Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DARA: Distributed Agent Reinforcement-based Adaptive routing in Flying Ad-hoc Network Environments

Badia Bouhdid

badiaa.bouhdid@ensi-uma.tn

ESPRIT School of Engineering

Adel Ben Mnaouer

Dr. Hussein AlSayyed Research and Innovation Center

Adnan Ahmed Abi Sen

Dr. Hussein AlSayyed Research and Innovation Center

Bassem Ouni

Technology Innovation Institute

Research Article

Keywords: Flying Ad-hoc Networks (FANETs), Distributed Agents, Reinforcement Learning, Adaptive Routing, Next-Hop Selection, Q-Learning, UAV Networks.

Posted Date: June 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9103864/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

DARA: Distributed Agent Reinforcement-based Adaptive routing in Flying Ad-hoc Network Environments

Badia Bouhdid[1,2*], Adel Ben Mnaouer[3†], Adnan Ahmed Abi Sen[3†], Bassem Ouni[4†]

[1*]HANA-Lab, University of Manouba, Tunisia Tunisia.
[2]ESPRIT School of Engineering, Ariana, Tunisia.
[3]Dr. Hussein AlSayyed Research and Innovation Center, University of Prince Mugrin, Medina, Saudi Arabia.
[4]Technology Innovation Institute, Abu Dhabi, UAE.

*Corresponding author(s). E-mail(s): badiaa.bouhdid@ensi-uma.tn;
Contributing authors: a.mnaouer@upm.edu.sa ; a.ahmed@upm.edu.sa ; bassem.Ouni@tii.ae;
†These authors contributed equally to this work.

Abstract

Flying Ad-hoc Networks (FANETs) are characterized by highly dynamic topologies, which make reliable, low-latency, and energy-efficient communication between Unmanned Aerial Vehicles (UAVs) and Ground Base Stations (GBSs) particularly challenging. Rapid three-dimensional mobility, frequent link disruptions, and the lack of persistent infrastructure significantly limit the effectiveness of conventional MANET-inspired and clustering-based routing approaches in aerial scenarios.

To address these challenges, this paper proposes DARA (Distributed Agent Reinforcement-based Adaptive Routing), a fully decentralized routing framework based on reinforcement learning. In DARA, each UAV functions as an independent learning agent that selects its next-hop relay using only locally available information, including link quality, relative progress toward the GBS, residual energy, and neighbor mobility. By distributing intelligence and decision-making across the UAV swarm, the framework minimizes routing overhead, mitigates bottlenecks, and enhances link stability.

Comprehensive simulations conducted using ns-3 and ns3-gym show that DARA consistently outperforms classical and recent FANET routing protocols in terms

of packet delivery ratio, end-to-end delay, and energy balancing. These findings highlight the potential of decentralized, reinforcement learning–driven routing as a scalable and robust solution for next-generation autonomous UAV networks.

Keywords: Flying Ad-hoc Networks (FANETs), Distributed Agents, Reinforcement Learning, Adaptive Routing, Next-Hop Selection, Q-Learning, UAV Networks.

1 Introduction

A Flying Ad Hoc Network (FANET) is a self-organizing wireless network formed by a group of Unmanned Aerial Vehicles (UAVs) or drones that communicate directly in an ad hoc manner without fixed infrastructure, enabling collaborative operations in three-dimensional aerial environments [1, 2]. FANETs find key applications in disaster monitoring and response [3, 5, 6], environmental monitoring [10], search and rescue missions [6], precision agriculture[8], and infrastructure inspection, where rapid deployment and flexibility are essential. Effective data routing is necessary in FANETs to ensure reliable end-to-end communication, maintain connectivity among highly mobile nodes, and support real-time data delivery to ground stations or other UAVs despite frequent topology changes. However, routing in FANETs faces significant challenges, including very high node mobility leading to rapid link breakages, dynamic and unpredictable network topology, scalability issues in large swarms and severe constraints in terms of energy budget, wireless channel variability, and computational resources[7, 9]. These characteristics make the design of stable, energy-efficient, and low-latency routing protocols a persistent research challenge [16].

