IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 January 2026, accepted 10 February 2026, date of publication 12 February 2026, date of current version 19 February 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3664198

METHODS

Ad-Hoc Network for Unmanned Mobile Vehicles: Implementation and Experiments

JIN-HEE SON[1], JEONG-MIN MA[1], HYUNG-GON LEE[1], NAM-JIN PARK[2], AND HYO-SUNG AHN[1], (Senior Member, IEEE)
[1]Department of Mechanical and Robotics Engineering, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea
[2]Department of Electrical Engineering, Kunsan National University, Gunsan-si, Jeollabuk-do 54150, South Korea

Corresponding author: Hyo-Sung Ahn (hyosung@gist.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by Korean Government through Ministry of Science and ICT (MSIT) under Grant 2022R1A2B5B03001459.

ABSTRACT This study presents an ad-hoc network system for unmanned mobile vehicles (UMVs), with an emphasis on system-level integration and experimental validation using customized hardware and software. A standalone ad-hoc module is developed by integrating open-source firmware and the *batman-adv* protocol, enabling multi-hop communication while remaining transparent to onboard control software. This design allows UMVs to participate in an ad-hoc network without requiring modifications to existing control or middleware stacks. Outdoor field experiments are conducted to evaluate multi-hop communication performance in terms of throughput, latency, and packet loss under representative experimental conditions. In addition, complementary case studies investigate the functional feasibility of supervisory UMV control and monitoring over multi-hop links. The experimental results demonstrate that the proposed system provides a practical reference implementation for integrated multi-UMV networking in outdoor experimental settings.

INDEX TERMS Ad-hoc network, multi-hop, unmanned mobile vehicles, multi-agent systems.

I. INTRODUCTION

Unmanned Mobile Vehicles (UMVs) have recently seen significant advancements and are being utilized across a wide range of applications. In particular, communication is one of the most critical aspects for operations involving multiple UMVs. To support this, various communication methods have been introduced, including not only cellular networks such as LTE, 5G, and the upcoming 6G [1], [2], but also UMV-to-UMV networks, which help overcome the limitations of cellular networks that are restricted to areas within the coverage of base stations [3], [4]. Ad-hoc networks are particularly suited for UMVs, where communication is relayed between neighboring vehicles. This setup allows UMVs to establish a network without central infrastructure, providing greater flexibility and reliability in environments where traditional cellular networks may be limited or unavailable.

A. RELATED WORK

Ad-hoc networking has been extensively studied across a variety of domains, including MANETs [5], [6], [7], [8], [9], [10], VANETs [11], [12], [13], [14], and FANETs [15], [16], [17], [18]. Beyond UMV-specific studies, related multi-hop mesh networking concepts have also been explored in networked control systems [19], [20] and field robotics applications, including emergency-response and exploration scenarios where relay-based mesh backbones are employed to maintain connectivity under infrastructure-degraded conditions [28], [29]. However, these studies primarily focus on mission execution or routing performance and provide limited discussion on system-level integration with UMV onboard control architectures.

Despite these advances, practical deployment of ad-hoc networking in UMV systems remains challenging. Existing solutions are often tightly coupled to specific hardware platforms or software stacks, which complicates integration as UMV onboard systems continue to evolve. Meanwhile, several routing protocols have been proposed for ad-hoc

The associate editor coordinating the review of this manuscript and approving it for publication was Adao Silva.

networks, including OLSR [21], *batman-adv* [22], and Babel [23]. While their routing performance has been evaluated in simulations and testbeds, empirical validation in outdoor multi-hop UMV operations remains limited.

For example, [24] evaluates *batman-adv* in UMV networks but does not address integration with UMV control systems or firmware customization. Simulation-based studies such as [25] analyze routing behavior under idealized assumptions, while other works focus on constrained environments, including indoor teleoperation [26] or specialized deployable infrastructures [27]. Overall, existing studies predominantly emphasize routing-level performance or application-specific deployments, whereas system-level integration between ad-hoc networking mechanisms and UMV onboard control architectures, together with empirical multi-hop performance under realistic robotic traffic, remains comparatively underexplored.

B. CONTRIBUTIONS

This paper presents an implementation-oriented study on the integration of ad-hoc networking with UMV control architectures. Specifically, a modular ad-hoc network system based on dedicated hardware and custom firmware is developed and integrated with heterogeneous UMV platforms. The proposed design isolates networking functionality within a standalone module, allowing UMVs to participate in a multi-hop ad-hoc network without requiring modifications to their internal hardware or software configurations. Outdoor field experiments are conducted to characterize multi-hop communication performance under representative UMV traffic.

The main contributions of this work are:

- The design of customized firmware that enables standalone operation of an ad-hoc module, allowing UMVs to join a multi-hop network without additional hardware or software modifications.
- The development and experimental validation of an ad-hoc network system for multiple UMVs, demonstrating its feasibility as a reference implementation in outdoor experimental environments.

II. PROBLEM FORMULATION AND SCOPE

This study considers the problem of enabling supervisory control and state monitoring of UMVs over a multi-hop ad-hoc network. Unlike low-level motion control loops, which are typically executed onboard under strict real-time constraints, the considered control architecture relies on high-level command, status, and telemetry exchange between a ground control station and UMVs. Such supervisory control architectures are widely adopted in practical UMV experiments, where real-time safety-critical control is handled locally, while mission-level commands and system monitoring are performed over wireless communication links.

