As device MAC addresses are increasingly randomized for privacy and security reasons, any protocol dependent on MAC-based addressing becomes unreliable for robust and privacy-preserving communication. Our protocol instead leverages content-based identification, using embedded application codes and cryptographic hashes to unambiguously classify and filter packets. This approach ensures that only the data pertinent to a given application or use case is processed, while completely abstracting away device identities at the network level. As a result, the system supports fully anonymous, association-less, and context-driven interactions, and enables a more flexible and scalable means of coordinating transient data exchanges among heterogeneous devices. In this we extend our lightweight BLE-based framework proposed, to support seamless communication over both BLE advertisements and WiFi Beacon Stuffing. By leveraging both radios on the ESP32 platform, our protocol enables nodes to broadcast and retrieve application data without requiring connections or MAC-based addressing, while providing redundancy and broader device compatibility. We Study the performance of such extension in terms of the power profile of the new combined system.

Framework Overview: For the sake of space, we provide the extended framework in a summarized form. We adopted a publish-subscribe (pub/sub) paradigm for our middleware because it provides strong decoupling between data producers and consumers. In pub/sub, applications publish events or data without knowledge of which nodes (if any) will receive them, and consumers subscribe to the data types or events they are interested in—this is known as process decoupling. By contrast, low-level approaches based directly on network message APIs (such as sockets or direct BLE/WiFi advertising APIs) require explicit knowledge of endpoints and tight coupling between sender and receiver. They lack these decoupling properties and typically do not scale well for complex, event-driven, or dynamic IoT scenarios.

Adopting a data-mediation approach abstracts away the details of network communication, enabling modular, flexible, and robust applications. This is especially important in inter-technology systems where data should mediate interactions, not device addresses or static connections.

Nodes act as providers (publishers) or requesters (scanners), broadcasting or listening for context-relevant data simultaneously over both BLE advertisements and WiFi beacons. Nodes operating in shared physical and temporal proximity can receive application data over either radio. In this proof of concept study, both radios are always on.

Unified Packet Structure: For both BLE and WiFi, we use a common, technology-agnostic packet format (Figure 1), enabling cross-decoding and unified verification. Packets are structured as follows:

- **Role (2 bits):** Indicates node function (e.g., provider/requester).
- **Application Code (2 bits):** Classifies the application (e.g., warning, sensor, etc.).
- **Generic Field (18 bits):** Reserved for extension and future use.

- **Payload (up to 11 bytes):** Encodes application data.
- **Hash Code (2 bytes):** MurmurHash2 output for membership verification.

Note that the total packet size in this extension becomes 16 bytes instead of 31 bytes. This will be discussed in details in the experimental results section.

WiFi Beacon Encoding: Using the Vendor Specific Information Element (VSIE) in WiFi management frames theoretically allows arbitrary data to be embedded in beacon or probe response frames, providing a higher payload capacity and avoiding some SSID limitations. However, in practice, most off-the-shelf mobile devices, laptops, and embedded WiFi stacks do not expose VSIE content to user applications via standard APIs. Scanning applications on Android, iOS, Windows, and even many Linux distributions can typically only read basic AP fields like SSID, BSSID, RSSI, and security settings—while VSIE data remains inaccessible or filtered for security and privacy reasons. This lack of universal support and programmability makes VSIE unsuitable for widely-deployable inter-technology broadcast solutions, especially when the target devices and apps must operate without root or special drivers. As a result, encoding data in the SSID, despite its constraints, remains the most interoperable option for such scenarios.

Directly embedding raw bytes into the WiFi SSID for data broadcasting proved unreliable, as most operating systems, drivers, and device APIs treat the SSID as a printable string and truncate or mangle any non-UTF-8 or non-printable characters—especially null bytes—resulting in corrupted or incomplete payloads at the receiver. To ensure portability and reliability across all platforms, we encode the payload as a printable hex string before embedding it in the SSID. This approach universally preserves the data but reduces the maximum effective payload by half, since each byte now requires two characters. Despite this limitation, hex encoding is the most robust and widely adopted solution for SSID-based data channels in practical inter-technology scenarios.