While FANETs evolved from the principles of Mobile Ad Hoc Networks (MANETs)[12] traditional MANET routing protocols, such as AODV [13], DSR [14] and OLSR [15], were not designed to cope with rapid and frequent topology changes, three-dimensional spatial dynamics and limited energy budgets,factors far more severe than in ground-based MANETs. The performance of MANET based routing approach degrades sharply in FANET scenarios, resulting in frequent route breakages, high end-to-end delay, and increased routing overhead [7, 11, 16].

To overcome these limitations, a wide range of FANET-tailored routing protocols has emerged. These primarily fall into two main paradigms: clustering-based and flat routing strategies. Cluster-based approaches organize UAVs into logical groups or clusters. Each cluster is managed by a designated cluster head (CH) responsible for intra-cluster coordination and inter-cluster data forwarding [17, 18]. This hierarchical organization substantially reduces control message overhead, improves scalability for large swarms, and enhances overall network stability amid frequent topology changes. Although effective in moderate mobility regimes, clustering approaches suffer from high re-clustering cost, control-message overhead, and sensitivity to topology disruptions—making them unsuitable for highly dynamic and decentralized aerial networks. In contrast, flat routing approaches[19] treat all UAVs as peers, typically employing reactive or position-based mechanisms. While these approaches offer simplicity and

lower initial setup complexity, they often incur higher flooding or route discovery overhead, particularly in dense or highly dynamic large-scale UAV deployments.

Recent developments in artificial intelligence (AI) and machine learning (ML) have made them strong enabler technologies for improving conventional routing techniques for FANETs. By moving beyond static heuristics and predefined models, learning-based routing approaches enable protocols to adapt autonomously to the highly dynamic and volatile nature of UAV networks.. Recent survey papers, [20, 21],highlight this paradigm shift, categorizing ML-driven routing enhancements—such as supervised learning for link quality prediction, unsupervised clustering for topology inference, and neural network–based traffic forecasting—as key innovations that significantly improve performance metrics including packet delivery ratio, end-to-end delay, energy efficiency, and robustness against mobility-induced disruptions in UAV swarms.

Among these techniques, reinforcement learning (RL) has received significant attention as it enables fully autonomous routing policy learning through a model-free, trial-and-error interaction with dynamic network environments [22–24]. Within the RL family, Q-learning has been extensively adopted for FANET routing owing to its algorithmic simplicity, off-policy learning capability, and convergence toward optimal action–value functions (Q-values) under partial observability and non-stationary conditions typical of aerial networks. Recent studies leverage Q-learning to improve next-hop selection [25], cluster-head election in hierarchical architectures [26], trajectory-aware geographic routing [27], and adaptive bitrate control under varying link conditions [27], consistently outperforming classical routing protocols in terms of delivery reliability, delay minimization, and adaptability to high-speed UAV mobility. These findings collectively demonstrate the effectiveness of Q-learning in enabling proactive, self-optimizing routing strategies driven by real-time experiential learning.

However, most recent RL contributions to FANET routing still predominantly assume centralized learning architectures, often relying on a ground station or central agent for policy training and decision aggregation, or depend on frequent global state exchanges among UAV nodes to maintain coherent Q-values and policy updates. This introduces significant communication overhead and vulnerability to single points of failure in highly dynamic environments [24]. Alternatively, many approaches integrate RL within clustering frameworks, where reinforcement learning is employed for cluster-head selection, re-clustering policies, or inter-cluster forwarding, often in a centralized or semi-centralized manner. While this leverages hierarchical organization for localized decisions, it reintroduces classic bottlenecks associated with cluster maintenance (e.g., periodic re-election and reconfiguration triggered by mobility), excessive message overhead for cluster status synchronization and state sharing, and reduced scalability in large-scale or ultra-high-mobility UAV swarms, where frequent topology disruptions amplify these issues [28, 29]. These limitations hinder the full potential of RL in truly decentralized, real-time adaptive routing for FANETs, motivating the need for lightweight, distributed alternatives that minimize global dependencies and clustering overheads.

