Enhancing Scalability in Wi-Fi IoT Networks with Logical Data Plane Segregation Using SDN Principles

Gabriel Vieira[a], Ana Rita Ortigoso[b], Daniel Fuentes[c], Luis Frazão[d], Nuno Costa[e] and António Pereira[f]

Computer Science and Communication Research Centre, Polytechnic University of Leiria, Portugal
{gabriel.m.vieira, ana.l.ortigoso, daniel.fuentes, luis.frazao, nuno.costa, apereira}@ipleiria.pt

Keywords: Data Plane Segregation, IoT, Wi-Fi 6, ESP-NOW, SDN, Scalability.

Abstract: The increasing demand for wireless connectivity, with IoT devices projected to exceed 20 billion by 2025, reinforces Wi-Fi as the dominant technology. However, the standard star topology limits coverage for widely dispersed IoT devices. While mesh networking extends coverage, most IoT endpoints lack native mesh support, posing scalability and security challenges. This study proposes an IoT architecture leveraging a dual-stack and dual logical data plane approach based on SDN principles. It separates control and data traffic into three planes: IoT data, Wi-Fi extension, and control. The control plane employs ESP-NOW for mesh optimisation, while Wi-Fi ensures compatibility and expanded functionality. A prototype using ESP32-C6 DevKitC-1 modules demonstrates cost-efficiency, supporting stable connectivity with performance degradation beyond 50 metres. Experimental results confirm the architecture's ability to establish a self-organising, resilient mesh network with dynamic reconfiguration, offering a scalable and flexible solution for IoT mesh networks.

1 INTRODUCTION

Wi-Fi technology is evolving rapidly, driven by the increasing societal demand for wireless connectivity. It has become the preferred choice for Internet of Things (IoT) devices, with the number of connected devices worldwide expected to surpass 20 billion by 2025 (Analytics, 2024). Several initiatives have been implemented to establish heterogeneous networks that enable communication between different devices (Zegeye et al., 2023). However, the increasing demand for enhanced security and efficient device configuration highlights the limitations of a single data plane, particularly in terms of scalability and vulnerability to security threats. As a result, the use of multiple planes to forward data can help mitigate (Kaljic et al., 2019). Providing Wi-Fi coverage for IoT devices, which are often widely dispersed, presents significant challenges. This is especially relevant in environments and smart buildings such as hotels, corporate offices, factories, hospitals, and university campuses. (Khanna and Kaur, 2020). A solution that involves extending the Wi-Fi network by leveraging existing IoT devices through a mesh network, as there are already numerous devices available (e.g., microwaves, refrigerators, sensors, and other actuators) is favourable. Ideally, this could be achieved by simply updating the firmware of compatible devices, enabling networks to become more configurable and extend their coverage area.

This research aims to provide a solution that enhances current IoT devices with dual logical data planes with the capability to create an IoT Wi-Fi mesh network using the latest fully specified Wi-Fi protocol (Ferro and Potorti, 2005), the Wi-Fi 6 / 802.11ax (Islam and Kashem, 2022), and the low-level communication protocol ESP-NOW, which facilitates managing IoT devices and their mesh network. The solution makes the following contributions:

- **Development of a Dual Data Plane Architecture:** Conceptualised an architecture that makes the use of two logical data planes possible, for IoT devices and for Wi-Fi end devices users.
- **Development of a Dual Stack Single Radio Mesh Architecture:** Designed an architecture that integrates two different communication pro-

[a] https://orcid.org/0009-0000-2300-8441
[b] https://orcid.org/0009-0001-7529-5857
[c] https://orcid.org/0000-0001-9726-1087
[d] https://orcid.org/0000-0003-2571-7940
[e] https://orcid.org/0000-0002-2353-369X
[f] https://orcid.org/0000-0001-5062-1241

Vieira, G., Ortigoso, A. R., Fuentes, D., Frazão, L., Costa, N. and Pereira, A.
Enhancing Scalability in Wi-Fi IoT Networks with Logical Data Plane Segregation Using SDN Principles.
DOI: 10.5220/0013428800003944

In *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, pages 378-385
ISBN: 978-989-758-750-4; ISSN: 2184-4976

tocols (Wi-Fi 6 & ESP-NOW), to provide both data segregation and extend coverage by using other IoT devices as part of the network.