Verification Mechanism: Upon receiving a packet (from BLE advertisement or extracted from WiFi SSID), the node decodes the payload and recomputes the hash over all fields except the appended hash. A match validates membership in the ephemeral dataspace and ensures the integrity of the transmitted message (see Table 1). This process is technology-agnostic and works identically for both radios. Therefore, the scanner application will see the same data twice if it has both wifi and ble enabled.

1 byte			1 byte	11 bytes	1 byte	2 bytes
2 bits	2 bits	4 bits	Generic Field	Payload	Generic Field	Hash
Role	Application Code	Generic Field				

Fig. 1. Unified advertisement packet structure for BLE and WiFi.

Table 1. Unified Packet Field Definitions (BLE/WiFi)

Field	Description	Size
Role	Node role (provider/requester)	2 bits
App Code	Application class	2 bits
Generic Field	Reserved/Extension	18 bits
Payload	Application data	up to 11 bytes
Hash Code	MurmurHash2 output	2 bytes

With a 2-byte hash, the probability of collision remains below 2% for up to 50 concurrent transmitters, as was shown in [4]. If needed, the generic field can be partially reallocated to further extend the hash and reduce collision risk.

The integrated framework is implemented on ESP32, enabling devices to advertise and scan on both radios within a single application cycle. Application logic is agnostic to the underlying transport: data can be published or received via either BLE or WiFi, and filtering is performed using the same hash-based verification.

4 Experimental Procedure

When enabling inter-technology communication (e.g., BLE advertisements or WiFi using beacon stuffing) over the same hardware, it’s not obvious how the combined use of these radios would affect the power draw and the overall output. There may be interactions, overhead, or unexpected inefficiencies that would only appear in real deployments. It is thus important to establish a performance study as a function of the input system parameters. In this article, we study the power consumption in terms of the current draw of the combined system, and compare it to a BLE only system. We then study the effect of the advertisement interval on the current draw of the combined system. Also, we study the boundary cases where the advertisement intervals of BLE and WiFi are equal. Power consumption remains a critical metric for wireless communication systems, especially in mobile and wearable devices that are power-constrained. Thus, knowing the actual power consumption under different modes (BLE-only, WiFi-only, BLE+WiFi concurrently, etc.) is essential for choosing the best technology for a given application scenario, optimizing the battery life, and informing the decision on when and how to activate both WiFi and BLE, or to operate only one of them. Finally, we provide some insights on the performance and tradeoffs of having both radios to operate concurrently.

We implement the system using ESP WROOM32D. ESP32 is used within the scope of this project due to its cost-effectiveness, large community, and flexible development environment. Moreover, it features integrated Wi-Fi and dual-mode Bluetooth capabilities, thus allowing a dual functionality between the two techniques on the same hardware. The current draw is measured using a Joulescope JS220 Precision Energy Analyser⁴. We setup the experiment as shown in figure

⁴ <https://www.joulescope.com/products/js220-joulescope-precision-energy-analyzer>

2, where the esp's are connected to the joulescope one at a time, and a DC power supply of 5V is used.

During the implementation of concurrent BLE and WiFi advertisement and scanning on the ESP32, several challenges stemming from the hardware radio-sharing model were encountered. Notably, while it was possible to advertise over both BLE and WiFi simultaneously, attempting to scan for BLE advertisements and WiFi access points in parallel led to conflicts, system instability, or the need to explicitly suspend one stack to allow the other to function. This limitation is enforced by the ESP-IDF, which restricts simultaneous scanning operations to prevent resource contention on the shared radio.