To address these challenges, we introduce **DARA**, a fully decentralized routing framework where every UAV acts as an independent and autonomous RL agent. DARA eliminates clustering structures and any form of centralized control, resulting

in reduced routing overhead and improved real-time adaptability to dynamic network topologies. Each UAV learns its optimal next-hop decision based purely on what it can observe locally such as current link quality, its own movement patterns, distance to any high-altitude platforms (HAPs), how full its buffer is, and remaining energy levels. This keeps data delivery reliable and robust, even when nodes are moving fast or the swarm has very different conditions across UAVs.

The key contributions of our work can be summarized as follows:

- We present a distributed RL-based routing framework tailored to highly dynamic FANET environments, eliminating the reliance on clusters or central coordinators that often cause bottlenecks.
- We develop a compact yet effective state representation together with a multi-goal reward function that balances link reliability, routing stability, congestion avoidance, and energy usage.
- We build a set of lightweight, practical algorithms for beacon exchange, RL-driven next-hop selection, and relay behavior, which together create a clear, three-phase workflow that any UAV can run efficiently.
- We evaluate DARA in realistic ns-3 and ns3-gym simulation setups, testing it across various mobility models and traffic patterns, where it consistently shows better results than established baselines like AODV, EMASS, and ESOFCluster..

The remainder of this paper is organized as follows; Section 2 reviews the most relevant routing strategies proposed for FANETs and highlights their limitations in highly dynamic aerial environments. Section 3 presents the network model and the set of assumptions related to the proposed approach. Section 4 presents the proposed DARA framework, describing the network model, the three operational stages, and the reinforcement learning mechanisms for next-hop selection. Section 5 details the simulation setup, scenarios, and performance metrics, followed by an in-depth analysis and comparison with baseline routing protocols. Finally, Section 6 concludes the paper and outlines promising directions for future research.

2 Challenges, Study of the Existing and Motivation

Routing in FANETs is considerably more complex than in traditional MANETs or Vehicular Ad-Hoc Networks (VANETs) due to the combination of three-dimensional mobility, rapidly evolving topology, strict on-board energy limitations, and highly variable wireless channel conditions. During the past five years (2021–2025), research on FANET routing has progressed along several major directions, ranging from adaptations of classical MANET protocols, to geographic and mobility-prediction techniques, to clustering-based and fuzzy-logic architectures, and more recently to Software-Defined Networking (SDN)-based and RL–driven solutions. Although each family of approaches offers valuable contributions, they still exhibit significant limitations when applied to highly dynamic UAV networks.

In this section, we first identify and discuss the fundamental *routing challenges* that characterize FANET environments. We then examine the major categories of existing routing strategies—including MANET-derived protocols, clustering-based hierarchical

approaches, SDN-centric architectures, and intelligent learning-based methods—and analyze their strengths and limitations. Finally, we highlight the gap in the current literature and motivate the need for a fully distributed, adaptive, and collaborative routing framework such as the proposed DARA protocol.

2.1 Key Routing Challenges in UAV Networks

Fig. 1 presents an overview of the primary routing hurdles in UAV networks (FANETs), which researchers commonly organize into five main categories based on recent surveys and reviews [2, 7, 9, 16]. These include:

Mobility and Topology Dynamics: UAVs in FANETs exhibit highly dynamic and heterogeneous 3D mobility, leading to rapid topology shifts, very short link lifetimes, frequent neighbor changes, and unpredictable line-of-sight (LoS) variations. Unlike road-constrained VANETs or lower-mobility MANETs, this spatial volatility causes constant link breakages and network partitioning. Routing protocols must therefore support near-instantaneous adaptation to topology changes, a requirement that exceeds the capabilities of most MANET-derived protocols [30].

Communication Challenges: Communication in FANETs is inherently unreliable due to frequent link disruptions, interference, and limited bandwidth. Environmental factors such as Doppler shifts, multipath propagation, and electromagnetic interference significantly degrade link quality, leading to intermittent connectivity [31]. Link failures commonly occur when UAVs move beyond communication range, encounter physical obstacles, or undergo abrupt mobility changes. Moreover, the scarcity of available spectrum, particularly when UAVs operate alongside commercial or mission-critical systems, requires routing protocols to be highly bandwidth-efficient and robust under constrained network conditions [9].

