

A Novel UAV-assisted VANET Routing Protocol for Post-Disaster Emergency Communications

Zhijie Fan, Mansi Zhang, Yue Cao, *Senior Member, IEEE*, Zilong Liu, *Senior Member, IEEE*, Omprakash Kaiwartya, *Senior Member, IEEE*, Yasir Javed, *Member, IEEE*, and Faisal Bashir Hussain

Abstract—After natural disasters, such as earthquakes or tsunamis, terrestrial communication networks often become in-operative due to infrastructure collapse. Simultaneously, damage to roads and transportation systems inevitably isolates different parts of the affected area, making it challenging for emergency vehicles to reach critical locations and deploy mobile Base Stations (BSs). In such scenarios, Unmanned Aerial Vehicles (UAVs) serve as a flexible and efficient solution. With the capability to establish temporary communication links, UAVs can provide emergency coverage for ground entities. In this paper, we propose a Dynamic Priority-based UAV-assisted Vehicular Ad-hoc Network (VANET) Routing (DPUVR) protocol for post-disaster message transmission. Specifically, DPUVR is a trajectory-based method for controlling the direction of message forwarding. DPUVR utilizes a multi-attribute decision-making method to adaptively evaluate the message delivery capability of candidate nodes (in this paper, nodes refer to both UAVs and vehicles), taking into account trajectory similarity, surplus energy, link survival time, remaining distance cost and queuing delay. In addition, we propose a dynamic prioritization delivery model. It evaluates the priority of messages in node buffers, selects appropriate candidate nodes and then chooses the best relay for message forwarding to trigger timely and efficient message delivery. Extensive simulation results show that DPUVR significantly outperforms other baseline methods in terms of delivery ratio, overhead, average delivery latency and average buffering time.

Index Terms—Unmanned Aerial Vehicle (UAV), Vehicular Ad-hoc Network (VANET), trajectory, emergency communications.

I. INTRODUCTION

IN recent years, large-scale natural disasters such as the 2023 Turkey-Syria earthquake, the 2022 Pakistan floods and 2021 Haiti earthquake have exposed the severe vulnerability of terrestrial communication infrastructures. These events

often result in the collapse of network infrastructure, leading to communication blackouts in the most critical hours for life-saving operations [1], [2]. Without timely information exchange, survivors struggle to request emergency assistance, while first responders face critical challenges in coordinating search and rescue efforts [3]. To mitigate these consequences, establishing a rapidly deployable, robust and reliable emergency communication network is essential.

After severe natural disasters, terrestrial communication infrastructures may be completely damaged, leading to information occlusions in emergency response. Crucially, the obstruction of rescue messages can cause significant casualties and economic losses. Unmanned Aerial Vehicles (UAVs), with their high mobility and deployment flexibility, have emerged as promising platforms for rapidly restoring communication in disaster-stricken areas. By acting as emergency BSs or relays, UAVs can rapidly and flexibly relay information from external networks to users in the disaster area [4], [5], [6]. However, UAVs are limited by their energy and load capacity, which makes it difficult to prolong the communication period and goods delivery.

To tackle this challenge, the collaboration between UAVs and ground vehicles has been extensively explored. For instance, the work [7] introduced an air-ground integrated emergency network, which improves connectivity and extends communication coverage in disaster-hit areas. In their model, UAVs act as aerial relays to bridge fragmented areas and assist Vehicle-to-Vehicle (V2V) communication. In addition, the study in [8] proposed a deep reinforcement learning based framework named DISCOUNT, which dispatches an organized fleet of UAVs to serve as long-term communication relays. This framework optimizes UAV trajectories by jointly considering energy efficiency, connectivity maintenance and collision avoidance in intermittently connected VANET environments. Other studies have adopted Store-Carry-and-Forward (SCF) mechanisms [9], where UAVs and vehicles cooperate to deliver messages across disconnected regions. In these methods, UAVs are typically responsible for collecting messages from isolated nodes and forwarding them to mobile ground relays or edge servers. Furthermore, bio-inspired and heuristic routing strategies have been introduced to improve delivery efficiency. These methods select optimal relays based on factors such as mobility prediction, energy metrics and link stability [10], [11]. Such works demonstrate that UAV-assisted VANETs can form a resilient communication infrastructure by leveraging the strengths of both aerial and ground nodes.

Despite extensive research on UAV-assisted VANET rout-

Zhijie Fan and Mansi Zhang are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430000, China. (e-mail: zhijie.fan@whu.edu.cn, mansi.zhang@whu.edu.cn).

Yue Cao is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430000, China, and also with the State Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing 100081, China. (e-mail: yue.cao@whu.edu.cn).

Zilong Liu is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: zilong.liu@essex.ac.uk).

Omprakash Kaiwartya is with the School of Science and Technology, Nottingham Trent University, NG11 8NS Nottingham, U.K. (e-mail: omprakash.kaiwartya@ntu.ac.uk).

Yasir Javed is with the Department of Software Engineering, Prince Sultan University, Riyadh 11586, Saudi Arabia. (e-mail: yjaved@psu.edu.sa).

Faisal Bashir Hussain is with the Department of Computer Science, Bahria University, Islamabad 44000, Pakistan (e-mail: fbashir.buic@bahria.edu.pk).

Corresponding author: Yue Cao (yue.cao@whu.edu.cn).

ing, existing methods still face significant limitations in real-world post-disaster applications. First, many protocols assume persistent or semi-persistent connectivity, which does not reflect the intermittent and fragmented nature of networks in disaster-stricken areas. Second, relay selection schemes often rely on geographic indicators such as node location and movement direction, overlooking the dynamic and irregular topologies in disaster environments, which can cause local optimization issues. Third, while UAVs are valuable for bridging disconnected regions, their coordination with ground vehicles is often loosely integrated. This lack of coordinated trajectory planning and relay scheduling leads to redundant paths, suboptimal relay selection, and missed transmission opportunities, which ultimately reduces communication efficiency, particularly in disaster environments. In addition, message scheduling mechanisms in prior work seldom account for the temporal urgency or delivery priority of emergency messages. Without such differentiation, significant messages may suffer from delays or packet drops, impeding effective rescue coordination.

To address the above challenges, we propose DPUVR: a dynamic priority-based UAV-assisted VANET routing protocol tailored for post-disaster communication. The key contributions of this paper are summarized as follows:

- **Trajectory-based multi-metric relay selection strategy:** While many existing UAV-assisted VANET routing protocols incorporate basic geographic indicators such as node location or movement direction, they often overlook key dynamic factors including energy availability, link survival time, and real-time buffer status. These limitations hinder accurate relay assessment in highly dynamic and fragmented post-disaster networks. To address this, we design an adaptive relay selection strategy that evaluates candidate nodes using five critical attributes: trajectory similarity, surplus energy, link survival time, remaining distance cost and queuing delay. By jointly considering these spatiotemporal and status-aware indicators, our method supports relay decisions under uncertain and constrained environments, significantly enhancing routing robustness and message delivery performance.
- **Dynamic priority-based message delivery model with adaptive buffer management:** In post-disaster scenarios, time-critical messages often compete for limited transmission opportunities due to infrastructure damage and constrained network resources. To address this, we introduce a dynamic prioritization mechanism that considers both message urgency and real-time buffer status. Unlike static or FIFO-based queue models, our design dynamically adjusts the forwarding priority of each message by analyzing its delay sensitivity, Time-To-Live (TTL) and buffer queue length of potential receivers, thereby avoiding congestion-induced packet loss and long buffering times. Furthermore, a relay selection mechanism is embedded into the prioritization process to match high-priority messages with suitable relays, ensuring timely delivery under network fragmentation and uneven message loads.

The remainder of this paper is organized as follows. Section II illustrates the related work and motivation. Section III displays a detailed system model, establishes and quantifies the message delivery model. Then, the detailed methods of this routing protocol are described in Section IV. Section V evaluates the performance of proposed methods. Finally, Section VI concludes this paper.

II. RELATED WORK

To provide a comprehensive overview of related approaches, Table I presents a comparative summary of representative UAV, VANET and UAV-assisted VANET routing protocols, thereby providing a basis for subsequent discussion.

A. UAV Routing Protocols

Routing protocols for UAV networks and Flying Ad hoc Networks (FANETs) have been extensively studied due to the unique challenges posed by high mobility, 3D movement, dynamic topology and limited energy resources. BLPR [12] introduces a time-varying effective transmission range that adapts to wireless channel conditions and interference, combining with a 3D perimeter forwarding strategy to improve routing robustness. Similarly, the work [13] developed a Q-learning-based Topology-Aware Routing (QTAR) protocol that extends routing decisions beyond one-hop neighbors to two-hop information, enhancing reliability in dynamic FANET topologies. The work [14] applied Machine Learning (ML) using Long Short-Term Memory (LSTM) to predict message arrival and avoid congested UAVs, improving packet delivery and delay performance. The work [15] proposed an energy-efficient RPL-based (Routing Protocol for Low power and Lossy networks) routing protocol for UAVs in Internet of Things (IoT) scenarios, balancing energy load and supporting both large-scale aggregation and urgent communications.

In addition to routing strategies, UAV deployment and message forwarding schemes have been explored to enhance communication performance and resilience. The work [16] addressed UAV deployment in emergency scenarios, where neither the wireless channel model nor the ground user positions are known. To cope with this uncertainty, a model-free online learning method was proposed to enable real-time UAV adaptation based on observed relay performance, without relying on predefined location information. The work [3] designed a UAV-assisted emergency Wi-Fi network leveraging Raspberry Pi boards to provide rapid disaster-area coverage and relay capabilities, with practical experiments validating communication range and network modes. The work [17] proposed a social relationship-based message splitting scheme for energy-efficient UAV-assisted delay tolerant networks. This scheme optimizes delivery ratio and latency by considering both social closeness and message timeliness. Moreover, the work [20] introduced a two-layer UAV routing strategy for post-disaster inspection and monitoring tasks, employing bilevel programming and mixed-integer linear programming to balance real-time updates and large-scale route planning.

Recent studies further enhance UAV routing through multi-metric optimization and learning mechanisms. For instance,

TABLE I: Comparative analysis of related schemes and the proposed method

UAV Routing Protocols								
Schemes	Years	Key Method	Key Metrics	Simulator	P1	P2	P3	P4
[12]	2023	Geographical Routing	Link available period, required service duration, predicted locations	NS-3.29	✓	✓	✓	✗
[13]	2022	Q-Learning, Topology-Aware	Position, delay, velocity, energy	MATLAB	✓	✓	✓	✗
[14]	2022	Adaptive Routing, LSTM	Packet arrival prediction	-	✓	✗	✓	✗
[15]	2021	Information Aggregation	Link quality, distance, energy	-	✗	✓	✓	✗
[17]	2021	Social Network	Timeliness, social closeness, mobility	-	✓	✗	✓	✗
[18]	2023	Modified OLSR	Energy efficiency, position	NS-3	✓	✓	✓	✗
[19]	2023	Deep Q-Network, OLSR	Position, speed	OPNET	✓	✗	✓	✗
[21]	2025	TOPSIS-Aided AODV	Stability, pheromone accumulation	NS-2	✓	✓	✓	✗
[22]	2025	Hybrid Reactive Routing	Mobility awareness, congestion, queue age	NS-3	✓	✗	✗	✗
[23]	2025	Adaptive Virtual Tunnel	Energy, link quality, hop count	NS-3	✓	✗	✓	✗
[24]	2025	Physical Sensing, Intelligent	LSTM-based link prediction	-	✗	✗	✓	✗
[25]	2025	Improved Q-Learning Multihop	Energy efficiency, reliability	MATLAB	✗	✗	✗	✗
[26]	2025	DRL adaptive coding-aware	Adaptive coding-aware	NS3-Gym	✓	✗	✓	✗
[27]	2024	Trajectory-Optimized	Trajectory knowledge, MDWD algorithm	Python3	✓	✗	✓	✗
[28]	2024	Multi-Objective Markov Decision	Energy, Expected Delay, Power/Distance	MATLAB	✓	✗	✓	✗
[29]	2024	Joint Prediction	LSTM + entropy-weight multi-metric	-	✓	✗	✓	✗
[30]	2023	Cross-Layer Mobility-Aware	Multi-metric OLSR optimization	NS-3	✓	✗	✓	✗
[31]	2021	Full-Echo Q-Routing	simulated annealing	MATLAB	✗	✗	✗	✗
VANET Routing Protocols								
Schemes	Years	Key Method	Key Metrics	Simulator	P1	P2	P3	P4
[32]	2025	GPSR, Geographic Routing	Link duration, relative distance, etc.	NS-2.35	✓	✓	✓	✗
[33]	2025	Adaptive Routing	Message priority, free nodes, mobility direction	ONE	✓	✓	✓	✗
[34]	2025	Topology Evolution	Link reliability, SINR	-	✓	✗	✓	✗
[35]	2023	Dynamic Topology Evolution	SINR, buffer queue, link availability	-	✓	✗	✓	✗
[36]	2021	Heuristic Routing	Distance, connectivity, speed, direction, available buffer size, signal fading	VSIM	✓	✓	✓	✗
[37]	2022	Weighted Geographical Routing	Mobility, distances, speed, link time, channel quality, etc.	SUMO-0.32 NS-3.23	✓	✓	✓	✗
[38]	2018	Trajectory-Driven Routing	Proximity to trajectory	ONE	✓	✓	✓	✗
[39]	2023	Trajectory-Based	Spatio-temporal domain parameters	ONE	✓	✓	✓	✓
[40]	2025	Trajectory-Based, Anycast	Link quality, trajectory similarity, distance	ONE	✓	✓	✓	✗
[42]	2023	Reinforcement Learning	Trajectory, speed, user preferences	SUMO	✓	✗	✓	✗
[43]	2023	PDR-based, Broadcast	Vehicle distance and density	Python	✓	✗	✗	✗
[44]	2025	Robust Predictive Routing	Context-aware link estimation	SUMO	✗	✗	✗	✗
[45]	2025	QoS-Adaptive Routing	Throughput, adaptive delay, QoS-awareness	NS-2	✓	✗	✓	✗
[46]	2024	CDSPAN Multicast	Multicast backbone, geometric	NS-3.29	-	✓	✓	✗
[47]	2024	5G Self-Organizing Routing	Energy-efficient, low-speed ITS	NS-2	✓	✗	✓	✗
[48]	2024	Lightweight, Cross-layer	Delay, energy, real-time V2V	NS-2.34	✓	✓	✗	✗
[49]	2024	RSU-Assisted DRL	Hybrid road-aware, Q-learning	Unity 3D	✗	✗	✓	✗
[50]	2023	Intersection-based QoS Routing	Reinforcement learning, intersection metrics	NS-2	✓	✓	✓	✗
[51]	2023	Weight-Based Clustering	Speed-degree weighted cluster head	SUMO	✓	✓	✓	✗
[52]	2023	Trust-Based EM Transmission	Node trust, link SNR, reliability	NS-3	✓	✗	✓	✗
[53]	2023	Agent-Based Reactive Geographic	Mobile agents, low overhead	-	✓	✓	✓	✗
[54]	2021	SPEA-Based Multimetric	Multi-objective optimization, delay	SUMO	✓	✓	✓	✗
UAV-assisted VANET Routing Protocols								
Schemes	Years	Key Method	Key Metrics	Simulator	P1	P2	P3	P4
[5]	2022	Interior Penalty Function	Delay, energy, data rate	MATLAB	✗	✗	✓	✗
[8]	2021	Deep Reinforcement Learning	Coverage, connectivity	Python	✓	✓	✓	✗
[9]	2022	Heuristic	Geography parameters, mobility	ONE	✓	✓	✓	✗
[10]	2021	Delay Tolerant Routing	Encounter probability, connection time	ONE	✓	✓	✓	✗
[56]	2019	Flooding, Prediction	Path expiration time	NS-2	✓	✓	✓	✗
[57]	2017	Delay Tolerant, Reactive routing	Traffic density, connectivity	NS-2	✓	✓	✓	✗
[58]	2021	Classification Learning	Handoff failure, access unavailability	-	✓	✗	✓	✗
[59]	2024	Bilateral Matching Game	Transmission rate, energy, etc.	MATLAB	✗	✗	✗	✗
[60]	2021	Artificial Intelligence	Speed, position, trust value	NS-2.35	✓	✓	✓	✗
[61]	2020	Q-learning	State transition probability, consumption	-	✓	✓	✓	✗
[62]	2021	Q-learning, Geographic	Distance, node degree	SUMO, NS-3	✓	✗	✓	✗
[63]	2025	DRL-Enhanced	Trajectory planning, MASAC optimization	OMNeT++	✓	✗	✓	✗
[64]	2024	Energy-Efficient UAV-aided	Unequal clustering, dynamic sectors	MATLAB	✗	✗	✗	✗
[65]	2024	Landslide Area UAV-aided	Fuzzy logic, energy balance	NS-2	✓	✗	✓	✗
[66]	2023	Low-Delay Routing	Optimized OLSR, smart city	NS-3	✓	✓	✓	✗
[67]	2023	UAV-Assisted Communication	LOS enhancement, throughput, delay	NS-2	✗	✗	✓	✗
[68]	2023	UAV-IoT Small-World Network	Delay-constrained, energy-throughput	-	✓	✗	✓	✗
[69]	2023	Q-Learning Cognitive Swarm	Energy, SNR, utility maximization	-	✗	✗	✗	✗
[70]	2024	Drone-Assisted Cooperative Routing	Dual-radio IoT integration, seamless connectivity	NS-2	✓	✓	✓	✗
Ours	2025	Trajectory-Based, Priority-Based	Trajectory similarity, energy, link time, distance, queuing delay	ONE	✓	✓	✓	✓