- **Logical Separation of Data and Control Planes:** The architecture proposed introduces a separation between control and two logical data planes. This introduces an Software-Defined Networking (SDN) inspired virtual separation that allows independent management of data, reducing interference and enhancing security.
- **Cost-Effective Prototype Implementation:** Development of a functional mesh network prototype using low-cost IoT devices that implement the Wi-Fi 6 and ESP-NOW protocols concurrently.
- **Comprehensive Functional and Performance Evaluation:** Testing the feasibility of the solution and the prototype to evaluate feasibility, functional integrity, and performance metrics.

The article is structured as follows: section 2 provides an overview of the technologies employed and their alternatives. Section 3 offers a literature review, placing this research in the context of similar solutions and prior work. Section 4 describes the proposed architecture, integrating existing IoT devices into the Wi-Fi infrastructure to provide network access to nearby end-user devices. In Section 5, the implemented prototype is presented with details on hardware components, testing setup, control plane and data plane. Section 6 outlines the results obtained from the prototype, which are further analysed and discussed in section 7 to provide insights into the solution's performance and functionality. Finally, section 8 concludes the research by summarising key findings and highlighting potential future work.

2 BACKGROUND

Wi-Fi is the most widely used wireless system, known for its low cost, ease of installation and use, and minimal skill requirements. The latest standard for IoT devices, Wi-Fi 6 (802.11ax), is increasingly adopted in industrial and IoT markets due to its high throughput, low latency, precise localisation capabilities, efficient resource management through Resource Units (RUs), support for speeds up to 9.6 Gbps, and tri-band operation (2.4, 5, and 6 GHz) (Karamyshev et al., 2024). These recent advancements enable its use in battery-powered IoT devices, solidifying its role as one of the preferred choices for connecting both IoT and user devices (Oughton et al., 2021) (Mozaffariahrar et al., 2022b) (Perez-Ramirez et al., 2022).

ESP-NOW is a protocol designed to deliver low latency, low power consumption, and long-distance communication on the 2.4 GHz spectrum (Eridani et al., 2021). The ESP-NOW protocol can transmit up to 250 bytes of data and can coexist with both Wi-Fi and Bluetooth (Espressif, 2023d) (Espressif, 2023c). While the 250-byte payload may seem restrictive, it is sufficient for many IoT use cases, such as sending sensor readings or control data. The following tables 1 and 2, detail the structure of ESP-NOW frame.

Table 1: ESP-NOW frame.

MAC Header	Category Code	Organisation ID	Random	Vendor Specific	FCS
24 bytes	1 byte	3 bytes	4 bytes	7-257 bytes	4 bytes

Table 2: ESP-NOW vendor specific packet.

Element ID	Length	Organisation ID	Type	Version	Body
1 byte	1 byte	3 bytes	1 byte	1 byte	0-250 bytes

Bluetooth is a Personal Area Network (PAN) that operates within the 2.4 GHz spectrum using frequency hopping to transmit data. There are two versions of this technology, Classic and Bluetooth Low Energy (BLE) (Bisdikian, 2001). However, when using a dual-stack technology, BLE must share antenna time, which results in reduced performance. (Espressif, 2020b).

Thread is an Internet Protocol (IP) solution using the IPv6 on the standard IEEE 802.15.4. Thread aims for reliable, flexible, low-power, device-to-device, agnostic communication, and a self-configuring network that enhances functionality and reduces maintenance costs (Thread Group, 2020) (Thread Group, 2024). According to the scope of the work and to authors in (Espressif, 2023f), concurrency with the other mentioned protocols will not work.

ZigBee, based on IEEE 802.15.4, provides a low data rate and simple connectivity, designed for low-powered (battery-operated) devices. ZigBee supports standard mesh networking, enhancing range and reliability (ZigBee Alliance, 2017). However, it operates on the same 2.4 GHz band as Wi-Fi, risking interference. Its dual PHY adds complexity but offers limited performance gains. In contrast, Wi-Fi, particularly Wi-Fi 6, provides higher throughput, lower latency, and better scalability for modern IoT networks, especially when leveraging SDN principles for logical data segregation (Zigbee2MQTT, 2024).

