Design and Experimental Evaluation of Wi-Fi Mesh Architectures for Robust Multi-Robot Communication in Riverine Monitoring

University of Turku
Department of Computing
Master of Science in Technology Thesis
Robotics and Autonomous Systems
June 2026
Chathuranga Liyanage

Supervisors:
Assistant Prof. Widhi Atman
Prof. Tomi Westerlund

UNIVERSITY OF TURKU
Department of Computing

Chathuranga Liyanage: Design and Experimental Evaluation of Wi-Fi Mesh Architectures for Robust Multi-Robot Communication in Riverine Monitoring

Master of Science in Technology Thesis, 89 p., 2 app. p.
Robotics and Autonomous Systems
June 2026

Environmental pollution and climate change present critical challenges to global sustainable development, underscoring the urgent need to protect riverine ecosystems that support vital biodiversity. While Multi-Robot Systems (MRS) offer an advanced approach to continuous river monitoring, their deployment is constrained by communication reliability. Unmanned aerial, ground, and surface platforms must exchange control, telemetry, and high-volume sensor data in environments across infrastructure-less corridors plagued by vegetative shadowing, NLOS conditions, and dynamic topology changes. This thesis investigates whether commercial Wi-Fi mesh architectures can meet the stringent communication requirements of multi-robot riverine monitoring.

The work designs and implements a practical communication testbed using MikroTik Groove A52 mesh nodes, Robonode-M mesh modules, and Ubiquiti P2P links. The system is evaluated through different scenarios: mobility, distance, LOS, NLOS, frequency band, and QoS experiments. Network level performance is measured using latency, throughput, packet loss, RSSI, jitter, forwarding behavior and link recovery. The evaluation is further extended to application level robotic validation by transmitting ROS sensor topics, including Livox LiDAR point clouds, RealSense colour images, and ZED 4K camera streams.

The results show that Wi-Fi mesh architectures provide useful connectivity, continuity under mobility and topology changes, making them suitable for riverine monitoring where maintaining communication for coordination is more important than maximizing peak throughput. MikroTik provides lower latency and more stable moderate range performance, while Robonode offers longer range and more dynamic mesh behavior but suffers from higher latency and lower application layer throughput. Apart from these, Ubiquiti P2P links achieve the best performance in transmitting high bandwidth sensor data, but are less flexible under mobility and obstruction. Overall, Wi-Fi mesh architecture satisfies riverine monitoring requirements. The findings further supports a hybrid design in which mesh networking provides mobile local access and topology resilience, while P2P backhaul supports high bandwidth sensor data transfer. Based on these findings, this research contributes a reproducible experimental benchmark and practical deployment guidelines for Wi-Fi mesh-based communication in riverine multi-robot monitoring systems.

Keywords: Wi-Fi mesh communication, Multi-robot systems, Riverine Monitoring, Robonode, Mikrotik, Robotic Sensors

Contents

List of Figures

List of Tables

List of acronyms

ACHORD Autonomous and Collaborative High Bandwidth Operations with Radio Droppables

AI Artificial Intelligence

AODV Ad hoc On-Demand Distance Vector

BATMAN-adv Better Approach To Mobile Adhoc Networking advanced

CA Collision Avoidance

CaCN Communication-Aware Control and Navigation

CBFs Control Barrier Functions

CSMA Carrier Sense Multiple Access

DDS Data Distribution Service

FEC Forward Error Correction

HWMP Hybrid Wireless Mesh Protocol

IEEE Institute of Electrical and Electronics Engineers

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

LAN Local Area Network

LiDAR Light Detection and Ranging

LOS Line of Sight

LTE Long Term Evolution

MAC Media Access Control

MANET Mobile Adhoc Network

MIMO Multiple-input and multiple-output

MPR Multi-Point Relay

MRS Multi-Robot System

NLOS non-line-of-sight

NR New Radio

OFDMA Orthogonal Frequency Division Multiple Access

OFDM Orthogonal Frequency-Division Multiplexing

OGMs Originator Messages

OLSR Optimized Link State Routing

OSI Open Systems Interconnection

P2P Point-to-Point

PHY Physical Layer

QAM Quadrature Amplitude Modulation

QoS Quality of Service

RGB Red, Green, Blue

ROS Robot Operating System

RSSI Received Signal Strength Indicator

RTT Round Trip Time

SD Standard Deviation

SSH Secure Shell

TCP Transmission Control Protocol

TC Topology Control

UAVs Unmanned Aerial Vehicles

UDP User Datagram Protocol

UGVs Unmanned Ground Vehicles

USVs Unmanned Surface Vehicles

WLAN Wireless Local Area Network

1 Introduction

1.1 Background and Motivation

The modern world is increasingly shaped by technologies that aim to improve human lives and address complex social challenges. At the same time, environmental degradation, habitat loss, pollution, and the declining health of natural ecosystems have become major global concerns. In this context, robotics offers promising opportunities not only for industrial and commercial applications, but also for protecting and monitoring the natural environment [1].

An important area where robotics can make a meaningful contribution is river monitoring. Riverine environments are ecologically significant and support biodiversity, water resources, and human livelihoods. However, these environments are often difficult to access, especially in areas with dense vegetation, unstable terrain, shallow water, mud, or seasonal flooding. In addition, traditional methods are heavily dependent on manual sampling and data processing, which is spatially limited, labour-intensive, and unable to capture the specific dynamic of riverine environments [2], [3], [4]. As a result, continuous monitoring by humans alone can be costly, time-consuming, and in some cases impractical.

Multi-Robot System (MRS) provides a practical way to address these challenges. Multiple robots can cooperate to perform monitoring tasks more efficiently in a wider area. In a riverine monitoring scenario, different robot platforms, such as

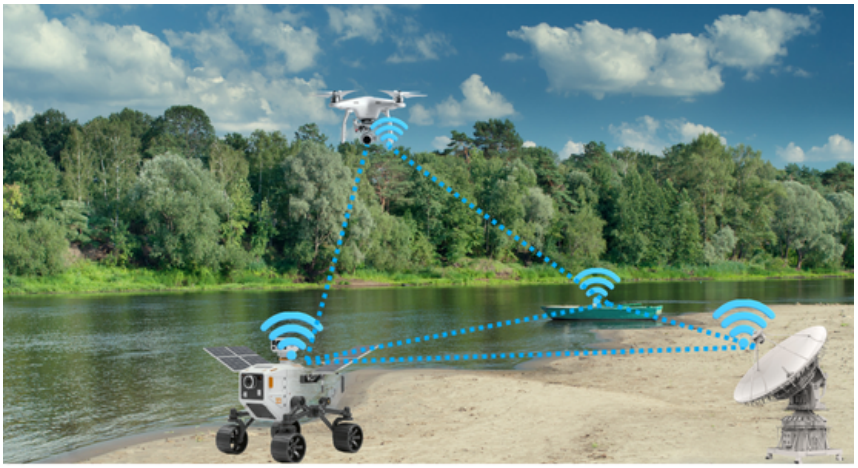

Figure 1.1: Unmanned Aerial Vehicles (UAVs), Unmanned Surface Vehicles (USVs), and Unmanned Ground Vehicles (UGVs) are working cooperatively as a MRS to monitor a riverine environment © 2026 IEEE

UAVs, UGVs, and USVs can be used together to collect visual, environmental and situational data from different perspectives as illustrated in Figure 1.1. This cooperative approach can improve coverage, redundancy, and task efficiency, especially in environments where one type of robot alone may not be sufficient [5].

However, the effectiveness of a MRS depends not only on the robot platforms themselves, but also on the quality of communication between them. To operate as a team, robots must be able to exchange information, coordinate actions, maintain connectivity, and support decision-making in real time. Without reliable communication, the benefits of using multiple robots are extremely reduced. Therefore, communication becomes a central requirement for the successful deployment of MRS in riverine monitoring applications.

1.2 Problem Statement

Although MRS has strong potential for riverine monitoring, establishing reliable communication in such environments remains a significant challenge. Riverine monitoring missions often generate large amounts of data, including video streams, sensor readings, control messages, and status updates. This requires a communication so-

lution that can provide sufficient bandwidth and throughput while also maintaining stable connectivity over a relatively large area.

In practice, riverine environments are especially challenging for wireless communication. These environments may include obstacles, NLOS conditions, changing topology due to robot movement, and limited existing infrastructure. In many deployment areas, advanced communication infrastructure such as 5G or 6G networks may not be available. As a result, the communication system must be able to operate independently while still supporting reliable, low-latency, and high-throughput data exchange [6], [7].

Given that, the core problem addressed in this thesis is to design and evaluate a communication architecture for MRS applications that can provide adequate bandwidth, long-range connectivity, and stable operation under infrastructure limited conditions. In the absence of cellular infrastructure, Wi-Fi mesh has been evaluated as an alternative communication technology solution for MRS. For example, in the DARPA Subterranean Challenge, Wi-Fi mesh has been tested in challenging environments by different teams [8], [9], [10]. In particular, this thesis focuses on investigating whether a Wi-Fi mesh-based solution can meet these requirements for real-time riverine monitoring tasks.

1.3 Research Questions

This thesis is guided by the following research questions,

Main RQ: How effectively can a Wi-Fi mesh communication architecture support reliable, low latency, and high throughput communication in multi-robot riverine monitoring scenarios?

RQ1: What are the requirements for communication network for river monitoring, and how much can Wi-Fi mesh technology fulfil those requirements?

RQ2: How to design, implement, and evaluate suitable Wi-Fi mesh architecture

towards riverine monitoring with MRS?

RQ3: How to integrate Wi-Fi mesh architecture on heterogeneous robotic platforms for riverine monitoring, and to what extent does the implemented system satisfy the operational requirements?

1.4 Thesis Contributions

This thesis designs and integrates a Wi-Fi mesh communication architecture using commercial hardware and network protocols, and systematically evaluates it under controlled and field-oriented conditions. The work focuses on practical deployment considerations for multi-robot riverine monitoring and aims to bridge the gap between communication theory and real robotic implementation. The main contributions of this thesis are as follows:

- Identification of the communication needs of multi-robot riverine monitoring systems, with emphasis on throughput, latency, reliability, and coverage.
- Justification of Wi-Fi mesh as a suitable communication approach for MRS in infrastructure-limited environments.
- Design and implementation of a practical communication architecture using commercial mesh hardware and computing platforms.
- Experimental evaluation of the proposed system using metrics such as throughput, latency, packet loss, routing stability, and mobility-induced performance variation.
- Validation of the proposed architecture through different scenario-based testing, which can occur in riverine environments.
- Development of practical deployment insights, performance characterization, and recommendations for riverine monitoring applications.

In addition, part of this work was accepted for publication and has been presented at the 2026 Asian Control Conference (ASCC). Overall, this thesis contributes

a validated Wi-Fi mesh-based communication framework for riverine MRS and provides a foundation for further work in communication-aware robotic monitoring and autonomous environmental applications.

1.5 Thesis Structure

The remainder of this thesis is organized as follows.

Chapter 2 presents the background of MRS, riverine monitoring, and the communication requirements of such applications. It also reviews relevant wireless communication technologies, existing networking devices and explains the motivation for selecting Wi-Fi mesh as the focus of this work.

Chapter 3 describes the system design and implementation of the proposed communication architecture, including the selected hardware platforms, network configuration, integration with computing devices and robotic systems, and the measurement framework used for evaluation.

Chapter 4 presents the experimental evaluation in controlled environments and field-relevant deployment conditions. The performance of the system is analysed using communication metrics such as throughput, latency, packet loss, and routing stability, as well as it examines the effects of mobility, topology changes, and practical operating conditions on communication performance. Furthermore, chapter 5 provides a comparison of different hardware, integration with robotic platforms and integration of different architecture such as P2P network along with mesh network.

Chapter 6 discusses the results in relation to the research questions, highlights the strengths and limitations of the proposed system, and provides practical recommendations for deployment in real-world riverine monitoring scenarios. Finally, Chapter 7 concludes the thesis by summarizing the main findings, reflecting on the significance of the work and outlining possible directions for future research.

1.6 Use of Artificial Intelligence Tools

During the preparation of this thesis, AI tools were used in a limited and supporting role. ChatGPT was used to help with rephrasing sentences, improving the flow of paragraphs and proofreading drafts, while Grammarly was used for grammar and spelling checks. These tools were applied to text I had already written myself, and every AI assisted suggestion was reviewed, edited, and accepted only when I judged it consistent with my intended meaning. Two figures, Figure 3.1 and Figure 4.17, started as manual drawn sketches and were redrawn in a cleaner, more formal style with the help of ChatGPT. AI was also used as a coding aid. I wrote the data analysis and measurement scripts myself, and used AI tools only to help locate bugs, understand error messages, and look up syntax. No script, result, or interpretation in this thesis was generated by an AI tool without my direct review and verification.

All technical content of this thesis, including the problem formulation, the research design, the experimental work, the measurements, the data analysis, and the interpretation of the results, were developed independently and reflects my own work. AI tools were used only as writing and coding assistants, and not as a source of ideas, citations, or finding. I take full responsibility for the content of every section of this thesis and for any errors that may remain.

2 Literature Review

Riverine ecosystems are highly affected by climate change, pollution, and anthropogenic activities, creating a need for riverine monitoring approaches that can capture changes across large areas over time [11]. At the same time, modern river studies highly depend on high-resolution and large-spatial coverage datasets (e.g., UAV photogrammetry, topo-bathymetric LiDAR, and other dense sensing outputs). These data streams and derived products are often large (GB-scale per mission in practice) and frequently processed offline, creating a gap between data acquisition and potential oversight in the field [12].

MRS is a potential solution to these riverine constraints because they can distribute sensing and coverage over multiple platforms, e.g., via UAV, UGV, and USV, reducing mission time and improving redundancy [13]. However, the practical value of MRS in riverine monitoring is strongly limited by communication. River corridors combine long operational extents, intermittent LOS, vegetation attenuation, and reflections from water surfaces conditions impact the bandwidth, packet loss, routing instability, and latency [14]. This literature review narrows from (i) why MRS matters, to (ii) why rivers are a demanding monitoring domain, to (iii) what communication requirements emerge, and finally to (iv) why Wi-Fi mesh networks are a compelling communication substrate for high data rate riverine multi-robot monitoring.

2.1 Monitoring Riverine Environment

MRS offers significant opportunities for riverine monitoring by enabling parallel and distributed data collection. In river corridors, multiple robots can survey different segments simultaneously, reducing the time and effort required as compared to sequential single robot mapping. This is especially valuable in dynamic riverine environments, where hydrological, geomorphological and ecological conditions vary in several kilometres.

2.1.1 Importance of Riverine Monitoring

Freshwater rivers are among the most ecologically and economically significant components of the global environment. Riverine environments provide habitats for a substantial proportion of the world's biodiversity, supplying water for agriculture and human consumption, and regulating regional hydrological cycles. However, industrial advancement and anthropogenic activities have significantly accelerated climate change, leading to the rapid deterioration of freshwater ecosystems. Tripathi et al. document the convergence of climate change and pollution pressures on river systems, describing cascading effects on biodiversity, water quality, and hydrological stability that demand consistent, high-resolution monitoring [11].

The information required to support studies of riverine ecosystems is broad and multidimensional. Relevant parameters include water level, flow velocity, sediment transport, temperature, turbidity, dissolved oxygen, pH, and the presence of chemical pollutants. In addition, high resolution spatial products including bathymetric maps, shoreline morphology data, and vegetation cover classifications are essential for understanding erosion dynamics, habitat distribution, and hydraulic behaviour [15]. Traditional riverine monitoring has relied on manual sampling campaigns and offline post-processing pipelines. This approach is spatially limited, labour-intensive, and fundamentally unable to capture the rapid temporal dynam-

ics of riverine environments [4]. The practical consequence of offline workflows is that data gaps or sensor errors are often discovered only during post-processing, which may necessitate additional field-testing.

2.1.2 Existing Technologies and Sensor Payloads

Advances in autonomous mobile platforms have substantially changed how riverine data is collected. UAVs, USVs, and UGVs are now routinely used to carry diverse sensor payloads across riverine terrain. The sensors deployed in these systems span a wide range of modalities, each with distinct data volume and processing characteristics, as shown in Table 2.1.

Table 2.1: Sensor payloads and data characteristics in riverine monitoring © 2026 IEEE

Sensor	**Weight**	**Platform**	**Data Modality**	**Data Volume**	**Processing**
LiDAR (FARO Photon 120) [4]	14.5 kg	Boat, cart, or backpack	3D point cloud	Moderate (Several MB)	Offline
LiDAR (YellowScan Navigator) [16]	3.7 kg	UAV (DJI Matrice 600)	3D point cloud	High (80 GB/ 1 km)	Offline
RGB (Red, Green, Blue) camera (Nikon D5100) [4]	1.5 kg	800 class radio-controlled helicopters	RGB imagery, 3D point models	Moderate (4 GB/ flight)	Offline
Multi-spectral camera (Mavic 3) [17]	0.951 kg	DJI Mavic 3 Multispectral UAV	RGB and multispectral orthomosaics	High (Several GB /flight)	Offline
RGB camera (Phantom 4) [18]	0.138 kg	DJI Phantom 4 Pro V2.0 UAV	RGB orthophotos and point cloud	High (Several GB)	Offline
Hyper-spectral (Visible and Near-Infrared) camera [19]	0.5–4 kg	DJI Matrice 600 or fixed-wing UAV	Hyperspectral data cube	High (MB–GB /flight)	Offline
Single beam sonar (acoustic) [20]	0.1 kg	Boat/USV-mounted	Scalar depth (range) soundings (plus optional intensity/waveform)	Low (Few MB)	Online (limited)
Multi-beam sonar (acoustic) [21]	1-3 kg	Boat/USV-mounted	Scalar depth (range) soundings (plus optional intensity/waveform)	High (MB–GB /survey)	Online (limited)
Optical Probe (multi parameter optical sensor) [22]	3.44 kg	Buoy system	Physicochemical Readings	Low (<25 MB /day)	Hybrid (online/ offline)
Subsurface Ground Penetrating Radar (GPR) [20]	4.5 kg	UAV / Drone	Radar Profile	Moderate (Several MB)	Offline

Light Detection and Ranging (LiDAR) is among the most data intensive sensor types used in riverine monitoring. Flener et al. demonstrated seamless high-

resolution river channel mapping by combining mobile LiDAR with UAV photogrammetry, producing dense 3D point clouds that capture both above water terrain and shallow bathymetry with sub-decimetre accuracy [4]. A UAV mounted LiDAR system reported data volumes of approximately 80 GB per flight [16], illustrating the scale of raw data produced in even a single survey session.

RGB and multispectral cameras mounted on UAVs provide complementary spatial data at lower unit cost. Akstinas et al. evaluated UAV based RGB and multispectral photogrammetry for riverbed topography reconstruction and determined that these systems generate datasets of several gigabytes per flight in the form of orthomosaics and dense point clouds [17]. Hyperspectral sensors, which capture reflectance across dozens of narrow spectral bands, are increasingly used for water quality classification and substrate mapping. These generate high dimensional spectral cubes ranging from hundreds of megabytes to multiple gigabytes per flight session [19].

Drone-borne Ground Penetrating Radar (GPR) has been demonstrated as a viable method for wide bathymetric profiling in inland waters, producing bathymetric profiles with accuracy comparable to water-coupled GPR when validated against RTK GNSS measurements [20]. *In situ* water quality probes measuring turbidity, fluorescent dissolved organic matter (FDOM) and temperature generate comparatively low data volumes, typically less than 25 MB per day, but require continuous connectivity for real-time alerting applications [22].

The combined effect of sensor payloads on communication requirements is substantial. A single multi-sensor riverine survey mission can generate datasets ranging from tens to several hundreds of gigabytes in total. Different sensor streams impose different communication requirements. As an example, LiDAR and imagery demand high throughput links for bulk offload, while telemetry and water quality parameters require low latency continuous channels. This heterogeneity of the data characteris-

tics is a primary driver of the communication architecture choices examined in the following sections.

2.1.3 MRS for Riverine Monitoring

The combination of large spatial extent, multi-modal sensor requirements and the need for near real-time data quality assessment strongly motivates the use of MRS for riverine monitoring. Pinto et al. demonstrated an early but influential instance of heterogeneous multi-robot riverine monitoring through the *RIVERWATCH* system, which deployed a marsupial surface aerial team comprising a USV and a UAV for coordinated riverine environmental monitoring [23]. This system demonstrated the value of simultaneous multi altitude perception combining surface level and aerial viewpoints by underscoring the practical importance of robust inter robot communication for coordinating the two platforms.

In a general framework, a MRS for riverine monitoring enables a group of UAVs to simultaneously survey different segments of a river corridor. Further, it effectively completes the spatial coverage in a given time window while improving both the spatial and temporal resolution of the collected dataset. Simultaneously, a heterogeneous deployment combining UAVs (for aerial mapping and relay), USVs (for surface water quality and sonar measurements) and UGVs (for bank side reference measurements) can generate a multimodal dataset that captures the river system from multiple perspectives.

Crucially, MRS operation enables an on-site data quality evaluation workflow that is not feasible with single robot or offline approaches. By continuously streaming measurement data from mobile platforms to a capable ground station computer over the communication network, data gaps and sensor errors can be identified in real-time. This allows the mission to be adapted before the robots return to base. All in all, it transforms the monitoring paradigm from a purely offline and post process

into a responsive real-time process. However, it requires an advanced communication subsystem, which must support both coordination traffic and high throughput data streams simultaneously.

2.2 Communication for Riverine Monitoring

This section examines the communication requirements, the distinctive challenges of riverine monitoring and reviews wireless technologies. The existing literature identifies plausible options for the communication requirements in riverine monitoring.

2.2.1 Requirements, Constraints, and Challenges

The communication requirements of a riverine MRS is determined by the intersection of two functional needs, namely, coordination traffic and data transfer. Coordination traffic encompasses relatively small messages exchanged at high frequency, such as task negotiation, positional updates, shared map increments, and safety-critical collision-avoidance signals. These messages are latency-sensitive, delays of even hundreds of milliseconds can compromise navigation safety or degrade the quality of coordinated coverage. Data transfer, on the contrary, involves large but less time-critical payloads such as LiDAR point clouds, high-resolution imagery, and multispectral data that may total tens of gigabytes per platform per session. Hayat et al. identify three primary traffic types in robotic systems, real-time control traffic, periodic telemetry, and delay-tolerant bulk data. Each of these types requires different (QoS) guarantees [24].

A communication architecture that optimizes exclusively for one category at the expense of others is unsuitable for the full range of riverine MRS operations, which implies a need for different traffic prioritization. With that, simple single-channel solutions, such as LoRaWAN, are not suitable for effective monitoring. In

[14] discusses this as a challenge in co-design, the coordination strategy must be aware of what the communication system can deliver, while the communication system must be dimensioned to support the coordination requirements. Hence, the mission planner must account for bandwidth availability when deciding how often to exchange shared maps, how much data to stream in real-time versus storing onboard for later offload, and how to gracefully adjust coordination when a link fails.

Meeting these requirements is difficult due to the set of physical and operational constraints. Unlike urban or controlled environments, rivers have long and narrow corridors that can extend for many kilometres with little or no permanent infrastructure, and cellular coverage can be absent in remote rural areas [25]. This corridor geometry forces robots into an essentially linear formation, so that maintaining a connected network across a long stretch of river requires either very long communication ranges or multi-hop relay architectures. Dense riparian vegetation can attenuate radio signals by tens of decibels over short lateral distances, while frequent NLOS conditions cause poor connectivity. Moreover, the water surface adds multi-path fading, where direct and reflected signals interfere destructively and cause intermittent loss even when a direct line of sight nominally exists. Finally, as robots drift or reposition during a mission, the network must adapt to the continual topology change, if this adaptation is too slow, the result is packet loss or unacceptable latency in coordination messages [6].