To address this, the implementation adopted a serialized approach, whereby BLE scanning and WiFi scanning were performed sequentially rather than concurrently. This was accomplished in the application code by explicitly deinitializing and disabling the BLE stack after a BLE scan completed, before reinitializing the WiFi stack for scanning, and vice versa. By carefully orchestrating the radio usage and respecting the necessary setup sequences between BLE and WiFi operations, it was possible to collect data from both technologies within a single application cycle, albeit at the cost of increased latency and complexity in the software flow. This solution enables inter-technology applications, but highlights the need for careful radio management on platforms with shared wireless hardware.

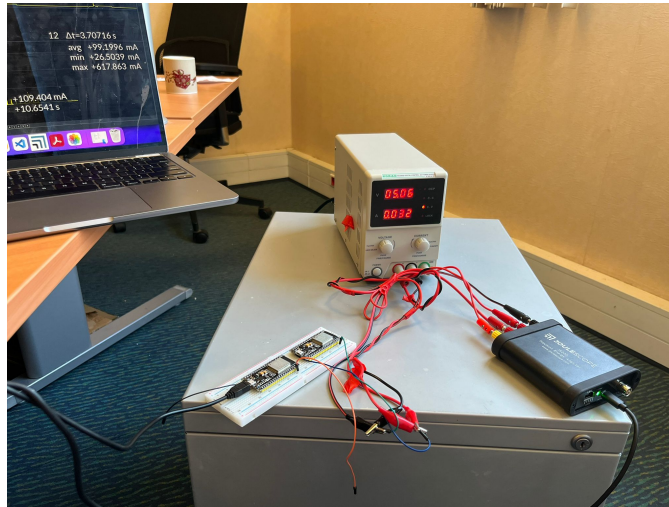


Fig. 2. Setup

For verifying the successful reception and data decoding, we use a dummy data vector to be advertised.

Figure 3 presents the terminal output of a inter-technology BLE-WiFi broadcast and scan session on the ESP32, demonstrating the successful end-to-end propagation and detection of a custom payload across both radios. The BLE scan result at the top highlights the reception of a raw advertising packet (43 FF FF 56 31 37 32 35 38 30 34 2E 33 31 79 8C), which is recognized as belonging to the shared dataspace, and the relevant data is extracted and displayed. Subsequently, in the WiFi scan output, the same payload is observed after being hex-encoded and broadcast as an SSID, which is then decoded back into its original byte sequence (43 FF FF 56 31 37 32 35 38 30 34 2E 33 31 79 8C). The system correctly identifies this SSID as part of the shared dataspace and parses out the advertised role, app code, and the intended data payload. The highlighted sections illustrate the seamless interoperability between BLE and WiFi advertisement and scanning, validating the framework’s ability to carry and verify a custom payload end-to-end across both radio technologies, while also confirming the effectiveness of the encoding/decoding and hash-matching process.

```

TERMINAL Python +
Scan started
ESP_GAP_BLE_SCAN_RESULT_EVT
Reason: Found=4000,AdvDataLen=16,DataLen=16
Raw Advertising Data: 43 FF FF 56 31 37 32 35 38 30 34 2E 33 31 79 8C
This packet is within the shared dataspace
This device provides data I'm interested in. Extracted Advertising Data:
FF FF 56 31 37 32 35 38 30 34 2E 33 31 79 8C

[!] In listen_wifi()
E (1503) wifi_init: Failed to deinit Wi-Fi driver (0x3001)
[!] Initializing WiFi
[!] WiFi mode set to STA
[!] WiFi started
[!] Starting WiFi scan (blocking)...
[!] WiFi scan complete!
WiFi Scan Found 13 APs:
Received SSID (as string): 43FFFF56313732353830342E3331798C
Received SSID (hex bytes): 34 39 46 46 46 46 39 36 33 31 33 37 33 32 33 35 33 38 30 33 34 32 45 33 33 33 31 37 39 38 43
Decoded SSID bytes: 43 FF FF 56 31 37 32 35 38 30 34 2E 33 31 79 8C
Hash in SSID: 79 8C | Calculated Hash: 79 8C
This packet is within the shared dataspace
Advertised Role: 01 | Advertised App Code: 00
This AP provides data I'm interested in. Extracted Data:
56 31 37 32 35 38 30 34 2E 33 31 79 8C
Received SSID (as string): F800C0WIFI
Received SSID (hex bytes): 66 72 65 65 62 6F 78 5F 57 59 53 59 47 49
Hash in SSID: 00 00 | Calculated Hash: CC D0
Received SSID (as string): F800C0WIFIsecure
Received SSID (hex bytes): 46 72 65 65 57 69 66 69 5F 73 65 63 75 72 65
Hash in SSID: EC 00 | Calculated Hash: a9 70
Received SSID (as string): B0x-DABFB26
Received SSID (hex bytes): 42 62 6F 78 2D 44 41 42 46 42 42 32 36
Decoded SSID bytes: B0 00 F3 AB FB B2
Hash in SSID: FB B2 | Calculated Hash: ED 4D
    
```