1. "✓" indicates that the corresponding parameter is included in the simulation, and "✗" means it is not considered.

2. P1: Delivery Ratio. P2: Overhead Ratio. P3: Delivery Latency. P4: Buffer Time.

TAAODV [21], AVTR [23], and Cross-Layer OLSR [30] introduce stability-aware and congestion-reducing enhancements, while PSIF [24], IQMR [25], and DRL-AdCAR [26] apply deep and reinforcement learning to predict link quality and optimize forwarding. Trajectory-oriented and decision-based frameworks, including TORP [27], MOBMDP [28], and JPE [29], leverage predictive mobility and multi-objective routing optimization. Moreover, Full-Echo Q-routing [31] combines simulated annealing and adaptive learning for dynamic UAV networks. These approaches collectively advance the adaptability, delay performance, and energy balance of UAV routing under complex aerial conditions.

Despite progress, many UAV routing protocols still rely on overly simplified assumptions such as stable channel conditions and reliable neighbor availability, which are difficult to guarantee in highly dynamic and interference-prone UAV networks. Additionally, ML-based methods show potential but often need large training data and high computation, limiting real-time utilization.

B. VANET Routing Protocols

VANETs have received considerable attention for enabling intelligent transportation systems through reliable inter-vehicle communication. To address the challenges of dynamic topologies and unstable links, a wide range of routing protocols have been developed. For instance, the HRLIGR protocol [32] improves the classic GPSR protocol with enhanced link stability metrics and low-latency forwarding, outperforming AODV and OLSR in terms of delivery ratio and latency. Similarly, MPFD [33] leverages urgency-aware message prioritization and relay selection based on community affiliation and motion patterns to boost delivery efficiency. Meanwhile, multi-objective optimization-based methods such as [34] employ dynamic topology evolution models. The author utilized the Dijkstra algorithm to select stable paths in highly variable urban environments. The DTE-RR protocol [35] further expands on topology dynamics by modeling link interactions with Markov-based state transitions, thereby improving packet delivery ratios and delay performance. Moreover, HERO [36] proposes a heuristic road-segment and vehicle selection method. It considers speed, direction, buffer status and signal quality, leading to notable gains in throughput and lower overhead. For time-sensitive data like health information, W-GeoR [37] utilizes a weighted metric considering traffic mobility, inter-vehicle distances, speed differences, communication link expiration time and channel quality, to expedite next-hop selection, especially in post-disaster or low-connectivity urban scenarios.

Beyond topology-driven solutions, several works emphasized opportunistic and trajectory-based designs. The TDOR protocol [38] utilizes vehicle GPS data to guide message relays along probable movement trajectories, increasing reliability in sparse networks. Likewise, STALB [39] integrates buffer state and local congestion awareness into a trajectory-based routing model, achieving up to 99.9% delivery ratio under sparse conditions and dramatically reducing end-to-end delay. The TBAR protocol [40] proposes an anycast strategy targeting MDRUs in post-disaster scenarios, using a multi-attribute relay

evaluation metric that considers link quality, trajectory similarity and message cost, significantly outperforming baseline protocols. Moreover, the Social Internet of Vehicle (SIOV) based emergency diffusion model [41] incorporates social stability into V2V link construction. It enhances the influence spread through b-matching and the selection of influential vehicles. In parallel, recent studies also explore edge intelligence and communication efficiency in 6G-enabled vehicular environments. A deep reinforcement learning-based collaborative caching scheme [42] was proposed for vehicular edge networks with cell-free massive MIMO. It jointly considers vehicle mobility and content preferences to optimize caching decisions, reducing delivery latency and improving cache hit ratios. Furthermore, the PDR-MD protocol [43] introduces a packet delivery ratio-based dissemination strategy for V2V emergency messages. By balancing PDR and inter-vehicle distance, it achieves timely, reliable broadcast performance in dense vehicular scenarios.

More recent efforts extend VANET routing with predictive, QoS, and learning-based optimizations. For instance, ROPE [44] enhances link reliability via context prediction, and PRP [45] dynamically adjusts QoS to maintain throughput. Clustered and RSU-supported designs such as CDSPAN [46] and RHRA-DRL [49] exploit network geometry and deep learning for multicast and hybrid road routing. Energy-efficient and low-latency mechanisms like LRECR [48], 5G-SOR [47], and IQRR [50] target real-time traffic scenarios. Similarly, trust and agent based frameworks TLEM [52], ARGENT [53], and SPEA MRP [54] focus on secure, multi metric, and adaptive routing. Together, these studies highlight a shift toward lightweight, intelligent, and trustworthy VANET routing paradigms.

While existing VANET routing protocols demonstrate improvements in delivery ratio, latency and adaptability, most remain tailored to specific scenarios or rely on assumptions such as predictable mobility, stable connectivity, or uniform traffic density. This limits their robustness across diverse and rapidly changing environments like post-disaster areas or highly congested urban areas.

C. UAV-assisted VANET Routing Protocols

Routing in VANETs assisted by UAVs has garnered considerable interest. This method enhances communication robustness under dynamic and fragmented network conditions, particularly in post-disaster scenarios. Early contributions, such as the on-demand aerial BS strategy proposed in [55], demonstrate the feasibility of employing UAVs to support VANET communication in emergencies. The authors modeled both ground-to-ground and ground-to-air channels under single and multiple UAV scenarios. The message flooding scheme in [56], allow vehicles to cooperate in an ad hoc fashion with existing UAVs. This kind of collaboration provides reliable routing paths and ensures alternative solutions in the case of path failures. The work [57] proposed a dual-mode communication strategy, where UAVs operate either as ground-assisted relays to enhance V2V routing or as independent aerial forwarders in the sky, enabling flexible routing across sparse or obstructed VANETs. The work [58] introduced a non-recurrent

classification learning-based dissemination model to tackle unstable connectivity and handoff failures in drone-assisted urban VANETs. Additionally, the work [59] introduced an air-ground cooperative framework utilizing multiple UAVs and Unmanned Ground Vehicles (UGVs) for disaster information collection and forwarding. It models UAV-UGV coordination as a bipartite graph and solves the resource allocation problem using a bidirectional matching game.

Recent studies have also focused on enhancing routing efficiency and security in UAV-assisted VANETs through optimization and intelligent decision-making. For example, the work [10] introduced a protocol that jointly considers encounter probability and persistent connection time. The protocol aims to improve link stability and reduce end-to-end delay in UAV-assisted VDTNs. Bio-inspired methods, such as the social spider optimization [9], seek to find a better selection of next SCF relays specifically for Vehicle-to-UAV (V2U) and UAV-to-Vehicle (U2V) connections, which aim to improve delivery ratio and latency in sparse urban environments. Similarly, the VRU protocol [60] incorporates distinct V2U and UAV-to-UAV (U2U) routing strategies. It achieved significant gains in message delivery and security metrics. To further enhance throughput, the work [71] integrated Non-Orthogonal Multiple Access (NOMA) with UAV deployment optimization via particle swarm optimization, targeting efficient subcarrier allocation and secure high-rate U2V communication. Similarly, the work [72] studied a NOMA-based integrated satellite-UAV-terrestrial network, showing improved spectrum efficiency and transmission capacity under overlay access, suggesting that NOMA can effectively support high-throughput communication in layered UAV-assisted architectures. Additionally, the work [73] proposed a lightweight two-factor authentication and key agreement scheme using chaotic maps and fuzzy verifiers. The scheme ensures secure UAV-vehicle communication in UAV-assisted VANETs during emergency rescue, balancing security with computational and communication overhead. In terms of Medium Access Control (MAC), protocol-level enhancements have also been proposed to optimize message prioritization and reliability [74], [75].

Complementary studies expand UAV-assisted VANET routing toward energy efficiency, learning, and IoT integration. ADMR [63] applies deep reinforcement learning for aerial trajectory planning, while energy-aware UAV-WSN routing frameworks [64], [65] enhance endurance in environmental monitoring. Low-delay and ubiquitous communication schemes [66], [67] optimize OLSR and multi-channel routing for smart cities, and small-world IoT architectures [68] reduce delay under energy constraints. Q-learning swarm coordination [69] and cooperative drone frameworks like DACR [70] further improve reliability, spectrum efficiency, and connectivity continuity. These advances illustrate the convergence of UAV-assisted routing with intelligent, energy-conscious, and IoT-driven communication systems.

D. Motivation

Unfortunately, the aforementioned works still have limitations. Specifically, (1) many existing protocols assume persistent or semi-persistent connectivity, which is unrealistic in

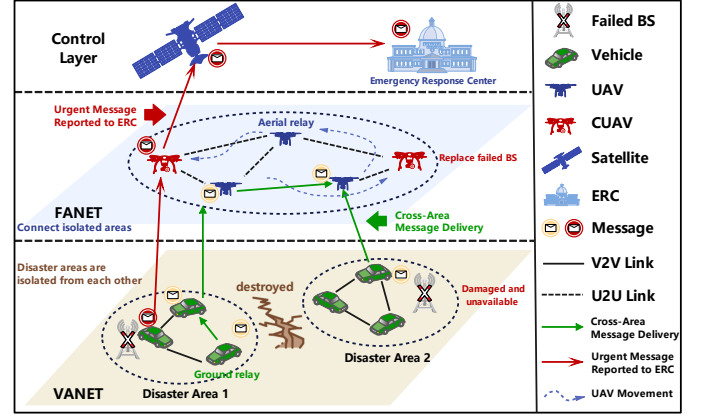


Fig. 1: Network Architecture.

isolated and dynamic disaster areas, (2) UAV-vehicle coordination is frequently loosely integrated, resulting in inefficient collaboration and underutilization of UAVs, and (3) most message delivery models do not incorporate urgency-aware mechanisms, treating all message equally regardless of its priority. This can lead to delays in the transmission of time-sensitive messages critical for effective emergency response. To overcome these challenges, this paper proposes a UAV-assisted VANET routing protocol based on scheduled UAV mobility across disconnected areas. It evaluates relays using multi-attribute metrics, such as trajectory similarity and queuing delay. In addition, it incorporates a dynamic priority-based delivery model to ensure timely and efficient dissemination of emergency messages.

III. SYSTEM MODEL

A. Network Architecture

The post-disaster emergency vehicle network architecture is shown in Fig. 1. This architecture consists of three main components: the VANET, the FANET and the Control Layer. These components form a hierarchical communication framework to ensure robust and continuous information exchange in disaster-affected areas. Traditional cellular infrastructures, such as BSs, are often damaged or unavailable in these scenarios.

After a disaster, ground communication infrastructure is likely fragmented due to collapsed bridges, road blockages, or terrain ruptures, resulting in disconnected network partitions. Each partition contains a functioning VANET composed of vehicles, which are isolated from each other. To bridge these partitions, UAVs are deployed to form a FANET, serving as mobile relays between VANETs. At a higher level, communication satellites and the Emergency Response Center (ERC) form the Control Layer, which manages the global data flow and mission coordination.

1) *VANET*: Each isolated region on the ground hosts a local VANET, composed of vehicles equipped with Dedicated Short-Range Communications (DSRC) modules or IEEE 802.11p interfaces. The VANET enables real-time message exchange within the region, including vehicle status, environmental sensing and survivor information. Due to the failure of ground BSs

(denoted as Failed BS in Fig. 1), these VANETs have no direct access to remote communication services. The VANET follows a delay-tolerant routing mechanism, where message generated in one region is stored until a relay becomes available.

2) *FANET*: The FANET is composed of multiple UAVs, which form an airborne communication network to relay messages between geographically disconnected VANETs. The UAVs operate on fixed trajectories. They periodically traverse the boundaries of each VANET to collect and deliver buffered message. In addition to mobile UAVs, the architecture includes a set of stationary Coordinated UAVs (CUAVs), shown in red in Fig. 1. These CUAVs are deployed at fixed positions to replace failed BSs and serve as temporary infrastructure nodes. Each CUAV provides persistent connectivity within its coverage area and facilitates local communication for the corresponding VANET.