Resource Limitations: UAVs are subject to stringent resource constraints, including limited battery capacity, lightweight processing hardware, and restricted memory. Routing related operations such as table maintenance, periodic dissemination of control messages, exchange of global network metrics, or cluster management can can impose considerable energy resources demands. Recent literature [7, 16], consistently identifies these resource constraints as a fundamental challenge in FANET protocol design, underscoring the necessity for lightweight, energy-efficient algorithms that effectively reconcile power economy with robust data delivery in highly dynamic aerial environments

Network Management Constraints: FANETs must accommodate a wide range of heterogeneous mission requirements, often with strict Quality of Service (QoS) demands [6]. Effective load balancing plays a crucial role in avoiding congestion hotspots at individual nodes, ensuring an equitable distribution of resources across the swarm, and maintaining overall network stability under varying traffic loads. Scalability becomes a major issue in large UAV deployments, where excessive control signaling, frequent global metric exchanges, or reliance on centralized coordination mechanisms can overwhelm the network [16]. Consequently, fully distributed routing approaches become a necessity for supporting scalable, resilient operations in expansive swarms, as they prevent single points of failure and minimize the communication overhead associated with centralized or hierarchical management structures.

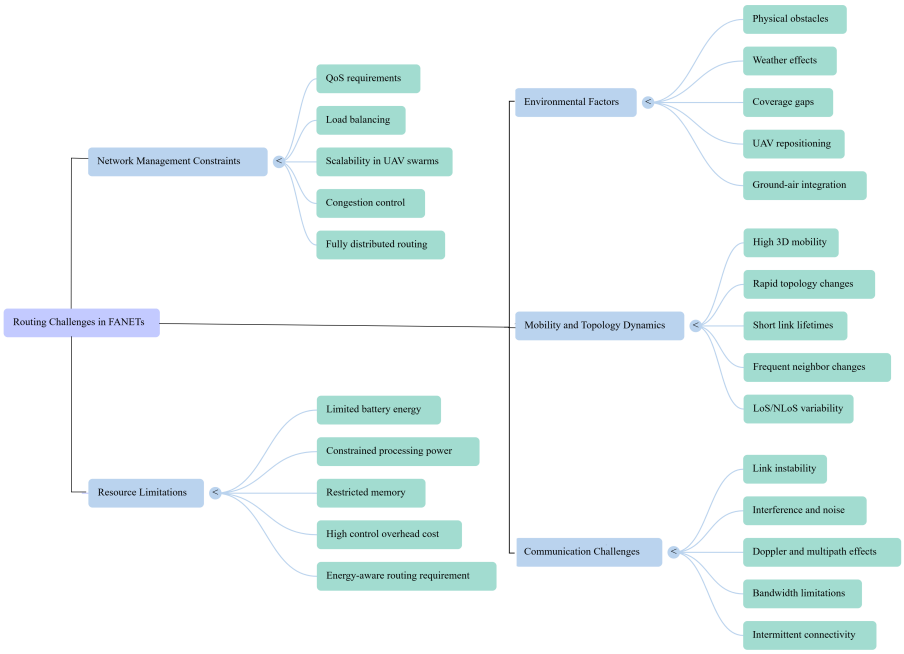


Fig. 1 Key routing challenges in UAV networks

Environmental Factors: Environmental factors pose additional significant challenges to reliable routing in FANETs. In fact, physical obstacles including buildings, hills,vegetation and other structures frequently create shadowing effects and non-line-of-sight (NLoS) propagation paths. In rural, remote, or disaster-affected regions, these factors often result coverage gaps requiring UAVs to adaptively reposition themselves or activate relay forwarding behaviors to restore end-to-end connectivity. Moreover, integrating FANETs with ground-based terrestrial networks remains an open challenge for hybrid terrestrial–aerial networks, requiring routing protocols that can operate seamlessly across multiple layers of infrastructure [16].