Fig. 3. Output

To begin with the power profile study, we measure the average currents for advertising and scanning for BLE and for the composite system (WiFi+BLE). We put the results in table 2.

Examining in table 2, the measured current values for BLE only and BLE+WiFi operation modes reveal several important trends. When BLE operates alone and over the advertisement interval of 40ms, the average current consumption during advertising is 54.37 mA, with a minimum of 38.59 mA and a peak of 317.32 mA. For scanning over the scan interval of 40ms, the BLE-only mode shows an average of 87.39 mA, a minimum of 36.95 mA, and a maximum of 191.58 mA.

In contrast, when BLE operates concurrently with WiFi (BLE+WiFi), the average current during advertising rises substantially to 129.6 mA over the adver-

	BLE Only		BLE+WiFi	
	Adv	Scan	Adv	Scan
Avg (mA)	54.37	87.39	129.6	97.47
Min (mA)	38.59	36.95	76.45	26.3
Max (mA)	317.3	121.58	545.31	559.28

Table 2. Current consumption (mA) comparison for BLE and WiFi in different operating modes.

tising interval of WiFi (100ms), and scanning jumps to 97.47 mA over the whole scan period. The minimum currents also increase, with advertising at 76.45 mA and scanning at 26.3 mA. Most strikingly, the maximum observed current surges to 545.31 mA for advertising and 559.28 mA for scanning—nearly doubling the peaks seen in BLE-only mode.

Then, to zoom over what is happening with more details, we trace in figure 6(a,b) the current of the advertiser and the scanner while operating both WiFi (AP mode) and BLE. In Figure 6-a, we observed the current trace for 300ms. I.e., 3 advertisement intervals of WiFi. Each is advertising at a distinct interval—100 ms for WiFi and 40 ms for BLE. The current trace shows a steady baseline of around 100–150 mA, punctuated by periodic spikes. The largest spikes, reaching up to approximately 533 mA and occurring every 100 ms, correspond to WiFi advertisements, which are known for their higher power consumption. In between these, smaller yet regular spikes appear at roughly 40 ms intervals, representing BLE advertisements that consume less current but occur more frequently. The overall average current is elevated due to these periodic transmissions, and the pattern confirms that both WiFi and BLE are operating at their configured intervals without overlap. This demonstrates the additive impact of concurrent wireless advertising on power consumption.

Figure 6-b shows the current trace of the scanner. As will be explained in the observations subsection, we operate the scanner sequentially between BLE and WiFi. Over a 3-second window, the trace starts with a brief idle phase at a low baseline (around 50 mA), followed by a large spike peaking at nearly 585 mA when scanning activity begins—likely corresponding to the initial activation or a WiFi scan, which typically draws significant current. After this spike, the current settles to a higher idle level (100–150 mA), reflecting the system being in an active scanning state. Then, a sequence of periodic smaller current spikes at regular intervals can be seen. These repeated pulses represent the bursts of activity as the device performs active scanning for BLE advertisements. Throughout this period, the overall average current consumption is elevated (about 120 mA). Then, as WiFi scanning starts as can be seen for the 11 spikes, the current is elevated to 250mA for each spike. This indicates the significant energy cost of WiFi scanning. We can deduce the cumulative energy impact of concurrent wireless scanning for both WiFi and BLE, where the baseline never fully returning to the pre-scan idle level during active operation.