3) *Control Layer*: The Control Layer includes communication satellites and an ERC. UAVs collect disaster-related information and transmit it to the ERC via satellite. The ERC is responsible for overall mission coordination, including UAV scheduling, network configuration updates and strategic resource deployment. It also processes real-time situational message for decision-making by disaster relief authorities. Commands from the ERC can be disseminated to UAVs through satellite, ensuring an adaptive response to dynamic post-disaster scenarios.

4) *Movement Model*: For vehicles in VANETs, their movement is constrained by the road topology within each isolated region. The disaster-affected region is fragmented into multiple disconnected areas due to infrastructure collapse. Within each area, vehicles follow adaptive trajectories along existing roads. Vehicles periodically exchange beacon messages to maintain neighbor awareness and update local routing tables only within their region. Due to the disconnection between regions, V2V communication is limited to intra-area interactions.

For aerial communication, UAVs follow pre-defined, fixed movement routes that are determined before deployment based on the geographic distribution of isolated disaster regions. These routes are carefully planned by the control layer, taking into account terrain constraints and the spatial layout of the disconnected VANETs. Each UAV is assigned a specific route that traverses multiple isolated areas, enabling it to collect, carry and deliver message between regions that lack direct communication. Unlike adaptive or dynamically scheduled UAV mobility models, the UAV trajectories in our system are static and known in advance. This design reduces system complexity, eliminates the need for in-flight coordination and ensures predictable communication latency.

B. Message Delivery Model

In the proposed DPUVR framework, we model the emergency message transmission process as a unicast-based, trajectory-guided, priority-aware delivery model under the support of UAV-assisted disconnected area relaying. Each message is initially generated by vehicles in local VANET areas upon the detection of an emergency event. The message contains a destination region, a TTL, priority level (emergency,

TABLE II: NOTATIONS AND SYMBOLS

Notations	Explanation
m	The message that needs to be transmitted
i, j	Nodes (vehicles or UAVs) in the network
$\mathcal{T}_i$	The current moving trajectory of i
$\mathcal{W}_i$	The sliding window of $\mathcal{T}_i$
$\mathcal{T}_m$	The reference trajectory of m
$\mathcal{W}_m$	The sliding window of $\mathcal{T}_m$
$TS(\mathcal{W}_i, \mathcal{W}_m)$	The trajectory similarity between $\mathcal{W}_i$ and $\mathcal{W}_m$
u	An UAV
E_u^P	The propulsion energy consumed by u
E_u^C	The communication energy consumed by u
E_u^I	The initial energy of each UAV
E_u	The surplus energy of u
R	The communication radius of nodes
$T_{i,j}$	The communication duration between i and j
$DC_{i,m}$	The remaining distance cost for i to relay m
$Q_{i,m}$	The queuing delay of m in i 's buffer
$u_i(x)$	The utility value for x th attribute of i
U_i	The utility function value of i
w_x^i	The weight value of the x th attribute of i
θ_i^m	The trajectory similarity angle between $\mathcal{W}_i$ and $\mathcal{W}_m$
S_m	The survivability score of message m
$B_{transmit}$	The transmission bandwidth

important, or regular) and payload size, and may be forwarded in multiple copies to enhance delivery robustness.

The message forwarding includes three main steps:

(1) candidate node evaluation, (2) priority-based queuing and (3) message delivery. Each node evaluates its neighbors based on trajectory similarity, surplus energy, link survival time, distance cost and queuing delay. These metrics are normalized and weighted via a deviation-based method. The node with the highest utility score is selected as the next-hop relay. Once candidate nodes are selected, all valid message-node pairs are inserted into three logical forwarding queues according to message priority. Within each queue, message-node pairs are sorted in descending order by utility value. This ensures that high-priority messages with strong forwarders are transmitted first, improving timely delivery under constrained resources.

C. Mathematical Model

Notations with their descriptions are listed in Table I.

1) Trajectory Similarity:

Definition (Trajectory): Let the road network be modeled as a graph $G = (V, E)$, where V denotes the set of road intersections and E the road segments. A trajectory is a continuous path in G , which can be represented by n road points ($\mathcal{T} = \langle p_1, p_2, \dots, p_n \rangle$). Each point $p_t \in \mathcal{T}$ satisfies $p_t \in V$, and each consecutive pair (p_t, p_{t+1}) satisfies $(p_t, p_{t+1}) \in E$, where $1 \leq t < n$. We utilize $\mathcal{T}$ to describe the reference trajectory of a message or the mobility of a node.

The trajectory similarity, denoted as TS , quantifies the proximity between a node's moving trajectory and the reference trajectory of a message. As shown in Fig. 2, $\mathcal{T}_i$ represents the current moving trajectory of the node i , while $\mathcal{T}_m$ represents the reference trajectory of message m . To compare these two trajectories locally, we define sliding windows $\mathcal{W}_i$ and $\mathcal{W}_m$, which are short subsequences of three consecutive road points extracted from $\mathcal{T}_i$ and $\mathcal{T}_m$, respectively.

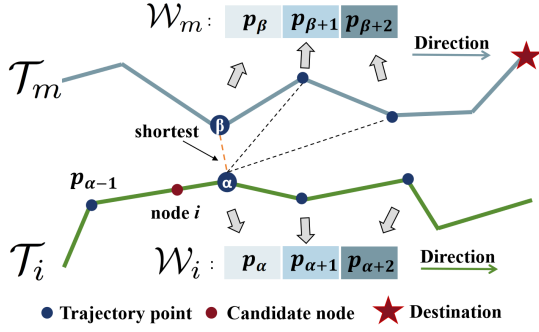


Fig. 2: An example of trajectory similarity.

Specifically, assume node i is currently located on road segment $(p_{\alpha-1}, p_{\alpha})$ of $\mathcal{T}_i$. Then, the sliding window $\mathcal{W}_i$ is defined as $\langle p_{\alpha}, p_{\alpha+1}, p_{\alpha+2} \rangle$. Similarly, $\mathcal{W}_m$ is selected from $\mathcal{T}_m$ such that the starting point $p_{\beta} \in \mathcal{T}_m$ is the trajectory point closest to the current position of node i , i.e., $\langle p_{\beta}, p_{\beta+1}, p_{\beta+2} \rangle$. Then, $\mathcal{W}_i$ and $\mathcal{W}_m$ are utilized to calculate the similarity of two trajectories $\mathcal{T}_i$ and $\mathcal{T}_m$.

The distance from a point p to a sub-trajectory $\mathcal{W}$ ($p \notin \mathcal{W}$), denoted as $D(p, \mathcal{W})$, is defined as follows,

$$D(p, \mathcal{W}) = \min_{p' \in \mathcal{W}} ED(p, p'). \quad (1)$$

where $ED(p, p')$ is the Euclidean distance between two trajectory points p and p' .

TS measures the average minimum distance from each point in a trajectory to another trajectory. The distance from each point in $\mathcal{W}_i$ to $\mathcal{W}_m$ is not necessarily equal to the distance from each point in $\mathcal{W}_m$ to $\mathcal{W}_i$, i.e., $TS(\mathcal{W}_i \rightarrow \mathcal{W}_m) \neq TS(\mathcal{W}_m \rightarrow \mathcal{W}_i)$. Therefore, $TS(\mathcal{W}_i, \mathcal{W}_m)$ is asymmetric and can be computed by:

$$TS(\mathcal{W}_i \rightarrow \mathcal{W}_m) = \frac{1}{|\mathcal{W}_i|} \sum_{p' \in \mathcal{W}_i} e^{-D(p', \mathcal{W}_m)}. \quad (2)$$

$$TS(\mathcal{W}_m \rightarrow \mathcal{W}_i) = \frac{1}{|\mathcal{W}_m|} \sum_{p' \in \mathcal{W}_m} e^{-D(p', \mathcal{W}_i)}. \quad (3)$$

$$TS(\mathcal{W}_i, \mathcal{W}_m) = \frac{1}{2} [TS(\mathcal{W}_i \rightarrow \mathcal{W}_m) + TS(\mathcal{W}_m \rightarrow \mathcal{W}_i)]. \quad (4)$$

Discussion: The exponential kernel in Eqs. (2)–(4) converts spatial distance into a similarity score in the range $(0, 1]$, where higher values indicate stronger trajectory alignment between a node and a message. Intuitively, when the geometric distance $D(p', \mathcal{W}_m)$ is small, the exponential term $e^{-D(p', \mathcal{W}_m)}$ approaches 1, meaning that node i is moving along nearly the same route as the message trajectory. Conversely, large spatial deviations lead to exponentially decreasing similarity, effectively filtering out nodes that are traveling in unrelated directions. The exponential form is chosen because it provides a smooth decay with distance and automatically normalizes the similarity to the range $(0, 1]$ without requiring additional scaling.

Since each window $\mathcal{W}_i$ and $\mathcal{W}_m$ contains only three consecutive road points, the computational complexity of

Eqs. (2)–(4) is $O(|\mathcal{W}_i| \times |\mathcal{W}_m|) = O(9)$ per comparison, which is negligible for real-time operation. When integrated into DPUVR, trajectory similarity is recalculated every 100 s during trajectory updates, ensuring that the routing decision reflects the most recent node movement patterns while keeping the overhead low.

Overall, this asymmetric yet averaged similarity formulation captures both local directional consistency and spatial proximity between node and message trajectories, thereby enhancing relay selection accuracy in highly dynamic post-disaster environments.

2) *Surplus Energy:* UAVs have limited energy capacity as a result of their constrained onboard batteries. The battery lifespan is influenced by various factors, such as the UAV's energy source, type, weight and speed. Usually, the UAV's energy usage can be categorized into two main components: propulsion energy and communication energy [76].

To model the propulsion energy, we utilize the propulsion-power model designed for rotary-wing UAVs as in the work [77]. The propulsion energy of UAV u is calculated as follows:

$$E_u^P = \left(\eta_1 \sqrt{\left(\sqrt{1 + \frac{v_u^4}{4v_0^4}} - \frac{v_u^2}{2v_0^2} \right)} + \eta_2 \left(1 + \frac{3v_u^2}{v_{tip}^2} \right) + \frac{1}{2} f_0 \varphi r D_u v_u^3 \right) \times t, \quad (5)$$

where η_1 refers to the blade profile power and η_2 refers to the induced power. v_u is the speed of flight of u , v_{tip} indicates the speed of u 's rotor blade, v_0 is the rotor induced velocity, f_0 refers to fuselage drag ratio, r refers to the rotor solidity, φ refers to the air density, D_u is the area of rotor disc, and t is the work time of u .

The communication energy consumption is the total energy consumed by each UAV to collect message, and the energy required to disseminate the updated state to the BS. Then, the communication energy consumed by u is expressed as:

$$E_u^C = P_{u,comm}(T_u + T_{BS,u}), \quad (6)$$

where $P_{u,comm}$ is the communication power of UAV, T_u is the time required to transmit data, and $T_{BS,u}$ is the time required to transmit the UAVs's state to the central station.

Each UAV is assumed to have an initial energy of E_u^I . Hence, the surplus energy of u can be calculated as follows:

$$E_u = E_u^I - E_u^P - E_u^C. \quad (7)$$

3) *Link Survival Time:* It indicates how long the link is expected to remain connected between two nodes. According to the definition in work [78], the method for calculating expected link survival time is outlined below.

For any two nodes i and j , their respective positions and speeds are $p_i = (x_i, y_i)$, $v_i = (v_{x_i}, v_{y_i})$, and $p_j = (x_j, y_j)$, $v_j = (v_{x_j}, v_{y_j})$. The distance between i and j , denoted by $\Delta \mathbf{D}_{i,j}$, is $(x_j - x_i, y_j - y_i)$. Similarly, the speed difference between i and j , denoted by $\Delta \mathbf{V}_{i,j}$, is $(v_{x_j} - v_{x_i}, v_{y_j} - v_{y_i})$. The distance between i and j changes over time t is given as follows:

$$D_{i,j}(t) = \sqrt{[x_j(t) - x_i(t)]^2 + [y_j(t) - y_i(t)]^2}. \quad (8)$$

We assume the communication radius of nodes is R . If $D_{i,j}(t) \leq R$, i and j can communicate at time t . When the initial distance $D_{i,j}(t_0) \leq R$, the communication duration for i and j is denoted as $T_{i,j}$, calculated by the formula:

$$T_{i,j} = \begin{cases} 0 & |\Delta \mathbf{D}_{i,j}| = R, \cos \theta \geq 0 \\ \infty & \Delta \mathbf{V}_{i,j} = 0 \\ \frac{-B + \sqrt{B^2 - 4 \times A \times C}}{2 \times A} & \text{otherwise} \end{cases}, \quad (9)$$

where $A = |\Delta \mathbf{V}_{i,j}|^2$, $B = 2\Delta \mathbf{D}_{i,j} \Delta \mathbf{V}_{i,j}$, $C = |\Delta \mathbf{D}_{i,j}|^2 - R^2$, and $\cos \theta = \cos(\Delta \mathbf{D}_{i,j}, \Delta \mathbf{V}_{i,j})$.

4) *Remaining Distance Cost*: As shown in Fig. 2, suppose message m is transmitted to its destination using i as a relay. The remaining distance cost, denoted by $DC_{i,m}$, is calculated by the formula:

$$DC_{i,m} = d_{i,\alpha} + d_{\alpha,\beta} + d_{\beta,dest}, \quad (10)$$

where $d_{i,\alpha}$ represent the distance from the current position of i to p_α , while $d_{\alpha,\beta}$ describes the distance of from p_α to p_β . Additionally, $d_{\beta,dest}$ denotes the distance from p_β to the destination along the reference trajectory $\mathcal{T}_m$.

5) *Queuing Delay*: It refers to the time a message spends waiting in the buffer of a relay before it can be transmitted. Once a message is inserted into the buffer, it must wait until all preceding messages in the queue have been forwarded. The queuing delay is determined by the total size of the messages ahead in the buffer and the transmission rate.

Let $Q_{i,m}$ represent the queuing delay experienced by message m in the buffer of node i . $M = \{m_1, m_2, \dots, m_k\}$ represents the set of k messages that before m and are still waiting in the queue. Then, the queuing delay can be calculated as follows:

$$Q_{i,m} = \sum_{l=1}^k \frac{size_l}{v_{transmit}}, \quad (11)$$

where $size_l$ is the size (in bits) of message m_l ($m_l \in M$), and $v_{transmit}$ is the message transmission rate (in bits per second).

This delay represents the cumulative time required to transmit all prior messages before message m can be forwarded. Minimizing queuing delay is essential for the timely delivery of emergency messages, especially in resource-constrained post-disaster communication scenarios.

