

# Multi-Mesh: A Miniaturized Multi-Radio WLAN Mesh Testbed

Tim Brockmann , Michael Rethfeldt , Benjamin Beichler ,  
Frank Gولاتowski , Christian Haubelt 

*Institute of Applied Microelectronics and Computer Engineering, University of Rostock, Germany*  
Email: firstname.lastname@uni-rostock.de

**Abstract**—As Industry 4.0 transforms industrial automation, the need for reliable, low-latency wireless communication is paramount. This paper presents Multi-Mesh, a miniaturized multi-radio IEEE 802.11s testbed, designed to address key challenges in industrial wireless networks. The testbed supports simultaneous data transmission across orthogonal channels, improving throughput and communication reliability. A key focus of this work is on experimental validation of advanced multi-path communication strategies and time synchronization mechanisms, both critical for robust industrial communication. By leveraging multiple radios, the testbed enables flexible exploration of redundancy and fault tolerance in mesh networks. Through comprehensive testing, the study demonstrates the testbed’s capability to evaluate and optimize future approaches to synchronization and multi-path strategies, providing a strong foundation for enhancing communication in Industry 4.0 environments.

**Index Terms**—WLAN mesh network, IEEE 802.11s, testbed, multi-radio, time synchronization

## I. INTRODUCTION

The rapid advancement of Industry 4.0 technologies has driven the need for robust, reliable, and low-latency wireless communication systems in industrial environments. As factories and industrial plants become increasingly automated and interconnected, the demand for networks that can support high-throughput, time-sensitive applications has never been greater. Wi-Fi mesh networks, particularly those based on the IEEE 802.11s (.11s) standard amendment, have emerged as a promising solution to meet these demands due to their scalability, flexibility, and inherent fault tolerance [1–3].

However, traditional single-radio mesh networks face significant challenges in environments where high data throughput and low latency are critical. To address these challenges, this paper introduces Multi-Mesh, a miniaturized multi-radio .11s mesh testbed designed to develop and evaluate approaches for communication reliability and efficiency in industrial scenarios. By leveraging multiple radios, the testbed allows for simultaneous data transmission across orthogonal channels, significantly improving network performance.

The Multi-Mesh testbed is designed to replicate real-world industrial environments, offering a versatile platform for evaluating network performance under various conditions. The testbed is particularly suited for scenarios requiring mobile lab deployments, ensuring that it can be applied to a wide range of industrial use cases. The system is constructed on

a cost-effective, extendable hardware platform with commercially off-the-shelf (COTS) components, ensuring adaptability and scalability for evolving needs. Its software platform is highly configurable, allowing for extensive customization of operating systems and drivers.

In industrial wireless networks, achieving precise time synchronization remains a significant challenge, particularly in mesh topologies where nodes frequently join and leave the network. The Multi-Mesh testbed is specifically designed to address this challenge by providing a platform for the development and testing of advanced time synchronization mechanisms. Additionally, its support for multi-path communication enhances network redundancy and reliability by enabling simultaneous data transmission over multiple paths.

## II. WI-FI MESH NETWORKS AND IEEE 802.11S

Wireless Mesh Networks (WMNs) are decentralized networks in which devices connect dynamically, forming a flexible and self-healing communication infrastructure [4]. Unlike traditional wireless networks, WMNs do not rely on a central router; instead, each node acts as both a host and a router, forwarding data for others. This multi-hop communication provides significant advantages in areas where setting up wired infrastructure is impractical or costly. WMNs are widely used in scenarios like disaster recovery, rural internet access, and urban sensor networks, offering high reliability and scalability through distributed routing and fault-tolerance.

Ratified in 2011, the IEEE 802.11s standard amendment established the first formalized framework for WMNs, built on the widely used IEEE 802.11 specifications [5]. It enables vendor-independent, infrastructure-less multi-hop communication by embedding essential mesh functions, such as routing and peering, directly into the MAC layer.

This integration represents a significant improvement over earlier, non-standardized mesh protocols that operated at the network layer. To ensure interoperability, every IEEE 802.11s node must support the Hybrid Wireless Mesh Protocol (HWMP) and Airtime Link Metric (ALM). HWMP, which is derived from the Ad-Hoc On-Demand Distance Vector (AODV) routing protocol, allows for reactive path discovery, creating routes only when needed. Each node maintains links to neighboring nodes within radio range and paths to destinations only when communication has been initiated.

Linux has been a leading platform for implementing and testing IEEE 802.11s, offering one of the most advanced open-source implementations of this standard. The `mac80211` kernel module provides the backbone for mesh networking, handling essential MAC-layer operations. This module is paired with soft MAC drivers such as `ath9k`. The use of soft MAC drivers ensures that most of the mesh functionality is handled in software, offering greater flexibility for network configurations.

Configuration and monitoring of mesh networks on Linux are realized by the `cfg80211` kernel module, which allows userspace applications to interact with the MAC layer via the Netlink socket interface [6]. The popular `iw`<sup>1</sup> tool leverages this interface, enabling users to manage physical WLAN devices, set mesh parameters like mesh IDs and channels, and inspect link and path lists in real time.

### III. RELATED WORK

In recent years, differnet works have focused on the miniaturization of WLAN mesh testbeds.