In the scope of this work, UMV control and monitoring are realized using MAVROS, which interfaces onboard autopilots with external control and monitoring software. MAVROS communication consists of periodic state updates, sensor information, and command messages exchanged at moderate data rates, and thus represents a typical supervisory communication workload in outdoor UMV experiments. Accordingly, the feasibility of supervisory UMV control over an ad-hoc network can be assessed by examining whether the communication system can reliably support such message exchanges under multi-hop conditions. Based on this setting, the communication requirements for supervisory UMV operation over a multi-hop ad-hoc network can be summarized as follows:

Condition 1: To support supervisory UMV control and state monitoring via MAVROS over a multi-hop ad-hoc network, the communication system should satisfy the following conditions:

- **(C1)** sufficient end-to-end throughput to sustain continuous MAVROS message exchange,
- **(C2)** tolerable latency, jitter, and packet loss for supervisory control and state monitoring;
- **(C3)** robustness to multi-hop propagation effects and environmental interference in outdoor experimental settings.

The objective of this work is to experimentally validate whether a modular ad-hoc networking system can reliably satisfy Condition 1 under representative outdoor multi-UMV operations.

III. METHOD AND DESIGN

Based on the problem scope and communication requirements defined in the previous section, this section describes the design and implementation of the proposed ad-hoc networking system for multi-UMV systems. The focus is on system-level integration and practical implementation, including the architectural separation of communication and control functions, custom firmware design for standalone operation, and the system configuration for multi-hop communication.

A. NECESSITY OF AD-HOC MODULE

The onboard computing systems of UMVs continue to evolve to support increasingly complex autonomy and control functions. However, integrating communication software across heterogeneous and rapidly changing hardware-software stacks remains challenging. In existing UMV systems, communication algorithms are often deployed directly on the companion computer, where they are tightly coupled with internal software dependencies and libraries. This tight coupling can lead to compatibility issues during system integration. Such challenges may result in communication failures or unintended software breakdowns, potentially leading to safety concerns.

To address these challenges, the proposed ad-hoc module is designed as a dedicated communication unit that operates independently of the UMV's primary control system. The module connects to the companion computer via a standard Ethernet interface and autonomously manages network transmission, multi-hop routing, and related communication tasks. By decoupling communication functionality from the companion computer, the proposed design enables independent operation of networking and control subsystems, thereby simplifying integration across heterogeneous UMV platforms.

To meet the computational and connectivity requirements of ad-hoc networking, commercial off-the-shelf wireless routers equipped with SoC processors and Wi-Fi interfaces were selected and customized as prototype ad-hoc modules. While custom hardware could be developed for this purpose, the use of commercial devices provides a practical and reliable solution for experimental validation and rapid deployment.

Remark 1: Decoupling communication from the primary control system can improve overall system robustness. Even if the companion computer fails or reboots, the ad-hoc network remains operational, preserving connectivity among UMVs.

B. CUSTOM FIRMWARE

This subsection describes the custom firmware architecture implemented within the proposed ad-hoc module to enable standalone multi-hop networking operation. Commercial Wi-Fi routers are typically equipped with vendor-specific firmware optimized for consumer networking, which provides limited flexibility for system-level customization and integration in multi-UMV environments.

To support independent operation of the ad-hoc module, the firmware must be open, self-contained, and freely configurable. Accordingly, *OpenWrt*, an open-source Linux-based operating system for network devices, is adopted as the foundation of the custom firmware. Its modular architecture allows a commercial router to operate as an independent computing platform, managing all networking functions and dependencies locally. Using build-time customization, wireless drivers, kernel-level networking components, and device-specific configurations are integrated into a single firmware image tailored to the target hardware, as illustrated in Fig. 1.

Within the ad-hoc module, the firmware encapsulates all components required for multi-hop networking, including the Linux kernel, wireless drivers, network stack, and routing mechanism. In this work, *batman-adv* is adopted as the core routing component [22]. Unlike user-space routing protocols such as OLSR or Babel [30], [31], *batman-adv* operates as a kernel module at the data link layer, forming a virtual Layer-2 mesh interface that is transparent to IP-level communication. This design allows the ad-hoc module to present a conventional Ethernet interface to the companion computer, enabling it to join the multi-hop network without requiring explicit routing configuration or subnet management.

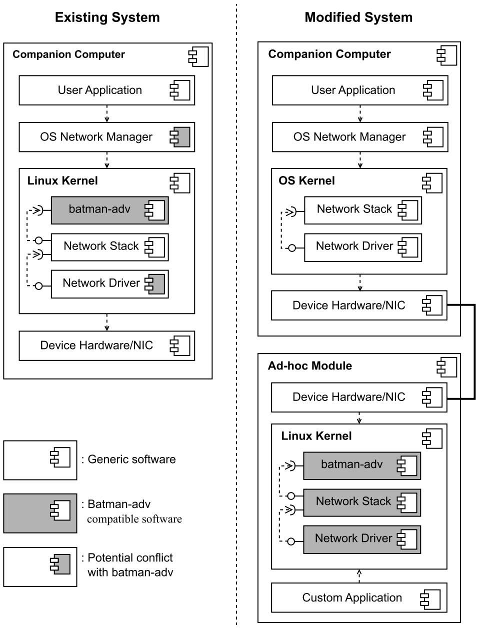


FIGURE 1. Comparison of existing and modified ad-hoc network systems. The modified design offloads networking tasks to a dedicated ad-hoc module. The custom firmware is available at https://github.com/GIST-DCASL/openwrt_mesh_configurate.

Based on this firmware design, the ad-hoc module bridges communication between its mesh interface and an external Ethernet port connected to the companion computer. As a result, the companion computer interacts with the ad-hoc network as if it were connected to a standard local area network, without installing additional drivers or modifying its internal software stack. By embedding routing mechanisms, wireless parameters, and related dependencies directly into the firmware, the ad-hoc module achieves portability across heterogeneous UMV platforms.