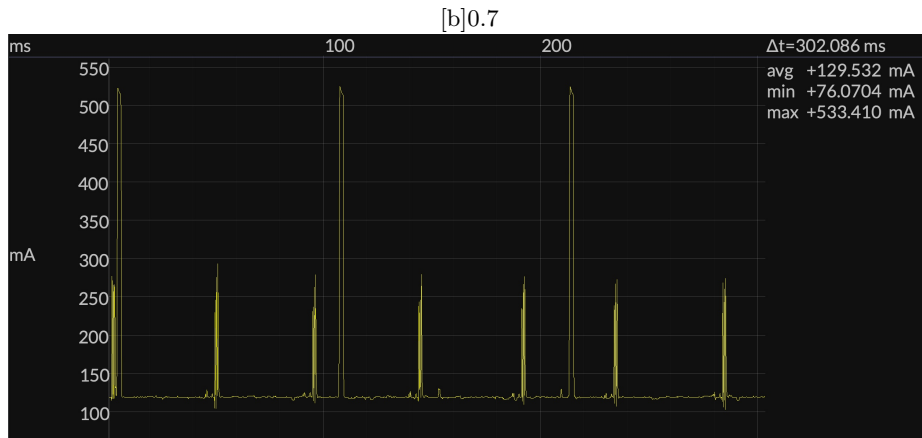


Fig. 4. Advertiser BLE+WiFi

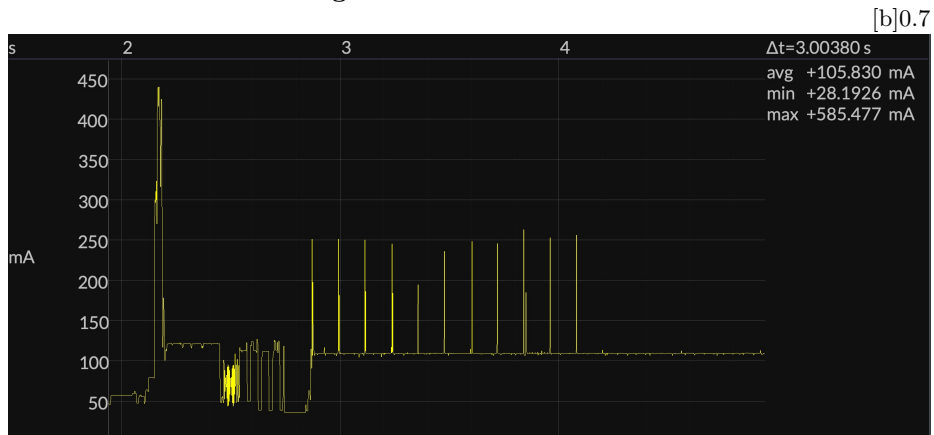


Fig. 5. Scanner BLE+WiFi

Fig. 6. Current traces BLE+WiFi (a) Advertiser, (b) Scanner

The comparison between the power profiles of the advertiser and scanner in the dual mode, clearly demonstrates that scanning (especially WiFi scanning) is more energy-intensive than advertising, both in average and peak current, due to longer radio-on times and the need to listen across multiple channels. Furthermore, while BLE advertising produces short, low-power pulses, WiFi advertising and scanning both introduce significant current surges, but scanning sustains a higher power envelope for longer durations. This comparative observation underlines the asymmetry in power costs between broadcast and receive operations across BLE and WiFi, which is a key design consideration for inter-technology middleware.

Having observed the elevated power profile for the composite system, we experiment with varying the advertising intervals.