In summary, Wi-Fi offers robust interconnectivity and performance, while ESP-NOW supports extended ranges with minimal latency. Bluetooth has a low power consumption, making it an optimal choice for long-term applications across a range of devices. ZigBee targets low-powered devices and has a built-in mesh topology. Among these technologies, ESP-NOW is noted for its low interference, better energy efficiency, and lowest latency (Eridani et al., 2021).

A comparative analysis was conducted to identify low-cost IoT devices that support communication technologies like Wi-Fi 6, ESP-NOW, Bluetooth, Thread, and ZigBee. The results highlight the communication capabilities and average prices of the selected devices, as presented in table 3.

Looking at the comparison in table 3 with device features and prices in conjunction of the aforementioned, we conclude that for the required, ESP32-C6 meets the desired criteria: Wi-Fi 6 and ESP-NOW support at a low cost. This device is a new addition to Espressif's product line, featuring support for Wi-Fi 6, BLE, Thread, and ZigBee. Additionally, it integrates the Espressif IoT Development Framework (ESP-IDF), an open-source project based on the FreeRTOS kernel, offering well-documented drivers, protocol support, and storage solutions that simplify IoT development (FreeRTOS, 2024).

3 LITERATURE REVIEW

The scientific community has yet to extensively explore the integration of complementary communication protocols in resource-constrained environments. Current research lacks comprehensive investigations into their combined application for achieving scalability, energy efficiency, and cost-effectiveness, highlighting a significant gap in the field. To the best of our knowledge, this is one of the few prototypes in this area.

In the study by (Muhendra et al., 2017), the Quick Mesh Project (QMP) framework was used to develop a WiFi mesh network with TP-LINK MR3020 routers, configured with the Quick Mesh Project (QMP) based OpenWRT and BMX6 routing protocol. The network utilised the MQTT protocol for machine-to-machine communication via a publish-and-subscribe model. An IoT WiFi Client, based on the low-power ESP8266 module, was also developed to connect electronic devices such as sensors and actuators to the network. In contrast to this, the present solution proposes using devices as network propagators with multiple planes, eliminating the need for dedicated mesh routers and creating its own managed mesh network.

Furthermore, the article by (Gergeleit, 2019) implements a mesh network using the painlessMesh library, relying on NAT for external network access but without incorporating a routing protocol. In a similar vein, Autotree utilises NAT alongside a distance vector-like routing mechanism tailored for ESP8266 modules, forming a dynamic tree topology with limited node mobility and simplified deployment. However, it lacks inter-node IP connectivity. By contrast, our solution introduces a dual-stack architecture with logical data plane segregation based on SDN principles. By isolating control and data traffic using ESP-NOW and Wi-Fi 6, the system improves scalability and route management. Moreover, it employs ESP32-C6 modules to support contemporary protocols, such as Wi-Fi 6, while enabling self-organising mesh networks.

In a related approach, the study by (Manvi and Maakar, 2020) develops a wireless mesh network with ESP-32 nodes, allowing decentralised communication without requiring internet access or routers. Data is exchanged through TCP servers on port 5555, and the network exhibits self-healing capabilities when routes fail. Our work differs by enhancing compatibility and implementing SDN principles to segregate control and data planes. Through the use of ESP-NOW and Wi-Fi 6, it achieves optimised traffic flow and more efficient resource utilisation, thereby surpassing the basic peer-to-peer methodology adopted in (Manvi and Maakar, 2020).

The work of (Khanchuea and Siripokarpirom, 2019) explores the integration of ZigBee, ESP-NOW, and ModBus protocols within a multi-protocol gateway to control IoT devices. ESP-NOW serves as the foundation for a low-power, multi-hop wireless network, while ZigBee forms subnetworks for device communication. Field trials highlight the effectiveness of ESP32 Wi-Fi/BLE chips combined with ESP-NOW in constructing energy-efficient, self-organising sensor networks. In contrast, our solution focuses on logical data plane segregation using SDN principles. By leveraging different protocols for each distinct planes, the system ensures more efficient traffic integration, moving beyond the modular, multi-protocol approach presented in (Khanchuea and Siripokarpirom, 2019).

Lastly, Espressif provides the *ESP-WIFI-MESH* implementation (Espressif, 2024b), which supports various configurations, including "Internal Communication", "IP Internal Network", and "Manual Networking" (Espressif, 2024a). However, its functionality is restricted to internal network communications, as external devices (other Wi-Fi clients) cannot connect to or interact with the mesh network.