One significant contribution is ScaleMesh [7], which presents a down-scaling approach for WMNs by using variable antenna attenuators. While it already evaluates the use of two radios, it only supports and analyzes the IEEE 802.11b/g technology and does not utilize .11s mesh networking.

Another significant work is Mini-Mesh [2], which is based on the down-scaling approach presented by ScaleMesh and further extends it by a combination of antenna attenuation and configurable transmission power. In contrast to ScaleMesh, the Mini-Mesh testbed only features a single-radio setup, which, however, supports the IEEE 802.11n and .11s technology.

The Multi-Mesh testbed, proposed in this paper, builds on these existing works by introducing a miniaturized, multi-radio .11s platform specifically tailored for the prototyping of industrial applications. By leveraging lessons learned from previous research, the Multi-Mesh testbed aims to address the unique requirements of Industry 4.0. Building on a more powerful hardware platform with multiple radios, sophisticated antenna positioning and interference mitigation concepts, and a comprehensive orchestration and deployment framework, Multi-Mesh will serve as foundation to develop and test optimization strategies for robust time synchronization, high throughput, and low latency in wireless industrial environments.

Table I gives an overview of the related works. Besides ScaleMesh and Mini-Mesh, there is further work that presented miniaturized .11s mesh testbeds [8, 9]. However, these only supported one radio, featured a much smaller number of nodes and had a lower degree of miniaturization, as only the transmission power of the devices was varied.

### IV. MULTI-MESH: TESTBED PREPARATION

In this section, we introduce our miniaturized .11s testbed, named Multi-Mesh, as illustrated in Figure 1. The primary

<sup>1</sup><https://wireless.wiki.kernel.org/en/users/documentation/iw>

TABLE I: Comparison of related work  
(AA: Attenuated Antennas, TP: Transmit Power,  
R: Radios, N: Nodes)

Works	.11 support	#R	#N	Scaling Method
ScaleMesh [7]	no	2	20	AA
Pojda et al. [8], Hayat et al. [9]	yes	1	3-6	TP
Mini-Mesh [2]	yes	1	36	AA + TP
Multi-Mesh	yes	2	25	AA + TP

goals of our approach focus on creating a real-world testbed that closely mimics practical conditions. Additionally, we provide robust support for .11s conform multi-radio configurations. A key capability of the system is its ability to handle simultaneous transmissions on orthogonal channels within the same frequency band, improving overall network performance. The testbed is versatile, designed as a mobile laboratory environment, while also applying to industrial scenarios. We emphasize high reproducibility of experiments to ensure consistent results, and the use of cost-effective hardware to make the system accessible without compromising functionality. Furthermore, the system supports a wide variety of network interface cards (NICs) and includes configurable software, such as customizable operating systems and drivers, to cater to specific needs.

#### A. Choice of Hardware and Software Components

The Multi-Mesh testbed utilizes 25 nodes based on PC Engines APU 2E2<sup>2</sup>, an established hardware platform for research and prototyping of network infrastructure. The hardware and software configuration of the nodes is shown in Table II. The APU 2E2 platform features multiple Gigabit Ethernet interfaces as well as multiple mPCIe interfaces, enabling the use of a wide range of wireless NICs. We chose to equip each device with two Atheros AR9280 NICs, supporting the IEEE 802.11n standard and IEEE 802.11s mesh networking under Linux. Each node is running Kali Linux OS<sup>3</sup> (version 2023.3) with a recent mainline Linux kernel (v6.6.0+).

The Atheros NIC chipset uses the `ath9k` driver, offering open-source flexibility and allowing deep control over the card's functionality. As a SoftMAC driver, `ath9k` relies on the `mac80211` subsystem for MAC-layer functionality and `cfg80211` for regulatory compliance and configuration management. These Linux subsystems provide a flexible framework for low-level control of wireless operations, including features like packet injection, monitor mode, and precise packet management, making it ideal for advanced experimentation and network performance analysis.

Due to its flexibility, the `ath9k` driver often serves as basis for research projects. For example, WMNs are a promising technology for applications ranging from smart infrastructure to large-scale industrial IoT systems, which often exhibit timing requirements. However, time synchronization, a fundamental component for reliable network operation, remains