These environmental-specific factors are not fully addressed by the general MRS communication literature. Gielis et al. identify dynamic topology change, bandwidth constraints, and synchronization as the dominant challenges of MRS communication [6], but the combination of vegetative shadowing and water-surface reflection produces a propagation environment substantially different from the open-field or urban scenarios studied in most wireless-networking work. Several studies discuss viable responses for these issues in different directions. Sousa et al. showed that link-

quality estimation and dynamically adjusting routing decisions based on measured signal quality substantially improves end-to-end delivery ratios in mobile aquatic networks [26]. This underscores the value of adaptive quality-aware routing under rapidly varying channels. Zolich et al. confirmed in a comprehensive survey that multi-hop ad-hoc architectures are the dominant approach for extending the range of aquatic robotic deployments, while identifying NLOS propagation and dynamic topology as the main degrading factors [27]. Abu-Aisheh et al. propose CARA, a connectivity-aware relay algorithm that dynamically designates a subset of robots as relays to preserve connectivity as the team traverses fragmented terrain [28], an approach directly applicable to the elongated geometry and intermittent obstructions of riverine deployments.

In summary, the communication technology selected for riverine MRS must satisfy a demanding combination of requirements. Sufficient bandwidth, low latency, heterogeneous traffic prioritization, and multi-hop support for dynamic topology change, while remaining infrastructure-independent and robust behaviour in a vegetated, reflective environment.

2.2.2 Existing Wireless Technologies and Their Suitability

A range of wireless communication technologies have been applied to MRS and autonomous vehicle deployments. Abderrahmane et al. presents an experimental comparison of Wi-Fi, Bluetooth, ZigBee, and LoRa for mobile MRS in hostile environments, providing empirical evidence of the trade-offs between these technologies in realistic outdoor conditions [29]. In Table 2.2 a summary of the comparison of wireless communication technology for MRS is explained.

Bluetooth is a short range and star topology default architecture make it unsuitable for multi-robot coordination across extended river reaches. ZigBee (IEEE 802.15.4) offers slightly more range (10–75 m), higher energy efficiency, and support

Table 2.2: Comparison of wireless communication technology for MRS © 2026 IEEE

Technology	Energy Efficiency	Connectivity	Data Rate	Coverage	Robustness	Decentralized
Wifi (802.11) [2], [3], [6], [29]	Low	High (supports many nodes and Internet Protocol networking in infrastructure or ad-hoc modes)	600 Mbps–7 Gbps	Medium (Mesh coverage high)	Medium (mesh self-healing)	Yes (mesh and Ad hoc modes)
Bluetooth [3], [29]	Medium	Low (Personal area network; connects only a few devices)	1–3 Mbps	Short (10-20 m)	Medium (default star topology)	Partial
Zigbee (802.15.4) [6], [29]	High	Medium (Supports dozens of nodes in mesh networks)	20–250 kbps max	Short (10-75 m)	High	Yes
LoRa /LoRaWAN [6], [29]	High	Medium (Excellent long-range node connectivity but scales poorly with many active robots)	0.3–50 kbps	Long (Several km)	High	No (Normally Star, requires a gateway)
Cellular (Mid-band 4 G, High-band 5 G) [6], [15]	Low	High (Wide-area network with carrier infrastructure; supports large device counts)	4 G: up to 1 Gbps; 5G: peak 20 7 Gbps downlink	4G:Long (2-5 km) 5 G: Medium (200-500 m)	High	No (Requires cellular infrastructure)

for mesh topologies, but its maximum data rate of 250 kbps is wholly inadequate for streaming data [29]. Sanghvi et al. confirm their performance analysis that ZigBee is appropriate only for low rate telemetry in small to medium scale deployments [30].

LoRa and LoRaWAN excel in low power, long-range telemetry and have been deployed for water level monitoring in riverine contexts [15]. Aho et al. reports a practical LoRaWAN deployment in Kuopio, Finland for water level monitoring and flood prediction, demonstrating kilometre-scale coverage with low-power consumption [31]. However, the data rates achievable with LoRaWAN are typically 0.3 to 50 kbps which is not suitable for bulk data transmission. Esfandiyar and Młodzikowski demonstrate a hybrid Wi-Fi and LoRa architecture in which LoRa provides a reliable control link beyond the Wi-Fi coverage range, but confirm that high-volume payload data must be deferred until a high-throughput Wi-Fi link is available [32].

Cellular networks (4G LTE and 5G NR) offer high data rates and a wide coverage area in populated areas, but their dependence on fixed infrastructure makes

them unreliable in remote river reaches where environmental monitoring is most needed. Stateczny et al. examined wireless LAN technologies as communication solutions for USVs and noted that while cellular links offer superior performance when available, the absence of coverage in many waterway environments requires infrastructure independent alternatives [33].

Wi-Fi (IEEE 802.11) provides a balanced combination of high throughput reaching 600 Mbps to 7 Gbps in its most recent variants, and infrastructure independence when configured in ad hoc or mesh mode. Sánchez-García et al. confirm in their survey that mesh architectures using the 802.11n and 802.11ac standards provide the best overall performance for outdoor UAV and USV deployments compared to cellular and satellite alternatives [25].

2.3 Wi-Fi Mesh as the Communication Platform

Wi-Fi Mesh extends the principles of the IEEE 802.11 wireless LAN standard by allowing multiple radio nodes to communicate with each other in a fully or partially connected graph, rather than requiring all traffic to pass through a single access point. In a mesh configuration, each node can act simultaneously as a client, an access point, and a relay that forwards packets on behalf of other nodes that are not in direct range of the destination. This multi hop forwarding capability allows the effective communication range of the network to scale with the number of deployed nodes, making Wi-Fi mesh inherently suitable for large-area deployments such as riverine surveys. The fundamental distinction between Wi-Fi mesh and conventional Wi-Fi infrastructure lies in the absence of a fixed backhaul requirement. In a traditional Wi-Fi network, all access points must be connected by a wired Ethernet backhaul. In a mesh network, the backhaul itself is wireless, allowing the network to be deployed in locations with no existing cable infrastructure, precisely the condition that prevails in remote riverine environments. The mesh network is self-organizing

as nodes discover each other, assess link quality, and establish routing tables without manual configuration, thus enabling rapid deployment from mobile platforms.

2.3.1 Layered Architecture: OSI Model Perspective

Understanding the full OSI model stack is important for designing Wi-Fi mesh architecture. The protocol choice in each layer impacts the overall communication performance experienced by the robotic middleware and applications.

At the Physical Layer (Layer 1), Wi-Fi mesh radios operate in the 2.4 GHz and 5 GHz bands using the IEEE 802.11 standard, with the most capable devices supporting 802.11ac (Wi-Fi 5) or 802.11ax (Wi-Fi 6). Wi-Fi 6 introduces orthogonal frequency division multiple access (OFDMA) and 1024-QAM, increasing spectral efficiency and theoretical throughput significantly, compared to earlier generations [34]. The choice of operating frequency involves trade-offs, the 2.4 GHz band provides better range due to its longer wavelength, while the 5 GHz band offers higher throughput but suffers greater propagation losses.

At the Data Link Layer (Layer 2), the IEEE 802.11 MAC protocol governs medium access using Carrier Sense Multiple Access with Collision Avoidance (CSMA /CA). The mesh-specific extension is defined by the IEEE 802.11s amendment, which adds the Hybrid Wireless Mesh Protocol (HWMP) as the default path selection mechanism. HWMP operates as a hybrid protocol, combining proactive tree-based path discovery from a designated mesh gate with reactive on-demand path discovery for nodes outside the proactive tree.

At the Network Layer (Layer 3), IP routing protocols enable inter-network communication and allow mesh nodes to be integrated into larger network architectures. In this layer, research-grade mesh routing protocols operate specially for robotics applications, including OLSR, BATMAN, and Babel. While middleware platforms such as ROS 2 and Zenoh interact with the mesh.

At the Transport Layer (Layer 4), the choice between TCP and UDP has significant implications for robotic communication. TCP provides reliable, ordered delivery with automatic retransmission, appropriate for bulk data transfers such as map offloads. However, TCP's congestion control mechanisms can severely degrade performance over wireless links with high packet loss, as the protocol misinterprets radio channel losses as network congestion and reduces its transmission rate unnecessarily. UDP provides unreliable but low-latency delivery suitable for real-time sensor streams and coordination messages, where latest information is more important than guaranteed delivery [33].

2.3.2 Mesh Routing Protocols

The routing protocol is arguably the most critical element in a Wi-Fi mesh deployment for MRS, as it determines how quickly the network adapts to topology changes, how much control overhead it consumes, and how reliably packets reach their destinations under mobility and NLOS conditions. A comparison of MRS routing protocols is analyzed in Table 2.3

Table 2.3: Comparison of mesh routing protocols for MRS © 2026 IEEE

Protocol	**Routing Type**	**OSI**	**Key Characteristics**	**Latency Profile**	**Typical Use Cases**
AODV [35]	Reactive	Layer 3	On-demand route discovery; low overhead; higher first-packet delay	High (5×OLSR)	Dynamic MANETs, sparse communication
OLSR [35]	Proactive	Layer 3	Periodic updates using HELLO and TC messages; low-latency routing	Low	Fixed or semi-mobile ad hoc networks
BATMAN [36]	Proactive	Layer 3	Local route decision; scalable and decentralized; does not need full topology	Low	Community mesh networks, mobile MRS
BATMAN-adv [35]	Proactive	Layer 2	MAC-layer mesh bridging; plug-and-play; no IP setup required	Low	UAV/UGV swarms, OpenWRT, ROS-based MRS
Babel [37]	Hybrid	Layer 3	Loop-free distance-vector; fast convergence; supports dual-stack (IPv4/IPv6)	Low–Moderate	Mobile IoT, embedded robotics, low-power mesh
HWMP [36]	Hybrid	Layer 2.5 (802.11)	IEEE 802.11s standard; combines on-demand and proactive tree-based routing	Moderate	Wi-Fi mesh routers, IEEE 802.11s networks
Meshmerize [38]	Hybrid	Layer 2 (802.11)	Opportunistic multi-path routing; seamless "make-before-break" handovers; sub 50 ms fail over	Ultra Low / Stable	Industrial MRS, Safety-critical drones, NLOS cases

BATMAN-adv (Better Approach To Mobile Adhoc Networking advanced) is a

proactive Layer 2 protocol that distributes routing decisions by allowing each node to independently determine the best next-hop towards any destination based on locally observed packet reception rates from originator messages (OGMs) [35]. Because BATMAN-adv operates at Layer 2, it does not require IP address assignment and can be deployed as a transparent Ethernet bridge, simplifying integration with robotic platforms. Sliwa et al. evaluated BATMAN v5 routing for aerial and ground-based mobile ad hoc networks and determines that BATMAN variants provide reliable packet delivery under node mobility, making them well-suited for UAV swarms operating in dynamic outdoor environments [35].

OLSR (Optimized Link State Routing) is a proactive Layer 3 protocol that maintains full topology knowledge at each node through the exchange of HELLO messages and Topology Control (TC) messages via a subset of Multi-Point Relay (MPR) nodes. Turlykozhayeva et al. conducted an experimental performance comparison of proactive routing protocols on Raspberry Pi 4 hardware and found that OLSR achieves higher throughput and more stable latency than BATMAN in relatively static network conditions [37].

AODV (Ad hoc On-Demand Distance Vector) is a reactive protocol that discovers routes only when traffic must be sent, reducing control overhead in sparse networks, but introducing route discovery latency for the first packet sent along a new path. Experimental comparisons show that AODV achieves competitive delivery ratios, but suffers higher per-packet latency than proactive protocols, which is problematic for real-time coordination messages in MRS [35].

Babel is a hybrid distance-vector protocol that converges quickly and supports both IPv4 and IPv6, making it well-suited for embedded robotic systems with resource constraints, although its performance degrades with increasing network size [37].

The IEEE 802.11s HWMP standard provides an interoperable baseline for Wi-

Fi mesh routing that is supported by commercial hardware without requiring custom firmware. Chovet et al. conducted a comprehensive performance comparison of ROS 2 middleware options over Wi-Fi mesh networks in planetary exploration scenarios, finding that the combination of IEEE 802.11s routing with Zenoh-based middleware provides the best overall performance for multi-robot sensor data exchange in large, dynamic topologies [36].

For riverine applications, BATMAN-adv provides the best balance of mobility resilience and deployment simplicity, while HWMP offers the advantage of hardware compatibility with a wide range of commercial outdoor Wi-Fi radios.

2.3.3 Communication Hardware for Riverine MRS

The physical hardware forming the mesh network nodes must satisfy stringent constraints, as size and weight must be compatible with mounting on UAVs or USVs. Power consumption must be manageable within the robot's energy budget and the hardware must be ruggedized for outdoor operation in potentially wet, dusty, or thermally challenging riverine conditions. The comparison of hardware devices for Wi-Fi mesh is shown in Table 2.4.

In the low-cost end, GL.iNet Slate and Raspberry Pi-based nodes running open-source mesh firmware (OpenWrt with BATMAN-adv) provide maximum software flexibility at minimal hardware cost. However, these devices lack the environmental protection ratings required for sustained outdoor riverine operation and have communication ranges of approximately 100–200 m, making it insufficient for extended river reach coverage without very dense node placement.

Mid-tier devices such as MikroTik Groove A52 and Ubiquiti UniFi AC Mesh offer weatherproof enclosures, moderate range (1–3 km under favourable conditions), and compatibility with open routing protocols. Pinto et al. used Ubiquiti airMAX radios in an early riverine robotic monitoring deployment and confirmed their suitability

Table 2.4: Comparison of Wi-Fi mesh communication devices for Multi-Robot/outdoor networks © 2026 IEEE

Device	**Size (in mm) / Weight (g)**	**Input Supply**	**Range**	**Appro: Price (in €)**	**Ruggedness and Specifications**
GL.iNet Slate (GL-AR750S) [34]	100 × 68 × 24 / 86	5 V / 2 A via Micro-USB	100 m	80	Non-rugged, Compact travel router, weather-proof
Doodle Labs Mesh Rider (Wearable Radio, Xtreme series) [39]	130 × 75 × 23 / 245	6–24 V DC input (USB-PD)	50+ km	1500+	Rugged MILspec mesh radio
Ubiquiti UniFi AC Mesh (UAP-AC-M) [23]	353 × 46 × 34 / 152	Passive PoE 24 V	180 m	100	IPX6, Outdoor-rated AP (weather-resistant)
Mobilicom MCU-30 (Rugged Mesh SDR) [40]	120 × 90 × 60 / 550	7.5–24 V DC	up to 15 km (omnidirection)	10000+	IP67 rugged , SDR-based mobile mesh unit
Rajant Breadcrumb (ES Series) [9]	155 × 149 × 41 / 455	9–30 V DC	More than 20 km	4000+	IP67, Ultra-rugged, wireless nodes for kinetic mesh
Silvus StreamCaster (4200 MANET radios) [41]	102 × 67 × 38 / 425	9–20 V DC	50+ km	10000–20000	IP68, High-end tactical MANET radios
MikroTik Groove A52 (HPn/ac) [42]	177 × 44 × 44 / 193	9-30 V DC	220 m LOS	60	IP55, Weatherproof outdoor

for outdoor aquatic environments [23]. Doodle Labs Mesh Rider radios represent a built solution for robotic mesh networking, combining ruggedness with 50+ km nominal range and native ROS 2 integration.

In the high end, Rajant Breadcrumb, Silvus StreamCaster, and Mobilicom MCU-30 provide ranges of 20–50 km and exceptional reliability in congested RF environments, using sub-GHz MIMO links and proprietary MANET protocols. These devices are well-suited as fixed base station or lead-boat nodes, but their size, weight, and cost limit their deployment on small UAVs.

The 8devices Robonode-M represents an emerging category of robotic mesh nodes: it combines Wi-Fi 6 hardware with Meshmerize firmware, which is a commercial mesh networking solution optimized for low-latency handovers and native integration with Zenoh-based ROS 2 middleware. Pandi et al. introduced Meshmerize as an opportunistic multi path routing approach for drone mesh networks, demonstrating reduced reconnection times during rapid topology changes compared to standard HWMP [38]. The Robonode-M achieves handover latencies below 50 ms and supports 1024-QAM modulation for high-throughput LiDAR and video streams,

making it an optimal candidate for riverine MRS deployments.

2.3.4 Middleware Integration and Communication Awareness

The hardware and the routing protocol of the mesh network form the lower layers of the communication stack, but robotic applications interact with the network through middleware platforms that abstract the underlying networking details. ROS 2 has become the de facto standard middleware for robotic systems, providing a publish-subscribe communication model built on top of the Data Distribution Service (DDS) standard [36]. Several DDS implementations are available including Fast DDS, Cyclone DDS, and Zenoh. Each of these has different performance characteristics under the dynamic, bandwidth-constrained conditions of a mesh network.

Zhang et al. compared middleware options for edge-to-edge and edge-to-cloud communication in distributed ROS 2 systems and determines that DDS-based transports perform well in stable wired networks but suffer from increased latency and message loss in the presence of network topology changes [43]. Zenoh, a peer-to-peer protocol designed explicitly for constrained and mobile environments, outperforms DDS-based transports in scenarios with intermittent connectivity and has been adopted as the recommended transport for Wi-Fi mesh-based ROS 2 deployments [36].

Beyond middleware selection, the most sophisticated approach to riverine MRS communication is the co-design of the communication network and the robot navigation strategy, a paradigm known as Communication-Aware Control and Navigation (CaCN). Lee and Panagou propose the use of Control Barrier Functions (CBFs) to provide formal guarantees of r-robustness in reconfigurable multi-robot networks, ensuring that the network topology remains connected even when individual robots fail or links degrade [44].

Saboia et al. presents ACHORD, a communication aware multi-robot coordination framework with intermittent connectivity, which explicitly co-designs the communication and coordination architecture to maintain mission progress despite frequent link outages [45]. Cladera et al. developed MOCHA (Multi-robot Opportunistic Communication for Heterogeneous collaboration), a gossip-based diffusion framework in which robots exchange accumulated telemetry opportunistically whenever two platforms come within the communication range, progressively propagating knowledge through the team even under severe connectivity limitations [12]. Liu and Miah propose decentralized multi cobot (Collaborative Robots) navigation under intermittent communication, using a graph-theoretic model of connectivity to guide path planning decisions in dynamic environments [46].

These CaCN approaches are directly relevant to riverine MRS, where the elongated deployment geometry and frequent NLOS conditions make continuous connectivity difficult to guaranty. By treating the mesh network state as an input to the planning algorithm, rather than treating communication as a fixed background service, the robot team can proactively manage its topology to maintain the communication quality required for real-time data streaming and coordination.

2.4 Summary and Research Gap

The review presented in this chapter establishes a clear line of motivation from riverine environmental monitoring needs, through the data volume and communication challenges these needs create, to the selection of Wi-Fi mesh networking as the most appropriate communication platform for riverine MRS. The analysis demonstrates that Wi-Fi mesh is the only technology that simultaneously satisfies all the identified requirements such as high throughput for bulk sensor data, low-latency for coordination messages, multi hop range extension, infrastructure independence, and adaptability to dynamic topologies. Final summary of the literature review and the

Table 2.5: Wi-Fi mesh communication platform for MRS© 2026 IEEE

Category	Hardware	Routing Protocol	Complexity & Integration	Connectivity Handling
Low-Cost Mesh	GL.iNet Slate/ Raspberry Pi [34]	BATMAN-adv (Layer 2) [35]	Requires custom Linux kernel patching and manual Originator message interval tuning.	High packet loss during node mobility; 2–5s reroute delay
Mid-Tier Research	MikroTik Groove [42]	HWMP (Hybrid - 802.11ac) [36]	Requires RouterOS configuration; limited by proprietary licensing constraints.	Stable for static relays; slow to adapt to fast-moving USVs
	Ubiquiti UniFi [23]	OpenWRT + BATMAN-adv [35]	Requires voiding warranty to flash OpenWRT; difficult antenna integration.	Proactive routing prevents total drops but adds CPU overhead
	Doodle Labs [39]	Proprietary Mesh Rider [39]	Purpose built for OEM integration; requires Sense platform for configuration.	Hardware-level packet prioritization (MANET-style)
Tactical/ Long-Range	Silvus StreamCaster [41]	Mobile Networked MIMO + Adaptive OLSR [37]	Proprietary Web-GUI; zero-config plug-and-play SDR architecture.	4x4 MIMO allows signal bouncing in river canyons
	Mobilicom [40]	Mobile-Mesh SDR [40]	Encrypted SDR link; requires specialized ICC (InterControl) software.	Space-frequency diversity ensures link stability.
	Rajant Breadcrumb [9]	InstaMesh [9]	Kinetic Mesh tech; requires BreadCrumb-specific management software.	"Make-before-break" technology for zero-packet loss

selection combination of all technologies are shown in the Table 2.5

Beyond the platforms reported in the literature and summarized in Table 2.5, emerging robotics-oriented Wi-Fi mesh devices, including Robonode-based systems, have not been evaluated in peer-reviewed outdoor MRS studies. Although such devices are designed for mobile robotic networking, their suitability for riverine monitoring cannot be determined from specifications alone. This thesis addresses this specific gap by benchmarking recent Wi-Fi mesh hardware under representative riverine MRS communication scenarios.

Although reviews of Wi-Fi mesh hardware, routing protocols, and middleware integration exist for general robotic applications [36], no previous experimental study has systematically characterized the throughput, latency, packet loss, and routing stability behaviour of a Wi-Fi mesh network under the specific conditions of riverine MRS deployment, including multi hop topologies with mobile nodes, simulated NLOS propagation, and mixed real-time and bulk data traffic. This experimental gap motivates the system design and evaluation work presented in the following chapters.

3 System Design and Implementation

This chapter introduces the design methodology, implementation procedure, and the evaluation framework of Wi-Fi mesh architecture. First, it explains the selection of the mesh category type and the hardware platforms for the experimental deployment. Then, it explains the selection of the mesh configuration type and routing strategy considering the feasibility of each hardware platforms. Finally, the evaluation metrics and the measurement framework are introduced. All in all, the objective of this chapter is to provide a hardware independent measuring framework, which will be used in the following Chapters 4 and 5 for the systematic evaluation process.

3.1 Architecture and Hardware Platforms

3.1.1 Mesh Category Selection

Wireless mesh deployments can be broadly classified into three categories. The first is the infrastructure (backbone) mesh category. This provides connectivity to conventional clients, which is the typical deployment in metropolitan or community Wi-Fi networks. The second is the client mesh category, where end devices themselves participate in routing and forwarding, without a dedicated backbone. In this

category, the nodes create their own virtual routing backbone on the fly. The third is the hybrid mesh category, which merges the previous two approaches by combining a dedicated backbone of mesh routers with mobile mesh-capable clients. This architecture provides the stability of wide-area coverage while allowing the network to expand dynamically. Hybrid mesh type is tested in this thesis with different mesh-capable hardware devices. Mesh-capable hardware modules are mounted on mobile robotic platforms (laptop, Raspberry Pi/Jetson, drone, ground robot) to extend coverage while the same hardware devices act as backbone nodes. Furthermore, a separate P2P link based Ubiquiti hardware also tested as a high-throughput baseline against mesh architecture.