2.2 Study of the existing

Existing routing protocols for FANETs)can be broadly classified into several categories, reflecting the evolution from traditional adaptations to more advanced, intelligent, and structured approaches.

2.2.1 MANET Drived routing approaches

Early research on FANET routing largely adapted well-established MANET protocols, which are primarily topology-based and traditionally classified as proactive, reactive, or hybrid. Proactive protocols, such as Optimized Link State Routing (OLSR)[15], rely on periodic routing table updates to maintain global network knowledge.Although several adaptations have been proposed to accommodate the faster topology dynamics

of FANETs [11, 32], these protocols remain sensitive to intermittent links and strict energy constraints inherent to aerial networks [7].

Reactive protocols, including Ad Hoc On-Demand Distance Vector (AODV)[13] and Dynamic Source Routing (DSR) [14], discover routes on-demand, making them more suitable for dynamic environments. FANET-specific extensions often incorporate position prediction or link stability metrics to counteract frequent route breakages caused by high UAV mobility [19]. While these modifications improve robustness, rapid topology changes continue to challenge their reliability.

Hybrid approaches, such as Zone Routing Protocol (ZRP), combine proactive intra-zone and reactive inter-zone routing schemes by maintaining local routes proactively while discovering distant routes on demand. Recent FANET-oriented variants focus on adaptive zone sizing to better manage routing overhead and scalability in three-dimensional aerial environments [11].

Despite offering a practical starting point, MANET-derived routing protocols struggle to cope with the extreme volatility of FANETs without extensive modifications, often resulting to increased packet loss and delay. Recent studies such as [34] acknowledge these improvements but consistently highlight ongoing limitations in real-time adaptability and fully decentralized operation.

2.2.2 Hierarchical/Cluster-Based routing Approaches

Hierarchical or cluster-based routing protocols organize UAVs into logical groupsknown as clusters, where cluster heads (CHs) manage intra-cluster coordination and handle inter-cluster data forwarding , thereby improving scalability in dynamic FANET environments [7].

CH election plays a pivotal role in the effectiveness of these protocols. Early approaches, largely adapted from wireless sensor networks, rely on heuristic-based selection strategies and have been extensively applied to FANETs. The classic Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol selects CHs probabilistically based on residual energy, combined with periodic role rotation to balance network load and extend network lifetime [35]. Building on this idea, EMASS algorithm introduces a weighted, multi-metric CH election strategy that jointly considers residual energy, mobility patterns, safety constraints, and link stability. This integrated approach enables more robust clustering performance under the high-speed, three-dimensional mobility characteristic of FANETs [17].

These heuristic methods offer advantages such as energy balancing through CH rotation and improved network stability in dense swarms compared to flat topologies [13]. However, they rely on simplistic probabilistic or weighted models that often fail to adapt well to FANET's rapid topology changes, leading to frequent re-clustering, increased overhead during high mobility, and suboptimal CH selection in heterogeneous conditions.

To address these limitations, fuzzy logic-based approaches have emerged for more adaptive and uncertainty-aware CH election. Recent works [18] introduce fuzzy logic frameworks that evaluate multiple fuzzy inputs (e.g., residual energy, mobility, connectivity, distance) to select stable and energy-efficient CHs. For instance, their

ESOFCluster scheme uses a fuzzy inference system for adaptive clustering, enhancing stability and energy efficiency in dynamic FANETs , while their fuzzy logic-based adaptive clustering further optimizes stability under varying UAV velocities and densities. These methods improve decision-making in uncertain environments but still face challenges in computational overhead for real-time fuzzy rule evaluation on resource-constrained UAVs.

Building on these heuristic and fuzzy foundations, reinforcement learning—particularly Q-learning—has been integrated for smarter, learning-based CH election. Q-learning enables UAVs to iteratively learn optimal CH selection policies by rewarding stable, energy-efficient choices based on network states (e.g., mobility, residual energy, link quality). A prominent example is the Q-Learning-based Smart Clustering Routing (QSCR) method presented in [26], which applies Q-learning to discover neighbors, elect CHs intelligently, and perform greedy inter-cluster forwarding, outperforming traditional weighted clustering in terms of stability and routing efficiency.