<sup>2</sup><https://www.apu-board.de/produkte/apu2e2.html>

<sup>3</sup><https://www.kali.org/>

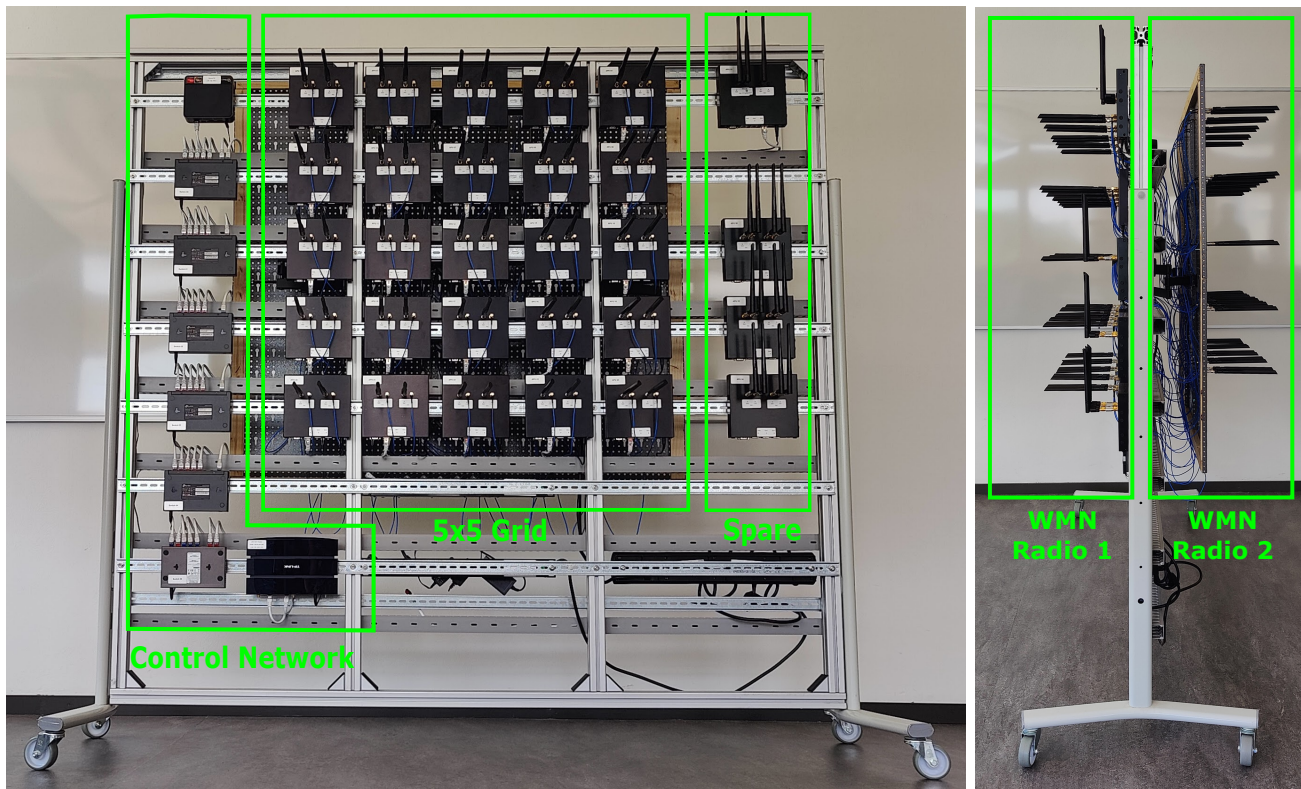


Fig. 1: Front and side views of the Multi-Mesh testbed

underexplored in the context of WMNs. One of our research goals is to explore time synchronization strategies in WMNs using, among others, the *Wi-PTP*<sup>4</sup> project [10]. This project comprises extensions for the *ath9k* driver to allow the usage of the Precision Time Protocol (PTP) with hardware timestamping in Wi-Fi networks. So far, its performance in WMNs has not been investigated.

Moving forward, we aim to deepen our investigation using the new testbed designed specifically for evaluating synchronization in real-world WMN environments. The new testbed will allow us to simulate and analyze various network conditions, node densities, and topological configurations, enabling a more comprehensive assessment of time synchronization challenges in WMNs. The multi-hop characteristics and dynamic nature of these networks requires adaptive synchronization mechanisms that go beyond existing solutions like *Wi-PTP*.

### B. Testbed Geometry and Miniaturization Concept

As can be seen in Figure 3, each wireless NIC is equipped with two 5 dBi omni-directional dual-band antennas, connected via 20 cm internal U.FL to RP-SMA pigtailed. The antenna pairs on the back of the testbed are connected with additional 100 cm long RG405 coaxial cables. The antennas can operate in both the 2.4 GHz and 5 GHz frequency bands, though for this experiment, we have opted to focus on the less congested 5 GHz

band. Specifically, we operate on the channels 149–161, which are generally underutilized in Europe, thus minimizing interference and optimizing data throughput. The system supports operation in both diversity and spatial multiplexing modes of Multiple-Input Multiple-Output (MIMO) communication, utilizing two receive (RX) and two transmit (TX) chains, with the capability of processing up to two spatial streams.

In order to reduce the range and achieve the associated miniaturization of the experimental setup, we adopted the range scaling approach of *Mini-Mesh*[2]. Due to the limited space available in the laboratory, we opted for a vertical setup. Furthermore, to allow for flexible use of the testbed in different environmental conditions (e.g., with or without presence of surrounding wireless networks) and for different purposes (research/student projects, demonstrations, teaching) we opted for a rollable mobile rack installation. To ensure that the setup fits our required dimensions, we chose an inter-node distance of 20 cm vertically and horizontally and, as a result, 33 cm diagonally (see Figure 2).

Analogous to the work *Mini-Mesh*[2], the combination of fixed antenna attenuators and configured transmission power required to be able to use the entire range of IEEE 802.11n data rates (MCS0-7) over the specified node distance was determined experimentally. This resulted in the choice of a fixed antenna attenuation of 20 dB. In conjunction with the configurable transmission power of the WLAN cards, suitable operating points can be found for stable data transmission