3.1.2 Hardware Selection

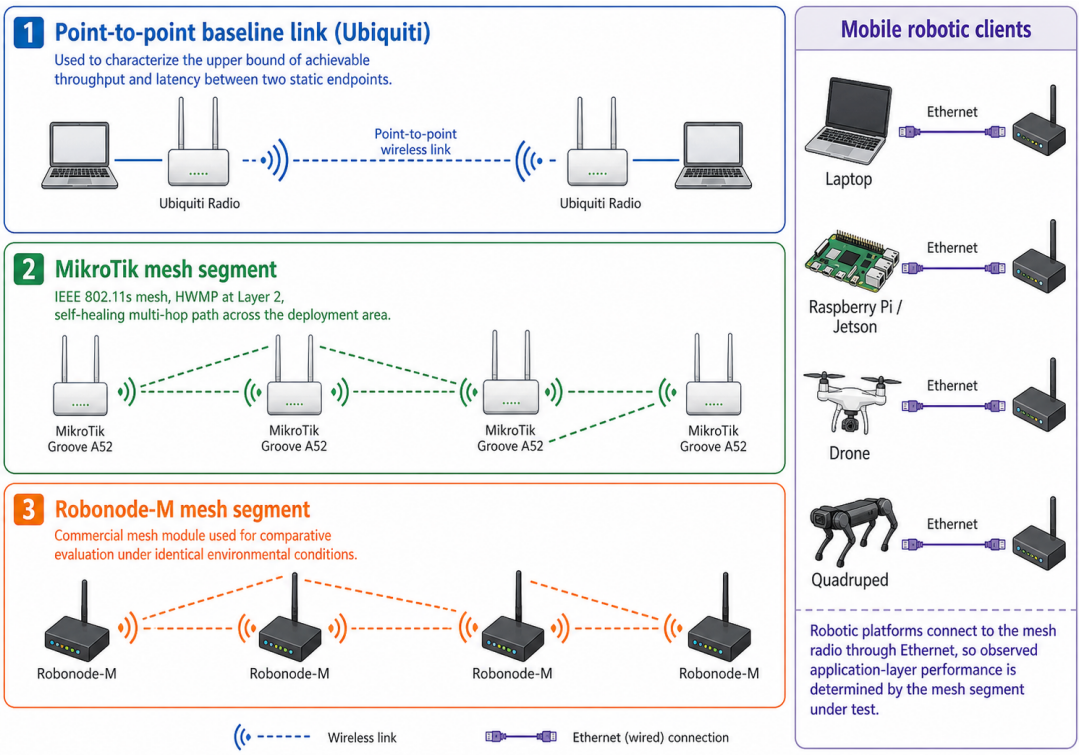

Figure 3.1: Overall network architecture combining a Ubiquiti P2P baseline, a MikroTik 802.11s mesh, and a Robonode-M mesh, each supporting mobile robotic clients (laptop, Raspberry Pi / Jetson, drone, ground robot).

The hardware selection was driven by three requirements, which are highlighted in Chapter 2. The first requirement is having the sufficient transmit power and antenna gain to operate reliably in NLOS riverine conditions. The second requirement is compatibility with standard robotics middleware through industry-standard mesh protocol. The third requirement is the practical mountability on small mobile robotic platforms with limited payload and power budgets. Considering these requirements, Ubiquiti was chosen for the P2P baseline, while MikroTik Groove A52 and Robonode-M were chosen for the mesh architecture. Figure 3.1 illustrates the overall logical architecture. Each robotic platform connects to a mesh radio through Ethernet. Using a wired connection mitigates external wireless factors, ensuring a uniform evaluation environment across all platforms. Furthermore, using an Ethernet interface ensures universal compatibility, allowing any robotic platform equipped with an Ethernet port to integrate into the mesh network.

3.1.3 Ubiquiti P2P Baseline

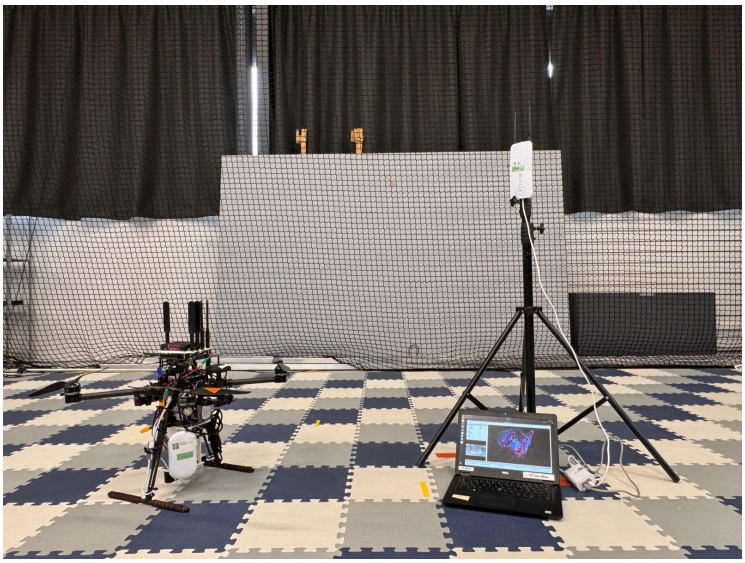

Figure 3.2: Direct P2P connection between the Ubiquiti device and the drone.

The Ubiquiti radios were configured as a dedicated point-to-point (P2P) link to serve as a performance baseline. This setup allows comparison of the mesh

network results directly against a standard P2P link handling the same traffic. It also confirms that laptops, cables, and testing tools (iPerf3 and ping) are not causing any network bottlenecks. Figure 3.2 shows the direct P2P connection between the Ubiquiti radio and the drone-side system. Similarly, Ubiquiti-based P2P links have been used in previous riverine monitoring studies [23]. Thus, ubiquti P2P provides a practical reference to compare the performance of the mesh network.

3.1.4 MikroTik Groove A52

Figure 3.3: MikroTik Groove A52 radio used as a mesh node. The unit is powered over passive PoE and carries an integrated dual-band antenna.

The MikroTik Groove A52 is an outdoor-rated Wi-Fi radio, well suited for the mobile robotic deployments in this experiment. It features a small form factor, an IP55 weatherproof enclosure, and supports Passive PoE (10–30 V), which simplifies integration with edge devices [47]. Internally, the radio is powered by a Qualcomm

Atheros chipset (720 MHz CPU, 64 MB RAM) and operates on both the 2.4 GHz band (802.11b/g/n with a 6 dBi antenna) and the 5 GHz band (802.11a/n/ac with an 8 dBi antenna) via RouterOS. Beyond its physical advantages, the Groove A52 was selected for its software flexibility. RouterOS exposes a comprehensive configuration interface (via CLI, WinBox, SNMP, and API) that is essential for automated, scripted measurements. Crucially, the radio natively supports the IEEE 802.11s standard with HWMP routing, allowing for open and standard experiments rather than relying on vendor-specific protocols. These capabilities are discussed in previous field-oriented studies [36], providing a reliable baseline for our deployments.

A key limitation of the Groove hardware is its fixed, integrated antenna pattern. This omits the possibility for adjustment via software, for example within obstructed environments. Key parameters extracted from the MikroTik datasheet are summarized in Table 3.1 alongside the specifications of the other evaluation platforms.

3.1.5 Robonode-M

The Robonode-M is a compact mesh module designed for embedded and robotic integration. It is enclosed in an aluminum casing that consists of Qualcomm QCS405 SoC (ARM Cortex-A53 quad-core, 1.4 GHz, 1 GB RAM). It features two Wi-Fi chipsets to handle the wireless traffic depending on the range, namely a long-range 4×4 MIMO radio (QCN9074) and a short-range 1×1 radio (WCN3980). Specifically, the long-range radio supports extended 2.4/5/6 GHz bands with channel widths up to 160 MHz, allowing up to 240 Mbps physical rate (5 GHz, MCS11). The Robonode-M has preloaded firmware that leverages 802.11ax features (OFDM, MU-MIMO, FEC) for robust mesh performance. Furthermore, it includes a Gigabit Ethernet port for data, which can be powered from 8.6–33 V DC. On the software side, the device provides advanced mesh functions (ack-less broadcast, self-healing links). As

Figure 3.4: Robonode-M mesh module used as a Linux-based mesh node, integrated directly on the robotic platforms via Ethernet.

a high-end mesh device, it exploits a powerful hardware and Wi-Fi 6 capabilities. It uses a Linux-based operating system on which the mesh stack runs natively using the `NAW mode` in Radio#0, supporting administrative access through SSH. Because the platform runs a fully featured Linux user space, telemetry can be retrieved using standard commands such as `iw`, `ip`, and custom scripts [48]. Overall, Robonode-M complements the MikroTik platform in the comparative study for various reasons. Primarily, it targets robotic and UAV use cases, with low weight and a robust mounting interface. Its Linux-based access allows scripting for measurement collection in the same language as on the robot's on-board computer, eliminating cross-platform parsing.

As a recent, research-oriented device, the Robonode-M has some practical limitations. Crucially, its documentation is more basic than that of MikroTik, and

the hardware itself is still in its early stages. Consequently, certain parameters, like airtime cost, must be measured using vendor-specific tools. Furthermore, full Meshmerize support was not available in the tested version. Instead, a customized firmware integration was deployed, which was lightly validated by the vendor. But a complete integration is expected in a future release.

Table 3.1 summarizes the hardware specifications and trade-offs of each platform used in this experiment. For example, the MikroTik Groove is inexpensive and easy to configure, but is based on an older Wi-Fi 5 radio with a limited 1×1 MIMO antenna design. In contrast, the Robonode is more expensive but offers advanced Wi-Fi 6 capabilities and specialized software built for industrial mesh networks.

Table 3.1: Summary of the hardware platforms

Parameter	**Ubiquiti (P2P)**	**MikroTik Groove A52**	**Robonode-M**
Role in this work	P2P baseline	Mesh node 802.11s with HWMP	Mesh node (vendor stack NAW mode)
Bands supported	5 GHz	2.4 / 5 GHz	2.4 / 5 GHz
Standards	802.11a/n/ac	802.11a/b/g/n/ac	802.11a/b/g/n/ac/ax
Typical Tx power	up to 25 dBm	up to 25 dBm	up to 23 dBm
Antenna	Integrated directional	Integrated 6 dBi dual-band	Integrated omni
Management	SSH / Web UI	RouterOS (CLI, SNMP, API) / Web UI	SSH (Linux user space)/ Web UI
Powering	PoE	Passive PoE	DC / battery pack

3.1.6 Practical Mounting and Integration

Mounting the radios on the robotic platforms revealed practical weight and power constraints. The MikroTik Groove requires an external mounting bracket and a passive PoE injector, adding approximately 250 g of payload. While powering it from a small portable battery (9 V boosted to 24 V) is feasible, it limits the experiment's duration. In contrast, the lighter Robonode-M integrates easily with battery packs, making it ideal for drone deployment. Managing the platform's power budget is crucial because the battery pack must simultaneously supports communication device, mechanical propulsion, and onboard computation. For power delivery, the Robonode uses a separate adapter, while the MikroTik relies on PoE. However, sharing a

common power supply is straightforward since both devices support a similar voltage range. By considering all these practicalities, the integration was completed. The integrated setup is shown in Figure 3.5, the left side of the figure illustrates the initial laboratory setup, while the right side displays the final drone integration.

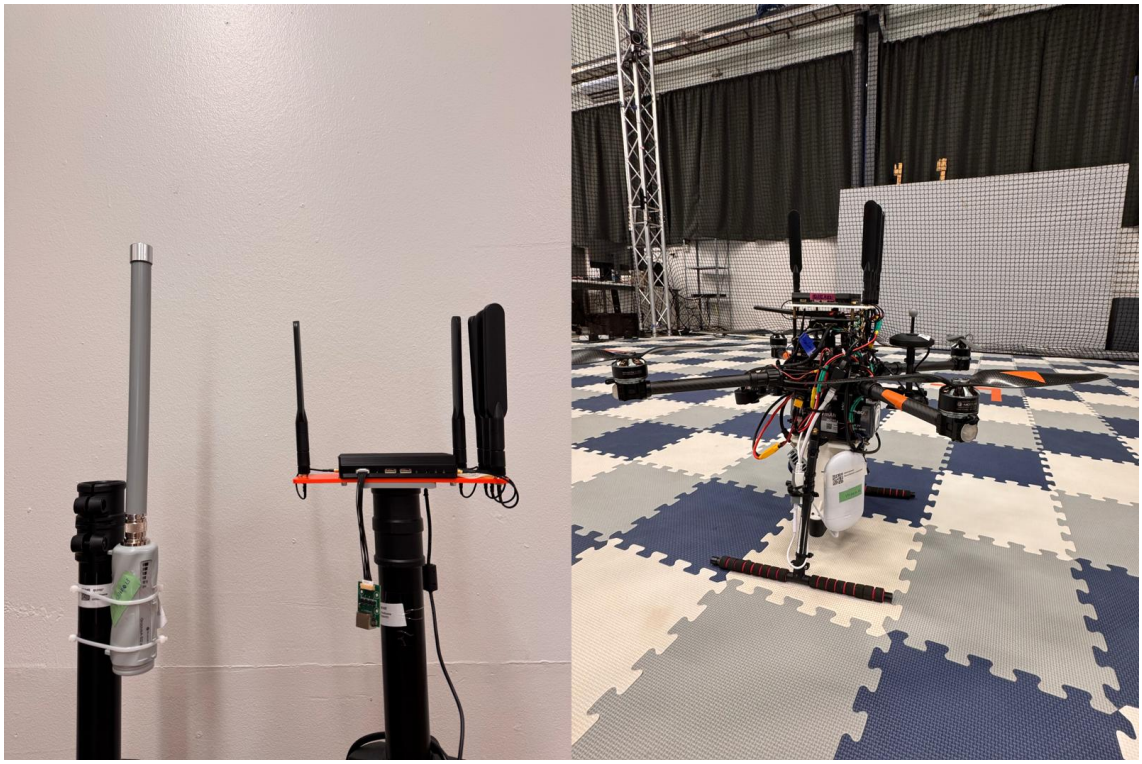

Figure 3.5: Initial laboratory test setup (left) and final drone deployment (right).

3.2 Network Configuration and Routing Strategy

3.2.1 OSI model Layer 2/Layer 3 Design

The Open Systems Interconnection (OSI) model separates network communications into distinct layers to handle different parts of the data transfer process. Layer 2 (the Data Link Layer) manages hardware-to-hardware communication between devices on the same local network using physical Media Access Control (MAC) addresses. Layer 3 (the Network Layer) handles routing across different network segments,

relying on logical Internet Protocol (IP) addresses to move data packets end-to-end. In this experiment, a specific configuration was deployed at both layers to create a seamless, flat network architecture for the mobile robots.

At layer 2 (Data Link Layer), all mesh nodes share a common mesh identifier (mesh ID) and operate on the same 5 GHz channel. A bridge interface is configured on each node to join the wireless mesh interface with the Ethernet port to which the edge device is connected. As a result, every device behind a mesh node appears on the same broadcast domain, which removes the need for inter-WLAN routing during the experiments and simplifies traffic captures. At layer 2 MAC addresses are explicitly recorded for each radio at deployment time. They are used to identifying nodes in the registration tables and to retrieve RSSI through SNMP.

At Layer 3 (Network Layer), every mesh node and every connected end device receives a static IPv4 address from the same /24 subnet (e.g., `192.168.1.0/24`). Static IP addressing was chosen over DHCP. Because it removes the dependency on a single DHCP server during mobility tests where any node may become temporarily isolated. Apart from that, it makes the result logs unambiguous, since each address corresponds to a fixed physical node throughout the experiment.

3.2.2 Choice of Routing Protocol: HWMP at Layer 2

Several routing strategies are available for mesh networks, ranging from layer 3 protocols such as OLSR, BATMAN, and AODV to layer 2 protocols such as the Hybrid Wireless Mesh Protocol (HWMP). HWMP is the default routing protocol of IEEE 802.11s. Most importantly, it is supported natively by both selected mesh platforms, ensuring interoperability and consistent comparison. HWMP forwards frames based on MAC addresses, which removes IP-layer routing complexity from the experiment and allows the mesh to behave as a single broadcast domain. HWMP supports both *on-demand* (reactive) and *proactive* path selection, which is

well suited for the riverine application. The literature reports that HWMP achieves a favorable trade-off between overhead and convergence time in small-to-medium mesh deployments [34]. This suggests the number of mesh nodes are around 3–5 nodes, which is quite suitable for the riverine monitoring. With all these aspects, HWMP was selected as the main protocol for this thesis.

3.2.3 MikroTik Configuration

The mesh interface is created by setting the wireless interface mode to `mesh` and configuring an identical `mesh-id` on every node. Then, the wireless interface is bridged with the Ethernet port. The minimum configuration used for each MikroTik node shown in Listing 3.1. The 5 GHz `a` frequency band is selected due to its higher bandwidth and lower interference compared with 2.4 GHz frequency band. The bandwidth-test server is enabled in advance, since it is the most accurate built-in throughput tool on RouterOS.

```
/interface wireless
    set [find default-name=wlan1] band=5ghz-a/n/ac \
        frequency=5180 mode=mesh ssid=riverine-mesh \
        mesh-id=riverine wireless-protocol=802.11
/interface bridge
    add name=br-mesh
/interface bridge port
    add bridge=br-mesh interface=wlan1
    add bridge=br-mesh interface=ether1
/ip address
    add address=192.168.1.11/24 interface=br-mesh
/tool bandwidth-server
    set enabled=yes authenticate=no
```

Listing 3.1: MikroTik RouterOS configuration for the wireless mesh node

3.2.4 Robonode-M Configuration

On the Robonode-M, the mesh interface is configured directly through the Linux `iw` commands, which exposes the IEEE 802.11s primitives natively. Robonode also configured with similar parameters as shown in the Listing 3.2. Thus, the two platforms can be compared under identical radio conditions. HWMP is enabled by default in the Linux 802.11s stack, and the path table can be inspected with `iw dev mesh0 mpath dump`.

```
iw dev wlan0 interface add mesh0 type mp
iw dev mesh0 set channel 36
ip link set mesh0 up
iw dev mesh0 mesh join riverine
brctl addbr br-mesh
brctl addif br-mesh mesh0
brctl addif br-mesh eth0
ip addr add 192.168.1.21/24 dev br-mesh
```

Listing 3.2: MikroTik RouterOS configuration for the wireless mesh node

3.2.5 Differences Between the Two Platforms

Although both platforms implement HWMP, three practical differences emerged during configuration. First, regarding metrics collection, MikroTik displays RSSI through SNMP and its registration table, whereas the Robonode-M displays this data through `iw`, `wifistats`, and `/proc/net/wireless`. Second, in terms of access management, RouterOS uses a declarative CLI, while the Robonode-M relies on imperative Linux networking commands. Finally, regarding protocol tuning, certain HWMP parameters such as the path refresh interval are locked in MikroTik but can be fully configured on the Robonode-M. To keep the comparison fair, all such parameters were kept at their default values.

3.3 Metrics and Measurement Framework

3.3.1 Parameters and Definitions

The experimental evaluation focuses on six primary metrics that characterize both the link quality and the routing behaviour of the mesh [34]. These metrics are throughput, latency, jitter, packet loss, hop count and RSSI. An explanation of each metric is given below.

Throughput is the actual data rate, measured in megabits per second (in this experiment) that the network can sustain end-to-end. Mathematically, it is calculated as,

$$T = \frac{B}{\Delta t}$$

where B is the total number of bits successfully delivered to the application layer during the measurement interval Δt. Throughput is measured with iPerf3 in TCP and UDP modes. The UDP measurement provides the upper bound of the link, while the TCP measurement reflects the realistic application-layer rate after accounting for protocol overhead and congestion control.

Latency is the round-trip time (RTT) between two endpoints. It is measured with ICMP echo (`ping`) with a packet size of 1500 bytes to simulate the real-world data payloads. Mathematically, it is calculated as,

$$\mathrm{RTT} = t_{\mathrm{recv}} - t_{\mathrm{sent}}$$

where t_{sent} is the departure timestamp and t_{recv} is the arrival timestamp of the reply. The mean, standard deviation, minimum and maximum values are reported for each run.

Jitter quantifies the statistical variation in packet arrival times, representing the stability of the delivery delay. It is measured as the average absolute differ-

ence between the inter-arrival times of consecutive UDP packets reported by iPerf3. Mathematically, it is calculated as,

$$J = \frac{1}{N-1} \sum_{i=2}^{N} |d_i - d_{i-1}|,$$

where d_i is the arrival time of packet i, and N is the total number of received packets

Packet Loss is the fraction of packets sent that are not successfully received. Mathematically, it is calculated as,

$$L = \frac{N_{\text{sent}} - N_{\text{recv}}}{N_{\text{sent}}}.$$

where N_{sent} and N_{recv} represent the number of transmitted and received packets, respectively. This metric is captured by both iPerf3 (in UDP mode) and `ping`, serving as the primary indicator of link instability.

Hop Count is the number of intermediate mesh routing steps between the source and destination nodes. On the Robonode-M, this value is extracted via `iw dev mesh0 mpath dump`, whereas on the MikroTik platform running RouterOS, it is retrieved using `/interface mesh fdb print`.

RSSI represents the strength of the wireless signal received at a node, reported in dBm. On the MikroTik platform, this is obtained from the `signal-strength` field of the registration table and via SNMP using OIDs under the `.1.3.6.1.4.1.14988 .1.1.1.2.1.3` subtree. On the Robonode-M, it is collected natively through the `wifistats` utility and `/proc/net/wireless`.

3.3.2 Measurement Tools and Methods per Platform

Table 3.2 summarizes the evaluation tools and platform-specific commands used for each metric. To ensure an identical application-layer environment across all tests, the same iPerf3 server was hosted on a dedicated measurement laptop throughout

the experiments.

Table 3.2: Mapping between metrics, tools, and platform-specific commands.

Metric	**Tool**	**MikroTik Command**	**Robonode-M Command**
Throughput (TCP/UDP)	iPerf3	`iperf3 -c <ip>` (from PC)	`iperf3 -c <ip>` (native)
Latency (RTT)	ping	`ping -s 1500 -c 100 <ip>`	`ping -s 1500 -c 100 <ip>`
Jitter	iPerf3 (UDP)	`iperf3 -c <ip> -u -b 10M`	`iperf3 -c <ip> -u -b 10M`
Packet loss	iPerf3 / ping	Same commands	Same commands
Hop count	route table	`/interface mesh fdb print`	`iw dev mesh0 mpath dump`
RSSI	vendor-native	SNMP / registration table	`wifistats`, `iw station dump`

For each scenario on the MikroTik platform, the procedure consists of:

1. Verifying the mesh topology with `/interface mesh fdb print` and the registration table.
2. Running a connectivity baseline with
   `ping -s 1500 -c 100 <ip>`.
3. Running a throughput measurement with the built-in tool,
   `/tool bandwidth-test <ip> protocol=udp direction=both`, and confirming with iPerf3 from the connected laptop.
4. Continuously monitoring the signal strength with
   `/interface wireless registration-table monitor 0`, logging the values every second.
5. Collecting SNMP telemetry from a remote machine with
   `snmpget -v2c -c public <ip> .1.3.6.1.4.1.14988.1.1.1.2.1.3.<MAC>`.

For each scenario on the Robonode-M platform, the same logical sequence is followed, but using Linux-native commands. Continuous logging is implemented with shell scripts that wrap the relevant tools. The fact that the Robonode-M runs a standard Linux user space allows the same scripts to run unmodified on the on-board computer of the robot, which is a clear advantage for the field-deployment.

4 Mesh Network Evaluation

4.1 Experiments

To evaluate the performance of the communication platforms in a riverine environment, a series of field experiments were conducted across scenarios. These scenarios replicate real-world challenges faced by MRS in river corridors, such as vegetation blockages, moving nodes, and heavy data payloads. Table 4.1 summarizes the experimental configurations, their specific relevance to riverine applications, and the evaluation metrics collected. Each scenario isolates unique operational requirements. For

Table 4.1: Experimental scenarios for robotic communication evaluation

Scenario	**Riverine application**	**Metrics**
1. Mobility - topology change and reconnecting	UGV blocked direct connection to USV by riverbank vegetation but reconnected via an UAV aerial relay	RTT, packet loss, outage time, hop
2. LOS vs NLOS	River bend or dense vegetation blocking LOS	RTT, packet loss, RSSI, jitter
3. Distance - RSSI decay	UAV moving upstream away from base station	RSSI, RTT, TCP/UDP throughput, jitter
4. QoS under load	UAV streaming LiDAR while receiving control commands	Latency under load, throughput, jitter
5. Device parameters (frequency band)	Choosing correct radio band for UAV in dense corridor	Throughput, RTT, RSSI

example, evaluating node mobility requires testing on dynamic multi-path topologies rather than basic ping tests alone. This confirms the network path remains stable and usable under moving conditions. In general, the evaluation is structured sequentially across three distinct hardware and architectural frameworks. First, the MikroTik Groove-based mesh network is analyzed, followed by the Robonode-M

system. Next, a standard P2P link is evaluated as a benchmark to compare link behaviour against a decentralized mesh network. Finally, a hybrid P2P and mesh architecture is assessed. By examining these three setups, this chapter systematically evaluates the structural advantages of mesh routing within demanding riverine environments.