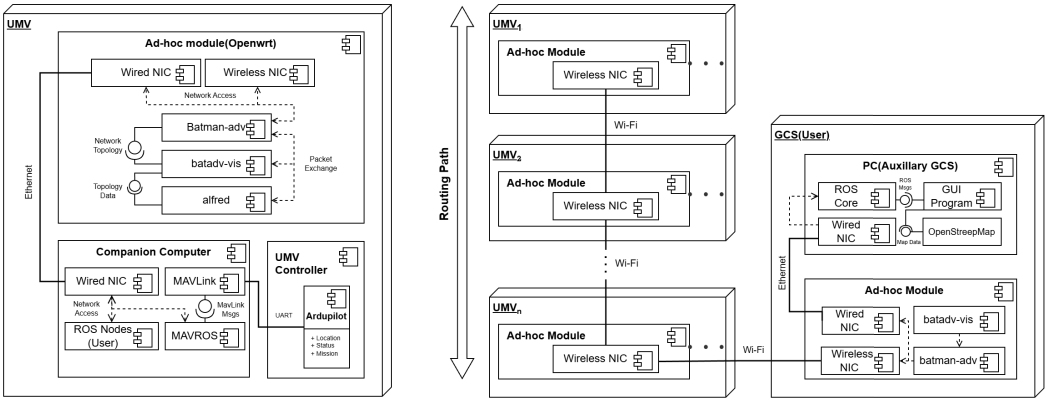
Remark 2: Embedding routing protocols and required dependencies directly into the firmware ensures version consistency across devices, preventing compatibility issues that commonly arise in multi-vendor UMV deployments.

C. SYSTEM ARCHITECTURE

This subsection describes the overall system architecture for supervisory control and state monitoring of multiple UMVs over the proposed ad-hoc network. The architecture is designed to support the communication requirements defined in Condition 1 by separating control, computation, and networking functionalities across dedicated system components. The resulting data flow and communication paths are illustrated in Fig. 2.

1) SINGLE-UMV ARCHITECTURE

As shown in Fig. 2(a), each UMV consists of three main components: a companion computer, a UMV controller



(a) UMV system architecture

(b) Multi-hop communication path

FIGURE 2. **System architecture illustrating UMV components and multi-hop communication.(a) Interaction between the companion computer, UMV controller, and ad-hoc module within a single UMV.(b) Multi-hop communication path between UMVs and the GCS via the ad-hoc network.**

(autopilot), and an ad-hoc module. The companion computer runs the Robot Operating System (*ROS*) and serves as the interface between user-level control applications and the onboard system. Supervisory commands, telemetry, and status messages are generated through *MAVROS*, which bridges *ROS* and the UMV controller via *MAVLink* communication.

All network traffic generated by the companion computer is forwarded through a wired Ethernet connection to the ad-hoc module. From the perspective of the companion computer, the ad-hoc module appears as a conventional network interface, requiring no modification to the onboard software stack. This design allows supervisory control and state monitoring messages to be transmitted transparently over the ad-hoc network, while time-critical low-level control loops remain locally executed on the UMV controller.

2) MULTI-HOP COMMUNICATION ARCHITECTURE

Figure 2(b) illustrates the multi-hop communication path established among multiple UMVs and the Ground Control Station (GCS). Each ad-hoc module communicates with neighboring modules via its wireless interface, forming a distributed multi-hop mesh network based on *batman-adv*. Supervisory control messages and telemetry data are relayed hop-by-hop through the ad-hoc network, ultimately reaching the GCS without requiring centralized infrastructure or predefined routing paths.

This architecture enables end-to-end communication between the GCS and each UMV, even when direct wireless connectivity is unavailable, and provides the structural basis for satisfying Condition 1 in outdoor multi-UMV operations.

3) GROUND CONTROL AND NETWORK MONITORING

The GCS connects to the ad-hoc network via a wired interface and supports standard TCP/IP-based access to UMVs. While a generic GCS is sufficient for supervisory control, it offers limited visibility into ad-hoc–specific network states such as connectivity and topology. For experimental analysis and network monitoring, an auxiliary interface was therefore employed to visualize and collect ad-hoc network information. Further details are provided in Section IV.

IV. EXPERIMENTS

This section describes the experiments conducted to validate the proposed system using actual hardware and software. The primary objective is to evaluate the multi-hop communication performance of the ad-hoc network in terms of latency, throughput, and packet loss, and to examine the operational feasibility of the system with respect to the communication requirements defined in Condition 1.

A. HARDWARE AND SOFTWARE

The experimental platform consists of multiple UMVs, including both Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). These commercially available platforms were selected to ensure sufficient payload capacity for carrying the companion computer and ad-hoc module while maintaining safe operation in outdoor experimental environments. The UAV and UGV platforms are based on the *DJI F450* and *SCX-10* frames, respectively. All UMVs are equipped with a *Pixhawk Cube Orange* autopilot running *Ardupilot 3.4.2*. The companion computer is a *Raspberry Pi 3* running *Ubuntu MATE 20.04.6*, integrated with *ROS Noetic*, *MAVLink*, and *MAVROS* for communication with the autopilot.

For ad-hoc networking, each UMV and the GCS is equipped with a *GL-AR300M16-Ext* wireless router, which serves as the dedicated ad-hoc module. The device features a single-core MIPS SoC, IEEE 802.11n (Wi-Fi 4) wireless support, and an external 2 dBi antenna. All ad-hoc modules run custom firmware based on *OpenWrt 22.03.5*, enabling automatic formation and maintenance of a multi-hop ad-hoc network. The hardware configuration of the UMVs and the ad-hoc modules is illustrated in Fig. 3.