<sup>4</sup><https://github.com/zlab-pub/wifi-ptp>

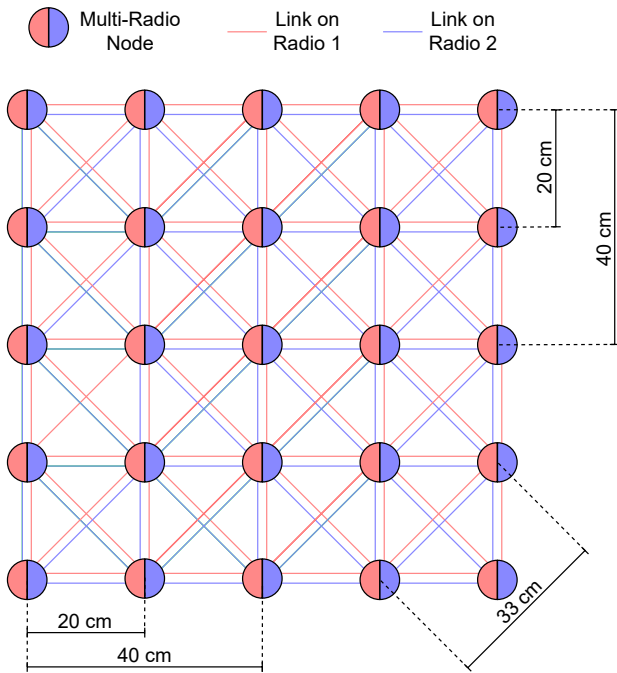


Fig. 2: 5x5 node mesh grid geometry

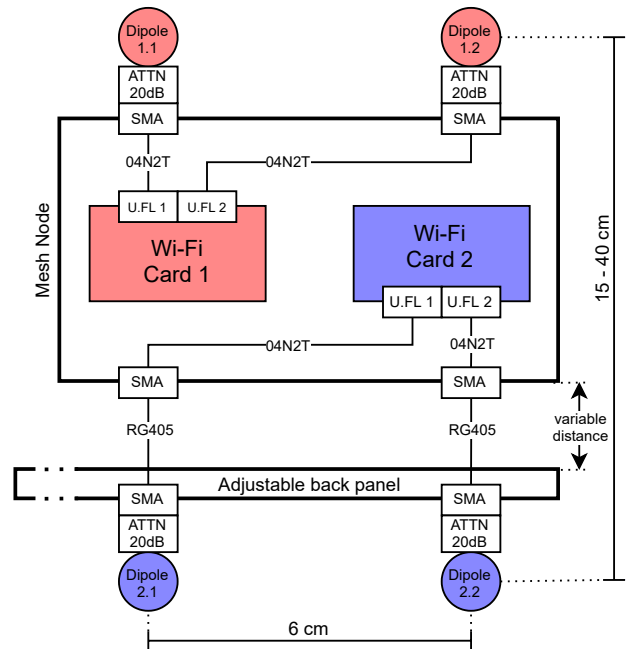


Fig. 3: Radio installation and spatial antenna separation (ATTN: Antenna Attenuator, U.F.L/SMA: Coaxial Connectors, 04N2T/RG405: Coaxial Cables)

between neighboring grid nodes. The `ath9k` driver allows the channel-dependent configuration of the transmission power between 0 dBm (1 mW) and 17 dBm (50 mW) via the Linux `debugfs` interface.

### C. Antenna Positioning, Shielding and Interference Mitigation Concepts

In our mesh testbed, we observed that without additional shielding, the attenuated signals were highly distorted at the radios, leading to significant reductions in data throughput and frequent packet drops. This issue arises primarily from electromagnetic interference (EMI) at critical points, particularly at the pigtail connectors. As explained in [11], shielding is only effective if properly grounded, which is often insufficient in high-frequency environments like Wi-Fi networks. To address this, we have implemented a technique inspired by [12], covering the pigtail connectors with additional copper foil, as can be seen in Figure 4 (both cards are covered in

the final setup). This method ensures continuous shielding at these vulnerable points, preventing the connectors from emitting too much unwanted radiation. Unshielded pigtails can interfere with Wi-Fi energy detection, resulting in unwanted carrier sense behavior and network degradation. By applying copper foil, we minimize leakage, improving electromagnetic compatibility (EMC) and enabling the independent operation of both radios on orthogonal channels. In conjunction with carefully spaced antennas (see Figure 1 and 3), as suggested in [13], this approach has significantly improved simultaneous data transmission within the same frequency band, reducing interference and packet loss.

The installation, as illustrated in Figure 3, shows the red and blue antenna pairs, each corresponding to a separate Wi-Fi card. The red antennas are installed on the front panel of the testbed, while the blue antennas are routed via 1-meter-long RG405 cables to a back panel. This setup ensures the same 6 cm antenna spacing for both pairs, matching the wavelength in the 5 GHz band. The identical geometry between the front and back panels makes the radio installations fully comparable. Additionally, this configuration allows us to vary the distance between the antenna pairs of all nodes between 15 cm and 40 cm, providing flexibility for testing different interference and performance scenarios. This precise



Fig. 4: Pigtail copper shielding

TABLE II: Testbed Configuration

Parameter	Value
Device	PC Engines APU 2E2
CPU	AMD GX-412TC (Quad-Core 1 GHz)
RAM	2 GB DDR3-1333 DRAM
OS	Kali Linux 2023.3 (Kernel v6.6.0+)
.11 NIC	Compex WLE200NX .11a/b/g/n
NIC Chipset	Atheros AR9280
Antennas	5 dBi Dual-Band Omni-Directional
Ant. Cables int.	20 cm U.F.L - RP SMA
Ant. Cables ext.	100 cm RG405 RPSMA - RPSMA
Attenuators	2 W SMA DC-6 GHz (20 dB)
Channels	149/161 (5745/5805 MHz, HT40+, Short GI)

arrangement minimizes interference and optimizes signal reception and transmission.