6) *Multi-attribute Decision Making*: In order to measure the transmission capability of a candidate node based on the multiple attributes, the multi-attribute utility function should satisfy the following constraints: (1) All decision attributes and function values should satisfy $[0,1]$. (2) If the value of any decision attribute is equal to 0, the utility function value must be 0. (3) When all decision attributes are equal to 1, the utility function value is equal to 1. Subject to these constraints, we can obtain the multi-attribute utility function as follows:

$$U_i = \prod_{x=1}^n (u_i(x))^{w_x}, \quad (12)$$

where the number of decision attributes in the utility function is n , w_x denotes the weight of x th decision attribute, and $\sum_{x=1}^n w_x = 1$. $u_i(x)$ is the utility value for x th attribute.

Based on the multi-attribute utility function, we can obtain the utility function of candidate node i based on trajectory similarity, surplus energy, link survival time, remaining distance cost and queuing delay as follows:

$$U_i = u_i(t)^{w_t^i} \times u_i(e)^{w_e^i} \times u_i(l)^{w_l^i} \times u_i(d)^{w_d^i} \times u_i(q)^{w_q^i}, \quad (13)$$

where $u_i(t)$, $u_i(e)$, $u_i(l)$, $u_i(d)$, and $u_i(q)$ denote the values of trajectory similarity, surplus energy, link survival time, remaining distance cost and queuing delay, respectively. The weights for these attributes are denoted by w_t^i , w_e^i , w_l^i , w_d^i , and w_q^i , with $w_t^i + w_e^i + w_l^i + w_d^i + w_q^i = 1$. Before calculating the utility function U_i , $u_i(\cdot)$ is normalised to $[0,1]$.

We adopt the [79] adaptive maximum deviation algorithm to determine the weight of each attribute in the multi-attribute decision-making process. This algorithm assigns larger weights to attributes that exhibit greater variability among candidate nodes, thereby improving the discrimination ability of the decision model. The theoretical foundation of this adaptive weighting concept is consistent with classical multi-attribute decision-making theories such as the Analytic Hierarchy Process (AHP) [80], which emphasizes the relative importance of attributes through pairwise comparison, and other objective weighting methods (e.g., entropy-based approaches). Compared with AHP, which requires subjective expert input, the deviation-based algorithm provides a fully data-driven and computationally lightweight mechanism suitable for dynamic and distributed post-disaster UAV-assisted VANET environments. Readers interested in detailed mathematical derivations of the adaptive maximum deviation algorithm are referred to the original work [79].

First, we can obtain the decision matrix $\bar{\Omega}$:

$$\bar{\Omega} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ u_{21} & u_{22} & u_{23} & u_{24} & u_{25} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{c1} & u_{c2} & u_{c3} & u_{c4} & u_{c5} \end{bmatrix}. \quad (14)$$

where u_{ix} represents the normalised value of the x th attribute of i , and c is the number of candidate nodes.

Then the weight value of the x th attribute of node i , denoted by w_x^i , is calculated as:

$$w_x^i = \frac{\sum_{a=1}^c \sum_{b=1, b \neq a}^c |u_{ax} - u_{bx}|}{\sum_{x=1}^5 \sum_{a=1}^c \sum_{b=1, b \neq a}^c |u_{ax} - u_{bx}|}, \quad (15)$$

We now discuss the practical impact of the deviation-based weight defined in Eq. (15) on the multi-attribute decision and related implementation considerations.

Intuitively, the numerator of Eq. (15) measures the total pairwise absolute deviation of attribute x across the current candidate set, while the denominator normalizes this measure

by the sum of deviations of all attributes. Therefore, an attribute that exhibits larger inter-candidate variability will receive a relatively larger weight and hence exert stronger influence on the composite utility U_i in Eq. (13). This behavior is desirable because attributes that discriminate more among alternatives are typically more informative for selecting the best relay under the current local context.

Several practical implications follow:

- **Sensitivity:** increasing w_x amplifies the sensitivity of the utility score to attribute x . In scenarios where one attribute (e.g., link survival time) varies substantially among neighbors, the algorithm will tend to favor nodes that score well on that attribute. Conversely, if an attribute shows little variation across candidates (near-uniform values), it will receive a small weight and have limited impact on selection.
- **Edge cases and numerical stability:** when all candidates have identical values for every attribute, the numerator and denominator in Eq. (15) become zero, which is undefined. In practice we handle this case by adding a small regularization constant ϵ to the denominator. This simple safeguard prevents numerical instability without introducing bias.
- **Operational tuning:** although Eq. (15) is fully data-driven, practical deployments may impose policy constraints (e.g., ensure a minimum weight for link survival time to maintain connection reliability). This can be implemented by clipping:

$$w'_x = \frac{\max(w_x, w_{\min,x})}{\sum_k \max(w_k, w_{\min,k})}, \quad (16)$$

where $w_{\min,x}$ is a small policy-driven floor.

IV. PROPOSED METHOD

The framework of method we proposed is shown as Fig.3. There are four main components: 1) reference trajectory path construction, 2) candidate node set construction, 3) transmission capacity assessment, and 4) dynamic prioritization delivery model.

A. Reference Trajectory Path Construction

Each emergency message is associated with a reference trajectory that guides its forwarding behavior. When message m is generated by a vehicle, the system computes a reference trajectory $\mathcal{T}_m$ from the source to destination. $\mathcal{T}_m$ is computed using Dijkstra algorithm based on pre-stored digital maps. It is then embedded in the header of m to guide subsequent forwarding decisions.

In dynamic post-disaster environments, the network topology may change frequently because of node mobility, link breakage, or temporary road blockages. To maintain routing adaptability, DPUVR periodically updates the reference trajectories of existing messages. The update process is executed every 100 s, which provides a practical balance between routing responsiveness and computational overhead according to empirical observations in our simulations. In addition, DPUVR performs event-driven trajectory updates whenever

significant topology variations occur. The triggering conditions include:

- A road segment becomes unavailable due to failure.
- A destination node (e.g., a rescue unit) changes its position or status.

When any of these conditions are detected, the affected messages reconstruct their reference trajectories based on the latest connectivity information. The overall process is presented in Algorithm 1.

Algorithm 1: Reference Trajectory Path Construction

Input: M_{new} : Message set for which the reference trajectory path has not been constructed.
 M_{old} : Message set requiring trajectory update.
 Δt : Periodic update interval (default = 100 s).

```

1 for message  $m \in M_{new}$  do
2   Get the destination  $dest$  from message header;
3    $map \leftarrow$  Get the connection graph of road;
4    $\mathcal{T}_m \leftarrow$  Dijkstra( $dest$ ,  $map$ );
5   Sets  $\mathcal{T}_m$  to the message  $m$  header;
6 for message  $m \in M_{old}$  do
7   if ( $current\ time\ mod\ \Delta t = 0$ ) or ( $trigger\ event\ occurs$ ) then
8     Get the destination  $dest$  from message header;
9      $map \leftarrow$  Update connection graph using latest network state;
10     $\mathcal{T}_m \leftarrow$  Dijkstra( $dest$ ,  $map$ );
11    Replace the trajectory field in the header of message  $m$  with  $\mathcal{T}_m$ ;

```

B. Candidate Node Set Construction

By constructing a candidate node set, nodes that are not suitable for forwarding messages are filtered. As shown in Fig. 4, the angle between $\mathcal{W}_i$ and $p_{\beta+2}$ is represented as θ_i^m , calculated by Eq. (17). If $\theta_i^m < \frac{\pi}{2}$, i will be put into the candidate set CR . More details of constructing the candidate set CR are summarized in Algorithm 2.

$$\theta_i^m = \frac{\theta_1 + \theta_2 + \theta_3}{3}, \quad (17)$$

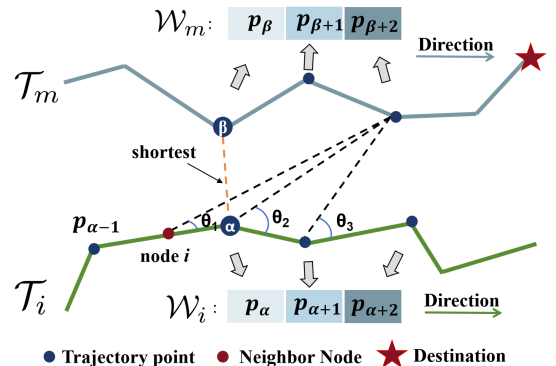


Fig. 4: An example of candidate node evaluation.

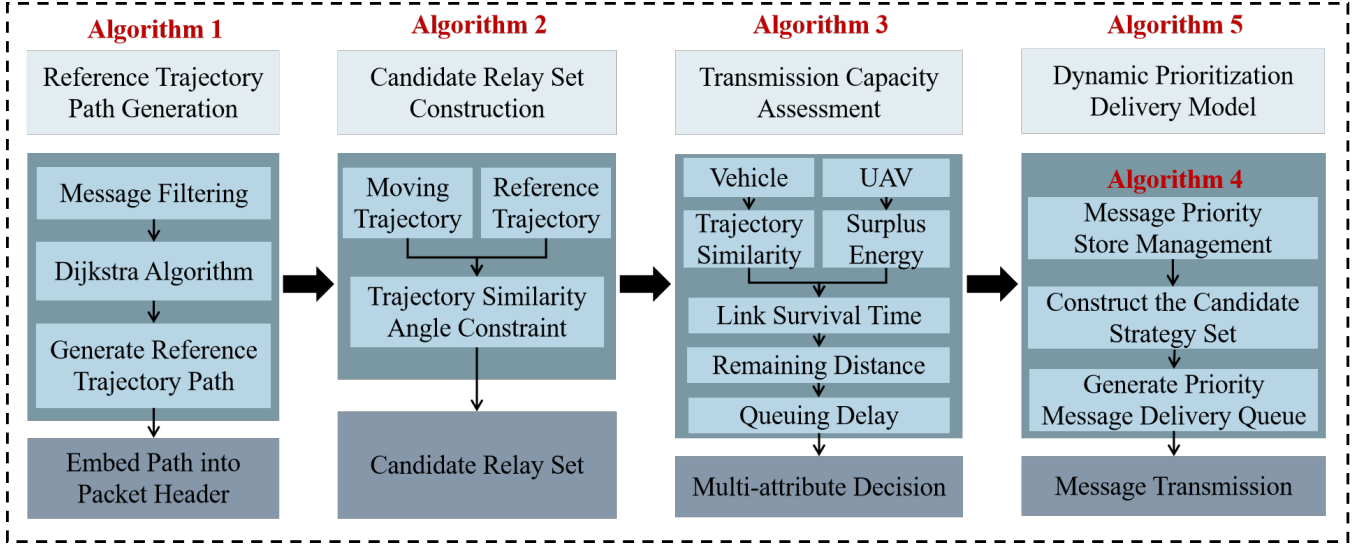


Fig. 3: The framework of proposed method.

Algorithm 2: Candidate Node Set Construction

Input: $\mathcal{T}_m$: The reference trajectory path of m .
 NR : The neighbour node set.

Output: CR : Candidate node set.

```

1 for neighbour node  $i \in NR$  do
2   Sliding Window  $\mathcal{W}_i \leftarrow$  Get the three sequential
   path segment in  $i$ 's moving trajectory ;
3   Sliding Window  $\mathcal{W}_m \leftarrow$  Get the three sequential
   path segment in  $\mathcal{T}_m$  ;
4    $\theta_i^m \leftarrow$  Calculate trajectory similarity angle using
   Eq. (17) ;
5   if  $\theta_i^m < \frac{\pi}{2}$  then
6     Add  $i$  to  $CR$ ;
7 return  $CR$ ;
  
```

C. Transmission Capacity Assessment

The status of candidate nodes is detected to evaluate the capability of message delivery. We score the local status of nodes from the candidate set, and judge the potential capabilities for transmitting messages. Notably, the score consists of trajectory similarity TS_{m_i} , surplus energy SE_{m_i} , link survival time LT_{m_i} , remaining distance cost RD_{m_i} and queuing delay QD_{m_i} . In order to store calculation results, some sets (TS_m , SE_m , LT_m , RD_m , and QD_m) are created, respectively. The scores in each set are normalized according to the Min-Max normalization. Then, calculate the utility function as Section III-C6 described. Algorithm 3 describes the process in detail and sorts the candidate nodes according to the utility function value.

DPUVR inevitably introduces additional computational overhead due to its multi-attribute relay evaluation process. For each candidate relay, the protocol jointly computes metrics such as trajectory similarity, residual energy, link survival time, residual distance cost, and queue delay. The overall computational complexity can be approximated as $\mathcal{O}(N \times A)$,

where N denotes the number of neighboring nodes and A the number of evaluated attributes. This design leads to a moderate increase in processing time compared with single-metric routing schemes. However, this additional cost is justified by the substantial improvement in routing adaptability and message delivery efficiency observed in dynamic post-disaster scenarios.

This trade-off is considered acceptable for UAV-assisted VANET scenarios, where computation resources are relatively sufficient on UAVs and roadside units. Furthermore, the computational cost can be further reduced by applying candidate pre-selection, where only a subset of promising neighbors are evaluated in detail, and incremental updates, where attribute values are recalculated only when significant state changes occur. Therefore, DPUVR achieves a balanced compromise between computational complexity and routing adaptability, ensuring both efficiency and scalability in real-world deployments.

D. Dynamic Prioritization Delivery Model

In this paper, each node maintains a local message queue, with messages being divided into emergency and ordinary. We define three priority levels: 0 (highest), 1 (medium) and 2 (lowest), corresponding to three storage queues: Q_0 , Q_1 and Q_2 . High-priority messages may be placed in lower-priority queues if necessary, but lower-priority messages are strictly restricted from entering higher-priority queues. When a message arrives at a node, it is put into a message queue according to its priority, and dynamic adjustment is performed to handle emergency messages.

1) *Message Priority Storage*: We consider three possible cases, as shown in Fig. 5.

- Regular case: Messages arrive at the forwarding node, and the message queues are not filled up.
- Emergency case: The emergency queue is full, the other two queues are not yet full.

Algorithm 3: Transmission Capacity Assessment

Input: CR : The candidate node set.
 m : Message.

Output: CR_{sorted} : CR sorted by utility function.