Table 4 provides a comparison of the key fea-

Table 3: Comparison of native communication technology support in IoT devices.

Features / Devices	Wi-Fi 6	ESP-NOW	Bluetooth	Thread	ZigBee	Avg. Price
Arduino Nano ESP32	-	✓	✓	-	-	20$
Arduino UNO Wi-Fi REV2	-	✓	✓	-	-	25$
Espressif ESP32-C6	✓	✓	✓	✓	✓	8$
Raspberry Pico W	-	-	✓	-	-	5$
Raspberry Pi Zero W	-	Adapted (Flayols, 2024)	✓	-	-	16$
Raspberry Pi 5	✓	Adapted (Flayols, 2024)	✓	-	-	70$

tures from the mentioned articles and the proposed architecture, including: *MQTT over ESP-NOW (MoE), Data-Plane Technology (DPT), Control-Plane Technology (CPT), Self-Organisation/Healing (SOH), SDN Capable (SDNC), Propagates Network (PN), Internet Connection, (IC), Data and Control Planes Separation (DCPS), Easy One-click Configuration (EOC), Monitoring (M), NAT Service (NS), ESP-WIFI-MESH (EWM), Allow External Device Connections (AEDC).*

4 PROPOSED ARCHITECTURE

The work proposes a new architecture that integrates IoT devices with Wi-Fi mesh network, to ensure network access to nearby end devices. The selection of Wi-Fi and ESP-NOW is critical for the architecture's scalability and efficiency. Wi-Fi ensures high data transmission capacity and broad compatibility for the data plane, while ESP-NOW enables low-latency and energy-efficient operations for the control plane. This combination leverages the strengths of both protocols to enable concurrent operations across multiple planes. This concurrent operation reduces interference from each other and is crucial for the concurrent use of both Wi-Fi and ESP-NOW.

Figure 1 illustrates the proposed architecture, where connections between existing IoT devices create a self-expanding mesh network. Wi-Fi and ESP-NOW will work concurrently in different roles, acting as the dual data plane and control plane, respectively, to establish distinct production and management networks at separate layers. These run in the same interface as shown in the figure 2.

The network includes a common router serving as the internet gateway (where is possible to have multiple gateways) and a simple Wi-Fi Access Point (AP) that connects directly to the router. Both allow wireless access for other devices, namely IoT devices with sensors and/or actuators (e.g., smart light bulbs, smart doors, or a smart appliance). The IoT devices use the ESP-NOW protocol to communicate, selecting the best route based on peer signal strength, reachable APs/routers or other metric. They monitor changes in their surroundings and adjust their connections, which are visible and configurable through an SDN Controller accessible to any device. After achieving convergence, IoT devices configure the Wi-Fi mesh

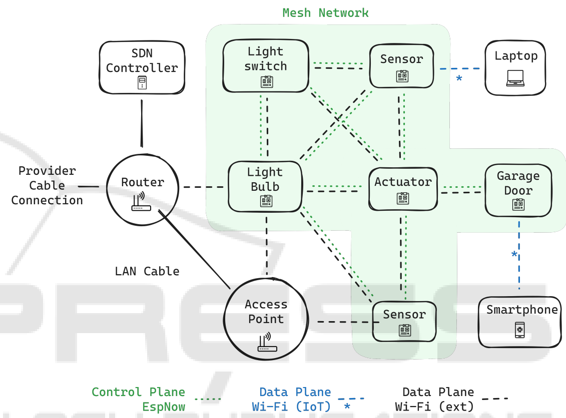

Figure 1: Proposed mesh architecture with devices as network propagators for both data and control planes.

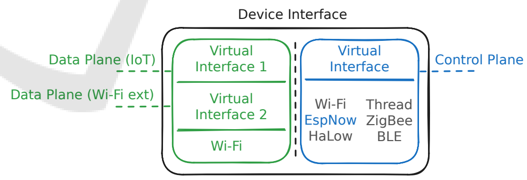

Figure 2: Device with a single interface for both dual data and control planes.

Table 4: Comparison between related work and proposed architecture features.