#### D. Testbed Control Network

The Ethernet-based control network of the testbed, as depicted in Figure 5, is designed to provide centralized control and management of all nodes within the testbed environment. The control network operates independently of the mesh network, ensuring a reliable and isolated infrastructure for managing testbed operations. The star topology ensures that the row switches can communicate directly with the central testbed server, allowing for streamlined control and data aggregation. A dedicated NAT (Network Address Translation) router provides connectivity between the testbed server and the wider university network, enabling external access as needed for diagnostics, configuration, and data collection. Additionally, a diagnostic computer is connected to the same NAT router, facilitating monitoring and control tasks, such as running experiments, collecting logs, and performing system diagnostics.

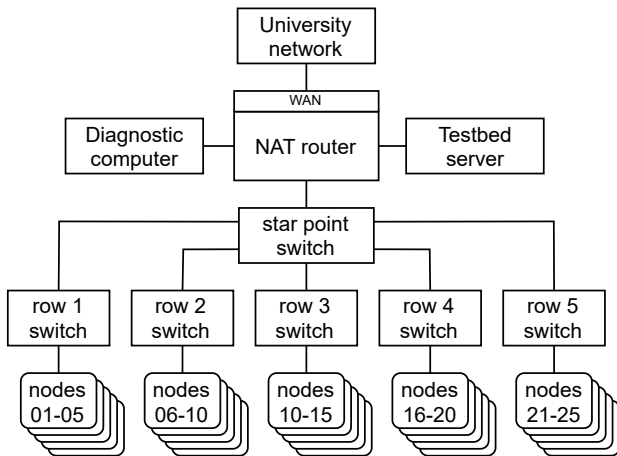


Fig. 5: Topology of the Ethernet based control network

### V. TESTBED ORCHESTRATION

This section is dedicated to the software and hardware management of the testbed and gives an outlook on the opportunities that this testbed offers.

#### A. Node Software Deployment

The primary operating system used on the SSDs of the Multi-Mesh testbed nodes is Kali Linux. We decided in favor of Kali Linux because it offers very extensive driver support, many network tools and a high degree of flexibility. The OS is customized to support the specific requirements of the testbed, including .11s mesh networking. We used our local GitLab DevOps infrastructure server to organize the deployment and development of the customized Linux kernel used in the testbed. In doing so, we automated the tasks for compilation and testing in a CI/CD pipeline that outsourced the building process, enabling a rapid and reliable development process for Linux kernel modifications. By relieving the strain of

manual intervention and utilizing external server capacities, we improved the efficiency of the development cycle and allowed for faster iteration and evaluation of different kernel modifications in controlled environments.

#### 1) PXE Boot and Backup Process

To facilitate efficient software deployment and updates, a PXE (Preboot Execution Environment) is set up. Initially, the nodes boot via PXE, retrieving a boot script and a TinyCoreLinux<sup>5</sup> RAM disk image (v14.0) from the testbed server as shown in Figure 6. Once booted into TinyCoreLinux, the system adjusts the time using NTP (Network Time Protocol) and creates a local log file. An iterative backup process is then initiated using `rsync`<sup>6</sup>. Depending on the backup plan, the process may involve partitioning the SSD, formatting partitions, synchronizing the remote backup, updating the kernel, or change the firmware of the system board. This backup plan ensures that the latest configurations, software updates, and necessary files are synchronized from a central testbed server to the nodes' SSDs. Importantly, even a node with a completely blank SSD can be fully initialized and configured using this backup system. Additionally, any node in the network can be defined as a reference node to update the remote backup, ensuring flexibility and redundancy in managing the system's configuration. After completing the backup and update tasks, the nodes push the console output log to the testbed server and await a reboot command via SSH. This method ensures that all nodes are consistently updated and can be quickly reconfigured as needed (see Figure 6). Once the backup process is complete, the PXE configuration can be switched to a local boot and all nodes will boot from their local Kali Linux installation.

#### 2) Testbed Monitoring

To monitor the node performance and the network health, an integrated system is designed using MQTT (Message Queue Telemetry Transport), Node-RED<sup>7</sup>, InfluxDB<sup>8</sup>, and Grafana<sup>9</sup> as shown in Figure 7. Python scripts utilizing the PahoMQTT<sup>10</sup> library facilitate the transmission of node data to a central EclipseMosquitto<sup>11</sup> MQTT broker, which serves as a message hub, allowing all testbed nodes to publish performance metrics like CPU usage, memory utilization, and network traffic to designated topics. Node-RED is employed to subscribe to the MQTT topics, where it processes incoming data and forwards it to the storage layer. This processing involves filtering and formatting the data for compatibility with InfluxDB, a time-series database that efficiently stores the performance metrics for long-term analysis. The stored data in InfluxDB is then visualized using Grafana, which offers real-time dashboards and historical insights into system performance.