4.2 MikroTik Groove A52 ac

4.2.1 Mobility - Topology Change, Disruption, Reconnection

This experiment evaluates whether the mesh network can maintain communication under mobility-induced topology changes. The moving node was displaced from close (20 m) to long (60 m) distance and then returned, while the middle node remained fixed as a potential relay. Continuous ping measurements were used to assess reachability and round-trip time (RTT), while the Received Signal Strength Indicator (RSSI) was recorded to observe link-quality variation. MikroTik mesh forwarding database (FDB) snapshots were also captured to examine changes in forwarding behaviour.

Figure 4.1 shows that, during the movement from close to long distance, the RSSI of the moving node decreases from approximately −50 dBm to below −70 dBm and later approaches −80 dBm. In contrast, the middle node remains stable at around −49 dBm. This degradation is also reflected in the RTT results, where early packet losses are followed by a large latency spike of more than 1000 ms and several smaller variations. These effects indicate a transition period in which the original link becomes unreliable and the forwarding path is updated. During the movement from long to close distance, the RSSI of the moving node improves as the distance decreases, and the RTT stabilizes after only a few initial variations, showing recovery

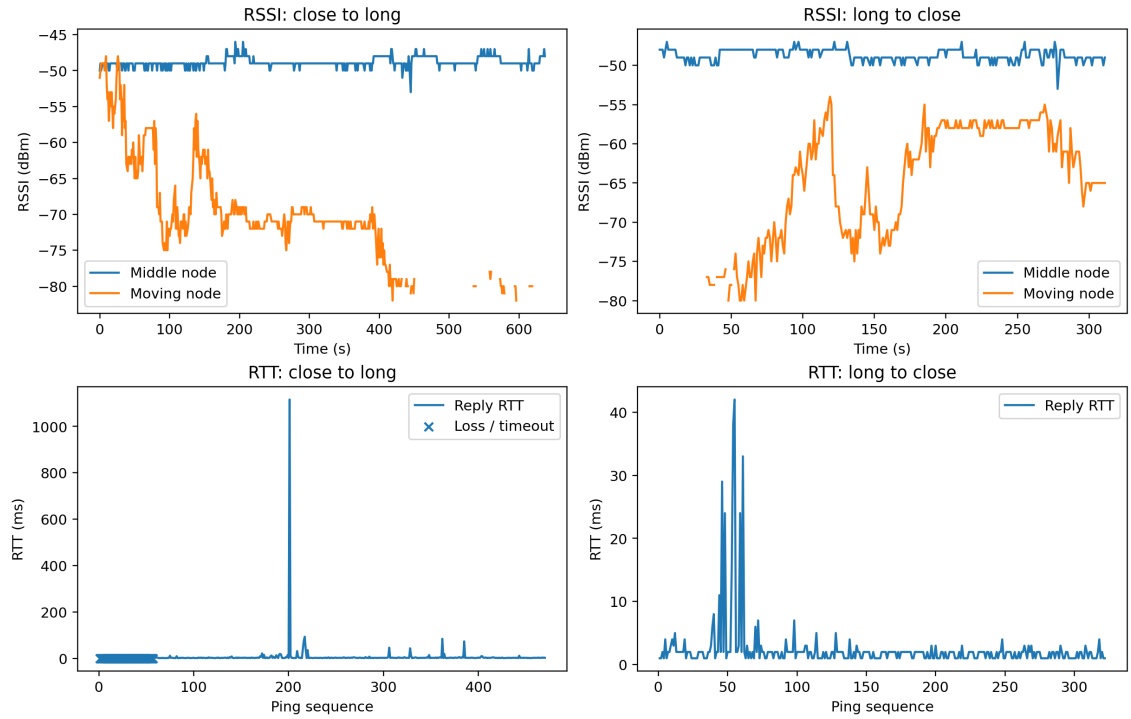

Figure 4.1: Combined mobility results for topology change. The top row shows RSSI variation for the middle node and moving node during close to long distance and long to close distance movement. The bottom row shows the corresponding RTT measured using continuous ping.

```
[admin@MikroTik] > /interface mesh fdb print detail
Flags: A - active, R - root
 A   mesh=mesh1 type=mesh mac-address=10:65:30:20:95:29 on-interface=wds1
     lifetime=0s age=11m43s metric=0 seq-number=0

 A   mesh=mesh1 type=outsider mac-address=10:7C:61:67:2B:D0
     on-interface=ether1 lifetime=0s age=21m32s metric=10 seq-number=0

     mesh=mesh1 type=larval mac-address=C8:F7:50:46:5C:36 lifetime=0s
     age=12m2s metric=0 seq-number=0

 AR  mesh=mesh1 type=neighbor mac-address=F4:1E:57:1B:5E:AB
     on-interface=wds1 age=21m37s metric=75 seq-number=158

 A   mesh=mesh1 type=local mac-address=F4:1E:57:1B:5E:C6 age=21m32s
     metric=0 seq-number=2

 A   mesh=mesh1 type=local mac-address=F4:1E:57:1B:5E:C7 age=21m37s
     metric=0 seq-number=2
```

Figure 4.2: FDB state after node movement, showing active neighbours and a larval entry during forwarding transition.

of link quality.

Figure 4.2 shows that active neighbour entries remain present on the WDS interfaces after node movement. The appearance of a larval entry suggests that the mesh was updating its forwarding state during the transition, rather than losing the forwarding structure completely. Therefore, the combined RSSI, RTT, and FDB observations show a clear sequence of link degradation, temporary disruption, and automatic recovery.

Overall, the results demonstrate that the middle node provides a stable relay while the moving node experiences link degradation during displacement. This behaviour is relevant to riverine robotic deployments, where mobile platforms may frequently encounter temporary link instability due to distance, vegetation, terrain, or other environmental obstacles. Existing literature in [26], [28], [36], [45] highlights the critical need for network connectivity that adapts without manual intervention. This mobility experiment directly fulfils that requirement, demonstrating that a mesh network can self-heal dynamically to maintain reliable communication during riverine monitoring.

4.2.2 LOS vs NLOS

The LOS and NLOS scenarios were evaluated under identical conditions at an approximate separation distance of 20 meters, with the primary difference being the presence of obstruction in the NLOS case caused by floor separation and structural barriers. This comparison is important because wireless propagation is strongly affected by blockage, attenuation, and multi path effects, which can degrade received signal strength and higher-layer communication performance such as latency, throughput, and packet loss [7], [27].

Figure 4.3(a) shows the received signal strength over time for both scenarios. The LOS condition maintained a consistently strong stable signal around -47.57 dBm

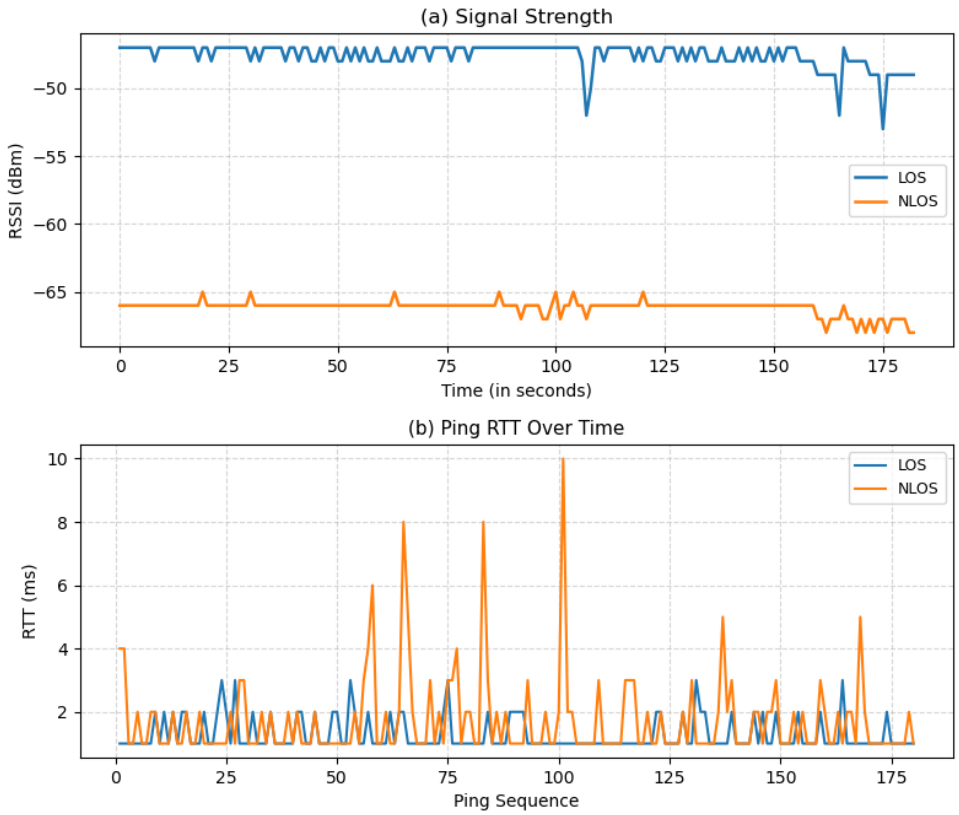

Figure 4.3: Signal strength and Ping RTT over time for LOS and NLOS conditions.

with minimal variation, whereas the NLOS condition resulted in a significantly weaker signal around -66.15 dBm indicating a persistent propagation loss. This corresponds to an attenuation of approximately 18.6 dB due to obstruction. As shown in Figure 4.3(b), the LOS scenario exhibits stable latency with minimal fluctuations, while the NLOS scenario shows noticeable spikes and higher variability over time. Quantitatively, the average RTT increased from 1.28 ms (LOS) to 1.70 ms (NLOS), with the maximum RTT increasing from 3 ms to 10 ms.

Figure 4.4(a) presents the TCP throughput over time. The LOS scenario achieves a relatively stable throughput around 20.18 Mbps, while the NLOS scenario shows a reduced average throughput of 16.24 Mbps along with increased fluctuations. This corresponds to an approximate 19.5% reduction in reliable data transmission capacity. As illustrated in Figure 4.4(b), the LOS scenario maintains a stable bitrate close to the target rate, while the NLOS scenario exhibits fluctuations and degra-

Figure 4.4: TCP throughput, UDP througput and Packet Loss over time.

dation over time. More importantly, Figure 4.4(c) shows that packet loss remains near zero under LOS conditions, whereas the NLOS scenario experiences significant packet loss bursts. The total packet loss in the NLOS case reached approximately 35%, with peak interval loss up to 61%. This shows that NLOS conditions severely impact real-time communication performance, leading to intermittent data delivery failures.

The results clearly demonstrate that NLOS conditions degrade communication

performance across multiple metrics. While RSSI, latency and TCP throughput are moderately affected, UDP-based communication suffers significantly due to increased packet loss and instability. These findings highlight the importance of maintaining LOS or deploying relay nodes in riverine environments.

4.2.3 Distance - RSSI Decay

This experiment evaluates the impact of distance on mesh link quality in an outdoor deployment. For this experiment, mainly 3 mesh nodes were used as shown in Figure 4.5, they are node A, B and C. Node A and B were fixed nodes while node

Figure 4.5: Node A (60.445293, 22.314026), Node B (60.445013, 22.314854), Node C (60.446842, 22.317963). Map adapted from Google Maps © Google and relevant data providers.

C was moving. In addition, node B, was placed approximately 60 m from node A, providing a stable short-range reference link. Node C, was moved progressively away from node A along the Tierankatu road, to a maximum separation of approximately 315 m from node A. However, the effective communication range was about 250 m from node B. Continuous RSSI logging and ping measurements were collected while

the moving node travelled from the near region toward the farthest reachable point. The moving node covered approximately 250 m in about 5 minutes, which corresponds to an average speed of about 0.83 m/s. This time-based movement was used to map each RSSI and ping sample to an estimated distance from the base node.

Figure 4.6 shows the degradation of the moving link as the distance increased. At short range, the moving node maintained a strong signal. During the first 60 m, the mean RSSI was −51.7 dBm. In comparison, the fixed reference node A remained nearly constant at about −63 dBm throughout the experiment, confirming that the short LOS baseline was stable. The most significant RSSI drop occurred early in the

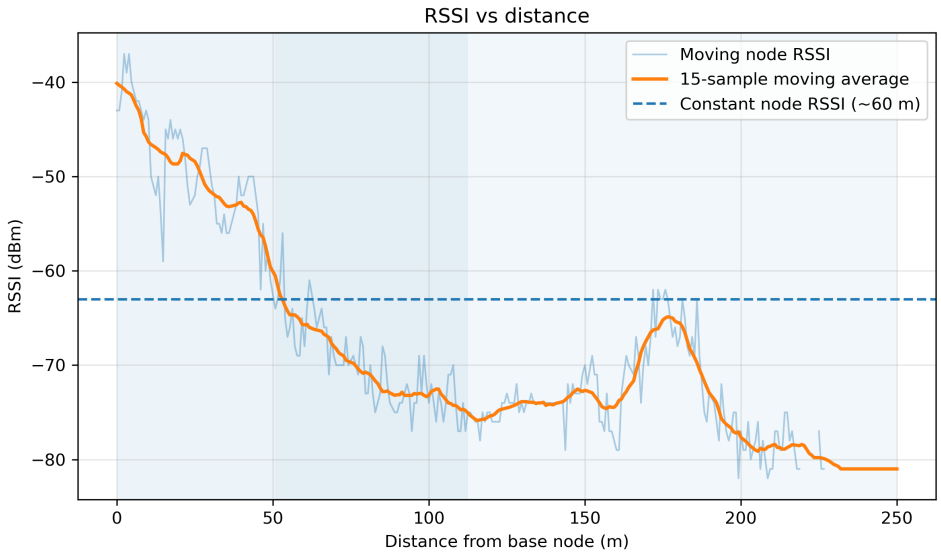

Figure 4.6: RSSI versus distance for the moving Node C.

movement. The smoothed RSSI crossed −60 dBm at approximately 50.8 m, which marks the end of the clearly LOS-dominant region. Between 60 m and 120 m, the mean RSSI fell to −71.7 dBm. This region represents the main transition zone, where the link moved from strong LOS propagation to partial obstruction. Besides the impact of the geometric distance, it indicates the impact of environmental blockage and multipath effects.

Beyond approximately 112.5 m, the link entered a weak but still usable region. Between 120 m and 200 m, the mean RSSI remained around −72.2 dBm, showing

that the signal no longer decayed rapidly with distance but instead plateaued at a lower level while having a sudden spike around 180 m. This behaviour occurs with NLOS-dominated propagation, where reflections and diffraction fluctuate connectivity even after the direct path is degraded.

Near the edge of the reachable range, the signal weakened further. From 200 m to 250 m, the mean RSSI decreased to −78.7 dBm, and the smoothed signal first crossed −80 dBm at approximately 225 m. This marks the beginning of the failure region, where the link approaches receiver sensitivity limits and connectivity becomes unstable.

Overall, the results show that the dominant performance change was not a gradual decay throughout the path. Instead, the key change occurred when the moving node left the strong LOS region and entered a partially obstructed region. After that transition, the RSSI stayed in a weak but relatively flat range for a substantial distance before collapsing near the edge of communication range. This behaviour agrees with earlier propagation and UAV-link studies, where obstacles and distance can cause rapid degradation in riverine monitoring [7], [24].

4.2.4 QoS Under Simultaneous Traffic Loads

Table 4.2: Filtered RTT summary before and after traffic load at 20 m.

Scenario	**Idle mean (ms)**	**Loaded mean (ms)**	**Loaded max (ms)**	**Loaded std (ms)**	**Timeouts**
LOS + UDP	1.20	1.99	47	3.25	0
LOS + TCP	1.10	24.59	47	3.72	1
NLOS + TCP	1.22	34.58	90	12.78	1

The QoS experiment evaluated the impact of network traffic on control-channel latency. The baseline set up corresponds to ping measurements without additional traffic, while the loaded set-up corresponds to ping measurements with traffic. The RTT remains low before the traffic started and it increases after the traffic begins.

Table 4.2 summarizes the RTT before and after the traffic is introduced. Under LOS conditions, the idle RTT remained at approximately 1–2 ms. When UDP traffic was added, the latency increased only slightly, indicating that the tested UDP stream introduced limited additional delay to the control channel. In contrast, simultaneous TCP traffic caused a clear increase in RTT and larger variation. The same effect becomes stronger under NLOS conditions. The baseline RTT was still low, but when TCP traffic was introduced, the mean RTT increased substantially. This shows, the obstruction amplifies the latency caused by concurrent traffic.

Table 4.3: TCP throughput summary at 20 m with and without simultaneous traffic.

Scenario	**Mean TCP (Mbps)**	**SD TCP (Mbps)**	**Max TCP (Mbps)**
LOS base	20.86	0.78	22.3
LOS + ping	20.20	1.73	23.0
NLOS base	16.39	3.55	21.7
NLOS + ping	15.58	3.49	20.5

TCP throughput changed marginally (from 20.86 to 20.20) while pinging, as shown in the Table 4.3. Thus, the main QoS effect is the latency hike. The UDP throughput results demonstrate stable bitrate, low jitter, and no packet loss. These results, confirm that, UDP traffic has only a limited impact on the link under LOS condition. Overall, the results indicate that simultaneous TCP traffic significantly degrades control-channel latency. The degradation becomes more pronounced under NLOS conditions, which is important for robotic systems that require stable low-latency communication.

Table 4.4: Distance-based QoS summary under TCP traffic load.

Distance	**TCP throughput (Mbps)**	**Idle RTT (ms)**	**Loaded RTT (ms)**	**Max RTT (ms)**
10 m	22.13	1.0	14.66	28
50 m	18.29	1.3	17.28	56
200 m (edge case)	1.18	395.5	162.98	3515

The distance-based QoS analysis evaluates the impact of distance on latency and traffic. Table 4.4 summarizes the results for 10 m, 50 m, and 200 m. At 10 m, the

link remains stable, with low RTT and high TCP throughput. At 50 m, a moderate accretion in RTT is observed, indicating the beginning of performance degradation. At 200 m, the effect becomes significantly more pronounced, with large RTT spikes and increased variability, indicating unstable communication.

Overall, the results show that increasing distance amplifies the impact of TCP traffic on latency, and beyond a certain range, the communication becomes unstable. This aligns with previous work highlighting QoS as a critical issue in multi-robot communication, where timely and reliable exchange is required for control and coordination [6]. These results indicate that acceptable throughput alone is not sufficient if latency degrades with distance for mixed traffic carrying robotic systems.

4.2.5 Device Parameters Impact (Frequency Band)

The previous experiments were conducted on 5 GHz band coupled with `5ghz-a`. To examine whether the observed behaviour was primarily related to the frequency band or the PHY mode, additional experiments were conducted by adjusting the frequency bands to `5ghz-a`, `5ghz-a/n/ac`, and `2ghz-onlyn`. The comparison was performed under both LOS and NLOS conditions through latency and throughput tests.

The latency results in Figure 4.7 show a clear separation between the 5 GHz and 2.4 GHz cases. With `5ghz-a`, the RTT remained consistently low in both LOS and NLOS. The measured average RTT was 1.28 ms in LOS and 1.70 ms in NLOS, with maxima of only 3 ms and 10 ms, respectively. In contrast, `2ghz-onlyn` exhibited substantially elevated latency. In LOS, the average RTT increased to 13.64 ms with spikes up to 493 ms, while in NLOS the average increased further to 16.97 ms with a maximum of 270 ms and one packet loss event. This indicates that the 2.4 GHz configuration maintained connectivity, but with significantly less stability.

The TCP throughput results in Figure 4.8 reveal a trade-off between peak

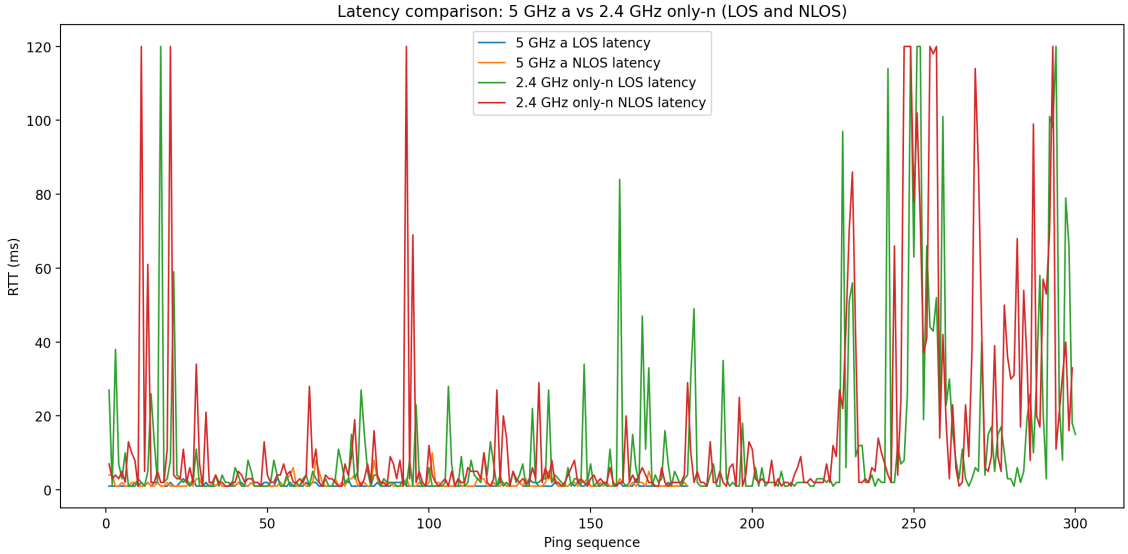

Figure 4.7: Latency comparison for 5ghz-a and 2ghz-onlyn in LOS and NLOS.

throughput and frequency band. In LOS, `5ghz-a/n/ac` achieved the highest average receiver throughput at 44.3 Mbit/s, clearly outperforming `5ghz-a` at 19.7 Mbit/s and `2ghz- onlyn` at 14.6 Mbit/s. However, under NLOS, the advantage of `5ghz-a/n/ac` disappeared. Its throughput dropped to 14.0 Mbit/s, which is lower than `5ghz-a` at 16.1 Mbit/s. The `2ghz-onlyn` configuration achieved 9.35 Mbit/s in NLOS, remaining connected but with the lowest overall data rate. These results explain why

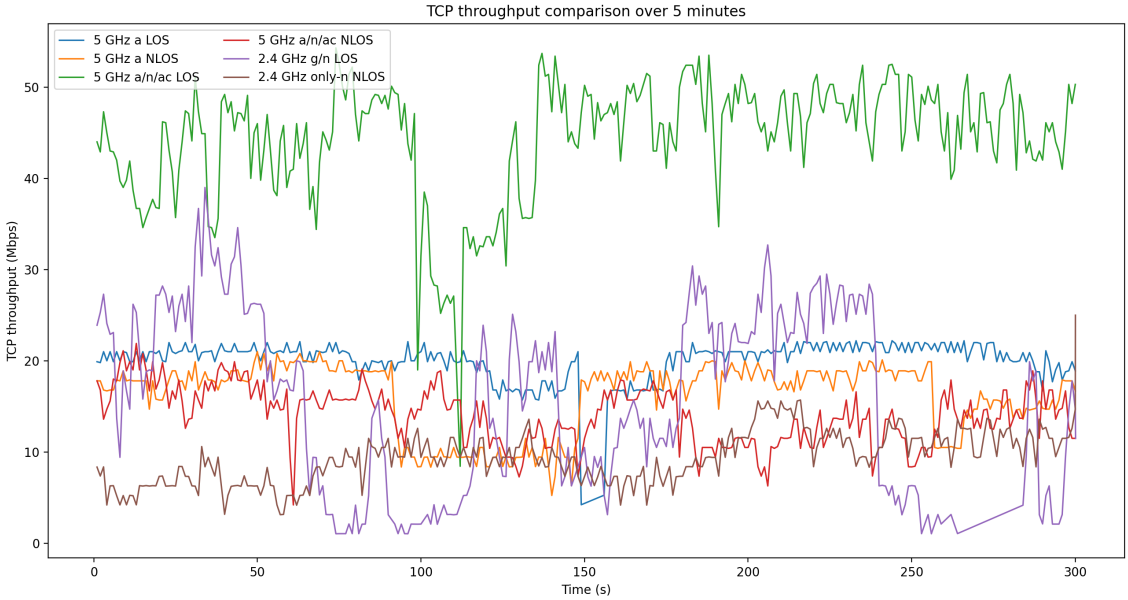

Figure 4.8: TCP throughput for 5ghz-a, 5ghz-a/n/ac, and 2ghz-onlyn.