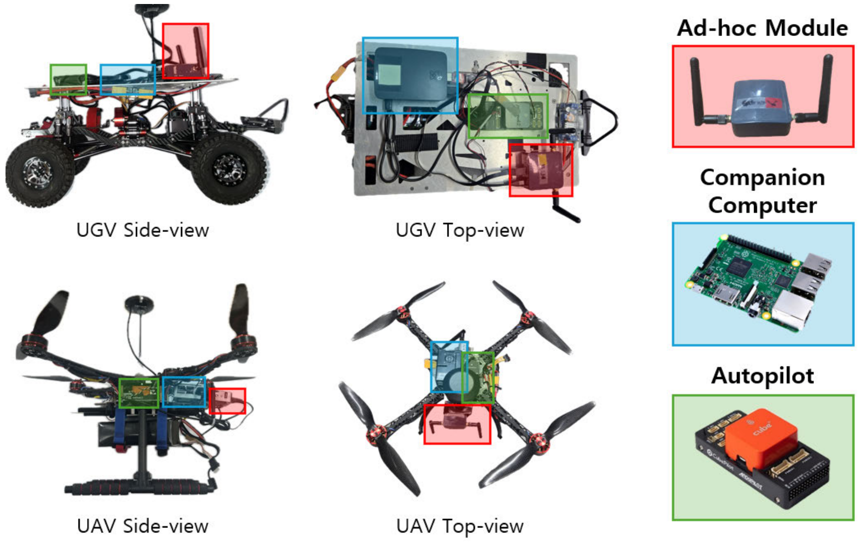
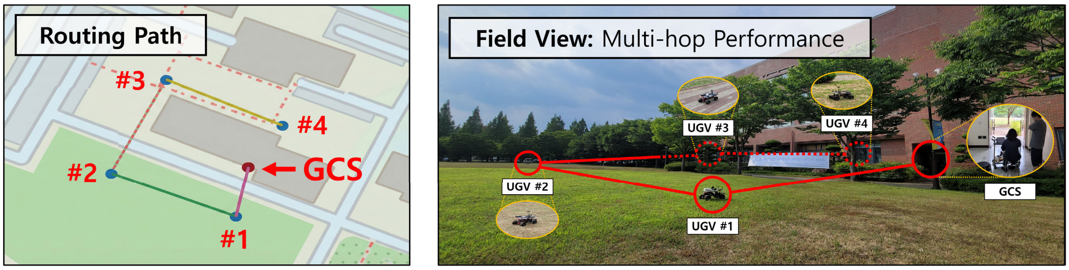
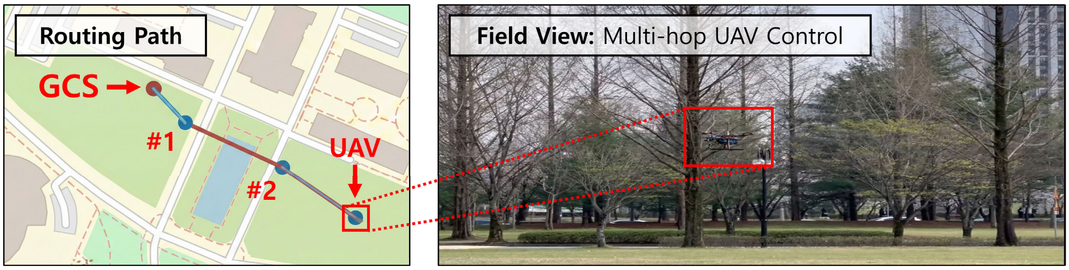


FIGURE 3. **Hardware configuration of the UGV and UAV. Each UMV integrates three core components: an ad-hoc module (*GL-AR300M16-Ext*, red)a companion computer (*Raspberry Pi 3*, blue), and an autopilot (*Pixhawk Cube Orange*, green).**



(a) Case 1



(b) Case 2

FIGURE 4. **Routing paths and field-test views for the two experimental cases. Case 1 illustrates the multi-hop routing paths among UGVs and the corresponding field setup used for quantitative evaluation. Case 2 shows the multi-hop routing path between the GCS and a UAV, with the field view of the UAV flight test.Video demonstrations are available at https://youtube.com/playlist?list=PLUIBjqIFRraHPR5gWhHmYErhmzvj_fMGH&si=Arq1TkAchhiOf_yT.**

B. EXPERIMENT SETUP

Two GCS units were connected to the ad-hoc network via dedicated ad-hoc modules, providing redundancy in case of system malfunction. The primary GCS ran *Mission Planner* on *Windows 10*, while the auxiliary GCS operated on *Ubuntu 20.04.6* and hosted the ad-hoc network monitoring GUI. The auxiliary GCS executed *roscore* and a WebSocket bridge to provide external access to ROS topics exchanged with the UMVs. A custom GUI developed using *Python 3.9* and the *PyQt5* framework was used to visualize ad-hoc network status, connectivity information, and positional data in real time. Finally, the ad-hoc modules ran additional scripts to extract mesh connectivity information using *alfred* and *batadv-vis*.

To evaluate the proposed ad-hoc networking system from complementary perspectives, two experimental cases were designed with distinct objectives, as illustrated in Fig. 4.

1) CASE 1 (MULTI-HOP PERFORMANCE)

Case 1 was designed to quantitatively evaluate the communication performance of the ad-hoc network under enforced multi-hop conditions. Four UGVs were deployed around the perimeter of a building, with approximately 60 m separation between adjacent vehicles. This deployment intentionally prevented direct one-hop wireless communication, thereby forcing all data traffic to be relayed through multi-hop paths. By constructing a controlled multi-hop topology with increasing hop counts, this case focuses on measuring network performance metrics, including throughput, latency, jitter, and packet loss.