- 1 $TS_m \leftarrow$ Create a set for trajectory similarity;
- 2 $SE_m \leftarrow$ Create a set for surplus energy;
- 3 $LT_m \leftarrow$ Create a set for link survival time;
- 4 $RD_m \leftarrow$ Create a set for remaining distance cost;
- 5 $QD_m \leftarrow$ Create a set for queuing delay;
- 6 $U \leftarrow$ Create a set for the utility function;
- 7 **for** $i \in CR$ **do**
- 8 Calculate TS_{m_i} using Eq. (4);
- 9 $TS_m \leftarrow TS_m \cup TS_{m_i}$;
- 10 Calculate SE_{m_i} using Eq. (7);
- 11 $SE_m \leftarrow SE_m \cup SE_{m_i}$;
- 12 Calculate LT_{m_i} using Eq. (9);
- 13 $LT_m \leftarrow LT_m \cup LT_{m_i}$;
- 14 Calculate RD_{m_i} using Eq. (10);
- 15 $RD_m \leftarrow RD_m \cup RD_{m_i}$;
- 16 Calculate QD_{m_i} using Eq. (11);
- 17 $QD_m \leftarrow QD_m \cup QD_{m_i}$;
- 18 Normalize TS_m , SE_m , LT_m , RD_m and QD_m ;
- 19 **for** $i \in CR$ **do**
- 20 Calculate $w_1^i, w_2^i, w_3^i, w_4^i$, and w_5^i using Eq.(15)
- 21 $U \leftarrow U \cup U_i =$
- 22 $TS_{m_i}^{w_1^i} \times SE_{m_i}^{w_2^i} \times LT_{m_i}^{w_3^i} \times RD_{m_i}^{w_4^i} \times QD_{m_i}^{w_5^i}$
- 23 $CR_{sorted} \leftarrow$ Sort CR in descending order of U ;
- 24 **return** CR_{sorted} ;

- Full case: All three queues are full, and emergency messages are waiting.

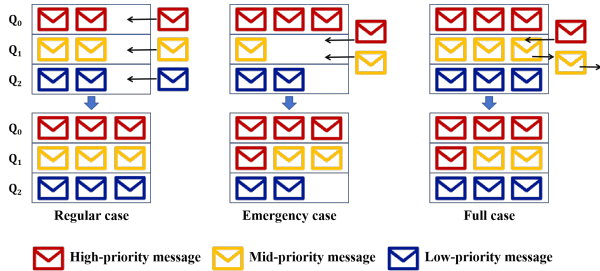


Fig. 5: Three cases of message priority storage.

Algorithm 4 describes how the current node processes a message upon reception. It determines the message's priority m_p based on the priority flag in the message header. If $m_p = 0$, then Q_0 is the highest-priority queue to store the message. If Q_0 is full, the node tries Q_1 and then Q_2 . If all queues are full, the message replaces the one with the lowest survivability score in $Q_1 \cup Q_2$ (Lines 2-11). If $m_p = 1$, then Q_1 is the primary queue for the message. If Q_1 is full, the message is inserted into Q_2 if space permits. Otherwise, it replaces the lowest-survivability message in Q_2 (Lines 12-19). If $m_p = 2$, then Q_2 is the only eligible queue. If Q_2 is full, the message is dropped (Lines 20-24).

Algorithm 4: Message Priority Storage

Input: m : Message needs to be put into the buffer.
 Q_0 : High priority queue in the buffer.
 Q_1 : Mid priority queue in the buffer.
 Q_2 : Low priority queue in the buffer.

- 1 m_p : Priority of message m ;
- 2 **if** $m_p = 0$ **then**
- 3 **if** $\neg Q_0.Full()$ **then**
- 4 $Q_0.Put(m)$;
- 5 **else if** $\neg Q_1.Full()$ **then**
- 6 $Q_1.Put(m)$;
- 7 **else if** $\neg Q_2.Full()$ **then**
- 8 $Q_2.Put(m)$;
- 9 **else**
- 10 $m_d \leftarrow \text{argmin}_{x \in Q_1 \cup Q_2} (S_x)$;
- 11 Replace m_d with m in its queue;
- 12 **else if** $m_p = 1$ **then**
- 13 **if** $\neg Q_1.Full()$ **then**
- 14 $Q_1.Put(m)$;
- 15 **else if** $\neg Q_2.Full()$ **then**
- 16 $Q_2.Put(m)$;
- 17 **else**
- 18 $m_d \leftarrow \text{argmin}_{x \in Q_2} (S_x)$;
- 19 Replace m_d with m in Q_2 ;
- 20 **else**
- 21 **if** $\neg Q_2.Full()$ **then**
- 22 $Q_2.Put(m)$;
- 23 **else**
- 24 Drop m ;

The survivability score of message m , denoted by S_m , reflects the likelihood that the message will reach its destination before expiration. It is calculated as:

$$S_m = \frac{TTL}{T_{ETA}}, \quad (18)$$

where TTL is the TTL of m , $T_{ETA} = \frac{d_{src-dst}}{B_{transmit}}$ is the estimated time to destination, $d_{src-dst}$ is the distance from the current node to the destination and $B_{transmit}$ is the transmission bandwidth. A higher S_m indicates a greater chance that m can be delivered before it expires, while a lower value suggests a higher risk of message loss due to delay.

2) *Dynamic Prioritization Delivery Model*: The dynamic prioritization delivery model of node i , denoted as $\Phi_i = (N_i, M_i, A_i, U)$:

- N_i : the set of neighbor nodes, $N_i = \{j_1, j_2, \dots, j_{|N_i|}\}$.
- M_i : the set of prioritized messages stored at node i , denoted as $M_i = \{m_1, m_2, \dots, m_{|M_i|}\}$.
- A_i : the set of candidate actions, where each action is a pair consisting of a message and a neighbor node selected as its relay. Formally,

$$A_i = \{(m, j) \mid m \in M_i, j \in N_i\}. \quad (19)$$

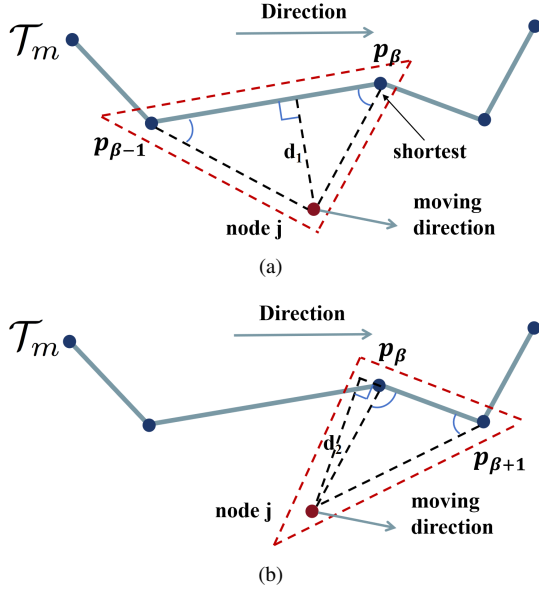


Fig. 6: Evaluation of related relationship.

- U : the utility function defined in Eq. (13), which evaluates the effectiveness of each action in A_i .

Node i selects the optimal action $(m^*, j^*) \in A_i$ that maximizes the utility function. The specific selection steps are shown in Algorithm 5.

In this model, node i maintains a buffer of undelivered messages and continuously evaluates delivery opportunities based on its local context. For each message $m \in M_i$ and each neighbor $j \in N_i$, the node determines whether j is a suitable candidate node for m . This evaluation considers both the message's priority and the trajectory relevance through the relation $(m \rightsquigarrow j)$.

Definition (Trajectory Relevance): Defined as $(m \rightsquigarrow j)$, it represents that either node j is moving along the reference trajectory of m , or geographically close to it. As shown in Fig.6, DPUVR identifies the reference trajectory point p_{β} that is closest to the current node j , as well as its adjacent points $p_{\beta-1}$ and $p_{\beta+1}$. Two triangles are then formed: $\triangle jp_{\beta-1}p_{\beta}$ and $\triangle jp_{\beta}p_{\beta+1}$. Let d_1 and d_2 denote the perpendicular distances from j to the base edges $p_{\beta-1}p_{\beta}$ and $p_{\beta}p_{\beta+1}$, respectively. If either $d_1 < R$ or $d_2 < R$ (where R is the communication radius of j), then m is considered to be related to j , i.e., $(m \rightsquigarrow j)$.

All candidate $(message, node)$ pairs are organized into three priority queues: *highQueue*, *midQueue*, and *lowQueue*, based on the message's priority level and whether the neighbor node is related to the message's reference trajectory. High-priority messages that match the trajectory are placed into *highQueue*, while other messages are allocated to *midQueue* or *lowQueue* accordingly.

Algorithm 5: Dynamic Prioritization Delivery Model

Input: NR_i : Neighbor node set of node i .

M_i : Message set in buffer of i .

Output: $Q_{Priority}$: Prioritized delivery queue of i .

```

1  $A_i \leftarrow$  The set of candidate strategies for  $i$ ;
2  $highQueue \leftarrow$  High storage queues for  $A_i$ ;
3  $midQueue \leftarrow$  Medium storage queues for  $A_i$ ;
4  $lowQueue \leftarrow$  Low storage queues for  $A_i$ ;
5 for message  $m \in M_i$  do
6   for neighbor  $j \in NR_i$  do
7     if  $j.isCandidate(m)$  then
8        $A_i \leftarrow A_i \cup \{(m, j)\}$ ;
9 for  $(m, j) \in A_i$  do
10   $p \leftarrow m.priority$ ;
11  if  $p = 0$  (High priority) then
12    if  $m \rightsquigarrow j$  (Matching condition) then
13       $highQueue.add(m, j)$ ;
14    else
15       $midQueue.add(m, j)$ ;
16  else
17    if  $p = 1$  (Medium priority) then
18      if  $m \rightsquigarrow j$  then
19         $midQueue.add(m, j)$ ;
20      else
21         $lowQueue.add(m, j)$ ;
22    else
23       $lowQueue.add(m, j)$ ;
24 for  $queue \in \{highQueue, midQueue, lowQueue\}$  do
25    $queue.sortBy(U)$ ;
26    $Q_{Priority}.extend(queue)$ ;
27 return  $Q_{Priority}$ ;

```

Each queue is then sorted based on the utility function defined in Eq. (13), which reflects the overall delivery benefit considering factors such as buffer occupancy, delay and trajectory similarity. Finally, all queues are concatenated in priority order to form the final transmission sequence $Q_{Priority}$.

This process ensures that high-priority and reference trajectory-relevant messages are delivered first, while also maintaining fairness and efficiency in multi-hop emergency message dissemination. Fig.7 is an example of this process.

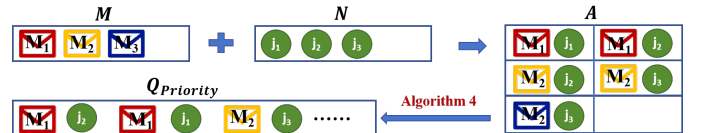


Fig. 7: An example of dynamic prioritization delivery.

V. PERFORMANCE EVALUATION

This section first describes the simulator for simulations and configuration. Then, extensive simulations are conducted to illustrate the performance of proposed method.

A. Simulation Setup

The evaluations are based on Opportunistic Network Environment (ONE) [81]. As shown in Fig. 8, the simulation area ($4.5 \text{ km} \times 3.4 \text{ km}$) is constructed from the real road topology data of Helsinki city. While obtaining the precise topology of an actual disaster zone is impractical due to safety and accessibility constraints, the Helsinki dataset provides a well-documented and complex urban layout that serves as a credible reference for evaluating communication performance under disaster conditions. Therefore, in this work, the Helsinki city topology is adopted as a representative urban-like post-disaster environment rather than an ordinary city traffic scene.

To emulate disaster impacts such as infrastructure collapse and traffic blockage, 20%~30% of road segments are randomly disabled, forming several disconnected sub-areas (Disaster Area 1~Disaster Area 3). Each area is served by one CUAV acting as a temporary aerial base station, while mobile UAVs patrol along predefined flight routes to bridge communication gaps between isolated areas.

We consider the simulation time set to 15,000s, and the total number of vehicles and UAVs is set to 120. Additionally, vehicles and UAVs are randomly deployed along the road on the digital map. Each vehicle follows the shortest path generated by the Dijkstra algorithm [82], and its moving speed is randomly chosen from $[18 \sim 54] \text{ km/h}$. According to the configuration in [39], the transmission range and bandwidth ($B_{transmit}$) of nodes are set to 100m and 500kbit/s, respectively. The buffer size of nodes is set to 1GB.

To reflect real rescue operations, three categories of emergency messages are defined:

- Priority 0 (Medical-rescue messages): Life-critical alerts that require the shortest delivery delay.
- Priority 1 (Supply-distribution coordination): Time sensitive logistics messages.
- Priority 2 (Situation reports): Non-critical updates such as environmental sensing.

These categories correspond to the priority-based forwarding model in Section IV-D, allowing evaluation of DPUVR's effectiveness in heterogeneous rescue scenarios.

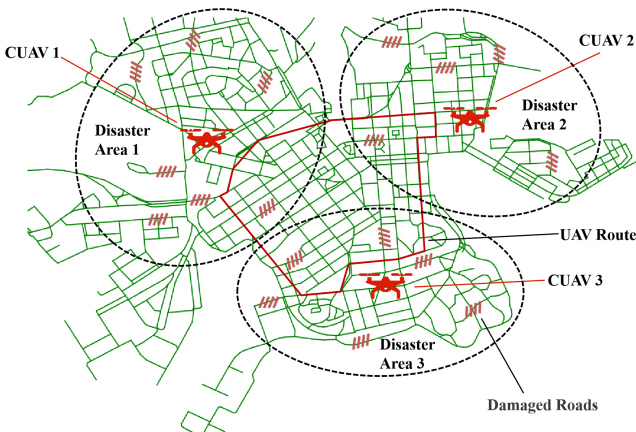


Fig. 8: Evaluation scenario.

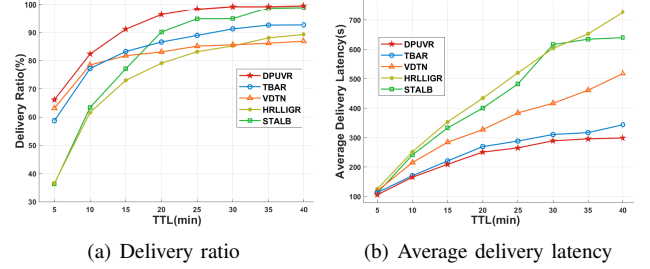


Fig. 9: The impact of TTL.

For performance evaluation, we compare the following routing protocols in various scenarios:

- **VDTN** [10]: A routing protocol for UAV-Assisted vehicular delay tolerant networks, which is based on the joint consideration of both the persistent connection time and message encounter probability.
- **HRLIGR** [32]: A high-reliability low-latency intelligent geographic routing protocol. The main mechanisms of HRLIGR include a reliable greedy forwarding method based on the evaluation of link stability metrics, a low-latency area prediction forwarding method for routing voids, and an intelligent routing update mechanism to reduce routing overhead.
- **STALB** [39]: A spatio-temporal domain autonomous load balancing routing protocol. STALB considers the spatio-temporal indicators and utilizes the weighted method to select the next-hop device in SDN.
- **TBAR** [40]: A trajectory-based anycast routing scheme. TBAR considers link communication quality, trajectory similarity and distance cost.
- **No-pri**: A simplified variant of the proposed method without the dynamic prioritization delivery model.