<sup>5</sup><http://tinycorelinux.net/>

<sup>6</sup><https://github.com/RsyncProject/rsync.git>

<sup>7</sup><https://nodered.org/>

<sup>8</sup><https://www.influxdata.com/products/influxdb/>

<sup>9</sup><https://grafana.com/>

<sup>10</sup><https://pypi.org/project/paho-mqtt/>

<sup>11</sup><https://mosquitto.org/>

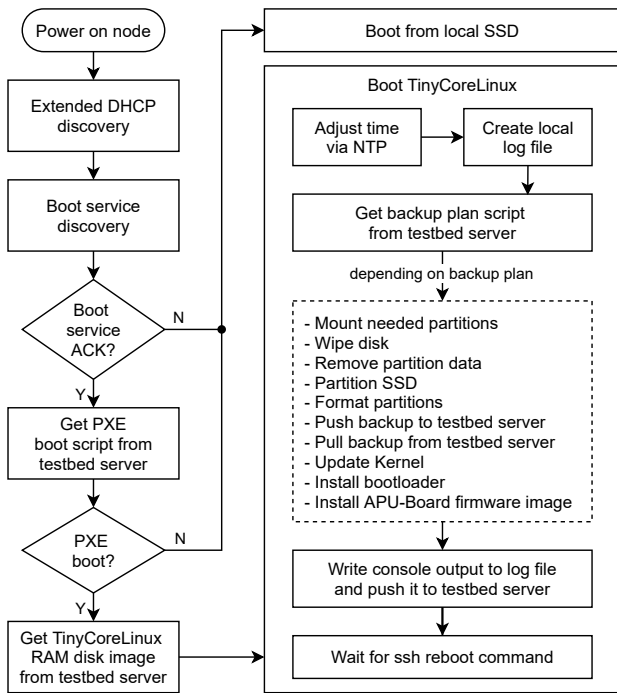


Fig. 6: Mesh node boot procedure

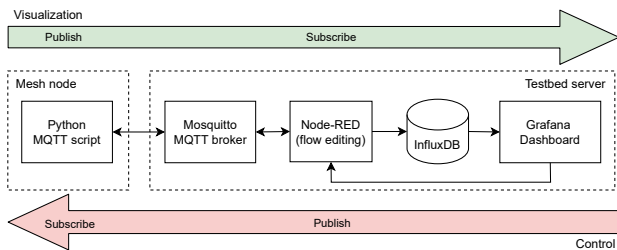


Fig. 7: Remote configuration and monitoring

## B. Configuration and Management

During the local boot process, the network configuration is downloaded from the testbed server. This configuration sets up the nodes' network interfaces to support multi-radio operation, allowing them to communicate simultaneously across multiple channels. Once the network configuration is applied, the .11s mesh network is dynamically formed, with nodes automatically establishing and maintaining a mesh topology. This setup involves defining mesh points, assigning mesh IDs, and configuring multiple parameters such as TX power, the allowed modulation and coding schemes (MCS), and potential traffic monitoring functions. In the next steps, the infrastructure described in Section V-A2 should also be used for network configuration. For this, the respective Grafana dashboards need to be adapted to transmit MQTT messages. These messages will then be received by the nodes, triggering the appropriate configurations. This could, for example, involve blocking links to neighboring nodes, which may lead to different path selections by other nodes.

## C. Persistent Peer Link Blocking

The need for a more reliable and controlled network topology in our wireless mesh network testbed drives the adoption of an enhanced MAC address filtering approach [14]. This approach is essential for mitigating the risks of unstable connections and unwanted link establishments in dynamic environments. Wireless mesh networks (WMNs), while promising due to their self-healing and robust nature, are susceptible to unpredictable link establishments, especially in dynamic environments. This can lead to unstable connections, oscillating link states, and potential security vulnerabilities. The existing Linux implementation of .11s provides only basic mechanisms for link blocking, which are insufficient for guaranteeing consistent topology control. By extending these capabilities, the approach we use ensures that unwanted connections are reliably blocked, thereby improving network stability and performance. This is particularly critical in scenarios requiring precise control over network behavior, such as in experimental testbeds or security-sensitive applications [15]. Ultimately, the enhanced peer-link blocking method supports more consistent and reproducible testing conditions, which are essential for evaluating new networking concepts and ensuring controlled network topologies for reproducible experiments.

## D. Deployment Verification and Testing

After software deployment, verification and testing procedures are conducted to ensure that each node running Kali Linux is correctly configured and operational. This includes network connectivity tests using the tool `ping`, time synchronization checks, and performance benchmarking. Any issues identified are resolved through iterative reconfiguration using the PXE environment and `rsync` for efficient updates. This process ensures that the testbed is fully functional and ready for reliable operation in experimental scenarios.

## VI. EXPERIMENTAL VALIDATION

### A. Testbed Configuration

The experimental testbed is comprised of 25 wireless nodes arranged in a 5 x 5 grid. Each node is equipped with two Wi-Fi radios, both of which have two antennas operating in a diversity MIMO configuration to enhance signal reliability. To control signal strength and mitigate interference between neighboring nodes, each radio is equipped with fixed 20 dB antenna attenuators.

The antennas placed at the front and back of the testbed form two independent mesh networks. To reduce the interference effects of both networks, we have maximized the distance between the front and back antenna pairs to 40 cm [13]. The networks operate on non-overlapping 5 GHz channels to prevent interference. The front radio uses channel 149 with a center frequency of 5 745 MHz, while the back radio uses channel 161 with a center frequency of 5 805 MHz. Both radios use a 40 MHz channel bandwidth, short guard interval (SGI), and are fixed to MCS6<sup>12</sup>, which provides a good

<sup>12</sup><https://mcsindex.com/>

balance between data rate and robustness by using 64-QAM modulation with a 3/4 coding rate.