Features	**(Espressif, 2024b)**	**(Muhendra et al., 2017)**	**(Gergeleit, 2019)**	**(Manvi and Maakar, 2020)**	**(Khanchuea and Siripokarpirom, 2019)**	**Proposed**
DPT	Wi-Fi	MoE	Wi-Fi ad-hoc	Wi-Fi	Zig-Bee	Wi-Fi
CPT	Wi-Fi	MoE	NAT	TCP	ESP-NOW & ModBus	ESP-NOW
SOH	✓	✓	✓	✓	✓	✓
SDNC	-	-	-	-	-	✓
PN	✓	-	✓	-	-	✓
IC	✓	-	✓	-	-	✓
DCPS	-	-	-	-	✓	✓
EOC	-	-	-	-	-	✓
M	-	-	-	-	✓	✓
NS	✓	-	✓	-	-	✓
AEDC	-	-	✓	-	-	✓

network by connecting to various APs/routers, extending the network using the same SSID. The Wi-Fi mesh network remains active among IoT devices, but broadcasting to end-user devices requires explicit controller configuration. The IoT devices can perform their primary functions, including acquiring sensor data, executing actuator tasks, and communicating with the IoT data collector. End-user devices (laptops, smartphones, etc.) can connect to the Wi-Fi network via any AP/router or any IoT device broadcasting that network.

Figure 3 illustrates the implementation process of the architecture: starting with raw operation, configuring interfaces and services, separating control and data planes, initialising mesh networks for device communication, and loading specialised software for sensors or actuators.

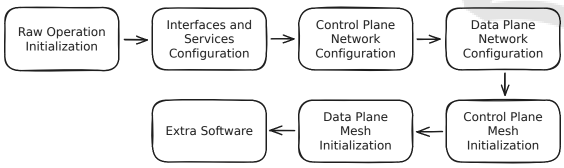

Figure 3: Proposal of the Software Flow the IoT devices.

5 IMPLEMENTED PROTOTYPE

Wi-Fi 6 is a relatively new technology in IoT terms (Mozaffariahrar et al., 2022a), and manufacturers are still in the process of adopting and releasing compatible devices. One such device is the Espressif ESP32-C6 DevKitC-1 v0.0, which has been selected for use in this project. This development board supports Wi-Fi 6, Bluetooth 5, Thread, and ZigBee communications, according to the official documentation (Espressif, 2023e), making it the first low-cost IoT device capable of implementing Wi-Fi 6 for mesh network development. A necessary note, is that the device used in the present work is a **sample version** (v0.0), it has power stability issues as noted in the sample paper accompanying the device, "The ADC in the current samples is not calibrated", and also doesn't support external antennas (The ESP32-C6-WROOM-1U version, which came out later (Espressif, 2024c) supports and has this problems addressed).

The implemented prototype was developed using the Espressif ESP32-C6, as it is currently the only low-cost IoT device meeting the necessary requirements for the intended functionality (Espressif, 2023g). This device can communicate using both protocol stacks concurrently via a single antenna, enabling the separation of the planes, with Wi-Fi serving the data plane (wireless clients) and ESP-NOW for the control plane (IoT devices).

The **dual data plane** is constructed atop a mesh network, where each device forwards data to the next one until it reaches a gateway (an access point or router). This implementation was necessary because native routing cannot be set up directly on the device (Espressif, 2023a) (Espressif, 2020a) (Espressif, 2015). By this, it is meant that the devices select the next hop by signal strength. The routing is handled simply by the inherent features of a NAT device. As such, it uses a NAT mechanism to forward traffic (Gergeleit, 2019). Implementing NAT in this specific context has performance costs, especially for devices closest to the gateways due to high traffic volumes and the number of dependent devices. Consequently, a restriction imposed by the manufacturer only allows up to 17 connections (Espressif, 2024d) to be connected to each network node. This value is further reduced to 8 devices per interface due to the requirement of two virtual interfaces per node (6 devices and 2 nodes per interface), and this can be rearranged according to the necessities up to 8 slots per interface.

For the **control plane**, several technologies and protocols are suitable for IoT communications, including ZigBee, Thread, BLE, HaLow, and ESP-NOW. Although all these protocols could theoretically implement a wireless communication architecture, only ESP-NOW fully meets the requirements for this specific use case. See section 2 for further details.

The device broadcasts a packet every 10 seconds to announce its presence to other nearby devices. There are two packet types: a search packet and a keep-alive packet, and both currently transmit identical information, differ only in retransmission intervals. The keep-alive packet having the longest interval. When a neighbouring device receives a packet, it checks whether the sender is already in the cached neighbours list, if it exists, its dead timer is set to 0, or else it increments.