The performance of our work is evaluated based on the following key parameters:

- **Delivery Ratio**: The ratio of messages successfully delivered to the total number of messages generated.
- **Overhead Ratio**: The ratio between the number of relayed messages and the number of delivered messages.
- **Average Delivery Latency**: The average time for a message to be forwarded from source to destination.
- **Average Buffer Time**: The average time a message is held in a network relay device after it has been received.

B. Parameter Sensitivity Analysis

To further verify the robustness of DPUVR under different simulation conditions, we analyze the effect of several key parameters on delivery ratio and latency, including TTL, node density, and transmission range. Each parameter is varied individually within a reasonable range while keeping others fixed at their default values. Aiming to evaluate whether DPUVR maintains stable performance trends under moderate parameter deviations and to provide insights for real-world parameter configuration.

1) *The Impact of TTL*: Firstly, we consider the impact of different TTL values on the delivery ratio of DPUVR and benchmark methods (VDTN, HRLIGR, STALB and TBAR). We change the TTL from 5 minutes to 40 minutes. Fig.9(a) shows that the delivery ratio of each method maintains stability with the increase of TTL. In case of lower TTL values, a significant number of messages expire before reaching their destinations, resulting in low delivery ratios for all methods. In particular, methods HRLIGR and STALB achieve only 36.54% and 36.31% delivery ratio when the TTL is 5 minutes. With increasing TTL, the delivery ratios of all methods show a gradual improvement. Obviously, DPUVR serves as the upper bound among benchmark methods. Its performance becomes near-optimal when TTL is greater than 20 minutes.

In addition, we analyze the effect of TTL on the average delivery latency, as shown in Fig. 9(b). When the TTL increases from 5 to 40 minutes, the average latency of all schemes grows. This is because a larger TTL allows packets to stay longer in the network, thereby increasing the probability of delayed deliveries. However, DPUVR consistently achieves the lowest latency among all methods, owing to its dynamic priority scheduling and multi-attribute relay selection that avoid congested or unstable links. In contrast, HRLIGR and STALB suffer from high delay accumulation due to their static relay strategies and longer queuing times, while VDTN experiences extra buffering delay from its store-carry-forward mechanism. When TTL exceeds 30 minutes, the latency gap between DPUVR and other methods becomes more pronounced—DPUVR reduces the end-to-end delay by approximately 50% compared with STALB and more than 60% compared with HRLIGR. This demonstrates that DPUVR can maintain a better balance between delivery success and timeliness under various TTL constraints, which is critical for time-sensitive emergency message dissemination.

From the above results, an appropriate TTL configuration is crucial for balancing delivery reliability and timeliness in real deployments. In practical post-disaster scenarios, TTL can be adaptively tuned according to message priority and network dynamics. For instance, high-priority emergency alerts (e.g., medical or life-critical messages) can adopt a short TTL of 10~15 minutes to guarantee low latency, while non-critical situation reports can use a longer TTL of 25~30 minutes to improve delivery completeness. Moreover, the TTL threshold can be dynamically adjusted based on observed network connectivity and UAV flight density—a higher node density or longer UAV endurance allows for shorter TTLs without compromising delivery rate. These parameter-tuning strategies enable DPUVR to be flexibly configured for diverse disaster environments, ensuring efficient message dissemination under realistic operational constraints.

2) *The Impact of Node Density*: We next evaluate the influence of node density on system performance by varying the number of nodes from 40 to 200. As illustrated in Fig. 10(a), the delivery ratio of all protocols improves as node density increases because a denser network offers more contact opportunities and shorter inter-node distances. DPUVR consistently achieves higher delivery ratios than other schemes owing to its adaptive relay selection that prioritizes stable and

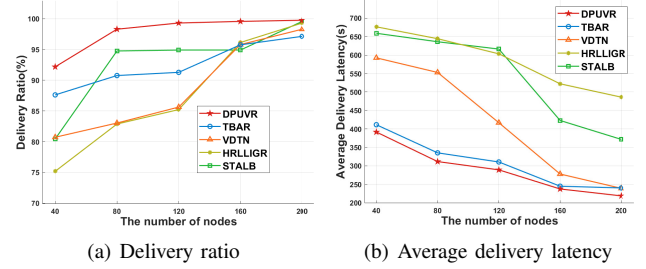


Fig. 10: The impact of node density.

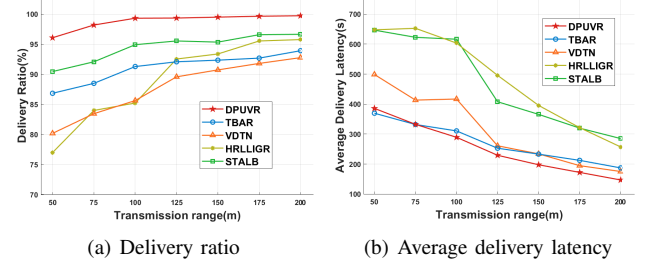


Fig. 11: The impact of transmission range.

energy-efficient links.

Fig. 10(b) shows that latency slightly decreases with increasing density since packets can reach their destinations through fewer hops and shorter detours. However, when the density becomes very high, contention and queuing effects cause a slight rise in latency for TBAR schemes. DPUVR mitigates this problem through its dynamic priority scheduling mechanism, which effectively suppresses low-importance packets under congestion.

Overall, these results demonstrate that DPUVR scales well with node density and maintains stable performance even in sparse or congested environments, which is essential for post-disaster scenarios with fluctuating traffic conditions.

3) *The Impact of Transmission Range*: We further analyze the impact of transmission range R_{tx} , which directly affects connectivity and link lifetime. Fig. 11(a) shows that increasing R_{tx} from 50m to 200m significantly enhances delivery ratio for all schemes, as nodes can maintain contact for longer durations and establish more stable routes. DPUVR maintains the highest delivery ratio because its multi-attribute relay selection leverages the extended connectivity to select links with both high stability and short expected delay.

In terms of average delivery latency, as illustrated in Fig. 11(b), latency gradually decreases with increasing transmission range. A larger R_{tx} allows messages to traverse the network with fewer hops and shorter waiting times, thus reducing the end-to-end delay across all schemes. However, the reduction trend differs among protocols: HRLIGR and STALB exhibit relatively high latency under small ranges due to frequent link breakages and buffer queuing, while DPUVR consistently achieves the lowest latency. This is because its dynamic priority-based scheduling and adaptive relay selection avoid congested relays and exploit stable high-quality links when the range expands. Even when R_{tx} exceeds 150m,

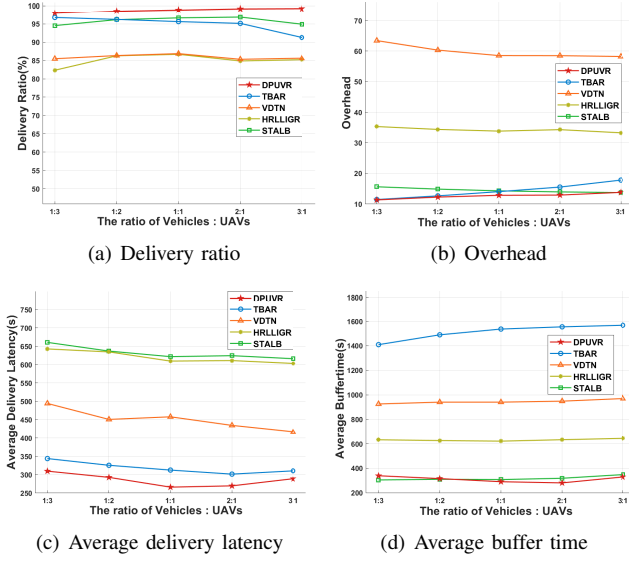


Fig. 12: The impact of ratio of node numbers.

DPUVR continues to maintain a latency advantage, showing that its decision model remains effective under both sparse and dense connectivity conditions.

These results confirm that DPUVR not only improves delivery reliability but also optimizes message timeliness under varying communication ranges, which is crucial for emergency message dissemination in post-disaster environments.

C. The Impact of Ratio of Node Numbers

In this section, the ratio of two types nodes (i vehicles and j UAVs) was denoted as $\mathcal{V}_{i,j}$. For example, there were 90 vehicles and 30 UAVs, and their ratio was $\mathcal{V}_{90,30}$. Then, we changed $\mathcal{V}_{i,j}$ from $\mathcal{V}_{90,30}$ to $\mathcal{V}_{60,60}$, and the message generation interval was 5s, keeping the total number of vehicles and UAVs were 120. Without loss of generality, the other constraints didn't change.

1) *Delivery ratio*: As shown in Fig. 12(a), the delivery ratio of DPUVR is higher than that of the benchmark method in all network scenarios, maintaining a high delivery ratio. Trajectory path-based DPUVR enables messages to be forwarded to their destinations along a reference trajectory. In addition, the reference trajectory reconstruction mechanism in DPUVR periodically updates the reference trajectory path that each message takes to reach its destination. The message is then transmitted along the new reference trajectory path. Thus, DPUVR ensures a high delivery ratio. The delivery ratio is lowest at $\mathcal{V}_{30,90}$ with 98.03% and highest at $\mathcal{V}_{90,30}$ with 99.32%. This also shows that a smaller number of UAVs can provide better communication services. Therefore, in the subsequent simulations, we utilize the node ratio of $\mathcal{V}_{90,30}$.

2) *Overhead*: Fig. 12(b) shows how the overhead changes with different ratios of two types nodes $\mathcal{V}_{i,j}$. In DPUVR, the number of message copies can be controlled by the message forwarding strategy, which naturally causes a limited number of message copies in the network. Similar to DPUVR, TBAR and STALB also control the number of message copies in the

network. DPUVR, TBAR, and STALB maintain a very low overhead, with the mean values of 12.61, 14.29, and 14.48, respectively. While VDTN and HRLIGR do not control the number of message copies in the network. Thus, the mean overhead is 59.81 and 34.22, respectively, which is higher than DPUVR by 374.31% and 171.37%. In DPUVR, the overhead increases slightly with the increase of V:U (Vehicle to UAV ratio). This is because the limited number of links constrains the forwarding of messages when the vehicles are sparse, resulting in some messages in the buffer waiting to be forwarded. The number of relayed messages is low and hence the overhead ratio is low.

3) *Average delivery latency*: Fig. 12(c) demonstrates the variation of average delivery latency in different $\mathcal{V}_{i,j}$. Attentively, the average delivery latency of DPUVR is the lowest in all scenarios. In most network scenarios, the average delivery latency of DPUVR was lower than 300s, and the lowest value was 289.18s ($\mathcal{V}_{60,60}$). To reduce the average delivery latency, DPUVR controls the direction of message forwarding and provides a reference trajectory update mechanism, which ensures a limited number of message hops. The above method enables messages to be forwarded to their destinations as soon as possible. The average delivery latency of DPUVR and TBAR stabilized around 285.49s and 318.89s, respectively. Specially, the average delivery latencies of VDTN, HRLIGR and STALB are 57.91%, 117.31%, and 121.41% per cent higher than that of DPUVR.

4) *Average buffer time*: Fig. 12(d) illustrates the impact of the ratio of two types of nodes $\mathcal{V}_{i,j}$ on the average buffer time. Since the total number of nodes in the network is constant in each simulation scenario, there is little variation in the average buffer time for each routing policy. The range of buffer times for DPUVR, VDTN, HRLIGR, and STALB is within 1 minute, while TBAR exhibits the largest variation, though its range does not exceed 3 minutes. The average buffer time of DPUVR and STALB stabilized around 311.23s and 318.15s, respectively. In particular, the average buffer time of DPUVR is 67.06%, 50.75%, and 79.42% lower than that of VDTN, HRLIGR and TBAR.

D. The Impact of Total Number of Messages

In this section, we consider the influence of different message densities on the efficiency of DPUVR and benchmark methods. We set the message density by controlling the message generation interval (from 1s to 10s), meaning the message volume ranges from 1.5k to 15k. The ratio of two types nodes $\mathcal{V}_{i,j}$ is fixed at $\mathcal{V}_{90,30}$, while other parameters are unchangeable.

1) *Delivery ratio*: Fig. 13(a) shows the delivery ratio versus the number of messages. It can be observed that for three message forwarding methods (DPUVR, TBAR, and STALB), the delivery ratio of each method stabilises with the increase in the number of messages. Obviously, the proposed DPUVR was the upper limit of all benchmark methods, which was approximately the optimal solution. In all simulations, DPUVR achieved the highest delivery ratio, with nearly 100% delivery ratio when the message generation interval exceeds

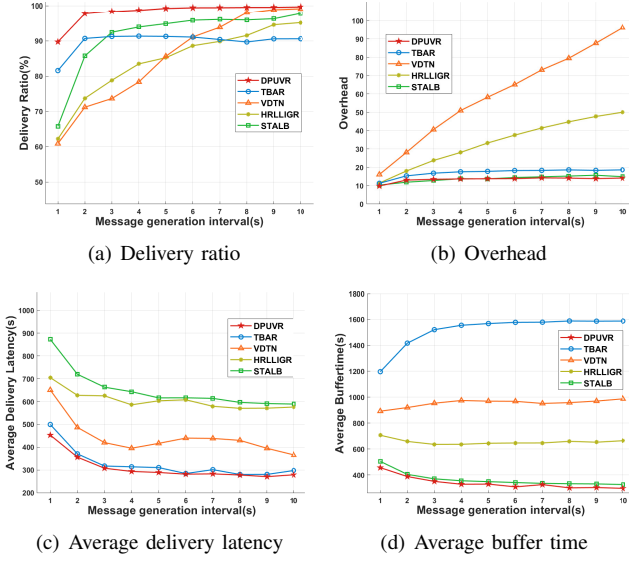


Fig. 13: The impact of total number of messages.

2 seconds. Although the delivery ratio of DPUVR declined in the presence of high message density, it was consistently above 90%. It is clear that DPUVR has a significantly higher delivery ratio than the other benchmark methods with high message density.

2) *Overhead*: Fig. 13(b) shows how the overhead changes with different numbers of messages. The overheads of DPUVR, TBAR, and STALB are basically stable, with mean values of 13.41, 17.10 and 13.76, respectively. In these methods, the number of message copies can be controlled by the message forwarding strategy, which naturally causes a limited number of message copies in the network. Furthermore, the number of message copies increases linearly in the core network because each vehicle merely forwards one message copy at a slot. The overheads of VDTN and HRLLIGR increase significantly because it does not limit the number of message copies, reaching a maximum overhead of 96.26 and 50.07, respectively. The overhead ratio of VDTN and HRLLIGR was even higher than STALB by 617.82% and 273.38%, respectively.