In terms of power configuration, we experimentally found that setting the transmission power for both radios to 7 dBm is the best choice, as it provides the ideal balance between transmission range and robust data throughput between neighboring nodes, while minimizing interference between radios and with nodes in multi-hop distance in the same network.

### B. Throughput Measurement

We conducted a throughput measurement method to evaluate the performance of the two independent mesh networks formed using the two Wi-Fi interfaces on each mesh node. The purpose of these measurements was to demonstrate the capability of simultaneous data transmission across both mesh networks, highlighting the potential for parallel communication without significant interference between the two interfaces.

The experiment was designed to assess throughput between various node pairs across different path lengths. To provide a comprehensive evaluation, we measured throughput for all possible node pairs at four distinct hop distances. Initially, throughput was measured between node pairs with a direct, one-hop connection, resulting in 144 node pairs. This was followed by measurements between two-hop node pairs (192 pairs), three-hop node pairs (168 pairs), and four-hop node pairs (96 pairs). These measurements were carefully planned to ensure bidirectional traffic flow between each pair of nodes, capturing performance in both directions across the network.

Sequentially for each node pair, throughput measurements were conducted on both mesh networks simultaneously, using the tool `iperf3`<sup>13</sup>. It is a widely used network measurement tool designed to test the performance of both TCP and UDP connections, providing detailed insights into the communication performance between network nodes. `iperf3` was configured to generate and monitor TCP traffic for a duration of 15 seconds per measurement. This duration was chosen to ensure that each test produced a stable and reliable estimate of the maximum achievable throughput. The tool was set up to fully utilize the link capacity during each measurement, simulating a high-demand traffic scenario.

### C. Results and Analysis

The experiment results are depicted in Figure 8, presenting a box plot for the entirety of TCP throughput measurements for IF1 and IF2, respectively, at each individual hop distance, ranging from 1 hop to 4 hops. The goal of these measurements was to assess the performance of each interface and demonstrate the possibility of parallel data transmission on orthogonal channels.

We utilized MCS6 with a 40 MHz channel bandwidth and short guard interval (SGI), which provides a maximum physical-layer data rate of 135 Mbps. This is the theoretical gross data rate under ideal conditions, assuming no interference and protocol overhead. However, as expected and

illustrated by the box plots, the measured net throughput consistently falls below this theoretical value due to various layers of protocol overhead and real-world limitations.

For example, in the one-hop scenario, which demonstrates the achievable link throughput, the median net throughput for both IF1 and IF2 is approximately 80-85 Mbps. This represents a reduction of around 35-40% compared to the theoretical maximum data rate of 135 Mbps, which was expected and also confirmed by numerous other studies [16]. A portion of this difference is due to TCP/IP overhead, Wi-Fi MAC layer contention, and Wi-Fi management frames, including the messages of the Hybrid Wireless Mesh Protocol (HWMP), i.e., the default routing protocol in .11s mesh networks.

The results confirm the expected trend: as the number of hops increases, the net throughput decreases. Specifically, the throughput is approximately halved when transitioning from one-hop to two-hop node pairs. This behavior can be explained by the nature of mesh routing, where each intermediate node must forward packets, leading to exclusive channel utilization for reception and transmission. Additionally, the phenomenon of hidden nodes between the endpoints of multi-hop paths can exacerbate the throughput degradation, as simultaneous transmissions can cause collisions.

For paths with three or more hops, the throughput reduction does not continue in a strict halving pattern. Instead, the degradation slows down, which can be partially explained by a "pipelining" effect: when the distance between nodes is sufficiently large, it is possible for intermediate nodes to transmit data in parallel. This effect reduces the severity of throughput degradation, preventing a simple exponential drop-off in performance. However, the impact of TCP congestion control and flow control mechanisms further influences the overall throughput. These end-to-end mechanisms can throttle the transmission window, especially over longer paths, as they react to perceived network congestion, packet loss, and increased latency.

The median net throughput for each node pair provides a clear indicator of the network's actual performance. As intended and observable, the achieved throughput for IF1 and IF2 is comparable across all hop distances. This demonstrates that the system effectively supports parallel dual-radio transmission, highlighting the ability to achieve independent data flows on orthogonal channels.

## VII. CONCLUSION AND OUTLOOK

The Multi-Mesh testbed, designed as a miniaturized multi-radio IEEE 802.11s platform, has proven to be an effective tool for evaluating the performance of wireless mesh networks in industrial scenarios. Through rigorous testing, including throughput measurements using `iperf3`, the testbed demonstrated its capability to deliver high data rates over multiple radios and multi-hop paths. The experimental results confirm that multi-radio setups have the potential to significantly enhance the performance and robustness of network applications, making them well-suited for the demands of Industry 4.0 scenarios. The use of commercially off-the-shelf hardware and