2) CASE 2 (MULTI-HOP UAV CONTROL)

Case 2 was designed to validate the functional feasibility of supervisory UAV control over a multi-hop ad-hoc network. In this scenario, a UAV was designated as the controlled target, while UGVs equipped with ad-hoc modules served as relay nodes to establish a communication path between the GCS and the UAV.

Unlike Case 1, which emphasizes quantitative performance benchmarking, this case focuses on verifying whether command-and-telemetry exchange required for UAV operation can be maintained reliably over the ad-hoc network. The experiment evaluates the continuity of control commands, telemetry feedback, and real-time monitoring during actual flight operation, thereby demonstrating practical feasibility.

C. MULTI-HOP PERFORMANCE

Multi-hop communication performance was evaluated in Case 1 with the objective of quantitatively assessing whether the proposed ad-hoc network satisfies the communication requirements for supervisory UMV control defined in Condition 1. To this end, three complementary tests were conducted to measure end-to-end throughput, latency, jitter, and packet loss under enforced multi-hop routing conditions.

- **TCP test:** TCP packets were transmitted using *iperf3*.[1] This test evaluates achievable end-to-end throughput and TCP retry behavior as the hop count increases, providing an upper bound on sustainable data rates under multi-hop operation.
- **UDP test:** UDP packets were transmitted at a fixed rate of 250 Kbps using *iperf3*. This rate intentionally exceeds the estimated bandwidth required for MAVROS-based command-and-telemetry traffic summarized in Table 1, thereby introducing a conservative margin for evaluating **(C1)** under multi-hop operation. This test measures UDP throughput, jitter, and packet loss, which are directly related to **(C2)**.
- **Latency test:** End-to-end delay was measured using *Lagscope*,[2] providing latency distributions and tail characteristics across different hop counts.

The measured performance metrics for Case 1 are obtained from the time-series results shown in Fig. 5 and are summarized in Table 2. These results collectively provide a quantitative basis for assessing the feasibility of supervisory UMV communication over multi-hop paths.

1) TCP TEST

The TCP test results are shown in Fig. 5(a) and Fig. 5(b). In the 1-hop scenario, throughput slightly exceeded 2 Mbps, which is well above the bandwidth requirements for supervisory UMV command and telemetry traffic considered in this study. As the hop count increased, throughput decreased to approximately 250 Kbps, yet remained sufficient for essential ROS-based communication. The increase in TCP retries with hop count indicates greater exposure to environmental interference along longer relay paths.

2) UDP TEST

Figs. 5(c)-5(e) present the UDP test results. Across all hop counts, more than 237 Kbps of UDP traffic was successfully delivered, exceeding the estimated MAVROS bandwidth requirements in Table 1. As the hop count increased, jitter and packet loss rose due to cumulative multi-hop effects and environmental interference. In particular, the loss rate increased to 3.74% at three hops, indicating that while overall delivery remained sufficient for MAVROS traffic, multi-hop propagation introduced occasional packet drops under external interference. Initial jitter spikes were observed as the ad-hoc modules stabilized their links, but values converged to within 100 ms during sustained transmission.

3) LATENCY TEST

Fig. 5(f) illustrates the latency distribution obtained from 10,000 TCP packets measured with *Lagscope*. Most packets exhibited low latency, but intermittent long-delay outliers produced a log-normal–like tail [32]. Both average and median latency increased with hop count, and the tail became more pronounced, indicating that environmental interference accumulates along extended multi-hop paths.

Remark 3: All experiments were conducted using commercially available devices operating in the 2.4 GHz band under regulatory transmission-power limits (20 dBm in South Korea). Surrounding buildings with enterprise-grade Wi-Fi access points operating in the same band introduced non-negligible interference, amplifying multi-hop effects and performance variability, particularly with respect to robustness against environmental interference as required in **(C3)**.

D. MULTI-HOP UAV CONTROL

This subsection reports the experimental results of Case 2, which focuses on verifying the functional feasibility of supervisory UAV control over a multi-hop ad-hoc network under outdoor operating conditions.

During the experiment, supervisory control commands were transmitted from the GCS to the UAV through the established multi-hop ad-hoc network, while telemetry data,

---

[1] *iperf3* is an open-source tool widely used for benchmarking TCP and UDP network performance.

[2] *Lagscope* transmits timestamped packets to measure one-way or round-trip latency.

TABLE 1. Topic frequencies and estimated bandwidths for MAVROS communication.

Topic (Message size in bytes)	Frequency	Bandwidth
`/mavros/imu/data` (100 bytes)*	50 Hz	50 Kbps
`/mavros/gps/fix` (128 bytes)	10 Hz	10 Kbps
`/mavros/setpoint_position/local` (84 bytes)	20 Hz	13 Kbps
`/mavros/battery` (40 bytes)	1 Hz	1 Kbps
`/mavros/rc/in` (35 bytes)	20 Hz	6 Kbps
`/mavros/state` (48 bytes)	1 Hz	1 Kbps
`/mavros/cmd/arming` (3 bytes)	Event-based	0.1 Kbps
Estimated Network Overhead (20%)		**Approx. 16.2 Kbps**
Total Estimated Bandwidth (Including Overhead)		**97.3 Kbps**

* Estimated packet size excludes IMU covariance fields.

TABLE 2. Summary of averaged network performance metrics for Case 1. Each entry represents the overall mean value of the corresponding results shown in Fig. 5.