In addition to the aforementioned, the device also scans periodically for Wi-Fi access points in the vicinity and then selects the nearest gateway/mesh. Before connecting to any node (device or gateway), a false positive check is performed three times, and if validated, the device attempts the connection, otherwise, it attempts on another node. The process repeats every 10 seconds for unconnected nodes and 30 seconds for connected ones. These values were chosen as default settings to balance responsiveness and resource usage. While they have not been experimentally validated, they can be adjusted based on specific application needs in future work.

6 RESULTS

The prototype testing aimed to record the overall network uptime, from boot to full operational mode, and assess stability and performance using tools like iperf3, hping3, nuttcp, and ping. The tests were conducted using four devices spaced between 1 and 15 metres apart (1, 5, 10, and 15 metres), covering up to 50 metres in a linear hop-by-hop mesh configuration. While this approach allowed for straightforward evaluation of individual hops, it does not fully exploit the resilience and redundancy typically associated with mesh networks. In this setup, failure of an intermediary node disrupts service for subsequent nodes. Although the development of the project used a full mesh topology, the testing phase employed a hop-by-hop NAT configuration for practical evaluation of individual node performance. The same mesh code was utilised in the test environment, ensuring consistency between the mesh architecture and the test configuration. Future testing should investigate the use of a mesh topology with more devices.

To streamline test execution, a Python script was developed, incorporating both client and server requirements in separate threads. The script's server mode provides an HTTP server to monitor device readiness in the network, a Mosquitto server for result sharing, a nuttcp server to test throughput, and an iperf3 server for bandwidth and data loss analysis. Figure 4 illustrates the setup and the communication between them, forwarded by the Espressif devices. All devices have a default route to ensure connectivity to the server and the Internet.

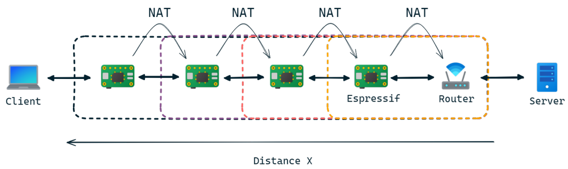

Figure 4: Arrangement of test equipment.

Table 5 summarises the results of the network setup readiness and convergence tests, measuring uptime from a cold boot at various distances. Each network configuration was tested 10 times with different device quantities, totalling 40 tests (4 devices × 10 tests each). The overall average uptime across all distances was 28.75 seconds.

Table 5: Average uptime submitted by the devices.

Devices \Distance	1m	5m	10m	15m
Device 1	30s	30s	30s	30s
Device 2	26s	26s	26s	28s
Device 3	29s	30s	51s	26s
Device 4	50s	50s	148s	28s

Table 6 displays the response times and packet loss rates measured by the ping tool, which gradually increase with the distance between devices. Nuttcp tools in 7, shows that results degrade as the distance increases, leading to higher response times. This results in longer Round-Trip Times (RTT), reduced throughput, and increased packet loss. Retransmissions were not observed. All the results are a compilation of 10 tests. The iperf3 results in table 8 also highlight the trend of significant performance degradation with increasing distance. However, when using UDP, the connection maintains a stable rate of 1.05 Mbits/sec.

Table 6: Ping tool results.

Ping \Distance	1m	5m	10m	15m
Minimum	6.982ms	10.859ms	8.834ms	6264.071ms
Average	17.482ms	33.028ms	18.113ms	7078.394ms
Maximum	51.641ms	130.871ms	42.200ms	8609.763ms
Packet Loss	0%	0%	0%	50%

Table 7: Nuttcp results.