`5ghz-a` was used as the main configuration for the previous experiments. Although `5ghz-a/n/ac` provides much higher throughput in favorable LOS conditions, its performance worsened more strongly in NLOS. In contrast, `5ghz-a` provided lower peak throughput but more stable behavior across LOS and NLOS, making it a more suitable baseline when the objective is to compare topology and propagation effects. The 2.4 GHz range did not provide an advantage in this setup, despite the expectation of better obstacle penetration, it showed significantly higher latency variation and lower throughput than the 5 GHz range.

The maximum-distance test was carried out at the same outdoor location in Figure 4.5. During movement, signal strength and ping latency were recorded for the `2ghz-onlyn` and `5ghz-a` modes. Both configurations reached a similar maximum communication distance of approximately 230 m. The `2ghz-onlyn` mode showed a slightly longer region, but it remained intermittently reachable near the maximum distance. However, it showed higher latency compared with the `5ghz-a` mode. The latency difference was visible, but not large enough to indicate a major performance gap between the two modes.

Overall, both MikroTik configurations provided similar maximum ranges in this scenario. The higher latency observed at 2.4 GHz, especially at 20 m, is likely due to the lower efficiency of PHY and increased contention sensitivity of the selected mode. In this experiment, `2ghz-onlyn` preserved reachability, but the link quality was suboptimal, it reflected by delay spikes and throughput reduction. Therefore, `5ghz-a` offered the best compromise in this test bed for robotics-oriented communication where it can support both control and payload traffic. This result is also consistent with prior WLAN-based literature in [33], [34].

4.3 Robonode-M

4.3.1 Mobility – Topology Change, Disruption, Reconnection

This experiment evaluates the Robonode mesh behaviour under mobility induced topology change. RSSI logs and RTT logs were collected for both stationary middle node and the moving node. Robonode mesh path snapshots were also recorded to observe route adaptation. Figure 4.9 illustrates the RSSI behaviour during close

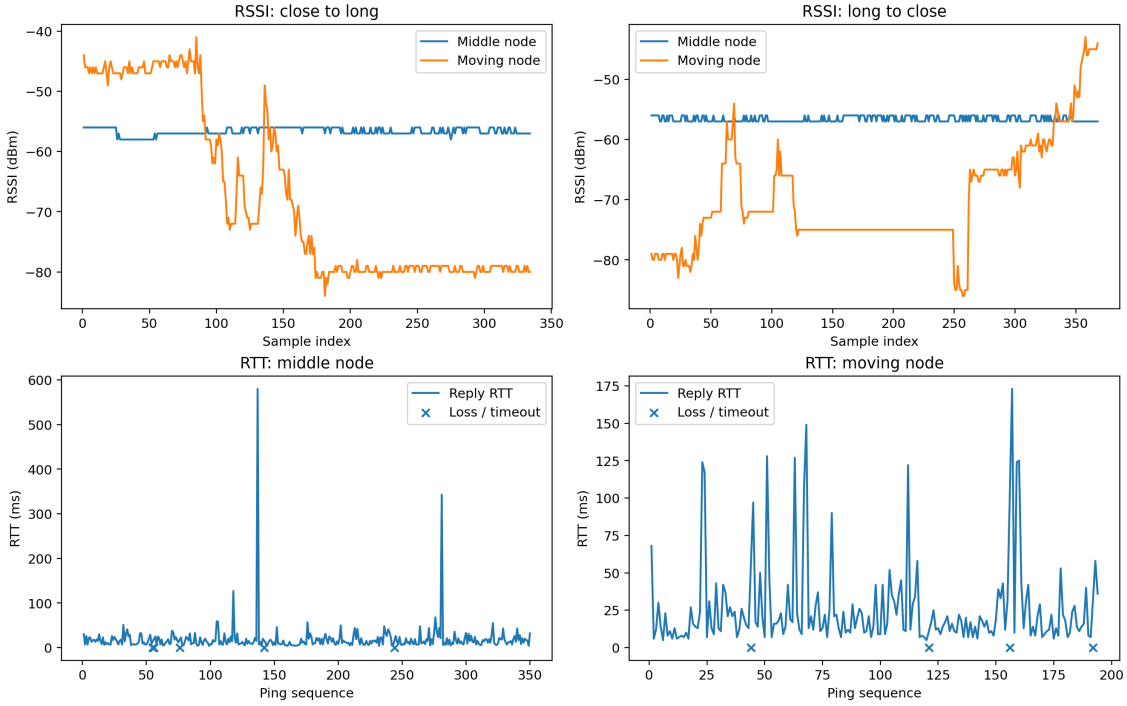

Figure 4.9: Combined results for topology change. The top row shows RSSI variation for the middle node and moving node. The bottom row shows the corresponding RTT measured using continuous ping for each node.

(20 m) to long (60 m) distance and returning. During close (20 m) to long (60 m) distance movement, the moving node shows a clear signal degradation, falling from approximately the mid -40 dBm range to below -80 dBm. In contrast, the middle node remains almost constant around -56 dBm and -57 dBm. This indicates that the fixed middle node provides a stable relay link while the moving node experiences

the main mobility induced degradation. In the long (60 m) to close (20 m) distance movement, the moving node initially starts with weak RSSI values near -80 dBm and then improves as it returns closer. Therefore, the Robonode signal behaviour follows the expected mobility trend, that the moving node is impacted by distance and position, while the middle node remained relatively stable.

Further, in the bottom row of the Figure 4.9 shows the corresponding RTT behaviour. For the moving node, 190 replies were received out of 194 packets, which is approximately 2.06% packet loss. For the middle node, 345 replies were received out of 350 packets, which is approximately 1.43% packet loss. Although occasional RTT spikes and isolated timeouts are visible, there is no extended outage in either case.

```
root@tobufi:~# iw dev wlan0.naw mpath dump
DEST ADDR         NEXT HOP          IFACE       SN      METRIC  QLEN    EXPTIME DTIM    DRET    FLAGS   HOP_COUNT
c4:93:00:57:c0:54 00:00:00:00:00:00 wlan0.naw   0       0       0       0       1600    4       0x4     0       0
c4:93:00:57:bd:c9 c4:93:00:57:bd:c9 wlan0.naw   1681    453     0       0       400     2       0x14    1       5
root@tobufi:~# iw dev wlan0.naw mpath dump
DEST ADDR         NEXT HOP          IFACE       SN      METRIC  QLEN    EXPTIME DTIM    DRET    FLAGS   HOP_COUNT
c4:93:00:57:c0:54 c4:93:00:57:bd:c9 wlan0.naw   49      859     0       0       1600    4       0x4     2       1
c4:93:00:57:bd:c9 c4:93:00:57:bd:c9 wlan0.naw   1681    683     0       2864    400     2       0x15    1       5
root@tobufi:~# iw dev wlan0.naw mpath dump
DEST ADDR         NEXT HOP          IFACE       SN      METRIC  QLEN    EXPTIME DTIM    DRET    FLAGS   HOP_COUNT
c4:93:00:57:c0:54 c4:93:00:57:c0:54 wlan0.naw   71      296     4       0       1600    4       0x6     1       2
c4:93:00:57:bd:c9 c4:93:00:57:bd:c9 wlan0.naw   1681    594     0       2608    400     2       0x15    1       5
root@tobufi:~# iw dev wlan0.naw mpath dump
```

Figure 4.10: The NLOS node connection establishment initial stage, secondly from the middle node after that finally direct connection.

Moreover, the Robonode mesh path dump Figure 4.10 supports the route adaptation. The recorded mesh path entries show changes in next-hop selection and hop count, including cases where a destination changes between one-hop and two-hop paths. This indicates that the Robonode mesh updates its forwarding path during movement rather than losing connectivity completely. Overall, the Robonode results show stable connectivity under mobility. The moving node experiences large RSSI variation, but the ping measurements show only isolated packet loss rather than a sustained link break. Compared with the MikroTik mesh, the Robonode mesh has a better topology adaptation with continuous connectivity.

4.3.2 LOS vs NLOS

For the LOS and NLOS experiment, similar setup was used like MikroTik, but there were some significant differences in the results. It was visible that the connection from base to the NLOS node is very weak, and it tries to connect via the middle node, even though it is expected to connect directly.

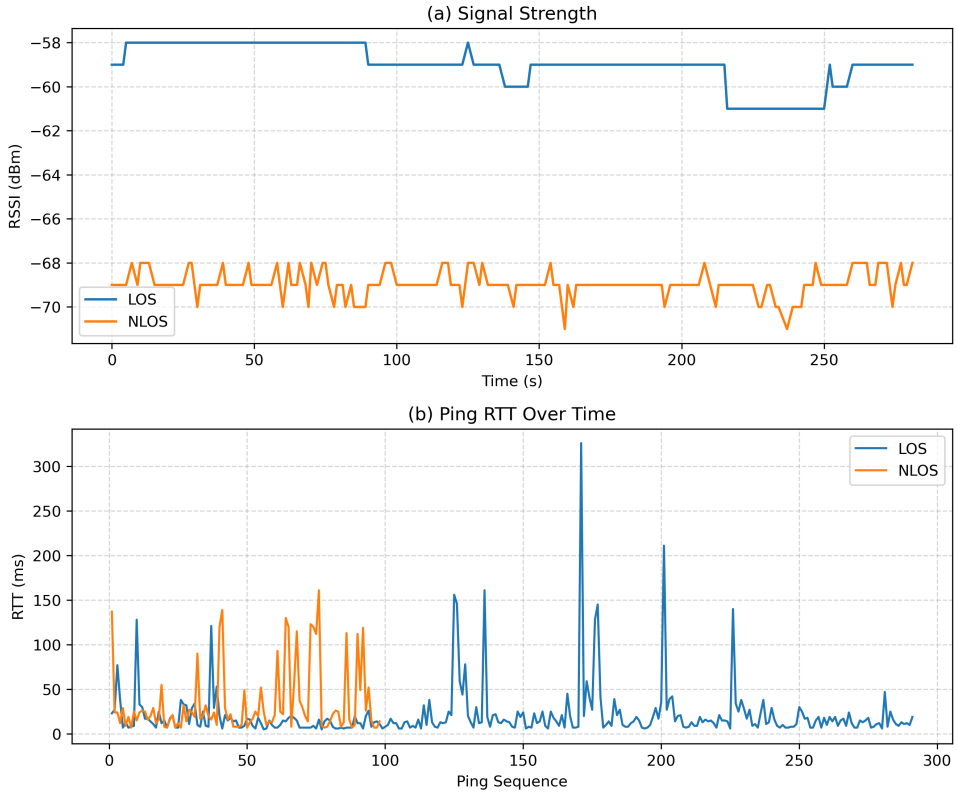

Figure 4.11: Signal strength and RTT over time for LOS and NLOS conditions.

Figure 4.11(a) shows the received signal strength over time. The LOS scenario maintains a strong signal with minor fluctuations, indicating stable propagation conditions. In contrast, the NLOS scenario exhibits a significantly weaker signal level throughout the experiment due to obstruction. The observed reduction in signal strength confirms the impact of physical barriers, which introduce additional path loss and multipath fading. Figure 4.11(b) illustrates the RTT behaviour. The LOS scenario shows low and stable latency with minimal variation over time. In contrast, the NLOS condition exhibits increased variability, with noticeable spikes

and fluctuations. This indicates that obstruction introduces both additional delay and instability in packet delivery, likely due to retransmissions.

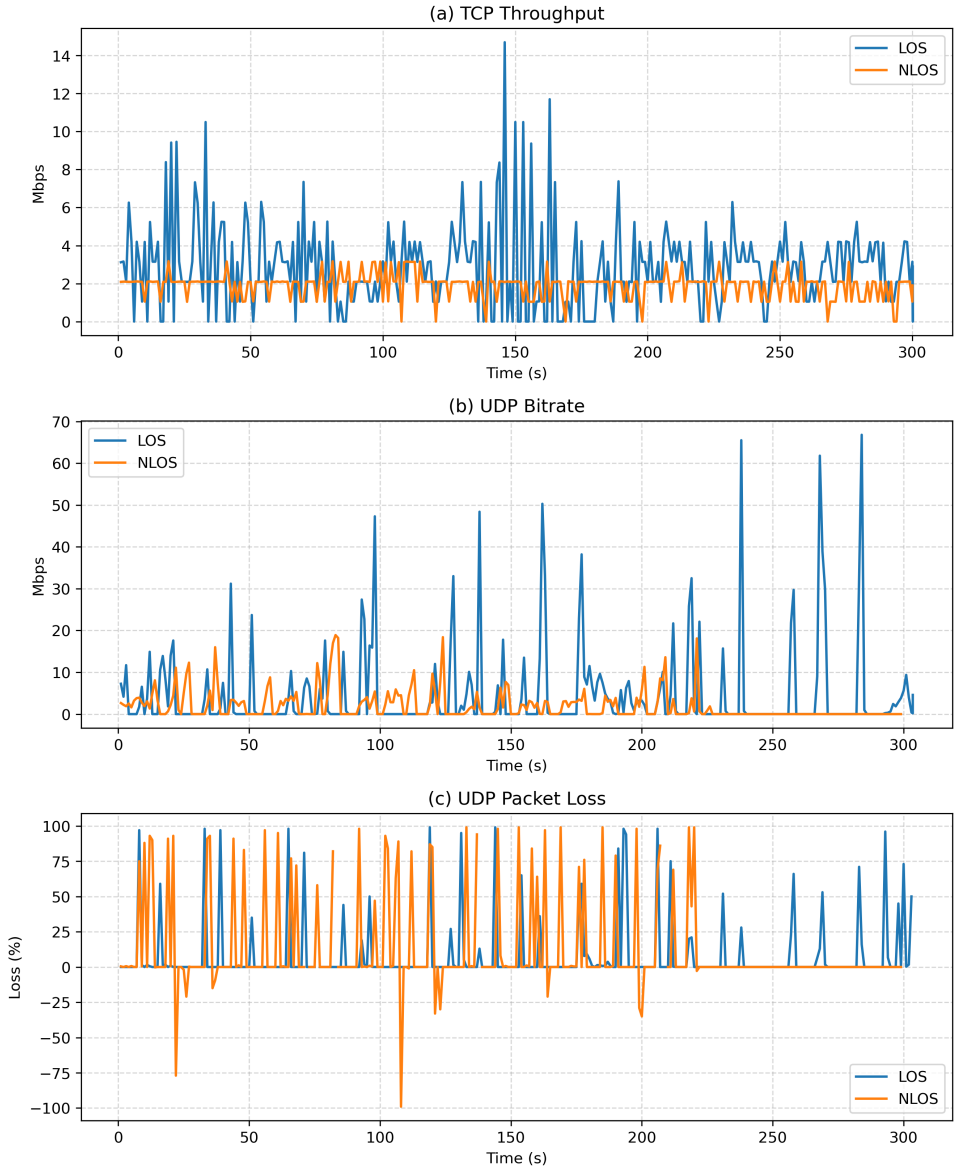

Figure 4.12: TCP throughput, UDP throughput and Packet Loss over time.

Figure 4.12(a) presents TCP throughput over time. The LOS scenario achieves higher throughput, indicating efficient data transmission with minimal retransmissions. In comparison, the NLOS scenario shows reduced throughput. This behaviour suggests that obstruction affects link reliability, leading to more frequent retransmissions and reduced effective throughput. UDP performance is shown in

Figures 4.12(b) and 4.12(c). Under LOS conditions, the bitrate remains relatively high, and packet loss is less compared to NLOS, indicating reliable real-time transmission. However, the NLOS scenario exhibits significant degradation.

The UDP bitrate becomes very low over time, while packet loss increases substantially, reaching approximately 50–60 percentage total loss with burst behaviour. This indicates that a large portion of transmitted packets fail to reach the receiver under obstruction. Such high packet loss is particularly critical for real-time applications, where UDP is commonly used for streaming, control, and telemetry. Compared to MikroTik, both LOS and NLOS behaving very inefficient. Moreover, minor negative UDP packet loss values were occasionally observed in the raw iPerf3 logs. This is a known limitation of the measurement tool, when packets are delivered out-of-order due to dynamic Layer 2 mesh path rerouting, or due to slight clock synchronization drift between the testing endpoints. The results demonstrate that NLOS conditions moderately degrade communication performance across all evaluated metrics.

4.3.3 Distance - RSSI Decay

This experiment evaluates distance-induced degradation in Robonode mesh deployment. Referring to the Figure 4.13, Node A and Node B were fixed nodes, while Node C was moving along Tierankatu road. The A–B separation was approximately 60 m and the maximum distance reached by the moving node was approximately 320 m. At the maximum distance, the link became unstable and then disconnected. Figure 4.14 shows a clear separation between the stable middle node and the degrading moving node link. The middle node RSSI remained nearly constant around -66.2 dBm, confirming that the fixed A–B reference path was stable. In contrast, the moving node RSSI decreased from -50 dBm to -93 dBm. Across the full log, the moving node RSSI had a mean value of -80.7 dBm, a median value of -85.0 dBm,

Figure 4.13: Node A (60.445293, 22.314026), Node B (60.445013, 22.314854), Node C (60.446814, 22.318861). Map adapted from Google Maps © Google and relevant data providers.

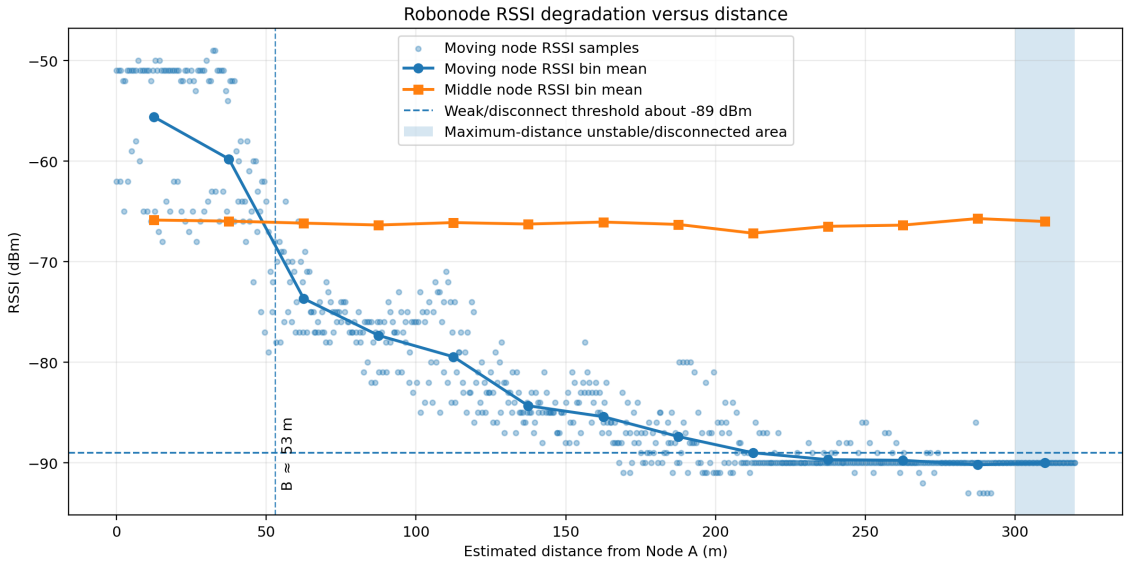

Figure 4.14: Robonode RSSI versus distance.

and the weakest observed value of -93 dBm.

The moving node signal first reached -80 dBm at about 120 m and reached the -89 dBm at about 170 m on the outward movement. After this point, the RSSI stayed around -89 to -90 dBm for a length period. Full summary of the distance and the RSSI, RTT behaviour is shown in the Table 4.5

The important observation is that a visible RSSI value around -89 or -90 dBm did not always mean that the link was usable. Near the maximum distance, the device could still report an RSSI value, but the ping log showed timeouts and destination host unreachable responses. All in all, the experiment shows the new Robonode configuration reached approximately 320 m, which is about 100 m farther than the earlier device configuration. However, the final part of the route was not stable. Once the moving node entered the maximum distance region, latency became bursty and target reachability failed. This confirms that, the range limit is not only determined by an RSSI value, but consistent target replies also requires for reliable communication.

Table 4.5: Summary of the distance, RSSI, and RTT.

Distance bin (m)	Moving RSSI (dBm)	Middle RSSI (dBm)	Median RTT (ms)	P95 RTT (ms)	Unavail. (%)
0–25	-55.6	-65.9	17.0	56.0	0.0
25–50	-59.8	-66.0	13.0	46.0	3.3
50–75	-73.7	-66.2	16.0	43.0	0.0
75–100	-77.3	-66.4	15.0	39.0	0.0
100–125	-79.5	-66.1	12.0	32.1	3.3
125–150	-84.3	-66.3	15.0	23.1	3.3
150–175	-85.4	-66.1	13.0	47.6	3.3
175–200	-87.4	-66.3	16.0	37.0	1.6
200–225	-89.0	-67.2	12.0	35.1	1.6
225–250	-89.7	-66.5	17.0	43.2	3.2
250–275	-89.7	-66.4	20.0	828.2	8.2
275–300	-90.2	-65.7	21.0	1166.4	45.9
300–320	-90.0	-66.0	31.0	1900.8	80.0

4.3.4 QoS Under Simultaneous Traffic Loads

The robonode QoS experiment evaluates the impact of simultaneous traffic with the control channel latency at 20 m under LOS and NLOS conditions. The initial

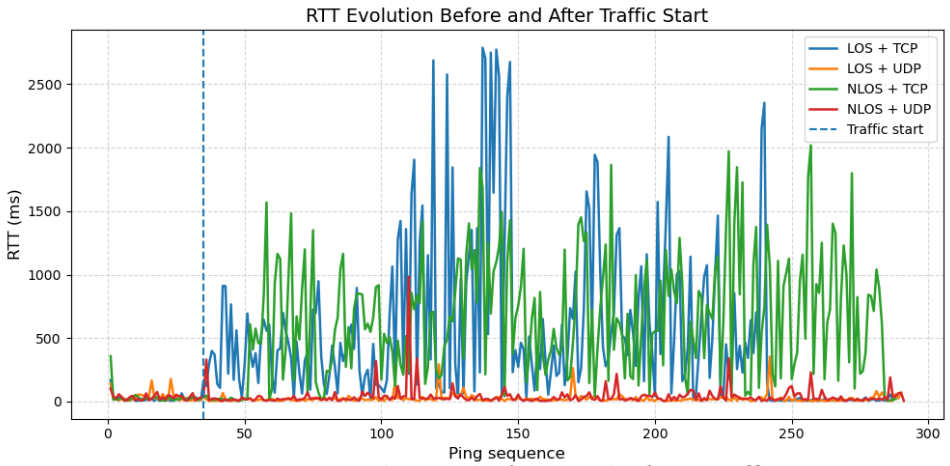

Figure 4.15: RTT evolution before and after traffic start.

low-latency part of the ping log was treated as the idle phase, while the RTT initial raising point was treated as the start of the traffic load. As shown in Figure 4.15, this transition occurs around ping sequence 35 for the LOS scenario and ping sequence 54 for the NLOS case.

Table 4.6: Robonode filtered RTT summary before and after TCP traffic load.