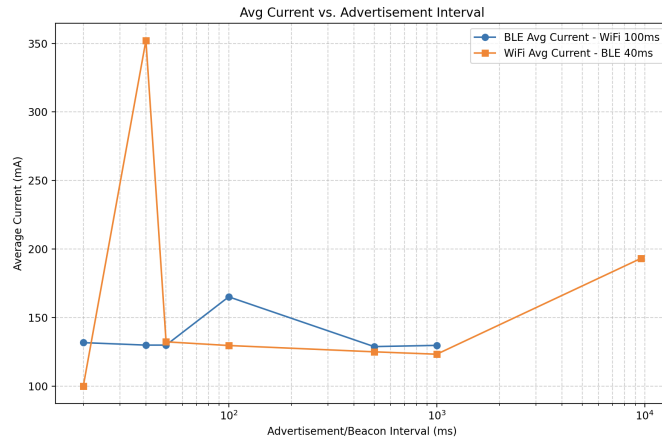


Fig. 7. Average current drawn as a function of different advertisement periods.

Figure 7 plots the average current drawn as a function of different advertisement periods. This experiment was done over two steps. In the first step, we fixed the WiFi advertisement interval to 100ms (the default value), and varied that of the BLE. The measured current was always between 128mA and 131mA, except for one point where the advertising period of BLE matched that of the WiFi, i.e, 100ms. At this instant, the average current raised to 165.122mA, indicating that a higher amount of processing was being done at the esp. In the next step, we repeated the same procedure but with fixing the advertisement interval of the BLE this time to 40ms, and varying that of the WiFi. We observed that when the advertisement intervals matched at 40ms, the average current raised significantly to 352.216 mA, almost doubling its value. Also, we observed that the peak current raised up to 2.52A. This indicates that more processing is being done at the esp, and even higher than the case of BLE. Then it was observed that the current drawn started to decrease as the advertisement interval was go-

ing beyond 100ms. We can deduce from this figure that WiFi affects the energy cost for a more frequent advertising, more than BLE.

Finally, to understand better what is happening when the advertisement intervals match (at 40ms, and at 100ms), we traced the current trace for these values in figures 8 and 9.