Nuttcp \Distance	1m	5m	10m	15m
TCP Throughput	6.7604 Mbps	2.1619 Mbps	0.0524 Mbps	-
TCP RTT	7.98ms	10.59ms	154.43ms	-
TCP Retrans	0	0	0	-
UDP Throughput	1.0002 Mbps	1.0012 Mbps	0.2756 Mbps	0.0164 Mbps
UDP Packet Loss	0%	0%	66.18%	98.69%

Table 8: Iperf3 TCP, UDP, retries and jitter results.

iperf3 \Distance	1m	5m	10m	15m
TCP	7.15 Mbits/sec	2.25 Mbits/sec	78.6 Kbits/sec	-
UDP	1.05 Mbits/sec	1.05 Mbits/sec	1.05 Mbits/sec	-
Retry	17	9	2	-
Jitter	1.096 ms	3.996 ms	-	-

7 DISCUSSION

The prototype demonstrates the use of a single antenna to accommodate the two data planes and a control plane. The antenna is used for three roles: IoT client, access point for IoT end devices, and management network. While the client and access point rely on Wi-Fi 6, the management network uses ESP-NOW. The use of a single-antenna design aims to deliver functionality for existing devices at a low cost. Multiple antennas could improve system performance, but may shift the focus from simple, low-cost solutions to complex implementations.

A concern on the prototype is the use of NAT in the Lightweight TCP/IP (LwIP) stack that does not support creating a network bridge, and as a solution NAT was used instead, resulting in an inherent ”delay” when many client devices are present in the network, as it works like a funnel. While there is an Ethernet Network Interface Card (NIC) based solution (Espressif, 2023b), the LwIP stack currently only supports Ethernet interfaces for bridging. Each node is limited to 8 connected devices (Espressif, 2024e) an can be configured up to 17 different devices.

The use of NAT was necessitated by the lack of support for direct interface bridging in the current development framework. While NAT offers advantages such as simplified route creation and privacy protection, its inherent limitations, including increased latency and reduced number of connected neighbors, highlight the need for future code framework improvements to support direct bridging for enhanced performance. However these inherent limitations do not affect the scalability of the mesh network.

The testing revealed coverage limitations possible due to a single antenna, IoT nature, and experimental software/hardware. The best stable outcome was achieved with devices with 10 metres, and reaching up to 50 metres for a usable network. The data shows that device uptime is relatively stable around 30 seconds, with constraints based on the connection time from lower layers. If layer 3 devices cannot boot due to layer 2 devices being inactive, subsequent layers wait for predefined time intervals to recheck and establish connections. Performance degradation beyond 10 metres is likely attributed to a combination of factors, including the prototype's development stage, lack of external antennas, unoptimised code, and the simultaneous use of multiple planes with a single antenna attempting concurrent operations. Addressing these hardware and software limitations could significantly enhance range and stability.

8 CONCLUSION

The paper presents a dual data plane architecture and functional prototype enabling new and existing IoT devices to establish their own mesh network while separating data streams. It achieves this by allowing devices to communicate via ESP-NOW for settings adjustments and decision-making, thereby fostering autonomous operation within the network, while the Wi-Fi planes handle IoT functionality and Wi-Fi access to end devices. The architecture supports concurrent operation of Wi-Fi 6 and ESP-NOW on ESP32-C6 (v0.0) devices, utilising a single 2.4 GHz interface. This setup enables devices to manage control independently without requiring an SDN Controller or central device, though it remains compatible with such configurations if needed.

Future work should focus on seamless integration across multiple vendors and broader testing under varied conditions. While this work does not encompass a detailed security analysis, future iterations could include exploring security challenges specific to ESP-NOW and mesh architectures. The integration of artificial intelligence techniques and a SDN controller can improve network performance through data-driven optimisation, failure prediction, and real-time parameter adjustments. Code revisions can improve environment scanning, routing, and updating neighbour statuses. A study should investigate efficiency challenges and security aspects in communications, particularly in the ESP-NOW protocol.

ACKNOWLEDGEMENTS

This work was funded by FCT – Fundação para a Ciência e Tecnologia, I.P. under the project UIDB/04524/2020.

REFERENCES

Analytics, I. (2024). State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally. [Online; accessed 8. May 2024].

Bisdikian, C. (2001). Bluetooth Technology Overview.

Eridani, D., Rochim, A. F., and Cesara, F. N. (2021). Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance. In *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 322–328.

Espressif (2015). Routing/bridging - esp32 forum.

Espressif (2020a). Layer 2 bridging - esp32 forum.

Espressif (2020b). Usage ble and wi-fi at the same program.

Espressif (2023a). Bridging wifi softap with ethernet - esp32 forum.

Espressif (2023b). esp-idf/examples/network/bridge at feature/lwip_bridge_w_wifi · kostaond/esp-idf.

Espressif (2023c). Esp-now - esp32-c3 - — esp-idf programming guide latest documentation.