Scenario	**Idle mean (ms)**	**Loaded mean (ms)**	**Loaded max (ms)**	**Loaded std (ms)**	**Timeouts**
LOS + TCP	18.45	528.07	2787	633.46	13
NLOS + TCP	21.20	686.31	2019	463.84	10
LOS + UDP	37.91	23.54	354	35.26	10
NLOS + UDP	34.23	40.02	984	76.96	9

Table 4.6 summarizes the RTT before and after the TCP traffic load. Before the load starts, the idle RTT remains relatively low, with mean values of 18.45 ms in LOS and 21.20 ms in NLOS. After TCP traffic starts, the RTT increases sharply. The loaded mean RTT rises to 528.07 ms in LOS and 686.31 ms in NLOS. The high maximum RTT values and packet timeouts show that the control channel becomes unstable during simultaneous TCP traffic, especially under NLOS conditions. The latency does not impact while maintaining the UDP connection. This was a similar result for the MikroTik as well.

TCP throughput changes minimally when ping traffic is presented, which is

Table 4.7: Robonode TCP throughput summary at 20 m for QoS.

Scenario	Mean TCP (Mbps)	Std TCP (Mbps)	Max TCP (Mbps)
LOS base	3.30	4.92	21.0
LOS + ping	3.20	4.56	21.0
NLOS base	3.17	4.35	10.5
NLOS + ping	3.03	4.67	21.0

shown in Table 4.7. In the LOS test, the mean TCP throughput changes from 3.30 Mbps to 3.20 Mbps, while in the NLOS test, it changes from 3.17 Mbps to 3.03 Mbps. This indicates TCP throughput does not impact by the QOS traffic, but the significant increase in latency shown in Table 4.6. The mesh network set to default bandwidth of 1.4 Mbits in UDP testing. With that setting, the LOS throughput was 1.04 Mbits/sec, jitter was 6.331 ms and the Lost datagrams is 290/27156 (1.1) at receiver side. For the NLOS experiment, throughput was 1.03 Mbits/sec, jitter was 8.635 ms and Lost datagrams were 389/27156 (1.4)at receiver side. With these results, LOS experiment has better results over the NLOS experiment. But, both of them exhibits very low performance compared to the TCP scenarios.

Table 4.8: Distance-based Robonode TCP throughput summary.

Distance	TCP throughput receiver (Mbps)	Sender throughput (Mbps)	Retransmissions	Zero intervals (%)	P95 interval (Mbps)
10 m	3.21	3.28	3458	63.67	10.50
20 m	2.89	3.00	2942	70.33	10.50
50 m	3.00	3.07	2601	58.67	9.91
100 m	2.87	2.95	2780	46.33	10.50
200 m	3.25	3.32	2243	65.67	10.50
300 m	2.29	2.31	263	14.33	4.08

The distance-based QoS analysis evaluates the impact of TCP throughput while the Robonode link distance increases. The test focuses on receiver side TCP throughput and its stability indicators. Table 4.8 summarizes the results at 10 m, 20 m, 50 m, 100 m, 200 m and 300 m. Overall, throughput remained relatively stable across most of the tested range. From 10 m to 200 m, the average receiver side throughput values are 3.21 Mbps, 2.89 Mbps, 3.00 Mbps, 2.87 Mbps, and 3.25 Mbps, respectively. This

indicates that the Layer-2 forwarding path maintained a broadly consistent end to end TCP capacity over these distances. A clearer reduction was observed only at 300 m, where the average throughput decreased to 2.29 Mbps. However, this reduction was moderate rather than a complete collapse, suggesting that distance alone did not strongly affect Robonode mesh throughput in this experiment.

4.3.5 Device Parameters Impact (Frequency Band)

This section compares Robonode mesh performance using 2.4 GHz and 5 GHz links under LOS and NLOS conditions. Latency, reachability and throughput results were analysed for each band and propagation condition.

Table 4.9: Robonode ping latency summary for 2.4 GHz and 5 GHz.

Test	Loss (%)	Avg RTT (ms)	Median RTT (ms)	Max RTT (ms)
2.4 GHz+LOS	0	57	40.5	621
2.4 GHz+NLOS	1	81	59.5	1758
5 GHz+LOS	2	25	13.0	826
5 GHz+NLOS	2	35	20.0	161

Table 4.9 shows that the 5 GHz link had lower average RTT than the 2.4 GHz link in both LOS and NLOS tests. In LOS, the average RTT decreased from 57 ms to 25 ms, which corresponds to change the frequency band from 2.4 GHz to 5 GHz. In NLOS, the average RTT decreased from 81 ms (at 2.4 GHz) to 35 ms (at 5 GHz). However, both 5 GHz tests had 2% ping loss, while the 2.4 GHz LOS test had no loss. The 2.4 GHz NLOS test produced the highest maximum RTT of 1758 ms, indicating a stronger delay spike under obstruction.

Table 4.10: Robonode TCP throughput summary of the receiver-side value result.

Test	Receiver avg	Sender avg	Retra:	Median	Zero intervals(%)
2.4 GHz+LOS	0.765	0.830	1097	0.00	57.33
2.4 GHz+NLOS	0.129	0.136	154	0.00	92.69
5 GHz+LOS	2.890	3.000	2942	3.12	16.28
5 GHz+NLOS	1.900	1.920	655	2.09	3.00

The 5 GHz band provided clearly higher TCP throughput than 2.4 GHz as per Table 4.10. In LOS, the receiver throughput increased from 0.765 Mbps (at 2.4 GHz)

to 2.890 Mbps (at 5 GHz). Under NLOS, the receiver throughput increased from 0.129 Mbps (at 2.4 GHz) to 1.900 Mbps (at 5 GHz). The 2.4 GHz NLOS test was the weakest TCP test, with 92.69% zero-throughput intervals. This indicates, the TCP connection remained active but spent most intervals stalled or waiting for recovery. The 5 GHz LOS test produced the highest retransmission count, which suggests that the link delivered higher throughput but still experienced packet recovery events during the test.

Table 4.11: Robonode UDP throughput, jitter, and datagram loss summary.

Test	Sender	Receiver	Jitter(ms)	Datagram loss(%)	Zero intervals(%)
2.4 LOS	1.05	0.944	17.004	10.00	5.98
2.4 NLOS	1.05	0.245	604.134	74.00	58.63
5 LOS	9.81	4.590	23.884	53.00	60.53
5 NLOS	8.90	1.952	548.679	77.34	53.85

UDP performance depends strongly on both the selected band and the offered load, according to the Table 4.11. At 2.4 GHz, the offered UDP rate was approximately 1.05 Mbps. In LOS, the receiver achieved 0.944 Mbps with 10% datagram loss, while in NLOS the receiver achieved only 0.245 Mbps with 74% datagram loss and 604.134 ms jitter. This confirms that 2.4 GHz NLOS was not suitable for stable UDP delivery at the tested rate. For 5 GHz UDP, the offered load was much higher, approximately 9–10 Mbps. The 5 GHz LOS test achieved the highest UDP receiver throughput of 4.590 Mbps, but it also had 53% datagram loss. The 5 GHz NLOS UDP value achieved approximately 1.952 Mbps with an estimated 77.34% datagram loss. Even though 5 GHz provided higher received throughput compared to 2.4 GHz, it has a significant packet loss, especially in NLOS.

Overall, the Robonode tests show that the 5 GHz configuration provided lower latency and higher TCP throughput than the 2.4 GHz configuration in the tested setup. However, UDP performance was more sensitive to the offered rate and link condition. The strongest overall performance was observed in the 5 GHz LOS scenario in terms of throughput and latency, while the weakest performance occurred

in the 2.4 GHz, NLOS scenario in TCP and the 2.4/5 GHz NLOS scenarios in UDP.

The maximum communication distance was evaluated separately for the two frequency bands. At 2.4 GHz, the maximum reachable distance was approximately 150 m, whereas at 5 GHz the link remained reachable up to approximately 320 m. Thus, the 5 GHz configuration achieved a substantially longer usable communication range in this experiment. The ping measurements further showed that the 2.4 GHz link experienced higher latency than the 5 GHz link. Although the 2.4 GHz RSSI remained relatively strong up to the disconnection point, the link became unstable and disconnected at a shorter distance. This indicates that RSSI alone did not fully explain the observed communication quality. Other factors, such as interference, channel congestion, link-layer retransmissions, and routing behaviour may have contributed to the higher latency and earlier disconnection. Overall, the 2.4 GHz link showed shorter maximum range, higher latency, and lower stability, whereas the 5 GHz link provided longer communication distance and better latency performance, reaching approximately 320 m before disconnection.

4.4 P2P and Mesh Combined Evaluation

This section evaluates a 5 GHz Ubiquiti AirMax P2P deployment consisting of one AirMax base station and two client stations, namely a LightBeam and a Nano Loco. The purpose of this evaluation is not to replace the mesh-network study, but to provide a controlled baseline for understanding how a conventional P2P architecture behaves under different spatial and channel conditions. This evaluation focuses on throughput, latency, packet loss, and fairness when the two client links are tested individually and when both are active simultaneously. The P2P study improves the overall reliability of this thesis by covering both conventional infrastructure and mesh-based communication. This is important in riverine monitoring, where some sections of a river may support direct P2P connectivity, while other sections may

require multi-hop forwarding.

4.4.1 P2P Evaluation Scenarios

For each test condition, measurements were taken using TCP throughput, UDP throughput at 100 Mbit/s, UDP packet loss, UDP jitter, and ICMP ping latency. Each test was performed in two modes such as Separate, where only one client link was active at a time. Together, where both client links were active simultaneously in order to evaluate contention and fairness at the base station.

A P2P architecture is optimal when LOS connectivity is available because it is simple to deploy, provides high capacity and easily connectable. The P2P experiments in this chapter were designed in three scenarios, summarized in Table 4.12.

Table 4.12: Summary of the P2P test scenarios.

Test	LightBeam position	Nano position	Visibility	Purpose
Test 1	1 m from base station	1 m from base station	LOS	Baseline comparison under favorable conditions and simultaneous link contention
Test 2	20 m from base station	1 m from base station	LOS	Evaluate the effect of distance asymmetry under LOS conditions
Test 3	20 m from base station	1 m from base station	NLOS:LightBeam, LOS:Nano	Evaluate the effect of obstruction and asymmetric channel quality under simultaneous access

Figure 4.16(a) shows that TCP throughput is relatively balanced in Test 1, becomes asymmetric in Test 2, and becomes highly unbalanced in Test 3. Figure 4.16(b) shows that UDP packet loss remains negligible in the baseline test, increases substantially for the longer LightBeam link in Test 2, and becomes extremely high under NLOS conditions in Test 3. Figure 4.16(c) shows that the latency variation follows the same trend, with the strongest instability appearing for the NLOS LightBeam link.

The summarized results of the P2P network is shown in Table 4.13. These results show that a direct P2P architecture can provide good performance when channel conditions are favourable and the links remain short and unobstructed. However,

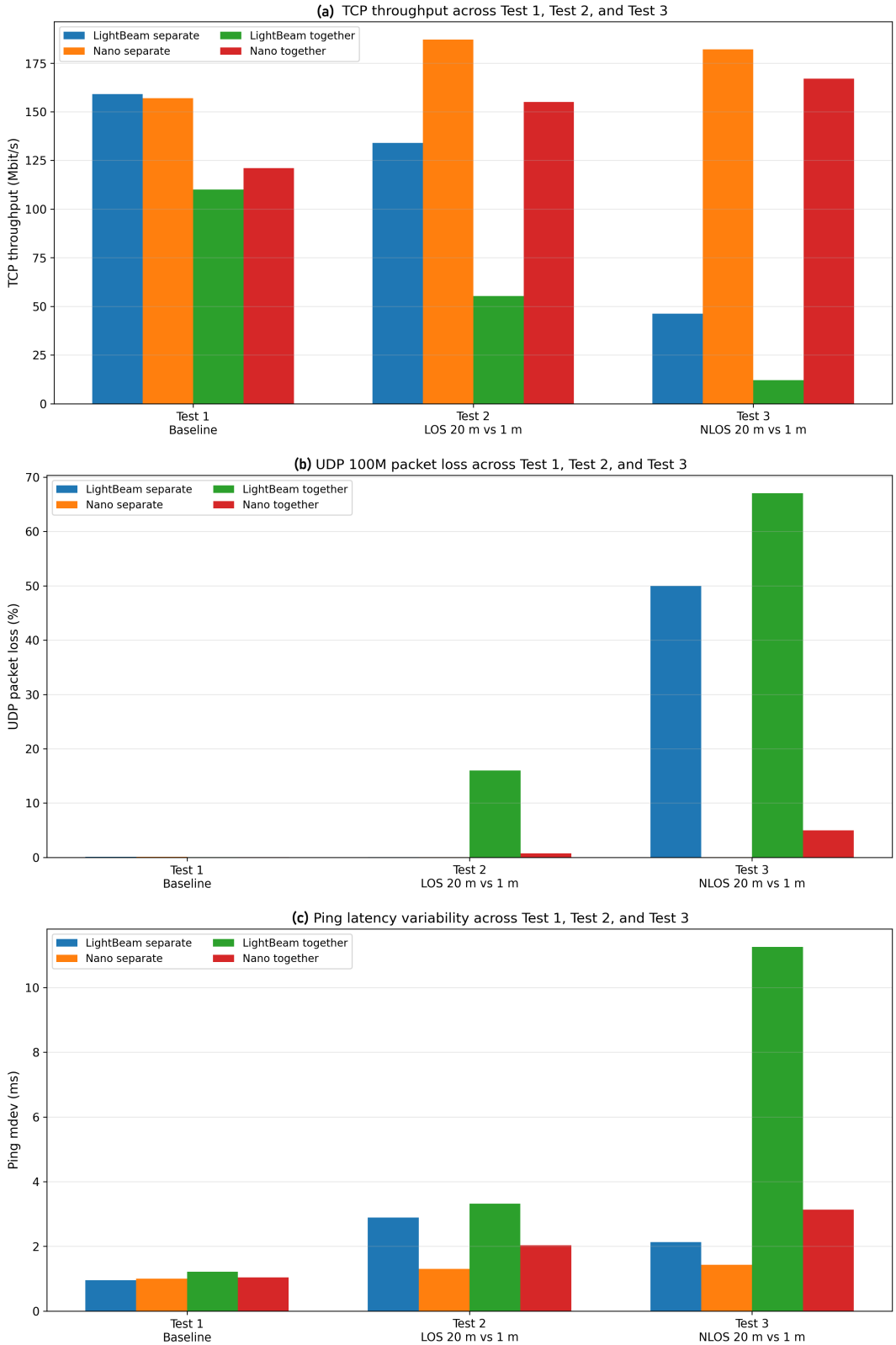

Figure 4.16: (a) TCP throughput, (b) UDP packet loss and (c) Ping latency variability expressed as mdev across Test 1, Test 2, and Test 3 for the LightBeam and Nano Loco under separate and together operating modes.

its behaviour becomes increasingly unfair when one client is farther away, and it becomes highly susceptible when one of the links is NLOS. In riverine environments, P2P is not usable as NLOS is the common scenario.

Table 4.13: Summary of key P2P performance results across the three tests.

Test	Scenario	Device	TCP (Mbit/s)	UDP 100M (Mbit/s)	UDP loss (%)	Ping avg (ms)	Ping mdev (ms)
Test 1	Separate	LightBeam	159.0	99.9	0.091	5.802	0.957
	Separate	Nano	157.0	99.9	0.130	6.294	1.005
	Together	LightBeam	110.0	99.9	0.032	5.760	1.211
	Together	Nano	121.0	100.0	0.026	6.439	1.042
Test 2	Separate	LightBeam	134.0	99.9	0.062	7.859	2.894
	Separate	Nano	187.0	99.9	0.055	6.128	1.306
	Together	LightBeam	55.4	83.8	16.000	7.920	3.322
	Together	Nano	155.0	99.2	0.760	6.203	2.037
Test 3	Separate	LightBeam	46.3	50.3	50.000	7.155	2.131
	Separate	Nano	182.0	99.9	0.056	5.862	1.428
	Together	LightBeam	12.1	32.7	67.000	8.762	11.248
	Together	Nano	167.0	95.0	5.000	6.143	3.136

Therefore, although P2P links are suitable for short, stable, and unobstructed backhaul segments, they are not sufficiently resilient for all riverine scenarios. In contrast, a mesh network can divide the communication path into shorter hops and provide alternate forwarding paths around obstructions. For this reason, the P2P evaluation in this chapter supports the motivation for mesh-network architecture.

4.4.2 Combined P2P Backhaul and Mesh Access Layer

The hybrid architecture shown in Figure 4.17 demonstrates that P2P and mesh networking should not be viewed as mutually exclusive solutions. Instead, they can complement each other, the P2P segment can serve as a high-capacity backhaul wherever a stable LOS path exists, while the mesh segment can provide flexible local access and maintain connectivity in areas where direct visibility is not guaranteed. Although the hybrid setup was evaluated only as a feasibility study in this thesis, it indicates that such an approach may be useful for future riverine deployments that require both bandwidth and coverage flexibility.

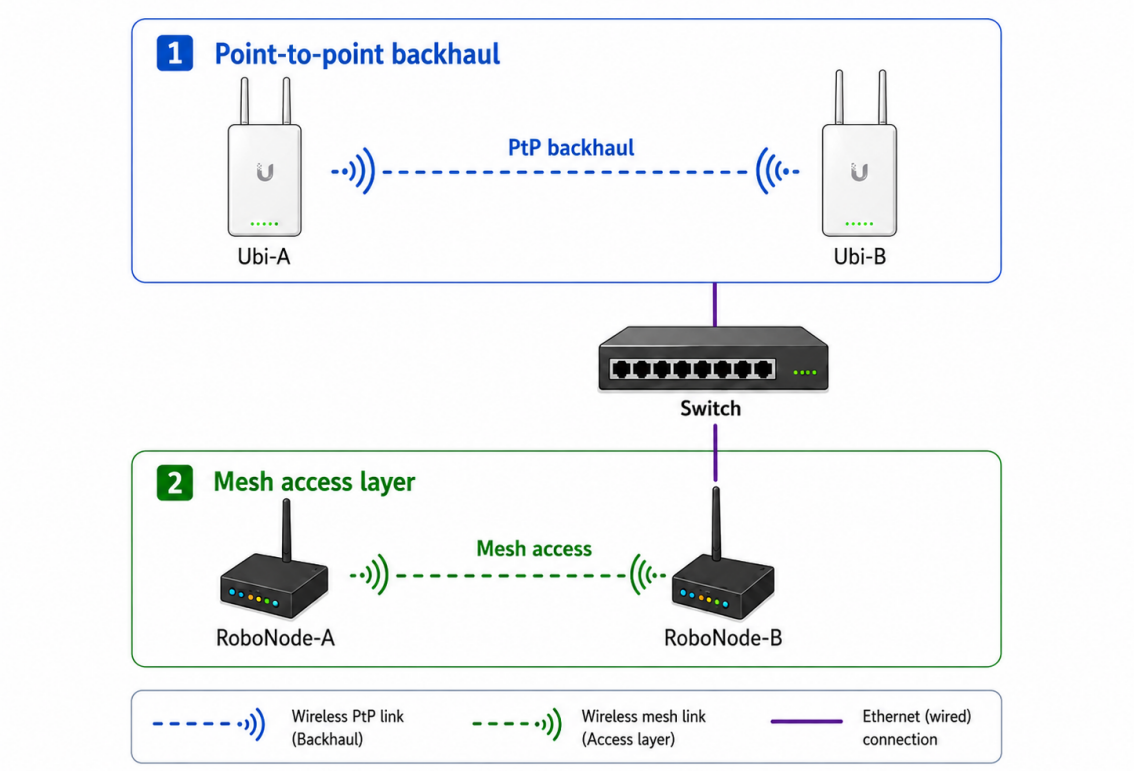

Figure 4.17: Hybrid communication architecture.

5 System-Level Robotic Sensor Data Transmission Evaluation

The previous communication benchmarks evaluated the radio links using network-level metrics such as ping latency, TCP/UDP throughput, packet loss, RSSI, and mesh forwarding behaviour. This section extends the evaluation to robotics-level validation by transmitting real ROS sensor topics from a drone/edge-computing platform to a receiver laptop through three communication architectures: MikroTik mesh, Robonode mesh, and Ubiquiti P2P backhaul. The objective is to determine whether the communication behaviour observed in the network benchmarks is reflected at the application layer when real robotic payloads are transmitted.

Three representative sensor streams were selected, they are Livox LiDAR point cloud, RealSense colour image stream, and ZED2i 4K camera image stream. These streams cover different payload sizes and timing requirements. LiDAR represents medium rate to high rate perception data, the RealSense camera represents image-based monitoring, and the 4K camera represents a high-bandwidth visual payload. The experiment evaluates not only whether packets can pass through the network, but whether the received ROS topics remain usable for monitoring and robot operation.

5.1 Experimental Measurement Procedure

The sensor platform was connected to the communication node and published ROS 1 Noetic topics. The receiver laptop subscribed to the topics via the wireless link and measured the topic rate, bandwidth, and delay using standard ROS tools, `rostopic hz`, `rostopic bw`, and `rostopic delay`. In addition, selected topics were recorded into rosbag files at the receiver side. The rosbag timestamps were then analysed by comparing each message header timestamp with the bag receive time, producing a direct estimate of end-to-end application-layer latency.

Before latency measurements, clock synchronization was checked using `chrony`. This step is essential because the rosbag latency calculation depends on the difference between the message header timestamp produced at the sender side and the received timestamp recorded by the receiver.

Table 5.1: ROS sensor topics used in the system-level validation.

Sensor stream	**ROS topic**	**Message type**	**Approx. message size**	**Main measurement purpose**
Livox LiDAR	`/livox/lidar`	PointCloud2	0.64 MB	Point-cloud delivery, rate stability, perception latency
RealSense colour camera	`/camera/color/ image_raw`	Image	6.22 MB	Raw image delivery and monitoring latency
ZED2i 4K camera	`/zed2i/zed_node/left/ image_rect_color`	Image	10.97 MB	High-bandwidth visual payload feasibility

5.2 Live ROS Topic Monitoring Results

Table 5.2 summarizes the live ROS topic measurements. The bandwidth values are reported in MB/s, which is taken from `rostopic bw`. The live `rostopic delay` column is treated as an online diagnostic, while the rosbag-based latency analysis in the next section is used as the main quantitative latency result. The live measurements show a clear separation between the architectures. For the LiDAR stream, MikroTik and Ubiquiti both delivered approximately 6.2 MB/s, corresponding to

Table 5.2: Live ROS topic measurements from rostopic bw, rostopic hz, and rostopic delay.

Sensor/topic	Msg size	Architecture	BW MB/s	BW Mbps	Rate Hz	Delay s	Status
LiDAR	0.64 MB	MikroTik	6.25	50.0	9.69	0.259	stable
LiDAR	0.64 MB	Robonode	0.401	3.2	0.289	9.331	severely delayed
LiDAR	0.64 MB	Ubiquiti	6.21	49.7	7.90	0.199	stable
RealSense camera	6.22 MB	MikroTik	6.52	52.2	1.079	1.410	stable
RealSense camera	6.22 MB	Robonode	0.182	1.5	–	84.074	no stable hz
RealSense camera	6.22 MB	Ubiquiti	7.09	56.7	1.156	1.301	stable
ZED2i 4K camera	10.97 MB	MikroTik	9.11	72.9	0.699	2.211	usable, delayed
ZED2i 4K camera	10.97 MB	Robonode	–	–	–	–	not feasible
ZED2i 4K camera	10.97 MB	Ubiquiti	18.54	148.3	1.117	1.344	best

about 50 Mbps, while Robonode delivered about 0.401 MB/s and the received rate dropped below 0.3 Hz. The RealSense camera produced a similar pattern, such that MikroTik and Ubiquiti sustained approximately 6.5 –7.1 MB/s, whereas Robonode did not provide a stable message rate and observed an excessive delay. For the 4K camera, Ubiquiti delivered the highest live bandwidth at 18.54 MB/s, while MikroTik remained usable but slower rate at 9.11 MB/s. Robonode was infeasible for 4K camera stream in this setup. Figure 5.1 shows the 4K camera image, taken of Ubiquity setup.