Test	Parameter	1 hop	2 hops	3 hops
TCP Test	Throughput [Kbps]	2195	359	243
	Retries (Average) [Count/s]	1.0	1.5	2.5
UDP Test	Throughput [Kbps]	246	249	237
	Jitter* (Average) [ms]	249.77	370.39	1766.58
	Jitter (Last) [ms]	9.19	40.38	55.11
	Loss Rate [%]	0.0	0.0	3.74
Latency Test	Latency (Average) [ms]	20.22	110.48	146.61
	Latency (Median) [ms]	7.72	38.67	76.81
	Latency (95% Percentile) [ms]	70.42	410.91	516.81

* Jitter indicates the variation in packet delay, representing transmission stability across consecutive packets.

including system status and position information, were continuously relayed back to the GCS. The developed monitoring GUI enabled real-time visualization of both UAV states and network connectivity throughout the flight operation. Unlike Case 1, which quantitatively evaluates network-level performance metrics, this experiment emphasizes operational continuity and basic controllability at the system level. In particular, the experiment examines whether command-and-telemetry exchange required for supervisory UAV operation can be sustained over a multi-hop wireless path without manual reconfiguration or additional software intervention.

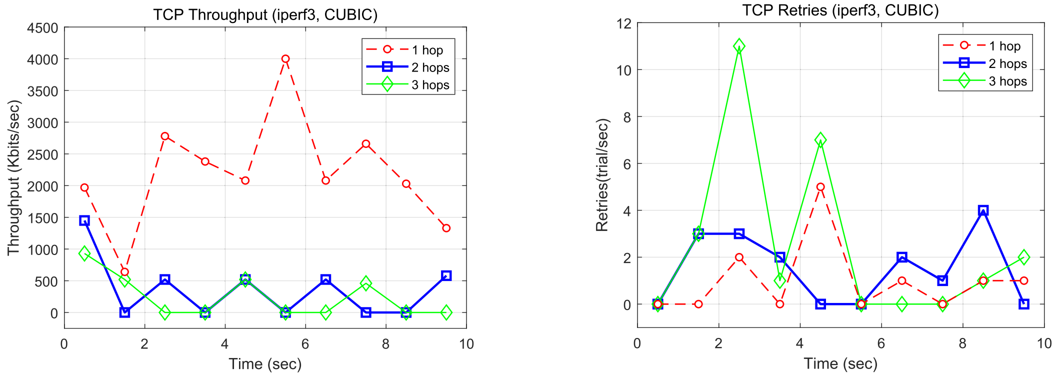
As observed during the flight test and confirmed by GUI outputs and the supplementary video, the UAV remained responsive to supervisory commands, and telemetry feedback was maintained throughout the experiment. No noticeable communication-induced interruptions or safety-critical anomalies were observed under the tested conditions. These results demonstrate that the proposed system can support supervisory UAV control over a multi-hop ad-hoc network in outdoor experimental environments, thereby validating its functional feasibility as a reference implementation for integrated UMV networking.

E. DISCUSSION

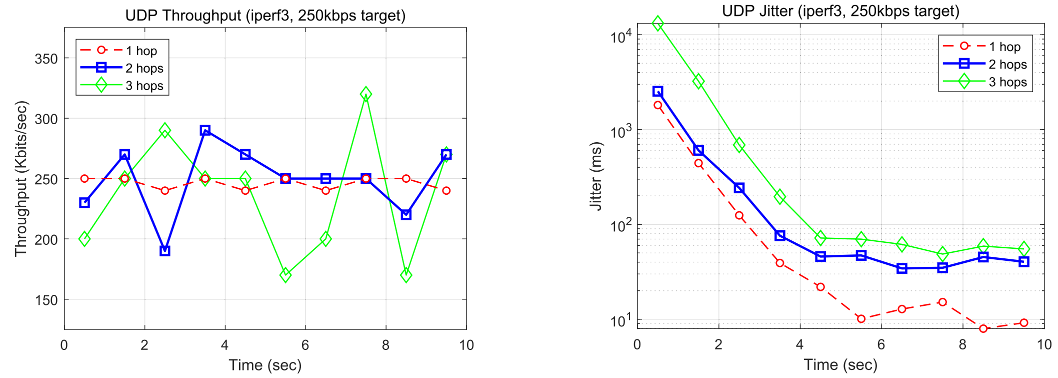
From a control perspective, the throughput across all hop counts exceeds the bandwidth requirements for MAVROS-based command-and-telemetry traffic summarized in Table 1. This indicates that, under the evaluated experimental conditions, the proposed ad-hoc network can support supervisory UMV control and state monitoring without bandwidth saturation.

Building on this observation, the experimental results demonstrate the feasibility of system-level integration and multi-hop operation using commercial off-the-shelf hardware and a custom ad-hoc networking stack. The evaluation was conducted with a limited number of nodes and hop counts (up to three hops) in outdoor environments using 2.4 GHz Wi-Fi hardware and representative command-and-telemetry traffic. Accordingly, the results should be interpreted as an empirical reference for integrated multi-hop UMV networking under the tested conditions, rather than as a comprehensive characterization across diverse environments.