<sup>13</sup><https://iperf.fr/>

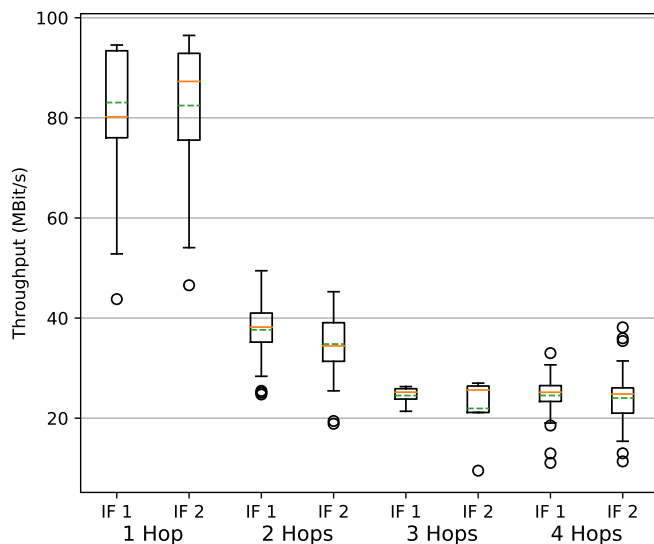


Fig. 8: TCP throughput with different mesh path lengths

a highly configurable software platform further underscores the testbed's practicality and adaptability.

Looking ahead, the Multi-Mesh testbed will serve as a basis for further research into advanced multi-radio mesh networking strategies, e.g., improved time synchronization techniques, and the integration of other wireless technologies currently under research, such as 6G. Additionally, future work will focus on scaling the testbed for larger deployments and investigating its performance in more dynamic industrial environments. The insights gained from this study will contribute to the ongoing development of robust, efficient, and scalable wireless communication systems, ultimately supporting the broader adoption of Industry 4.0 technologies.

#### ACKNOWLEDGMENT

This research has been achieved in the projects CargoAssist (funded by the maritime research program, an initiative of the German Federal Ministry of Economics and Climate Protection (BMWK) under the funding code 03SX601B) and the European ITEA project GenerIoT (funded by the German Federal Ministry of Education and Research (BMBF) under reference number 01IS22084F).

#### REFERENCES

- [1] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock Synchronization Over IEEE 802.11—A Survey of Methodologies and Protocols," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, Apr. 2017.
- [2] M. Rethfeldt, B. Beichler, H. Raddatz, F. Uster, P. Danielis, C. Haubelt, and D. Timmermann, "Mini-Mesh: Practical assessment of a miniaturized IEEE 802.11n/s mesh testbed," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2018.
- [3] M. Rethfeldt, B. Beichler, P. Danielis, T. Brockmann, C. Haubelt, and D. Timmermann, "CHaChA: Clustering Heuristic and Channel Assignment for IEEE 802.11s Mesh Networks," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2018.
- [4] S. M. Taleb, Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Nodes placement in wireless mesh networks using optimization approaches: A survey," *Neural Computing and Applications*, vol. 34, no. 7, Apr. 2022.
- [5] *802.11-2020 - IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks—Specific Requirements - Part 11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) S. S.1.*: IEEE, 2021.
- [6] P. Neira-Ayuso, R. M. Gasca, and L. Lefevre, "Communicating between the kernel and user-space in Linux using Netlink sockets," *Software: Practice and Experience*, 2010.
- [7] S. M. ElRakabawy, S. Frohn, and C. Lindemann, "ScaleMesh: A Scalable Dual-Radio Wireless Mesh Testbed," in *2008 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, San Francisco, CA, USA: IEEE, Jun. 2008.
- [8] J. Pojda, A. Wolff, M. Sbeiti, and C. Wietfeld, "Performance analysis of mesh routing protocols for UAV swarming applications," in *2011 8th International Symposium on Wireless Communication Systems*, Nov. 2011.
- [9] S. Hayat, E. Yanmaz, and C. Bettstetter, "Experimental analysis of multipoint-to-point UAV communications with IEEE 802.11n and 802.11ac," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Aug. 2015.
- [10] P. Chen and Z. Yang, "Understanding PTP Performance in Today's Wi-Fi Networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, Dec. 2023.
- [11] *The myth of shielded cable connections — Spanish Magazine of Electronics — redeweb.com*, <https://www.redeweb.com/en/Articles/components/the-myth-of-shielded-cable-connections-al/>, [Accessed 28-08-2024].
- [12] *EMC drawback of shielded cable pigtail — rfemcdevelopment.eu*, <http://www.rfemcdevelopment.eu/lv/emc-basics/emc-blog/147-emc-pigtail/>, [Accessed 28-08-2024].
- [13] M. Deryng, *WN Blog 022 – Distance Between 802.11 Radios – How Close Is Too Close? - WiFi Ninjas - Podcasts & Blogs — wifininjas.net*, <https://wifininjas.net/2019/12/12/wn-blog-022-distance-between-802-11-radios-how-close-is-too-close/>, [Accessed 28-08-2024].
- [14] T. Brockmann, M. Rethfeldt, B. Beichler, F. Golasowski, and C. Haubelt, "MAC-Filter based Topology Control for WLAN Mesh Networks," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2024, forthcoming.
- [15] A. Wall, H. Raddatz, M. Rethfeldt, P. Danielis, and D. Timmermann, "ANTs: Application-driven network trust zones on MAC layer in smart buildings," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV: IEEE, Jan. 2018.
- [16] J. Friedrich, S. Frohn, S. Gubner, and C. Lindemann, "Understanding IEEE 802.11n multi-hop communication in wireless networks," in *2011 International Symposium of Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks*, Princeton, NJ, USA: IEEE, May 2011.