Espressif (2023d). Esp-now wireless communication protocol — espressif systems.

Espressif (2023e). Esp32-c6 wi-fi 6 & ble 5 & thread/zigbee soc — espressif systems.

Espressif (2023f). Openthread border router.

Espressif (2023g). Rf coexistence - esp32-c6.

Espressif (2024a). esp-idf/examples/mesh at master · espressif/esp-idf. [Online; accessed 11. Jul. 2024].

Espressif (2024b). ESP-WIFI-MESH - ESP32 - — ESP-IDF Programming Guide v5.2.2 documentation. [Online; accessed 11. Jul. 2024].

Espressif (2024c). Esp32-c6-devkitc-1 user guide. [Online; accessed 10. Jul. 2024].

Espressif (2024d). Wi-Fi Driver - ESP32 - — ESP-IDF Programming Guide v5.2.1 documentation. [Online; accessed 10. May 2024].

Espressif (2024e). Wi-Fi Driver - ESP32-C6 - — ESP-IDF Programming Guide latest documentation. [Online; accessed 11. Jul. 2024].

Ferro, E. and Potorti, F. (2005). Bluetooth and wi-fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, 12(1):12–26.

Flayols, T. (2024). Linux-ESPNOW. [Online; accessed 8. May 2024].

FreeRTOS (2024). Freertos - market leading rtos (real time operating system) for embedded systems with internet of things extensions.

Gergeleit, M. (2019). Autotree: Connecting cheap iot nodes with an auto-configuring wifi tree network. *2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019*, pages 199–203.

Islam, G. Z. and Kashem, M. A. (2022). A proportional scheduling protocol for the ofdma-based future wi-fi network. *J. Commun.*, 17(5):322–338.

Kaljic, E., Maric, A., Njemcevic, P., and Hadzialic, M. (2019). A Survey on Data Plane Flexibility and Programmability in Software-Defined Networking. *IEEE Access*, 7:47804–47840.

Karamyshev, A., Liubogoshchev, M., Lyakhov, A., and Khorov, E. (2024). Enabling industrial internet of things with wi-fi 6: An automated factory case study. *IEEE Transactions on Industrial Informatics*, 20(11):13090–13100.

Khanchuea, K. and Siripokarpirom, R. (2019). A multi-protocol iot gateway and wifi/ble sensor nodes for smart home and building automation: Design and implementation. *10th International Conference on Information and Communication Technology for Embedded Systems, IC-ICTES 2019 - Proceedings*.

Khanna, A. and Kaur, S. (2020). Internet of things (iot), applications and challenges: A comprehensive review. *Wireless Personal Communications*, 114(2):1687–1762.

Manvi, M. M. and Maakar, M. S. K. (2020). Implementing wireless mesh network topology between multiple wi-fi powered nodes for iot systems. *International Research Journal of Engineering and Technology*, 7(10):1242–1244.

Mozaffariahrar, E., Theoleyre, F., and Menth, M. (2022a). A Survey of Wi-Fi 6: Technologies, Advances, and Challenges. *Future Internet*, 14(10):293.

Mozaffariahrar, E., Theoleyre, F., and Menth, M. (2022b). A Survey of Wi-Fi 6: Technologies, Advances, and Challenges. *Future Internet*, 14(10):293.

Muhendra, R., Rinaldi, A., Budiman, M., and Khairurrijal (2017). Development of wifi mesh infrastructure for internet of things applications. *Procedia Engineering*, 170:332–337.

Oughton, E. J., Lehr, W., Katsaros, K., Selinis, I., Bubley, D., and Kusuma, J. (2021). Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6. *Telecommunications Policy*, 45(5):102127.

Perez-Ramirez, J., Seijo, O., and Val, I. (2022). Time-Critical IoT Applications Enabled by Wi-Fi 6 and Beyond. *IEEE Internet of Things Magazine*, 5(3):44–49.

Thread Group (2020). [accessed 24. May 2024].

Thread Group (2024). Thread Benefits. [Online; accessed 24. May 2024].

Zegeye, W., Jemal, A., and Kornegay, K. (2023). Connected smart home over matter protocol. *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–7.

ZigBee Alliance (2017). Zigbee green power.

Zigbee2MQTT (2024). Improve network range and stability — zigbee2mqtt.