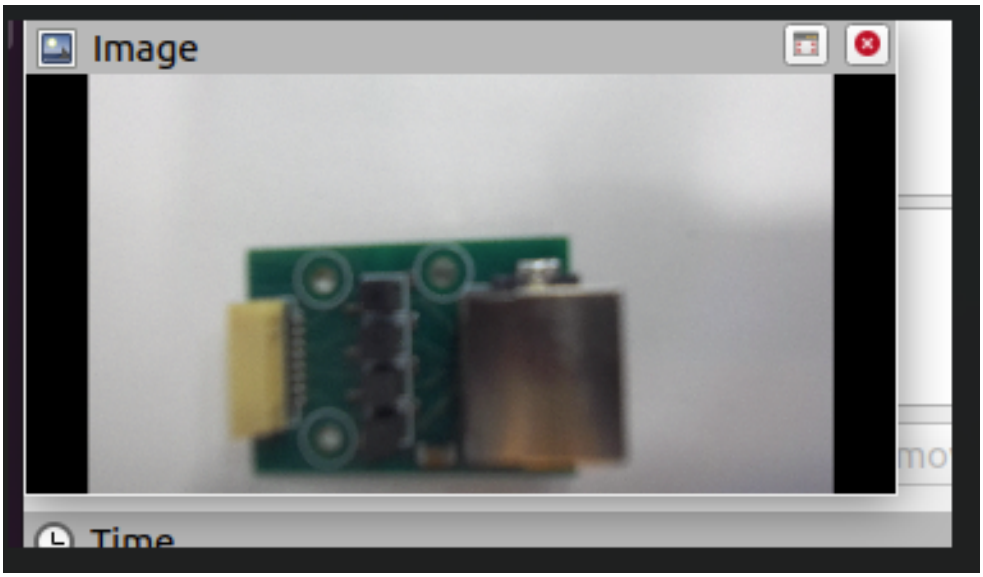

Figure 5.1: ZED2i 4K camera image visualised in RViz during the transmission test.

5.3 Rosbag-Based Latency Analysis

The rosbag analysis provides a more reliable view of end to end application latency because it uses the timestamp embedded in each ROS message and the received time recorded in the bag file. Two recording durations were used where possible, approximately 30 s and 60 s, to check whether the measured behaviour remained consistent over a longer window.

Table 5.3: Rosbag-based application-layer latency for sensor streams.

Sensor	**Architecture**	**Time**	**Samples**	**Average latency (ms)**
LiDAR	MikroTik	30 s	290	207.92
LiDAR	MikroTik	60 s	605	304.47
LiDAR	Robonode	30 s	11	19337.58
LiDAR	Robonode	60 s	31	20970.44
LiDAR	Ubiquiti	30 s	286	246.93
LiDAR	Ubiquiti	60 s	600	197.70
RealSense camera	MikroTik	30 s	32	1379.13
RealSense camera	MikroTik	60 s	71	1351.79
RealSense camera	Ubiquiti	30 s	39	1360.58
RealSense camera	Ubiquiti	60 s	83	1147.30
4K camera	MikroTik	30 s	30	1943.03
4K camera	MikroTik	60 s	41	1977.64
4K camera	Ubiquiti	30 s	45	1003.85
4K camera	Ubiquiti	60 s	101	901.98

The LiDAR stream is the most important perception topic because it represents spatial sensing rather than 2D visual monitoring. MikroTik and Ubiquiti both supported the LiDAR topics at a usable rate. In the 60 s recordings, MikroTik received 605 samples with an average latency of 304.47 ms, while Ubiquiti received 600 samples with an average latency of 197.70 ms. The sample counts are close to the expected order of magnitude for the measured 8–10 Hz LiDAR rate, which indicates that both links were able to carry the point cloud stream continuously.

Robonode behaved very differently. Only 11 samples were captured in the shorter

recording and 31 samples in the longer recording, with average latencies of 19.34 s and 20.97 s respectively. This confirms that the Robonode link was not suitable for this uncompressed LiDAR topic in the tested configuration, even though the topic could technically be received. The link appears to buffer and deliver messages slowly rather than dropping the topic immediately, which is undesirable for real-time perception.

The RealSense image topic produced a larger per-message payload than the LiDAR stream. MikroTik and Ubiquiti both supported the stream as a low-rate visual-monitoring channel. The 60 s recordings produced average latencies of 1351.79 ms for MikroTik and 1147.30 ms for Ubiquiti. These values are too high for tight closed-loop visual control, but they are usable for low-rate operator awareness or delayed monitoring. Robonode did not provide a stable RealSense stream in this configuration. Therefore, it was excluded from the rosbag latency comparison for this sensor.

The 4K image stream was the heaviest single stream. Ubiquiti was clearly stronger for this payload, achieving a 60 s average latency of 901.98 ms compared with 1977.64 ms for MikroTik. This result is consistent with the live bandwidth results, where Ubiquiti delivered approximately twice the 4K topic bandwidth of MikroTik. MikroTik can still carry the stream, but the resulting delay indicates that it is more appropriate for delayed monitoring than for time-critical control. Robonode was not feasible for the 4K stream.

5.4 4K Camera and LiDAR Combined

A combined stream test was performed to evaluate whether the network could support simultaneous perception data. In this case, the 4K camera and LiDAR topics were recorded together. This is closer to a realistic robotic monitoring scenario, where a robot may transmit both a visual payload and spatial sensing data at the

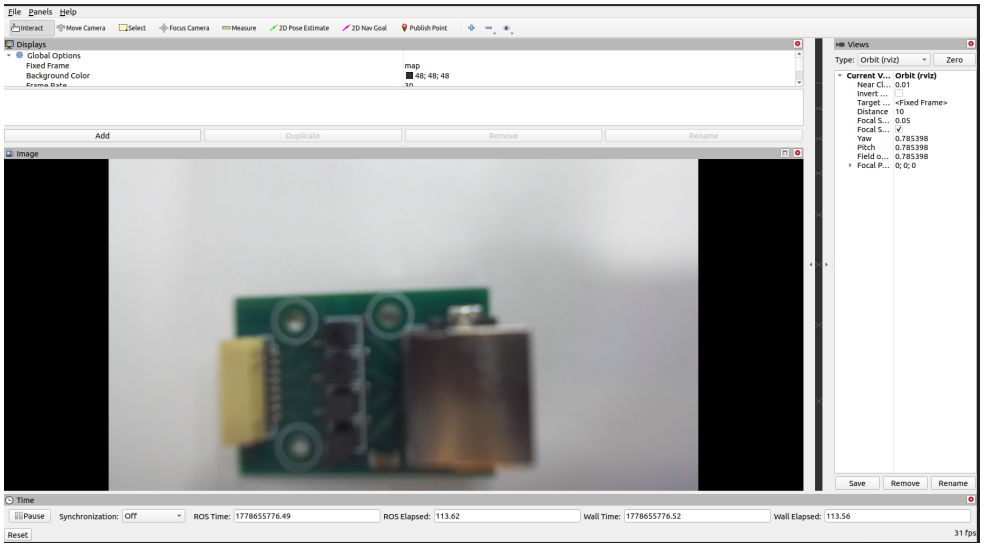

Figure 5.2: 4K camera image received through the Ubiquiti P2P architecture.

same time.

Table 5.4: Combined 4K camera and LiDAR rosbag latency.

Architecture	**Recording**	**Stream**	**Samples**	**Average latency (ms)**
MikroTik	30 s	4K camera	10	4529.05
MikroTik	30 s	LiDAR	262	6463.00
MikroTik	60 s	4K camera	19	3852.98
MikroTik	60 s	LiDAR	383	12667.61
Ubiquiti	30 s	4K camera	17	2399.43
Ubiquiti	30 s	LiDAR	201	7863.53
Ubiquiti	60 s	4K camera	28	2977.62
Ubiquiti	60 s	LiDAR	547	1718.33

The combined test shows that simultaneous high-payload transmission is much more demanding than individual topic transmission. For MikroTik, the 4K camera latency increased to approximately 3.85–4.53 s, while the LiDAR latency increased to 6.46–12.67 s. This indicates strong queueing and contention when both streams share the same mesh link. For Ubiquiti, the 4K latency was lower than MikroTik in the combined case, ranging from 2.40 s to 2.98 s. The LiDAR result was varying, the 30 s run produced 7.86 s average latency, while the 60 s run produced 1.72 s. This difference suggests that the combined pipeline was sensitive to transient buffer-

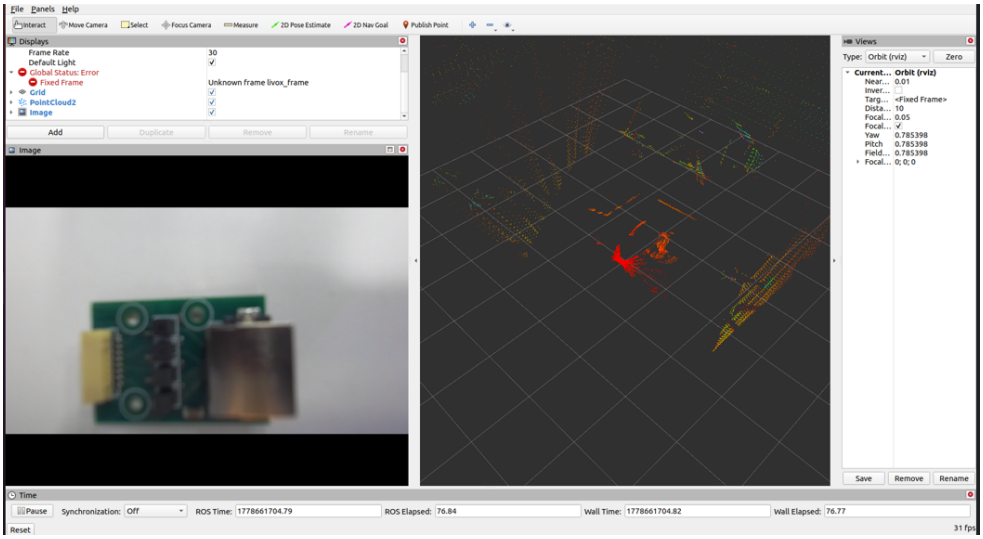

Figure 5.3: Simultaneous LiDAR point cloud and 4K camera visualisation in RViz.

ing, startup effects, and stream scheduling. The key finding is therefore that one combined-run value is absolute, but that simultaneous uncompressed visual and LiDAR streams create large application-layer delay even when individual streams are feasible.

5.5 Results of Drone Flying and Evaluation

After the static and ground-level benchmarks, the communication setup was validated in flight. The Robonode and Ubiquiti radios were mounted on the drone with careful attention to the power distribution and the antenna alignment discussed in the previous section, since both factors directly affect link quality once the platform is airborne. During the flight, the Livox LiDAR point-cloud topic was published from the on-board computer and recorded into rosbag files at the receiver. The application-layer latency was then computed from the rosbag timestamps for two conditions, before flying (static, on the ground) and during flying. The Ubiquiti link provides much higher bandwidth than the Robonode mesh, the full point-cloud path was additionally recorded over the Ubiquiti link, where it could be captured and visualized in real time well enough to confirm the spatial data product.

Table 5.5: Livox LiDAR point-cloud application-layer latency over the Robonode mesh, before and during flight. Latency is computed from rosbag header and receive timestamps.

Metric	**Before flying**	**During flying**
Samples	28	52
Minimum latency (s)	1.55	6.48
Average latency (s)	18.37	37.51
Median latency (s)	19.41	37.53
95th percentile (s)	31.88	47.36
Maximum latency (s)	33.60	48.43

Table 5.5 shows that the LiDAR latency over the Robonode mesh was already high on the ground (about 18.4 s average) and roughly doubled during flight (about 37.5 s average). The during-flight median was almost identical to the average (37.53 s versus 37.51 s), indicating a consistently high delay rather than a few isolated spikes, whereas the before-flight distribution was slightly more spread. The low sample counts (28 and 52) are themselves evidence of the problem. The Robonode mesh did not drop the topic, but buffered and delivered point clouds far more slowly than they were produced. These results confirm that the Robonode mesh at the flight level and in its current configuration cannot deliver an uncompressed LiDAR stream at a usable real-time rate.

By contrast, the higher-bandwidth Ubiquiti link was able to carry the high bandwidth point-cloud path and reconstruct it for real-time visualization, as shown in Figure 5.4. This reinforces the layered conclusion of this thesis, the mesh is valuable for maintaining connectivity during flight and topology change, while high-bandwidth perception payloads such as the LiDAR point cloud should be carried over a directional backhaul whenever LOS available.

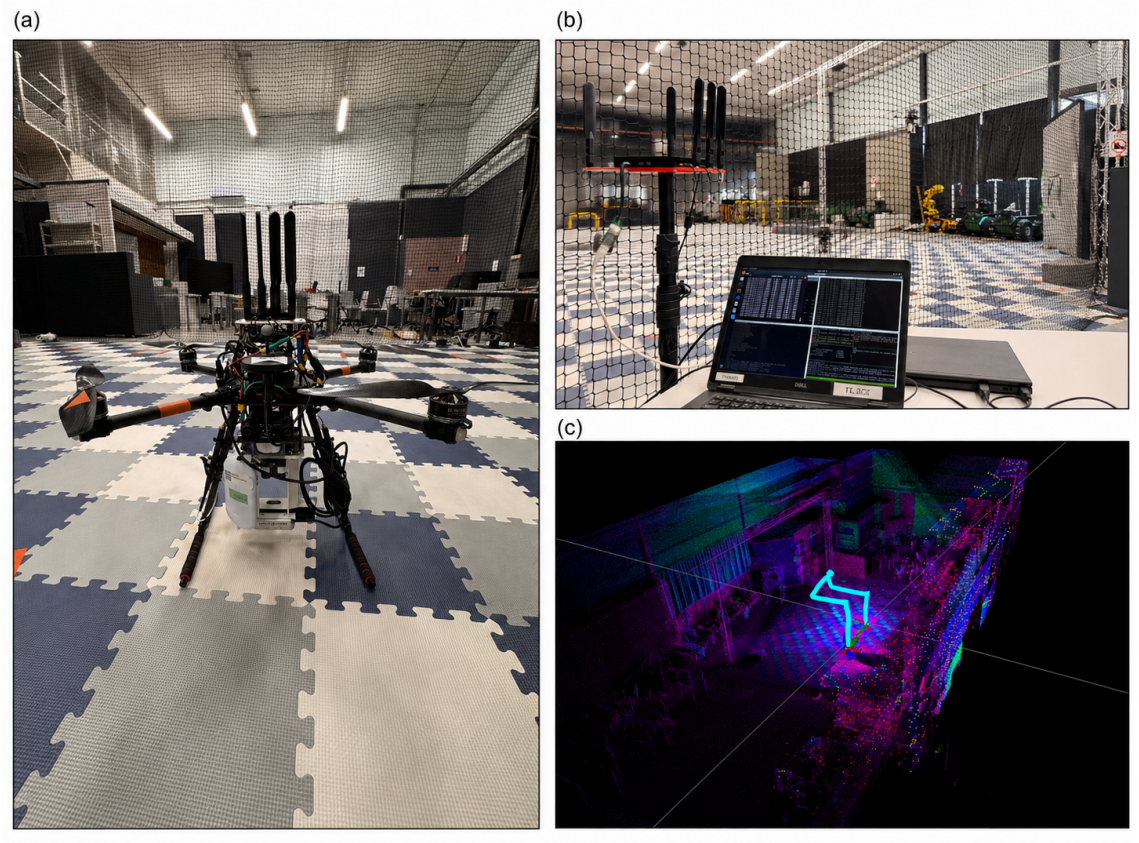

Figure 5.4: System-level drone communication evaluation. (a) UAV platform with onboard sensing and communication payload. (b) Ground-station setup used for monitoring, logging, and communication during the indoor flight test. (c) RViz visualization of the received point-cloud map and flight path.

5.6 Architecture-Level Interpretation

5.6.1 MikroTik Mesh

The MikroTik mesh provided the best balance among the mesh-based solutions. It supported all three individual sensor streams and delivered the LiDAR topic with sub-second rosbag latency. The camera topics were also receivable, although their latencies were extended. This makes MikroTik suitable for telemetry, point-cloud monitoring, low-rate camera inspection, and non-critical operator visual feedback. However, when the 4K stream and LiDAR were transmitted together, the delay increased sharply, showing that the mesh should not be used as an unlimited high-

bandwidth backbone without compression or stream prioritization.

5.6.2 Robonode Mesh

Robonode did not perform well for the uncompressed application-layer streams used in this test. Although previous network-level tests showed that Robonode can maintain connectivity and can be useful for longer-distance or dynamic mesh behaviour. This ROS payload test shows that the available application throughput was too low for raw LiDAR and camera streams. The LiDAR topic was received with tens of seconds of delay, and camera streams were not usable. This does not mean that Robonode is unsuitable for all robotic communication, but it suggests that it is better suited to low-rate telemetry, command messages, status updates, or compressed payloads unless its configuration is further optimized.

5.6.3 Ubiquiti P2P Backhaul

Ubiquiti provided the strongest application-layer performance for high-bandwidth streams. It delivered the lowest 60 s LiDAR latency, the lowest 4K camera latency, and the highest live 4K bandwidth. These results confirm that a directional P2P link is preferable when the communication path is static and LOS can be maintained. However, this strength must be interpreted together with the earlier mobility experiments P2P links are less flexible when robots move behind obstacles or require topology adaptation. Therefore, Ubiquiti is best treated as a high-throughput backhaul rather than as a replacement for mesh in all parts of a mobile riverine system.

5.7 Findings for Riverine Monitoring

The application-level tests support a layered communication design. Mesh communication is valuable for maintaining connectivity during mobility and topology

changes, but high-resolution raw sensor data can quickly exceed the practical capacity of a low-power mesh. A directional P2P backhaul is more appropriate for high-volume payloads such as 4K video, while mesh is more appropriate for local access, robot-to-robot connectivity, telemetry, and fallback communication. A combined architecture using Ubiquiti as a backhaul and MikroTik or another mesh solution as a local access layer is therefore the most practical interpretation of the results.

For real riverine deployment, the results imply that raw sensor streaming should be used selectively. LiDAR transmission is feasible over MikroTik and Ubiquiti when sent alone, but simultaneous raw LiDAR and 4K image transmission creates unacceptable delay. Therefore, a deployed system should use one or more of the following strategies, such as image compression, lower frame rate, region-of-interest streaming, local edge processing on the robot, event-triggered upload, or prioritization of command and telemetry traffic over bulk sensor streams.

Table 5.6: Practical deployment interpretation of the application-level results.

Architecture	**Best use in riverine robotics**	**Main limitation observed**
MikroTik mesh	Mobile mesh access, LiDAR monitoring, low-rate camera feedback, relay-based continuity	Large delay when multiple raw high-bandwidth streams are combined
Robonode mesh	Low-rate telemetry, command/status messages, dynamic connectivity experiments	Raw LiDAR/camera topics became severely delayed or unusable in this configuration
Ubiquiti PtP	High-throughput backhaul for 4K camera and sensor upload when LOS is available	Less suitable for mobile topology changes and obstruction without additional relay/mesh support
Combined architecture	Ubiquiti backhaul plus local mesh access for mobile robots	Requires traffic management and careful placement of backhaul and mesh nodes

The tests were conducted at short distance compared with the outdoor mobility experiments. Therefore, these results should be interpreted as application-layer capacity tests under controlled connectivity, not as maximum-range robotic deploy-

ment trials. The results provide important evidence because they connect the communication benchmarks to real robotic payloads and show which architectures can support actual ROS topic transmission.

6 Discussion

This chapter interprets the experimental results presented in the evaluation chapters, returning the analysis to the research questions formulated in the Introduction. Rather than restating individual numbers, the goal is to reason about what the measurements imply for the suitability of Wi-Fi mesh communication architecture in a multi-robot riverine monitoring system. The primary objective of this thesis, is to evaluate the effectiveness of a Wi-Fi mesh communication architecture for multi-robot riverine monitoring, particularly in infrastructure-limited environments. This chapter is structured around the central finding of this thesis, which is the Wi-Fi mesh communication architecture can provide the necessary reliability and adaptability to meet the specific operational demands, though careful architectural design is required to manage high-bandwidth payloads. The two mesh platforms exhibit complementary strengths, the P2P baseline is useful for high-bandwidth payloads but restrict the mobility, and a hybrid switched configuration combining these two, recovers most of the deficits of each.

6.1 Mobility, NLOS, and Network-Level Behaviour

Mobility and Topology Adaptation – A critical requirement for riverine MRS is the ability to maintain connectivity despite continuous movement and changing formations. The experiments demonstrated that mesh architecture successfully handles mobility-induced topology changes. The MikroTik setup, maintained connectivity

by forwarding traffic through intermediate nodes when direct links degraded. The Robonode platform exhibited intelligent routing behaviour by consistently connecting to the closest optimal node, as visualized through its internal hop count metrics. This confirms that mesh networks effectively prevent the complete communication breakdowns commonly observed in single-link setups, proving highly valuable for riverine exploration.

Resilience in NLOS Conditions – Riverine environments are inherently challenging due to signal occlusion from dense vegetation, river bends, and terrain. The evaluation clearly showed that while P2P connections offer superior throughput in clear LOS conditions, their performance degrades exponentially when distance increases or LOS is lost. Conversely, the mesh networks demonstrated robust resilience in NLOS scenarios. Because the mesh protocol can redirect traffic through unblocked nodes, it avoids the severe signal drop-offs experienced by the P2P baseline, establishing mesh as the superior strategy for navigating occluded river corridors.

Latency was satisfied for all moderate payload applications. The Robonode mesh exhibited higher steady state latency (typically tens of milliseconds rather than a few milliseconds), which is consistent with its design as an intelligent routing layer rather than a simple bridge. For control traffic, telemetry, and short status messages, both platforms remain well within the operational requirement. For tight closed loop visual control or low latency teleportation, the MikroTik mesh is the only viable mesh option.

6.2 Hardware Comparison and Architecture

Hardware Comparison: Range vs. Quality of Service (QoS) – Testing revealed a distinct trade-off between communication range and payload capacity among the tested hardware. The Robonode hardware achieved a maximum communication distance of 350 meters, outperforming the 230 m range of the MikroTik

devices. However, in terms of QoS and application layer throughput, MikroTik held a distinct advantage. System-level robotic tests showed that MikroTik could reliably deliver ROS sensor topics, such as LiDAR point clouds and 4K camera streams, with moderate latency. The Robonode, currently utilizing HWMP alongside Layer 2 WLAN networking, struggled with high bandwidth data, requiring further research into optimized routing firmware like Meshmerize to unlock its full potential.

The frequency band experiment produced a less obvious result. Using the MikroTik radios in 5ghz-a, 5ghz-a/n/ac, and 2ghz-onlyn modes, the maximum reachable distance was approximately 230 m in both 5 GHz and 2.4 GHz configurations. The 2ghz-onlyn mode showed a slightly more edge of connectivity region but at the cost of higher latency, and the 5ghz-a/n/ac mode delivered the highest throughput. Thus, switching to 2.4 GHz does not yield the dramatic range improvement. Considering these, the deployment recommendation is to default to 5 GHz for throughput and to consider 2.4 GHz only when interference avoidance in the 5 GHz band becomes the binding constraint.

P2P as a Baseline and Its Inherent Limits – The Ubiquiti P2P link was included as a high throughput reference rather than as a mesh deployment architecture. It met that purpose well and produced two findings that the mesh only experiments could not. First, the LOS Ubiquiti link sustained a 4K camera topic at 18.5 MB/s and 1.0 s of application layer latency. These features are out of reach of either mesh platform in the tested configuration. This confirms that the mesh throughput is bounded by the mesh radios and not by the laptops, cabling, or measurement pipeline. Second, the P2P behaviour under obstruction was qualitatively different from the mesh behaviour. Throughput collapsed approximately exponentially with NLOS distance, rather than degrading smoothly as observed for the mesh. Most importantly, the link did not automatically recover when the geometry changed. By construction, a directional P2P has no relay path, so the same obstruc-

tion that the mesh re-routed lead into a hard connectivity loss for the P2P link. The implication is that P2P is unsuitable as a standalone communication architecture for mobile multi-robot riverine monitoring. It cannot tolerate the topology changes that are intrinsic to the application. However, it is very useful as a component of the architecture when high-bandwidth payloads need to traverse a static, LOS segment of the path.