3) *Average delivery latency*: Fig. 13(c) illustrates how the average delivery latency of the four methods varies with message density. It demonstrated that the average delivery latency of five methods all increases as the number of messages increases. Attentively, the increase of the average delivery latency in DPUVR and TBAR was not obvious, while that in the other three methods had increased significantly. The average delivery latency of DPUVR and TBAR stabilized around 309.25s and 325.71s, respectively. Notably, DPUVR has the lowest average latency, at least 30.33%, 48.90% and 52.59% lower than VDTN, HRLLIGR and STALB, respectively. Highlighting its superiority in delivering emergency messages.

4) *Average buffer time*: Fig. 13(d) illustrates the impact of number of messages on average buffer time. It can be found that DPUVR had the lowest average buffer time, i.e., 338.88s, while that in TBAR, VDTN, HRLLIGR and STALB

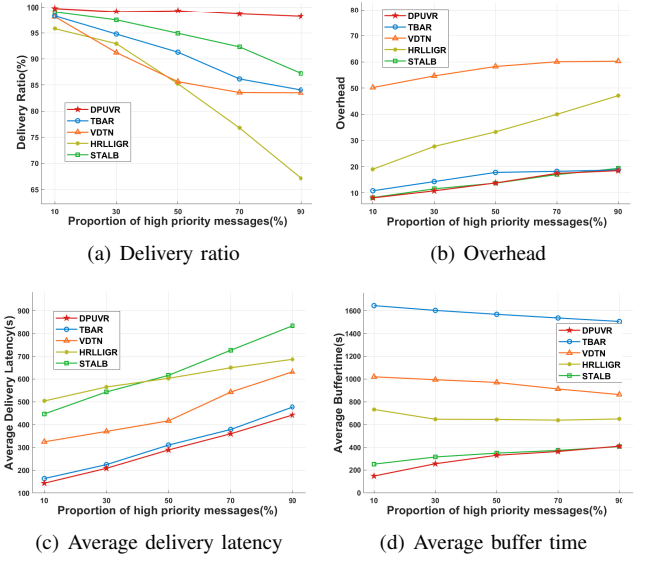


Fig. 14: The impact of different high priority message ratios.

were 1517.52s, 954.66s, 655.45s and 365.16s, respectively. Compared with TBAR, VDTN, HRLLIGR and STALB in the dense network scenario (the message generation interval is 1s), the average buffer time of DPUVR is lower than 61.85%, 48.82%, 35.36%, and 9.55%, respectively. The main reason for the lower average buffer time in DPUVR is the ability to quickly update the list of delivered messages in the network. In detail, the messages stored in this list are used to calculate the buffer time, defined as the waiting time between reception and deletion. The speed at which the list of delivered messages is updated determines the time horizon, and faster updates help reduce buffer time.

E. The Impact of High Priority Message Ratios

In this section, we consider the influence of different high message ratios on the efficiency of DPUVR and benchmark methods. We set the high message ratio from 10% to 90%, while other parameters are unchangeable.

1) *Delivery ratio*: As shown in Fig. 14(a), the delivery ratio decreases as the high-priority message ratio rises. In all cases, DPUVR consistently achieves the highest delivery ratio. This is due to DPUVR's dynamic prioritisation delivery mechanism: When a message arrives at a node, it is sorted according to message priority and survivability. When a message is transmitted, the most appropriate relay is selected based on the priority. This ensures the successful transmission of high-priority messages. When high-priority message is 90%, the delivery ratio of DPUVR is 14.13%, 14.66%, 31.02% and 10.95% higher than TBAR, VDTN, HRLLIGR and STALB, respectively.

2) *Overhead*: Fig. 14(b) shows how the overhead changes with different ratios of high-priority messages. As the ratio of high-priority messages increases, the overhead rises. This is because the increase in high-priority messages leads to more forwarding attempts and buffer usage, which in turn triggers greater overhead. DPUVR, TBAR and STALB have

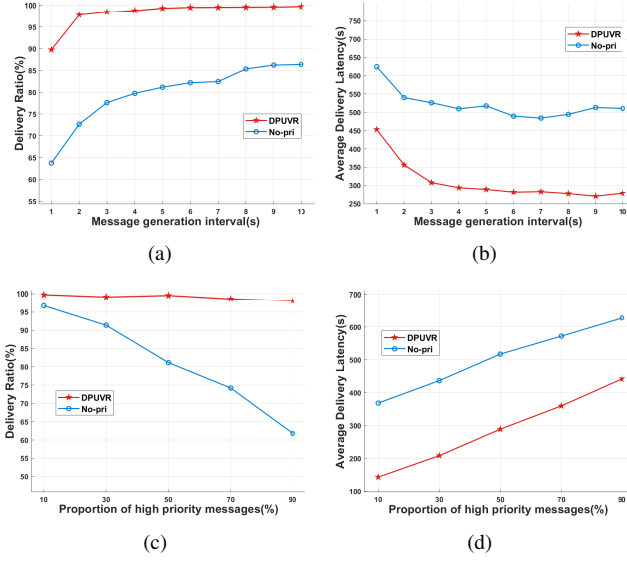


Fig. 15: (a) Delivery ratio versus the number of messages. (b) Average delivery latency versus the number of messages. (c) Delivery ratio versus the high-priority message ratios. (d) Average delivery latency versus the high-priority message ratios.

low overhead, with mean values of 13.71, 15.97, and 13.95, respectively. This is because the above methods control the number of message copies throughout the network at routing time. However, the overheads of VDTN and HRLIGR are 42.97 and 19.71 higher than DPUVR's.

3) *Average delivery latency*: Fig. 14(c) demonstrates that average delivery latency increases as the ratio of high-priority messages rises. Obviously, as the ratio of high-priority messages increases, more copies of messages exist in the network, which greatly increases the buffer time of the message. Thus, the delivery latency is increased. It is worth noting that the average delivery latency of DPUVR and TBAR stabilized around 288.77s and 311.12s, respectively. Specially, the average delivery latency of DPUVR is at least 36.87%, 52.01%, and 54.41% lower than that of VDTN, HRLIGR and STALB, respectively.

4) *Average buffer time*: Fig. 14(d) illustrates the impact of ratio of high priority messages on average buffer time. It can be found that DPUVR had the lowest average buffer time, i.e., 301.06s. When messages are forwarded to their destinations, their IDs will be recorded in a table in the network relay device. The table will be obtained by nodes when the connections are established. Then, the nodes will delete the messages which in the exchanged table. Therefore, the lower average delivery latency DPUVR has, the lower average buffer time DPUVR obtains. The average buffer time of DPUVR is 80.84%, 68.36%, 54.53%, and 11.11% lower than that of TBAR, VDTN, HRLIGR and STALB, respectively.

F. Impact of Prioritization Strategies

In this section, we consider the impact on DPUVR of having or not having the prioritisation strategy proposed in this paper.

1) *Versus the number of messages*: Fig. 14(a) shows how the delivery ratio varies with different message generation intervals. When messages are generated more frequently (i.e., shorter intervals), both methods experience performance degradation due to increased contention for limited network resources. However, DPUVR consistently achieves a higher delivery ratio. Specifically, when the interval is 2s, DPUVR delivers nearly 98% of messages, while the no-prioritization scheme delivers only about 72.68%. This highlights the effectiveness of DPUVR in managing message forwarding order. Fig. 14(b) illustrates that the latency decreases as the interval increases, due to reduced message load. DPUVR maintains a much lower latency throughout, indicating better resource efficiency. This is because DPUVR avoids redundant transmissions by forwarding fewer but more valuable messages based on prioritization.

2) *Versus the high priority message ratios*: Fig. 14(c) shows that as the proportion of high-priority messages increases, the delivery ratio of the no-prioritization method declines significantly, from around 96.74% to 61.78%. This is because the method treats all messages equally and fails to ensure the timely delivery of critical messages under high load. In contrast, DPUVR maintains a delivery ratio close to 100% across all ratios, demonstrating its robustness. Fig. 14(d) shows how the average delivery latency changes. For the no-prioritization method, latency increases sharply as high-priority messages dominate the traffic. Since all messages share the same queue and competition intensifies, delays grow. In contrast, DPUVR keeps latency significantly lower by dynamically adjusting delivery decisions to prioritize urgent messages.

To sum up, it can be found that DPUVR reduces the average delivery latency and overhead while ensuring a high average delivery ratio, but also achieves lower average buffer time. Therefore, we demonstrate that DPUVR is effective under different network scenarios.

VI. CONCLUSION

In this paper, we proposed a Dynamic Priority-based UAV-assisted VANET Routing (DPUVR) protocol to improve the efficiency and reliability of emergency message delivery in post-disaster scenarios. DPUVR was designed to address the challenge of fragmented ground connectivity by leveraging UAVs as scheduled aerial relays. Specifically, we developed a trajectory-aware, multi-metric relay selection strategy that evaluated candidate nodes based on trajectory similarity, surplus energy, link survival time, remaining distance cost and queuing delay. In addition, we introduced a dynamic prioritization model that classified messages by urgency and dynamically adjusted their forwarding order according to TTL and buffer state. Message-node pairs were ranked based on utility and priority, ensuring the timely delivery of critical message. Extensive simulation results demonstrated that DPUVR outperformed baseline protocols, including TBAR, STALB, VDTN and HRLIGR, achieving up to 99.7% delivery ratio, with lower average delay and overhead.

DPUVR effectively utilizes multi-attribute decision-making, it lacks adaptive learning capability. Reinforcement learning

(RL) integration may improve adaptive relay selection and weight tuning, but challenges such as computation cost, delayed convergence, and unstable network feedback must be addressed. Future work will focus on lightweight distributed RL frameworks that enable online adaptability with minimal resource overhead.