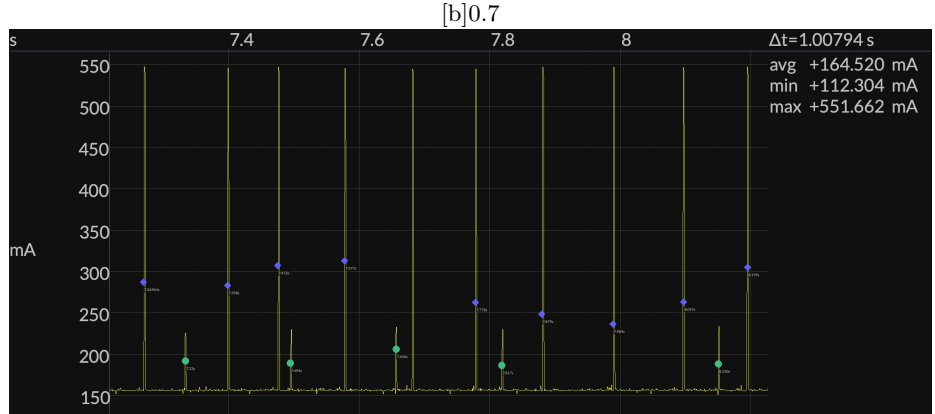


Fig. 8. BLE+WiFi 100ms

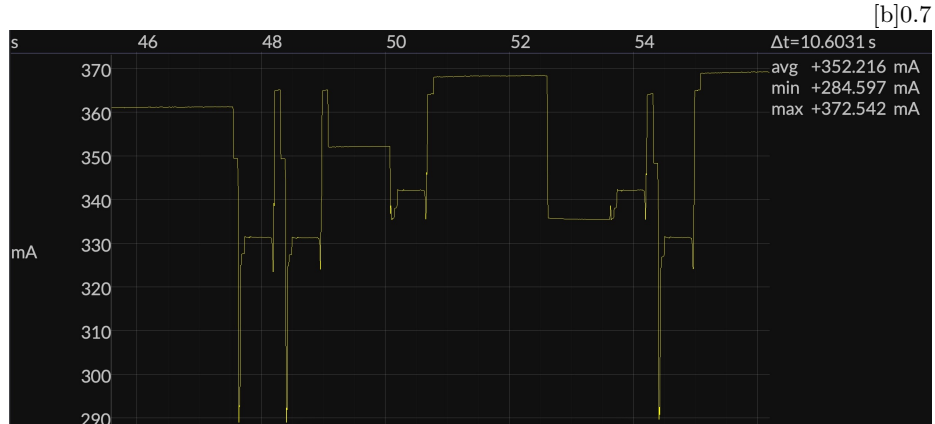


Fig. 9. BLE+WiFi 40ms

Fig. 10. Current traces at matching advertisement intervals: (a) 100ms, (b) 40ms.

When both advertisement intervals are set to 100ms and over a period of 1s, it can be seen from figure 10-a that the advertisements are no more uniformly spaced. During this period, 10 WiFi advertisements were advertised, whereas only 5 BLE advertisements were observed. This indicates that the direct effect of this overlap on the radio scheduling and sharing at the esp32. We mitigated

this problem at the scanner by making the scans occur sequentially, and by de-initializing one tech when the other is listening. Moreover, the values of the min and max currents drawn are of high values compared to the values when the advertisement intervals are different and apart.

We also set both advertisement intervals to 40ms, and observed in 10-b that the resulting current trace reveals a highly irregular and elevated current profile, with the average current sustained at approximately 352 mA and very little time spent at lower baseline levels. Unlike traces with longer intervals, which show distinct, periodic peaks corresponding to individual advertisement events, this configuration produces a continuous plateau-like pattern with rapid fluctuations, indicating that the ESP32’s radio is almost constantly active. This behavior is characteristic of severe resource contention, where the BLE and WiFi advertisement events frequently overlap or even collide, preventing the radio from entering idle or low-power states between transmissions. Such inefficient scheduling not only causes significantly higher average power consumption—greatly diminishing battery life—but also increases the likelihood of packet loss and scheduling errors. Overall, this trace demonstrates that aggressive, synchronized advertisement intervals on both BLE and WiFi lead to inefficient radio coexistence, underscoring the necessity of careful interval configuration or the implementation of staggered schedules to optimize energy efficiency and reliability.

5 Conclusions

In this paper, we have proposed a unified, advertisement-based communication framework enabling direct, inter-technology data exchange over BLE and WiFi without requiring radio-level modifications. Our implementation on ESP32 hardware revealed non-trivial interactions in power consumption during concurrent operation, highlighting the need for careful tuning of advertisement intervals and coexistence parameters. Experimental results show that power draw under hybrid BLE+WiFi modes is not simply additive, with contention and scheduling dynamics introducing measurable inefficiencies. Also, we have showed that the effective payload of broadcasted data decreases by half when interoperability among devices is crucial. These findings are critical for ephemeral, energy-sensitive applications, where communication must be opportunistic yet efficient. Our work sets the basis for a middleware layer that abstracts away low-level protocol complexity, harmonizes BLE and WiFi behavior, and exposes a simple, unified API to application developers—bridging the gap between heterogeneity and deployability in real-world scenarios. By tailoring the communication system to the requirements of ephemeral, opportunistic exchanges; prioritizing low latency, simplicity, and open access, we can enable new classes of applications that are lightweight, flexible, and responsive to the dynamics of the environment.