The Hybrid Architecture Strategy – Since high-bandwidth sensor payloads (e.g., simultaneous LiDAR and 4K video) induce severe latency on a pure mesh network, a hybrid approach is required. By deploying a P2P Ubiquiti connection as a high-throughput backhaul and utilizing the mesh network for localized edge-device access, the system can dynamically manage data loads. This allows the mobile robots to maintain continuous command and control connectivity via the mesh, while down streaming massive sensor payloads through the P2P link when LOS is available.

Radio class and physical layer impact – The MikroTik Groove A52 is a 1x1 802.11ac radio with a 6/8 dBi integrated antenna. The Robonode-M carries a 4x4 MIMO 802.11ax-capable radio with a higher claimed peak rate. The application-layer measurements do not reflect the Robonode's nominal advantage because the mesh mode of the firmware does not exploit the MU-MIMO and aggregation features of the underlying chipset to their full potential. This is a vendor software, not a hardware limitation, and it is one of the most actionable findings of the thesis. The platform is undersized in throughput today, but has the radio headroom to support a much higher rate if a different firmware stack (Meshmerize or a tuned 802.11s configuration) were available. By contrast, the MikroTik radio is closer to its physical layer ceiling already, so the room for improvement on that platform is smaller.

6.3 Practical Implementation and Validation

Practical Integration Constraints – Integrating this communication setup onto real robotic platforms, specifically drones, highlighted several mechanical and electrical constraints. Establishing the communication link itself was straightforward, however, accurately managing the power distribution between the drone's mechanical systems and the communication equipment was challenging. Furthermore, the payload weight and mounting alignment had to be carefully calibrated to ensure stable flight dynamics, underscoring that communication hardware must be explicitly co-designed with the physical robotic platform.

ROS application layer validation – The ROS level validation closed the loop between the network metrics and the robotic application. Three streams were tested, such as Livox LiDAR point cloud (0.64 MB per message), a RealSense colour image (6.22 MB), and a ZED2i 4K image (10.97 MB) and the results aligned with the network benchmarks at every level. The LiDAR topic, was delivered with sub second latency on MikroTik and Ubiquiti but seconds of buffered delay on Robonode, which means that the Robonode is currently slow as a perception data carrier without compression or a firmware upgrade. The camera topics were usable but delayed on both MikroTik and Ubiquiti, for the 4K stream the Ubiquiti was roughly twice as fast as MikroTik. The combined LiDAR with 4K stream test demonstrated that simultaneous high bandwidth payloads cannot be carried by a single mesh segment without prioritization, which is one of the arguments for the hybrid switched architecture. For a riverine deployment, the practical implication of the ROS-level tests is that the application stack should be designed for selective streaming rather than continuous high-rate upload. Local edge processing on the robot, event triggered upload of the heavy payloads when the geometry favours the P2P backhaul, and prioritization of command and telemetry traffic over bulk sensor streams are the three practices that the experimental data supports most directly.

6.4 Contributions and Deployment Guidelines

Given the all above points, the experiment supports a number of concrete recommendations for building Wi-Fi mesh communication into riverine multi-robot deployments.

- Use IEEE 802.11s with HWMP at Layer 2 as the default mesh stack. Both platforms tested provided self-healing behaviour under mobility.
- Select the mesh radio class based on the dominant operating mode of the mission. The MikroTik Groove A52 is preferable for moderate range, moderate throughput. The Robonode-M is preferable for mobile, longer range, low rate segments where connectivity matters more than peak throughput.
- Do not treat P2P as a replacement for mesh in mobile deployments. Use it instead as a backhaul attached to the mesh through a small managed switch.
- Design the application stack for selective streaming. Carry telemetry, commands, and LiDAR over the mesh. Offload high bandwidth video over the P2P segment when LOS is available and perform compression or event triggered upload for the rest.
- Default to 5 GHz for throughput and consider 2.4 GHz only when 5 GHz interference becomes the binding constraint. The maximum range gain from 2.4 GHz in these hardware classes is small and not worth the latency penalty.
- Analyze deployment with both an end to end metric (such as the application layer rosbag latency) and a link layer metric (RSSI, hop count). The two views are complementary and catch different classes of failure.

7 Summary and Future Work

7.1 Summary

This thesis researched the design and experimental evaluation of a Wi-Fi mesh communication architecture for multi-robot riverine monitoring application. This experiment was inspired by the lack of testing in the existing literature on Wi-Fi mesh for MRS in riverine monitoring environments. Limited mesh networking evaluations have been performed on commercial mesh modules connected with realistic robotic payloads for remote monitoring. Provided that, this thesis set out to determine whether Wi-Fi mesh architecture can provide a robust, self-healing communication by utilizing off-the-shelf hardware modules.

Apart from selecting Wi-Fi mesh architecture as the solution for riverine monitoring, this work made four other primary contributions. First, it identified and structured the communication requirements of MRS in riverine monitoring such as throughput, latency, reliability, and coverage. Second, it designed and implemented a hybrid mesh test bed and an Ubiquiti P2P baseline, which resulted in a reproducible configuration setup and measurement methodology. Third, systematic experiments were performed covering scenarios such as mobility-induced, LOS-NLOS, distance-dependent, QoS-loaded, and frequency-band configurations. In addition, these experiments were extended to robotics level validation by streaming ROS topics (LiDAR, RealSense image, and ZED2i 4K image). The ROS topics were trans-

mitted via each hardware and analysed through rosbag-based latency measurement. Fourth, it formulated and validated a hybrid switched architecture, where the mesh provides mobility and the P2P link provides high throughput. This combination resolves the throughput mobility tension that no single radio platform in the tested class could resolve alone.

The primary findings are as follows, Wi-Fi mesh architectures assure reliable, low-latency, and moderate-bandwidth data exchange, satisfying the mobility and topology changes. In addition, mesh routing successfully mitigates the severe signal degradation caused by NLOS (vegetation and physical obstructions). Furthermore, the two mesh networks do not exhibit identical performances, their performances depend on the selected scenario. While MikroTik supports for higher application-layer throughput (approximately 20 Mbps TCP under LOS) at moderate range, Robonode supports for longer-range operation (approximately 320 m versus 230 m) and more intelligent path selection under mobility. In fact, a hybrid network, which combines the mesh network and a P2P backhaul provides an efficient framework for real-time monitoring for high throughput data streams.

Returning to the research questions formulated in the introduction chapter 1, the thesis provides answers to the main research question (RQ). A Wi-Fi mesh communication architecture can support reliable, low latency, and moderate throughput communication for multi-robot riverine monitoring. It has been systematically accomplished from Chapter 2 to chapter 6.

The communication requirements are low latency, mobility tolerance, continuous connectivity and adequate throughput. Wi-Fi mesh stands out from other technologies because it is infrastructure free compared to cellular network while having better coverage and data rate. Since, Wi-Fi mesh has been already tested in infrastructure denied environment, it fulfils the requirements for communication network in riverine monitoring. This answers the first research question (RQ1), and it has been

extensively answered in the chapter 2.

The selection of the suitable hardware, software, and measurement parameters for Wi-Fi mesh architecture with relevant configurations are done in chapter 3. Moreover, the rigorous testing and evaluation of the network level performance have been conducted under different scenarios in chapter 4. The architecture performed as expected under outdoor mobility, distance, and topology change, noticeably the mesh reconnected within seconds and maintained continuous reachability throughout the experiment. These results strongly answer the second research question (RQ2).

For answering the third research question (RQ3), the mesh network is integrated to the robotic platforms. The integrated system is sufficient for telemetry, perception, and low to moderate rate visual monitoring. Moreover, when the hybrid switched architecture is used, it is sufficient for high-rate visual monitoring as well.

Overall, the contributions of the thesis can be stated as the complete review on Wi-Fi mesh, selection of hardware modules, the cross-platform measurement framework, evaluation and analysing of experimental scenarios, and the rosbag latency analysis. In particular, the framework and the evaluation methods are independent of the selected hardware. For this reason, the structure is reusable for future comparative studies of mesh implementations on robotic platforms. Ultimately, this research provides a foundational communication framework that demonstrates a promising future for the multi-robot riverine monitoring systems.

7.2 Future Work

While this thesis establishes a robust baseline for riverine mesh communication, several pathways remain for expanding the system's capabilities for scalable deployments.

- Implementation of Advanced Routing Protocols – the Robonode hardware

showed moderate range but struggled with high-throughput payloads under its current software configuration. Future research should implement and benchmark advanced industrial firmware, such as Meshmerize. And evaluate the possibility of performance enhancement in mesh network.

- Communication Aware Cooperative Control – building on the signal strength and connectivity data gathered in this work, future robotic platforms should implement communication-aware control algorithms. Instead of simply reacting to broken links, robots can use predictive modelling to alter their paths or distribute tasks cooperatively before a communication break occurs.
- Optimization of Hybrid Network Topologies – the proposed hybrid mesh-P2P connection should be further optimized. Future work should develop automated traffic-shaping methods that dynamically route low-latency control commands through the mesh and bulk sensor data through the P2P backhaul based on real-time link quality assessments.
- Evaluation in real river environment – due to the time and resource constraints, the thesis was conducted in limited outdoor area. Evaluation of the communication setup in real river environment will be completed in the future.

References

[1] A. Tripathi-II, S. Rai, D. K. Soni, A. Tripathi-I, and P. Pandey, "Combined challenges posed by climate change and pollution to river ecosystems: Exploring connecting links, hydro-biological impacts and future research approaches", *Environmental Sustainability*, vol. 8, no. 3, pp. 457–472, 2025.

[2] Z. Khosravi et al., "Designing High-Speed Directional Communication Capabilities for Unmanned Surface Vehicles", in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, Oulu, Finland: IEEE, Aug. 2019, pp. 651–655.

[3] "River Depth Monitoring Robot with Waste Collection Feature", *IARJSET*, vol. 12, no. 5, May 2025.

[4] C. Flener et al., "Seamless Mapping of River Channels at High Resolution Using Mobile LiDAR and UAV-Photography", *Remote Sensing*, vol. 5, no. 12, pp. 6382–6407, Nov. 2013.

[5] J. J. Roldán-Gómez, E. González-Gironda, and A. Barrientos, "A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety", *Applied Sciences*, vol. 11, no. 1, 2021.

[6] J. Gielis, A. Shankar, and A. Prorok, "A Critical Review of Communications in Multi-robot Systems", *Current Robotics Reports*, vol. 3, no. 4, pp. 213–225, 2022.

[7] D. Bonilla Licea, M. Ghogho, and M. Saska, "When Robotics Meets Wireless Communications: An Introductory Tutorial", *Proceedings of the IEEE*, vol. 112, no. 2, pp. 140–177, 2024.

[8] A. Agha et al., "Nebula: Team costar's robotic autonomy solution that won phase ii of darpa subterranean challenge", *Field Robotics*, vol. 2, pp. 1432–1506, 2022.

[9] M. Tranzatto et al., "Cerberus in the darpa subterranean challenge", *Science Robotics*, vol. 7, no. 66, eabp9742, 2022.

[10] H. Biggie and S. McGuire, "Heterogeneous ground-air autonomous vehicle networking in austere environments: Practical implementation of a mesh network in the darpa subterranean challenge", in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2022, pp. 261–268.

[11] A. Tripathi-II et al., "Combined challenges posed by climate change and pollution to river ecosystems", *Environmental Sustainability*, vol. 8, no. 3, pp. 457–472, 2025.

[12] F. Cladera, Z. Ravichandran, I. D. Miller, M. Ani Hsieh, C. J. Taylor, and V. Kumar, "Enabling large-scale heterogeneous collaboration with opportunistic communications", in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 2610–2616.

[13] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot Systems: A Classification Focused on Coordination", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2015–2028, 2004.

[14] A. Martorell-Torres, J. Guerrero-Sastre, and G. Oliver-Codina, "Coordination of marine multi robot systems with communication constraints", *Applied Ocean Research*, vol. 142, p. 103 848, Jan. 2024.

[15] K. S. Adu-Manu, F. A. Katsriku, J.-D. Abdulai, and F. Engmann, "Smart River Monitoring Using Wireless Sensor Networks", *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–19, Sep. 2020.

[16] C. J. MacDonell, R. D. Williams, J. White, and K. Roberts, "Seamless quantification of wet and dry riverscape topography using uav topo-bathymetric lidar", *Drones*, vol. 9, no. 12, 2025.

[17] V. Akstinas, K. Gurjazkaitė, D. Meilutytė-Lukauskienė, A. Kriščiūnas, D. Čalnerytė, and R. Barauskas, "Suitability of UAV-Based RGB and Multispectral Photogrammetry for Riverbed Topography in Hydrodynamic Modelling", *Water*, vol. 18, no. 1, p. 38, Dec. 2025.

[18] M. Rusnák et al., "A detection of channel bathymetry from drones: Accuracy of point clouds and rgb images for reconstruction of channel-bed topography", *Geografický časopis / Geographical Journal*, vol. 77, no. 4, pp. 183–199, 2025.

[19] T. Adão et al., "Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry", *Remote Sensing*, vol. 9, no. 11, p. 1110, Oct. 2017.

[20] F. Bandini et al., "Mapping inland water bathymetry with Ground Penetrating Radar (GPR) on board Unmanned Aerial Systems (UASs)", *Journal of Hydrology*, vol. 616, p. 128 789, Jan. 2023.

[21] O. Specht, "Bathymetric data coverage and density from single-beam and multibeam echo sounding surveys using unmanned surface vehicles in shallow inland waters", *Journal of Water and Land Development*, pp. 83–83, Feb. 16, 2026.

[22] T. Goblirsch, T. Mayer, S. Penzel, M. Rudolph, and H. Borsdorf, "In Situ Water Quality Monitoring Using an Optical Multiparameter Sensor Probe", *Sensors*, vol. 23, no. 23, p. 9545, Nov. 2023.

[23] E. Pinto, F. Marques, R. Mendonca, A. Lourenco, P. Santana, and J. Barata, "An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception", in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Bali, Indonesia: IEEE, Dec. 2014, pp. 443–450.

[24] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint", *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[25] J. Sánchez-García, J. García-Campos, M. Arzamendia, D. Reina, S. Toral, and D. Gregor, "A survey on unmanned aerial and aquatic vehicle multihop networks: Wireless communications, evaluation tools and applications", *Computer Communications*, vol. 119, pp. 43–65, Apr. 2018.

[26] D. Sousa, M. Luís, S. Sargento, and A. Pereira, "An Aquatic Mobile Sensing USV Swarm with a Link Quality-Based Delay Tolerant Network", *Sensors*, vol. 18, no. 10, p. 3440, Oct. 13, 2018.

[27] A. Zolich et al., "Survey on Communication and Networks for Autonomous Marine Systems", *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3, pp. 789–813, Sep. 2019.

[28] R. Abu-Aisheh, F. Bronzino, L. Salaün, and T. Watteyne, "CARA: Connectivity-Aware Relay Algorithm for Multi-Robot Expeditions", *Sensors*, vol. 22, no. 23, p. 9042, Nov. 2022.

[29] T. Abderrahmane, A. Nourredine, and T. Mohammed, "Experimental analysis for comparison of wireless transmission technologies: Wi-Fi, Bluetooth, ZigBee and LoRa for mobile multi-robot in hostile sites", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, p. 2753, Jun. 2024.

[30] N. Sanghvi, R. Niyogi, and R. Tripathi, "Performance analysis of communication protocols for multi-robot coordination in agricultural field", *Int. J. Pervasive Comput. Commun.*, vol. 21, no. 4–5, pp. 232–251, 2025.

[31] M. Aho, A. Happonen, M. Jäntti, and K. Pehkonen, "Private LoRaWAN Network Deployment in Kuopio, Finland: A Case Study on AI-Based Water-Level Monitoring and Urban Flood Prediction", *IOTAI*, pp. 7–13, 2025.

[32] I. Esfandiyar and K. Młodzikowski, "Dual-Link Data Resilient Edge-to-cloud Communication Framework for Agricultural Robots", *IOTAI*, pp. 27–34, 2025.

[33] A. Stateczny, K. Gierlowski, and M. Hoeft, "Wireless Local Area Network Technologies as Communication Solutions for Unmanned Surface Vehicles", *Sensors*, vol. 22, no. 2, p. 655, Jan. 2022.

[34] R. Liu and N. Choi, "A First Look at Wi-Fi 6 in Action: Throughput, Latency, Energy Efficiency, and Security", *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 1, pp. 1–25, Feb. 2023, ISSN: 2476-1249.

[35] B. Sliwa, S. Falten, and C. Wietfeld, "Performance evaluation and optimization of B.A.T.M.A.N. V routing for aerial and ground-based mobile ad-hoc networks", in *Proc. IEEE VTC2019-Spring*, 2019, pp. 1–7.

[36] L. P. Chovet et al., "Performance comparison of ROS2 middlewares for multi-robot mesh networks in planetary exploration", *J. Intell. Robot. Syst.*, vol. 111, no. 1, p. 18, 2025.

[37] D. Turlykozhayeva et al., "Experimental performance comparison of proactive routing protocols in wireless mesh network using Raspberry Pi 4", *Telecom*, vol. 5, no. 4, pp. 1008–1020, 2024.

[38] S. Pandi et al., "MESHMERIZE: An interactive demo of resilient mesh networks in drones", in *Proc. IEEE CCNC*, 2019, pp. 1–2.

[39] A. Ingole, B. Michaels, O. Pipano, P. Xu, R. Page, and T. Dhulipala, "Search technology for optimal rescue missions (storm)", National Aeronautics and Space Administration, Tech. Rep. NASA/TM–20240012868, 2024.

[40] T. Manoni, D. Albani, J. Horyna, P. Petracek, M. Saska, and E. Ferrante, "Adaptive arbitration of aerial swarm interactions through a Gaussian kernel for coherent group motion", *Frontiers in Robotics and AI*, vol. 9, p. 1 006 786, Dec. 2022.

[41] A. Coyle, A. Gupta, and B. Campbell, "RDMST- A Novel Distributed Topology Control Algorithm for Low Probability of Detection Mobile Communication Networks", *Procedia Computer Science*, vol. 205, pp. 68–77, 2022.

[42] D. Van Der Meer, L. P. Chovet, G. M. Garcia, A. Bera, and M. A. Olivares-Mendez, "REALMS2 - Resilient Exploration And Lunar Mapping System 2 – A Comprehensive Approach", in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2025, pp. 1495–1502.

[43] J. Zhang et al., "Comparison of middlewares in edge-to-edge and edge-to-cloud communication for distributed ROS 2 systems", *J. Intell. Robot. Syst.*, vol. 110, no. 4, p. 162, 2024.

[44] H. Lee and D. Panagou, "Maintaining strong r-robustness in reconfigurable multi-robot networks using control barrier functions", in *Proc. IEEE ICRA*, 2025, pp. 1–7.

[45] M. Saboia et al., "ACHORD: Communication-Aware Multi-Robot Coordination With Intermittent Connectivity", *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 184–10 191, Oct. 2022.

[46] Z. Liu and M. S. Miah, "Decentralized Multi-Cobot Navigation Under Intermittent Communication", *Robotics*, vol. 15, no. 1, p. 4, Dec. 2025, ISSN: 2218-6581.

[47] MikroTik, *RBGroove52HPnr2 Product Specifications*, https://mikrotik.com/product/RBGroove52HPnr2#product_specification, Accessed: 2026-05-05, 2026.

[48] RoboNode, *RoboNode-M Product Specifications*, https://www.8devices.com/products/robonode, Accessed: 2026-05-05, 2026.

Appendix A Configurations

A.1 Robonode-M-Specific Measurement Procedure

```
# Continuous ping
ping -s 1500 -c 600 192.168.1.21 > ping_$(date +%s).log


# Continuous iPerf3 (UDP, jitter)
iperf3 -c 192.168.1.21 -u -b 10M -t 600 -i 1 \
    > iperf_udp_$(date +%s).log


# Continuous mesh path
while true; do
    date >> mpath.log
    iw dev mesh0 mpath dump >> mpath.log
    sleep 1
done


# Continuous link quality
while true; do
    date >> wifistats.log
    wifistats >> wifistats.log
    sleep 1
done
```

Listing A.1: Robonode configuration for the wireless mesh node

A.2 Time Synchronization for Rosbag Collection

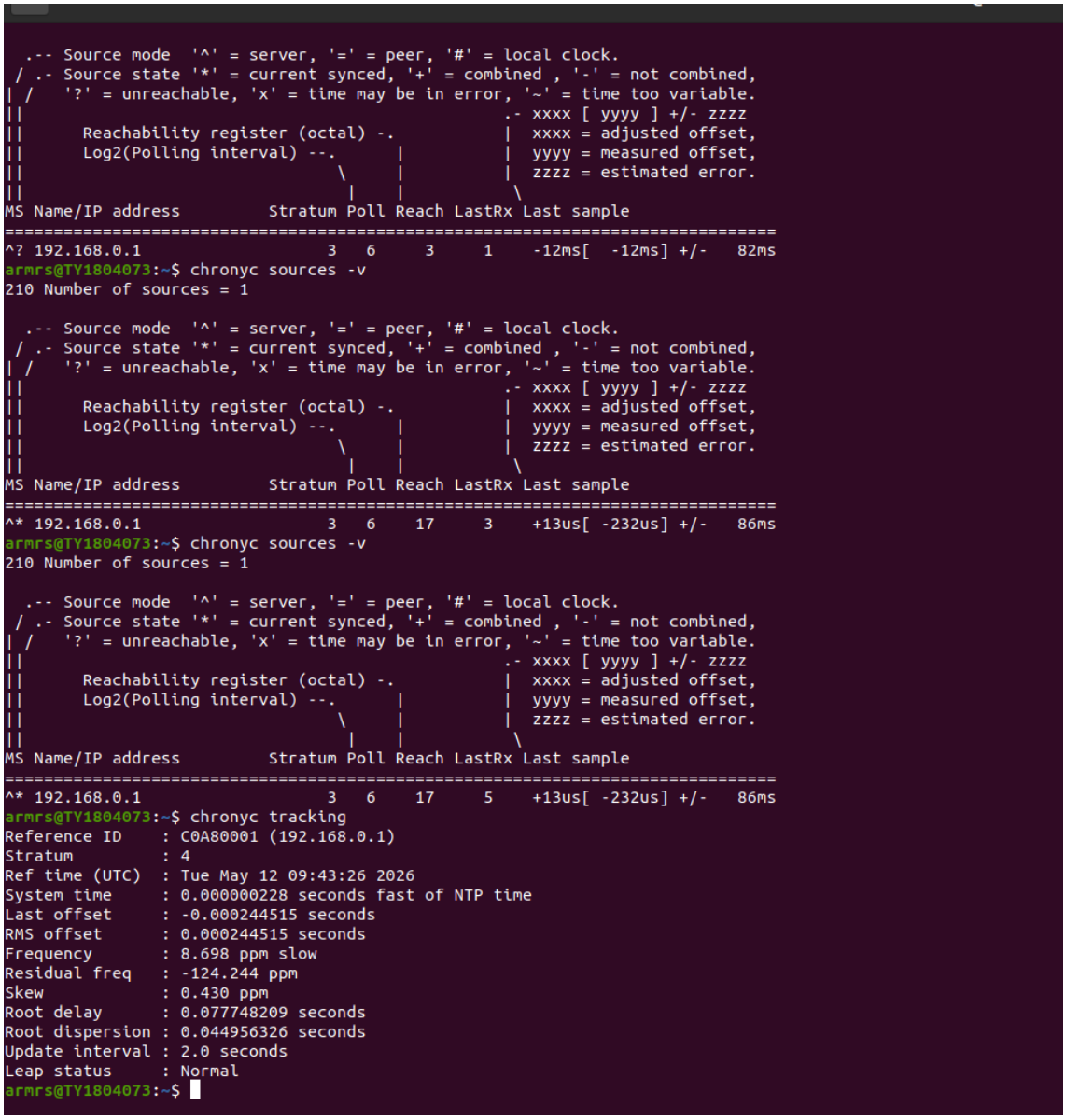

Figure A.1: Time synchronization with the drone and initial offset investigation.