REFERENCES

- [1] N. Zhao et al., "UAV-assisted emergency networks in disasters," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 45–51, Feb. 2019.
- [2] S. Wu, W. Xu, F. Wang, G. Li, and M. Pan, "Distributed Federated Deep Reinforcement Learning Based Trajectory Optimization for Air-Ground Cooperative Emergency Networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 9107–9112, Aug. 2022.
- [3] K. G. Panda, S. Das, D. Sen, and W. Arif, "Design and Deployment of UAV-Aided Post-Disaster Emergency Network," *IEEE Access*, vol. 7, pp. 102985–102999, 2019.
- [4] Y. He, D. Wang, F. Huang, R. Zhang, and L. Min, "Aerial-Ground Integrated Vehicular Networks: A UAV-Vehicle Collaboration Perspective," *IEEE Trans. Intell. Transport. Syst.*, vol. 25, no. 6, pp. 5154–5169, Jun. 2024.
- [5] S. Mokhtari, N. Nouri, J. Abouei, A. Avokh, and K. N. Plataniotis, "Relaying Data With Joint Optimization of Energy and Delay in Cluster-Based UAV-Assisted VANETs," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24541–24559, Dec. 2022.
- [6] Y. Su, M. Liwang, Z. Chen, and X. Du, "Toward Optimal Deployment of UAV Relays in UAV-Assisted Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 13392–13405, Oct. 2023.
- [7] L. Wang, R. Li, L. Xu, W. Zhu, Y. Zhang, and A. Fei, "Aerial-Ground Cooperative Vehicular Networks for Emergency Integrated Localization and Communication," *IEEE Network*, vol. 37, no. 4, pp. 323–330, Jul. 2023.
- [8] O. S. Oubbati, M. Atiquzzaman, A. Baz, H. Alhakami, and J. Ben-Othman, "Dispatch of UAVs for Urban Vehicular Networks: A Deep Reinforcement Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13174–13189, Dec. 2021.
- [9] Y. Azzoug and A. Boukra, "Enhanced UAV-aided vehicular delay tolerant network (VDTN) routing for urban environment using a bio-inspired approach," *Ad Hoc Networks*, vol. 133, p. 102902, Aug. 2022.
- [10] Z. Du, C. Wu, X. Chen, X. Wang, and K.-L. A. Yau, "A Routing Protocol for UAV-Assisted Vehicular Delay Tolerant Networks," vol. 2, 2021.
- [11] H. Alsolai et al., "Enhanced Artificial Gorilla Troops Optimizer Based Clustering Protocol for UAV-Assisted Intelligent Vehicular Network," *Drones*, vol. 6, no. 11, p. 358, Nov. 2022.
- [12] Y. Cui, H. Tian, and G. Nie, "New Geographical Routing Protocol for Three-Dimensional Flying Ad-Hoc Network Based on New Effective Transmission Range," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, vol. 72, no. 12, 2023.
- [13] M. Y. Arafat and S. Moh, "A Q-Learning-Based Topology-Aware Routing Protocol for Flying Ad Hoc Networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1985–2000, Feb. 2022.
- [14] M. Zhang, C. Dong, P. Yang, T. Tao, Q. Wu, and T. Q. S. Quek, "Adaptive Routing Design for Flying Ad Hoc Networks," *IEEE Commun. Lett.*, vol. 26, no. 6, pp. 1438–1442, Jun. 2022.
- [15] Z. Yang, H. Liu, Y. Chen, X. Zhu, Y. Ning, and W. Zhu, "UEE-RPL: A UAV-Based Energy Efficient Routing for Internet of Things," *IEEE Trans. on Green Commun. Netw.*, vol. 5, no. 3, pp. 1333–1344, Sep. 2021.
- [16] X. Zhong, Y. Guo, N. Li, Y. Chen, and S. Li, "Deployment Optimization of UAV Relay for Malfunctioning Base Station: Model-Free Approaches," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11971–11984, Dec. 2019.
- [17] C. Zhang, M. Dong, and K. Ota, "Heterogeneous Mobile Networking for Lightweight UAV Assisted Emergency Communication," *IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING*, vol. 5, no. 3, 2021.
- [18] S. Gangopadhyay and V. K. Jain, "A Position-Based Modified OLSR Routing Protocol for Flying Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 12087–12098, Sep. 2023.
- [19] Y. Zeng et al., "Stable routing protocol for unmanned aerial vehicle ad-hoc networks based on DQN-OLSR," *IET Communications*, vol. 17, no. 1, pp. 73–85, Jan. 2023.
- [20] J. Fu, A. Nunez, and B. De Schutter, "Real-Time UAV Routing Strategy for Monitoring and Inspection for Post-disaster Restoration of Distribution Networks," *IEEE Trans. Ind. Inf.*, vol. 18, no. 4, pp. 2582–2592, Apr. 2022.
- [21] Y. Li, Q. Li, H. Yu, and L. Li, "TAAODV: TOPSIS-Aided AODV Routing Protocol to Diffuse Mass Information in the Unstable and High-Density UAV Ad Hoc Networks," *IEEE Trans. Intell. Transport. Syst.*, vol. 26, no. 10, pp. 17255–17268, Oct. 2025.
- [22] S. Garg, A. Ihler, E. S. Bentley, and S. Kumar, "A Hybrid Reactive Routing Protocol for Decentralized UAV Networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 61, no. 5, pp. 11683–11701, Oct. 2025.
- [23] H. Tang, P. Wang, D. Li, Y. Zhang, and X. Chen, "An Adaptive Virtual Tunnel Routing Protocol With Eliminating Boundary Effects for Flying Ad-Hoc Networks," *IEEE Internet Things J.*, vol. 12, no. 16, pp. 34068–34085, Aug. 2025.
- [24] J. Chong, X. Jia, and Z. Yang, "Toward Routing in Low-Altitude Drone Networks: A Physical Sensing-Aided Intelligent Forwarding Mechanism With Deep Learning," *IEEE Internet Things J.*, vol. 12, no. 13, pp. 25442–25456, Jul. 2025.
- [25] N. P. Sharvari, D. Das, J. Bapat, and D. Das, "Improved Q-Learning-Based Multi-Hop Routing for UAV-Assisted Communication," *IEEE Trans. Netw. Serv. Manage.*, vol. 22, no. 2, pp. 1330–1344, Apr. 2025.
- [26] B. Song et al., "DRL-AdCAR: Adaptive Coding-Aware Routing With Maximum Coding Opportunities and High-Quality via Deep Reinforcement Learning in FANET," *IEEE Trans. Veh. Technol.*, vol. 74, no. 1, pp. 1280–1295, Jan. 2025.
- [27] Z. Zhou, J. Tang, W. Feng, N. Zhao, Z. Yang, and K.-K. Wong, "Optimized Routing Protocol Through Exploitation of Trajectory Knowledge for UAV Swarms," *IEEE Trans. Veh. Technol.*, vol. 73, no. 10, pp. 15499–15512, Oct. 2024.
- [28] P. Mahajan, B. Palanisamy, A. Kumar, G. S. S. Chalapathi, V. Chamola, and M. Khabbaz, "Multi-Objective MDP-Based Routing in UAV Networks for Search-Based Operations," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13777–13789, Sep. 2024.
- [29] M. Zhang et al., "Adaptive Routing Design for Flying Ad Hoc Networks: A Joint Prediction Approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2593–2604, Feb. 2024.
- [30] S. Garg, A. Ihler, E. S. Bentley, and S. Kumar, "A Cross-Layer, Mobility, and Congestion-Aware Routing Protocol for UAV Networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 3778–3796, Aug. 2023.
- [31] A. Rovira-Sugranes, F. Afghah, J. Qu, and A. Razi, "Fully-Echoed Q-Routing With Simulated Annealing Inference for Flying Adhoc Networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2223–2234, Jul. 2021.
- [32] X. Jian et al., "High-Reliability Low-Latency Intelligent Geographic Routing Protocol for Vehicle Road Cooperation System," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 1774–1788, Jan. 2025.
- [33] F. Xu, C. Yang, H. Chen, M. Ahmed, M. Deng, and Y. Lyu, "VANETs Routing Optimization: Combining Message Priority, Free Nodes, and Mobility Direction," *IEEE Access*, vol. 13, pp. 23331–23343, 2025.
- [34] H. Wang, "Dynamic Topology Evolution and Multi-Objective Routing Optimization for Efficient VANET Communication," *IEEE Access*, vol. 13, pp. 36124–36134, 2025.
- [35] Z. Han, C. Xu, S. Ma, Y. Hu, G. Zhao, and S. Yu, "DTE-RR: Dynamic Topology Evolution-Based Reliable Routing in VANET," *IEEE Wireless Commun. Lett.*, vol. 12, no. 6, pp. 1061–1065, Jun. 2023.
- [36] A. Hawbani et al., "A Novel Heuristic Data Routing for Urban Vehicular Ad Hoc Networks," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8976–8989, Jun. 2021.
- [37] P. Singh, R. S. Raw, S. A. Khan, M. A. Mohammed, A. A. Aly, and D.-N. Le, "W-GeoR: Weighted Geographical Routing for VANET's Health Monitoring Applications in Urban Traffic Networks," *IEEE Access*, vol. 10, pp. 38850–38869, 2022.
- [38] Y. Cao et al., "A Trajectory-Driven Opportunistic Routing Protocol for VCPS," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 2628–2642, Dec. 2018.
- [39] Y. Song, K. Jiang, Y. Cao, R. Zhou, C. Suthaputchakun, and Y. Zhuang, "STALB: A Spatio-Temporal Domain Autonomous Load Balancing Routing Protocol," *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 1, pp. 73–87, Mar. 2023.
- [40] Z. Fan, Y. Liu, M. Zhang, Y. Cao, Y. He, and K. Wang, "Trajectory-Based Anycast Routing Protocol with MDRUs Assistance in Disaster Response Network," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, Milan, Italy: IEEE, Mar. 2025, pp. 1–6.
- [41] S. Li, B. Wang, S. Qian, Y. Sun, X. Yun, and Y. Zhou, "Influence Maximization for Emergency Information Diffusion in Social Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8768–8782, Aug. 2022.
- [42] W. Chaowei, W. Ziye, X. Lexi, Y. Xiaofei, Z. Zhi, and W. Weidong, "Collaborative Caching in Vehicular Edge Network Assisted by Cell-Free

- Massive MIMO,” *Chinese J. Elect.*, vol. 32, no. 6, pp. 1218–1229, Nov. 2023.
- [43] C. Chen, W. Chenyu, C. Li, X. Ming, and P. Qingqi, “A V2V Emergent Message Dissemination Scheme for 6G-Oriented Vehicular Networks,” *Chinese J. Elect.*, vol. 32, no. 6, pp. 1179–1191, Nov. 2023.
- [44] Y. Chang and X. Wang, “Robust Predictive Routing for Internet of Vehicles With Context-Aware Link Estimation and Multipath Verification,” *IEEE Internet Things J.*, vol. 12, no. 20, pp. 43345–43358, Oct. 2025.
- [45] Q. V. Khanh, V. A. Dang, Q. N. Minh, and D. C. Nguyen, “A QoS-Adaptive Routing Scheme for Urban-MANETs,” *IEEE Sensors J.*, vol. 25, no. 13, pp. 26001–26008, Jul. 2025.
- [46] S. Babu and A. R. K. P., “CDSPAN: A Collaborative Distributed SPANner Backbone for Multiple Source Multicast Routing in Vehicular Network,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5213–5228, Apr. 2024.
- [47] V. K. Quy, A. Chehri, N. M. Quy, V.-H. Nguyen, and N. T. Ban, “An Efficient Routing Algorithm for Self-Organizing Networks in 5G-Based Intelligent Transportation Systems,” *IEEE Trans. Consumer Electron.*, vol. 70, no. 1, pp. 1757–1765, Feb. 2024.
- [48] P. Chithaluru et al., “A Lightweight Energy-Efficient Routing Scheme for Real-Time WSN-VANET-Based Applications,” *IEEE Trans. Consumer Electron.*, vol. 70, no. 1, pp. 3820–3826, Feb. 2024.
- [49] J.-H. Park, Q. Yang, and S.-J. Yoo, “RHRA-DRL: RSU-Assisted Hybrid Road-Aware Routing Using Distributed Reinforcement Learning in Internet of Vehicles,” *IEEE Access*, vol. 12, pp. 25385–25396, 2024.
- [50] L. Rui et al., “An Intersection-Based QoS Routing for Vehicular Ad Hoc Networks With Reinforcement Learning,” *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 9, pp. 9068–9083, Sep. 2023.
- [51] M. Sharma, P. Kumar, and R. S. Tomar, “Weight-Based Clustering Algorithm for Military Vehicles Communication in VANET,” *SAIEE Afr. Res. J.*, vol. 114, no. 1, pp. 25–34, Mar. 2023.
- [52] B. Su and L. Tong, “Transmission Protocol of Emergency Messages in VANET Based on the Trust Level of Nodes,” *IEEE Access*, vol. 11, pp. 68243–68256, 2023.
- [53] M. Mazouzi, K. Merashad, O. Cheikhrouhou, and M. Hamdi, “Agent-Based Reactive Geographic Routing Protocol for Internet of Vehicles,” *IEEE Access*, vol. 11, pp. 79954–79973, 2023.
- [54] C. Ghorai, S. Shakhari, and I. Banerjee, “A SPEA-Based Multimetric Routing Protocol for Intelligent Transportation Systems,” *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 11, pp. 6737–6747, Nov. 2021.
- [55] S. Jia and L. Zhang, “Modelling unmanned aerial vehicles base station in ground-to-air cooperative networks,” *IET Commun.*, vol. 11, no. 8, pp. 1187–1194, 2017.
- [56] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, “UAV-assisted supporting services connectivity in urban VANETs,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3944–3951, Apr. 2019.
- [57] O. S. Oubbati, A. Lakas, F. Zhou, M. Güne s, N. Lagraa, and M. B. Yagoubi, “Intelligent UAV-assisted routing protocol for urban VANETs,” *Comput. Commun.*, vol. 107, pp. 93–111, Jul. 2017.
- [58] G. Manogaran, C.-H. Hsu, P. M. Shakeel, and M. Alazab, “Non-Recurrent Classification Learning Model for Drone Assisted Vehicular Ad-Hoc Network Communication in Smart Cities,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2792–2800, Oct. 2021.
- [59] Y. Zhou, Z. Jin, H. Shi, L. Shi, N. Lu, and M. Dong, “Enhanced Emergency Communication Services for Post-Disaster Rescue: Multi-IRS Assisted Air-Ground Integrated Data Collection,” *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 4651–4664, Sep. 2024.
- [60] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, “Efficient and Secure Routing Protocol Based on Artificial Intelligence Algorithms With UAV-Assisted for Vehicular Ad Hoc Networks in Intelligent Transportation Systems,” *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 7, pp. 4757–4769, Jul. 2021.
- [61] Y. He, D. Zhai, Y. Jiang, and R. Zhang, “Relay selection for UAV-assisted urban vehicular ad hoc networks,” *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1379–1383, Sep. 2020.
- [62] S. Jiang, Z. Huang, and Y. Ji, “Adaptive UAV-Assisted Geographic Routing With Q-Learning in VANET,” *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1358–1362, Apr. 2021.
- [63] J. Chen et al., “Enhancing Routing Performance Through Trajectory Planning With DRL in UAV-Aided VANETs,” *Trans. Mach. Learn. Comm. Netw.*, vol. 3, pp. 517–533, 2025.
- [64] H. Liu, R. Chen, S. Ding, Z. Jiang, F. Liu, and J. Zhang, “An energy efficiency routing protocol for UAV-aided WSNs data collection,” *Ad Hoc Networks*, vol. 154, p. 103378, Mar. 2024.
- [65] K. Karunanithy, B. Velusamy, S. Krishnakumar, and V. J. Senthilkumar, “Energy-Efficient Data Routing in Landslide-Prone Area Using Wireless Sensor Networks With Drone,” *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4497–4507, Feb. 2024.
- [66] X. Wei, H. Yang, and W. Huang, “Low-Delay Routing Scheme for UAV Communications in Smart Cities,” *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18837–18843, Nov. 2023.
- [67] A. Raza, Z. Iqbal, and F. Aadil, “UAV-assisted ubiquitous communication architecture for urban VANET environment,” *J Supercomput.*, vol. 79, no. 13, pp. 14602–14632, Sep. 2023.
- [68] S. R. Yeduri, N. S. Chilamkurthy, O. J. Pandey, and L. R. Cenkeramaddi, “Energy and Throughput Management in Delay-Constrained Small-World UAV-IoT Network,” *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7922–7935, May 2023.
- [69] L. Zhang, X. Ma, Z. Zhuang, H. Xu, V. Sharma, and Z. Han, “Q-Learning Aided Intelligent Routing With Maximum Utility in Cognitive UAV Swarm for Emergency Communications,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3707–3723, Mar. 2023.
- [70] O. Chughtai, N. N. Qadri, Z. Kaleem, and C. Yuen, “Drone-Assisted Cooperative Routing Scheme for Seamless Connectivity in V2X Communication,” *IEEE Access*, vol. 12, pp. 17369–17381, 2024.
- [71] Y. He, L. Nie, T. Guo, K. Kaur, M. M. Hassan, and K. Yu, “A NOMA-Enabled Framework for Relay Deployment and Network Optimization in Double-Layer Airborne Access VANETs,” *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 11, pp. 22452–22466, Nov. 2022.
- [72] K. Guo, R. Liu, C. Dong, K. An, Y. Huang, and S. Zhu, “Ergodic Capacity of NOMA-Based Overlay Cognitive Integrated Satellite-UAV-Terrestrial Networks,” *Chinese J. Elect.*, vol. 32, no. 2, pp. 273–282, Mar. 2023.
- [73] J. Cui et al., “A Practical and Provably Secure Authentication and Key Agreement Scheme for UAV-Assisted VANETs for Emergency Rescue,” *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 1454–1468, Mar. 2024.
- [74] J. Li, T. Han, W. Guan, and X. Lian, “A Preemptive-Resume Priority MAC Protocol for Efficient BSM Transmission in UAV-Assisted VANETs,” *Applied Sciences*, vol. 14, no. 5, p. 2151, Mar. 2024.
- [75] X. Yang, Y. Mao, Q. Xu, and L. Wang, “Priority-Based Hybrid MAC Protocol for VANET with UAV-Enabled Roadside Units,” *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–13, Feb. 2022.
- [76] M. Dhuheir, A. Erbad, A. Al-Fuqaha, and A. M. Seid, “Meta Reinforcement Learning for UAV-Assisted Energy Harvesting IoT Devices in Disaster-Affected Areas,” *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2145–2163, 2024.
- [77] O. Ghdiri, W. Jaafar, S. Alfattani, J. B. Abderrazak, and H. Yanikomeroglu, “Offline and online UAV-enabled data collection in time-constrained IoT networks,” *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1918–1933, Dec. 2021.
- [78] G. Sun, Y. Zhang, D. Liao, H. Yu, X. Du, and M. Guizani, “Bus-trajectory-based street-centric routing for message delivery in urban vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7550–7563, Aug. 2018.
- [79] C. Xu, Z. Xiong, Z. Han, G. Zhao, and S. Yu, “Link reliability-based adaptive routing for multilevel vehicular networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11771–11785, Oct. 2020.
- [80] Saaty, Thomas L. , and K. P. Kearns . ”The Analytic Hierarchy Process.” analytical planning (1985).
- [81] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proc. Int. ICST Conf. Simul. Tools Techniques*, Rome, Italy, 2009.
- [82] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Edsger Wybe Dijkstra: His Life, Work, Legacy*, pp. 287–290, 2022.