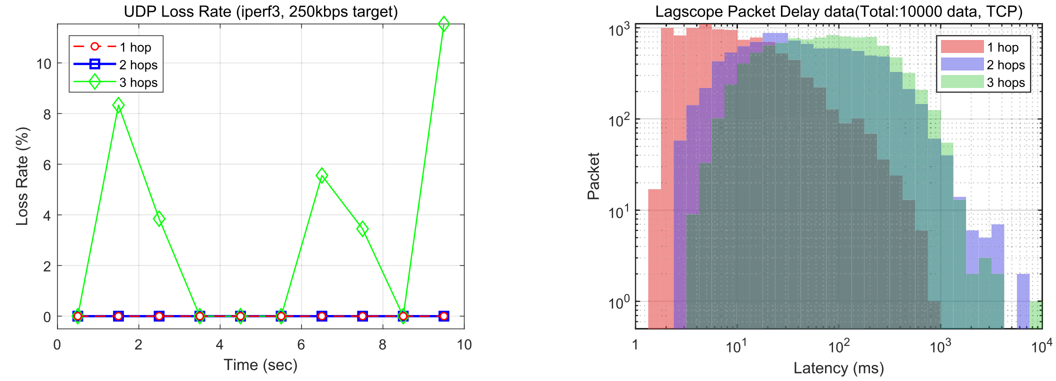
To obtain repeatable and interpretable measurements, mobility effects were minimized in the baseline experiments, enabling stable observation of throughput, latency, jitter, and packet loss under enforced multi-hop conditions. As with typical wireless ad-hoc networks, overall performance and scalability remain influenced by factors such as airtime contention, external interference, and cumulative multi-hop effects. These aspects constitute important practical considerations for broader deployment scenarios and motivate further



(a) TCP Throughput (b) TCP Retries



(c) UDP Throughput (d) UDP Jitter



(e) UDP Loss Rate (f) Latency Histogram

FIGURE 5. Case 1 experimental results for 1-hop, 2-hop, and 3-hop communication.(a)–(e) show time-averaged metrics computed over fixed 1-second windows, including TCP throughput and retries, UDP throughput, jitter, and loss rate.(f) presents a latency histogram obtained from 10,000 individually measured TCP packet delays.

evaluation under more diverse environmental and operational conditions.

V. CONCLUSION AND FUTURE WORK

This paper presented a practical ad-hoc networking system for UMVs, enabled by a standalone ad-hoc module and custom firmware, with an emphasis on system-level integration and experimental validation. By decoupling networking functions from onboard control and computation, the proposed design supports multi-hop communication without requiring modifications to each platform's internal hardware or software.

Two complementary experiments were conducted to evaluate the proposed system. Case 1 quantitatively characterized multi-hop communication performance, while Case 2 demonstrated the functional feasibility of supervisory UAV control over a multi-hop ad-hoc link. Together, these results provide an empirical reference for multi-UMV networking under the evaluated outdoor experimental conditions.

Beyond the conducted experiments, the proposed framework is applicable to UMV operations in environments where direct communication between a ground control station and vehicles is constrained by distance, occlusion, or wireless interference. In such settings, cooperative multi-hop relaying among UMVs can extend communication coverage and enable continued supervisory control and state monitoring.

Future work will focus on extending the current implementation toward performance-aware optimization, including energy-efficient communication strategies, evaluation under larger network scales, and additional field experiments in more diverse and dynamic environments.

REFERENCES

[1] Y. Qian, G. Liu, and H. Shi, "Deep learning method to optimize the quality of services in UAV-aided 5G/6G networks," *Wireless Netw.*, vol. 30, no. 8, pp. 7003–7013, Nov. 2024.

[2] D. Tang and Q. Zhang, "UAV 5G: Enabled wireless communications using enhanced deep learning for edge devices," *Wireless Netw.*, vol. 30, no. 8, pp. 7123–7136, Nov. 2024.

[3] X. Gu and G. Zhang, "A survey on UAV-assisted wireless communications: Recent advances and future trends," *Comput. Commun.*, vol. 208, pp. 44–78, Aug. 2023.

[4] F. Pasandideh, J. P. J. D. Costa, R. Kunst, W. Hardjawana, and E. P. de Freitas, "A systematic literature review of flying ad hoc networks: State-of-the-art, challenges, and perspectives," *J. Field Robot.*, vol. 40, no. 4, pp. 955–979, Jun. 2023.

[5] D. Ramphull, A. Mungur, S. Armoogum, and S. Pudaruth, "A review of mobile ad hoc network (MANET) protocols and their applications," in *Proc. 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2021, pp. 204–211.

[6] Z. A. Younis, A. M. Abdulazeez, S. R. M. Zeebaree, R. R. Zebari, and D. Q. Zeebaree, "Mobile ad hoc network in disaster area network scenario: A review on routing protocols," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 17, no. 3, pp. 49–75, Mar. 2021.

[7] D. N. M. Hoang, J. M. Rhee, and S. Y. Park, "Fault-tolerant ad hoc on-demand routing protocol for mobile ad hoc networks," *IEEE Access*, vol. 10, pp. 111337–111350, 2022.

[8] N. Shah, H. El-Ocla, and P. Shah, "Adaptive routing protocol in mobile ad-hoc networks using genetic algorithm," *IEEE Access*, vol. 10, pp. 132949–132964, 2022.

[9] M. Tahboush and M. Agoyi, "A hybrid wormhole attack detection in mobile ad-hoc network (MANET)," *IEEE Access*, vol. 9, pp. 11872–11883, 2021.

[10] D. Kafetzis, S. Vassilaras, G. Vardoulias, and I. Koutsopoulos, "Software-defined networking meets Software-defined radio in mobile ad hoc networks: State of the art and future directions," *IEEE Access*, vol. 10, pp. 9989–10014, 2022.

[11] S. Xiao, X. Ge, Q.-L. Han, and Y. Zhang, "Resource-efficient platooning control of connected automated vehicles over VANETs," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 579–589, Sep. 2022.

[12] S. Jayashree and S. V. N. S. Kumar, "An efficient group signature based certificate less verification scheme for vehicular ad-hoc network," *Wireless Netw.*, vol. 30, no. 5, pp. 3269–3298, Jul. 2024.