References

1. Chin, A., Wang, H., Xu, B., Zhang, K., Zhu, L.: Connecting people in the workplace through ephemeral social networks. In: 2011 IEEE 3rd International Conference

- on Privacy, Security, Risk and Trust and 2011 IEEE 3rd International Conference on Social Computing, pp. 527–530. IEEE (2011)
2. Chandra, R., Padhye, J., Ravindranath, L., Wolman, A.: Beacon-stuffing: Wi-Fi without associations. In: 8th IEEE Workshop on Mobile Computing Systems and Applications, pp. 53–57. IEEE (2007)
 3. Issa, M., Espinel-Galviz, P., Couderc, P., Bonnin, J.M.: WiFi beacon stuffing vs. BLE connectionless mode for ephemeral interactions in smart environments. In: 2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 402–407. IEEE (2024)
 4. Issa, M., Couderc, P., Bonnin, J.M.: Supporting an ephemeral shared dataspace with a BLE connectionless protocol. In: 21st EAI International Conference on Mobile and Ubiquitous Systems. Springer (2024)
 5. Zachariah, T., Jackson, N., Ghena, B., Dutta, P.: ReliaBLE: Towards reliable communication via Bluetooth Low Energy advertisement networks. In: Proc. 2022 International Conference on Embedded Wireless Systems and Networks (EWSN), vol. 22 (2022)
 6. Wu, M., Ma, B., Liu, Z., Xiu, L., Zhang, L.: BLE-horn: A smartphone-based Bluetooth low energy vehicle-to-pedestrian safety system. In: 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6. IEEE (2017)
 7. Gu, F., Niu, J., Jiang, L., Liu, X., Hancke, G.P.: SafePath: Exploiting ubiquitous smartphones to avoid vehicle–pedestrian collision. IEEE Internet of Things Journal 9(9), 6763–6777 (2021)
 8. Abedi, A., Abari, O., Brecht, T.: Wi-le: Can WiFi replace Bluetooth?. In: Proc. 18th ACM Workshop on Hot Topics in Networks, pp. 117–124 (2019)
 9. Ng, P.C., Spachos, P., Plataniotis, K.N.: COVID-19 and your smartphone: BLE-based smart contact tracing. IEEE Systems Journal 15(4), 5367–5378 (2021)
 10. Raes, R., Luxey-Bitri, A., Rouvoy, R., Frey, D., Taïani, F.: Venice: Eschewing the cloud by leveraging local communication channels. In: ICT4S 2024 – International Conference on Information and Communications Technology for Sustainability, pp. 1–4 (2024)
 11. Couderc, P., Maurel, Y.: Location corroboration using passive observations of IEEE 802.11 access points. In: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 1–7. IEEE (2019)
 12. Tosi, J., Taffoni, F., Santacatterina, M., Sannino, R., Formica, D.: Performance evaluation of Bluetooth low energy: A systematic review. Sensors 17(12), 2898 (2017)
 13. Townsend, K., Cufí, C., Davidson, R., et al.: Getting started with Bluetooth low energy: Tools and techniques for low-power networking. O’Reilly Media (2014)
 14. Mohammadi, S., Ismail, K., Ghods, A.H.: Investigating Wi-Fi, Bluetooth, and Bluetooth low-energy signal characteristics for integration in vehicle–pedestrian collision warning systems. Sustainability 13(19), 10823 (2021)
 15. Dede, J., Förster, A.: Comparative analysis of opportunistic communication technologies. In: International Conference on Interoperability in IoT, pp. 3–10. Springer (2016)
 16. Rodrigues, J., Marques, E.R.B., Lopes, L.M.B., Silva, F.: Towards a middleware for mobile edge-cloud applications. In: Proc. 2nd Workshop on Middleware for Edge Clouds & Cloudlets, pp. 1–6 (2017)
 17. Yun, Y.H., Choi, H.: Development of a multiuser interaction middleware based on D2D communications. In: 2017 International Conference on Information Networking (ICOIN), pp. 480–485. IEEE (2017)

18. Saputra, Y., Hua, J., Wendt, N., Julien, C., Roman, G.C.: Warble: Programming abstractions for personalizing interactions in the internet of things. In: 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp. 128–139. IEEE (2019)