[13] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, "A comprehensive review of authentication schemes in vehicular ad-hoc network," *IEEE Access*, vol. 9, pp. 31309–31321, 2021.

[14] M. A. Al-Shareeda and S. Manickam, "A systematic literature review on security of vehicular ad-hoc network (VANET) based on VEINS framework," *IEEE Access*, vol. 11, pp. 46218–46228, 2023.

[15] A. A. Munshi, S. Sharma, and S. S. Kang, "A review on routing protocols for flying ad-hoc networks," in *Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2018, pp. 1270–1274.

[16] D. S. Lakew, U. Sa'ad, N.-N. Dao, W. Na, and S. Cho, "Routing in flying ad hoc networks: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 22, no. 2, pp. 1071–1120, 2nd Quart., 2020.

[17] R. Kumar, B. Sharma, and S. Athithan, "TBMR: Trust based multi-hop routing for secure communication in flying ad-hoc networks," *Wireless Netw.*, vol. 23, pp. 1–17, Sep. 2023.

[18] O. T. Abdulhae, J. S. Mandeep, and M. Islam, "Cluster-based routing protocols for flying ad hoc networks (FANETs)," *IEEE Access*, vol. 10, pp. 32981–33004, 2022.

[19] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.

[20] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless mesh networking: An IoT-oriented perspective survey on relevant technologies," *Future Internet*, vol. 11, no. 4, p. 99, Apr. 2019.

[21] L. Ferranti, S. D'Oro, L. Bonati, E. Demirors, F. Cuomo, and T. Melodia, "HIRO-NET: Self-organized robotic mesh networking for internet sharing in disaster scenarios," in *Proc. IEEE 20th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2019, pp. 1–9.

[22] M. Tranzatto et al., "CERBERUS: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge," *Field Robot.*, vol. 2, pp. 274–324, Mar. 2022.

[23] T. H. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, document RFC 3626, Oct. 2003.

[24] D. Seither, A. König, and M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of BATMAN advanced," in *Proc. IEEE 36th Conf. Local Comput. Netw.*, Oct. 2011, pp. 897–904.

[25] J. Chroboczek and D. Schinazi, *The Babel Routing Protocol*, document RFC 8966, Jan. 2021.

[26] S. Liu, C. Dong, X. Zhu, J. Tang, and L. Zhang, "Performance evaluation of BATMAN-adv protocol on convergecast communication in UAV networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 5105–5110.

[27] Z. Zheng, T. Yong, J. Li, and Z. Wen, "Simulation research of UAANET based on BATMAN-ADV routing protocol," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2022, pp. 232–236.

[28] A. Velázquez-Mena, H. Benítez-Pérez, R. C. Rodríguez-Martínez, and R. F. Villarreal-Martínez, "Design and evaluation of indoor wireless ad hoc network using BATMAN-adv with mobile robots," in *Proc. IEEE 17th Conf. Ind. Electron. Appl. (ICIEA)*, Dec. 2022, pp. 376–383.

[29] 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 *Proc. 18th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2022, pp. 261–268.

[30] E. Kulla, M. Hiyama, M. Ikeda, and L. Barolli, "Performance comparison of OLSR and BATMAN routing protocols by a MANET testbed in stairs environment," *Comput. Math. Appl.*, vol. 63, no. 2, pp. 339–349, Jan. 2012.

[31] A. Guillen-Perez, A.-M. Montoya, J.-C. Sanchez-Aarnoutse, and M.-D. Cano, "A comparative performance evaluation of routing protocols for flying ad-hoc networks in real conditions," *Appl. Sci.*, vol. 11, no. 10, p. 4363, May 2021.

[32] I. Antoniou, V. V. Ivanov, V. V. Ivanov, and P. V. Zrelov, "On the log-normal distribution of network traffic," *Phys. D, Nonlinear Phenomena*, vol. 167, nos. 1–2, pp. 72–85, Jul. 2002.

JIN-HEE SON received the B.S. degree in electronics system engineering from Hanyang University, Ansan, South Korea, in 2016. He is currently pursuing the integrated M.S. and Ph.D. degree in mechanical and robotics engineering with Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. His research interests include multi-agent systems, communication systems, and distributed optimization.

JEONG-MIN MA received the B.S. degree in mechanical engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2019, where he is currently pursuing the integrated M.S. and Ph.D. degree in mechanical and robotics engineering. His research interests include multi-agent systems, graph theory, and consensus algorithms.

HYUNG-GON LEE received the B.S. degree in mechanical system engineering from Jeonbuk National University, Jeonju, South Korea, in 2019. He is currently pursuing the integrated M.S. and Ph.D. degree in mechanical and robotics engineering with Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. His research interests include multi-agent systems, distributed algorithms, and network decision making.

NAM-JIN PARK received the B.S. degree in control and instrumentation engineering from Pukyong National University, Busan, South Korea, in 2017, and the Ph.D. degree in mechanical and robotics engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. He is currently an Assistant Professor with the Department of Electrical Engineering, Kunsan National University. His research interests include graph theory, network controllability, multi-agent systems, consensus networks, and hardware/software development for autonomous multi-agent control.

HYO-SUNG AHN (Senior Member, IEEE) received the B.S. and M.S. degrees in astronomy from Yonsei University, Seoul, South Korea, in 1998 and 2000, respectively, the M.S. degree in electrical engineering from the University of North Dakota, Grand Forks, ND, USA, in 2003, and the Ph.D. degree in electrical engineering from Utah State University, Logan, UT, USA, in 2006. He is currently a Professor with the School of Mechanical and Robotics Engineering, Gwangju Institute of Science and Technology (GIST). His research interests include distributed control, aerospace navigation and control, network localization, and learning control.