While Q-learning-based methods offer superior adaptability and long-term optimization, many remain semi-centralized or embedded in clustering frameworks, reintroducing maintenance overhead and scalability issues under extreme mobility—motivating fully distributed alternatives like our DARA approach.

Recent surveys such as [28] highlight this progression from heuristics,fuzzy, RL-enhanced clustering as a key trend, yet emphasize persistent bottlenecks in overhead, re-clustering frequency, and true decentralization.

2.2.3 SDN-Based routing Approaches

Software-Defined Networking (SDN) introduces a logically centralized control plane that separates it from the data plane, enabling programmable, dynamic routing decisions in FANETs through a central controlleroften ground-based or on a high-altitude platform (HAP) [36]. In SDN-FANET architectures, the controller collects global network state (topology, link quality, traffic patterns) via communication protocols and installs optimized flow rules on UAV nodes for forwarding, load balancing, QoS enforcement, and rapid reconfiguration.

Recent proposals has increasingly explored SDN to address FANET-specific challenges, such as multi-path routing for improved reliability, traffic engineering for heterogeneous missions requirements, and the integration of UAV mobility prediction to proactively update forwarding rules before link failures occur. Representative examples include SDN-enabled FANET architectures that combine centralized path computation with lightweight local fallback mechanisms to maintain connectivity during controller delays or disconnections [37]. Other works adopt hybrid SDN–reinforcement learning (SDN-RL) approaches, in which the controller employs learning-based techniques to dynamically optimize flow rules in response to evolving network conditions [38].

The primary advantages of SDN-based routing stem from its global network visibility, which enables more efficient resource allocation and stronger Quality of Service (QoS) support. Centralized policy control allows rapid adaptation to topology changes, while also simplifying the management of large-scale UAV swarms. In addition, SDN

facilitates the seamless integration of advanced capabilities, such as network slicing and security policy enforcement. Collectively, these benefits make SDN-based routing particularly well suited for mission-critical FANET applications that demand predictable and reliable performance.

Despite these advantages, SDN-based routing introduces notable limitations due to its reliance on a central controller. Effective operation depends on stable controller–UAV connectivity, which is particularly vulnerable to link disruptions, interference, or jamming. In high-mobility scenarios, control-plane latency can increase significantly, delaying routing updates and reducing responsiveness. Moreover, the centralized control paradigm creates a single point of failure and may lead to scalability bottlenecks when the controller is overwhelmed by frequent state updates from large numbers of rapidly moving UAVs. In fully aerial or infrastructure-less FANET deployments, maintaining reliable control channels remains a major challenge, often necessitating a fallback to distributed operation modes that diminish the overall benefits of SDN [30].

Recent literature, [22], positions SDN as a powerful paradigm for programmable FANETs but consistently highlights the tension between centralized intelligence and the need for decentralization in highly dynamic, resource-constrained aerial environments—reinforcing the motivation for fully distributed, agent-based alternatives like DARA that eliminate controller dependency while retaining adaptive decision-making.

2.3 Motivation for a Distributed Reinforcement-Learning Routing Approach

As illustrated in Fig. 2 and summarized in Table 1,the review of existing FANET routing strategies highlights a fundamental gap between the requirements of highly dynamic UAV networks and the capabilities of traditional, cluster-based, or SDN-driven routing architectures. Although MANET-derived protocols, clustering mechanisms, and SDN controllers each contribute valuable ideas, they share structural limitations that prevent them from achieving robust performance under fast 3D mobility, intermittent connectivity, and decentralized mission scenarios.

This gap highlights the necessity for a routing mechanism that:

- **Enables each UAV to autonomously make next-hop decisions** based on local observations.
- **Continuously adapts routing behavior** to mobility, link quality, queue dynamics, and energy constraints.
- **Eliminates the need for centralized coordination or cluster structures**, thereby ensuring resilience and scalability.
- **Allows learning to emerge from distributed interactions**, such that local decisions collectively improve global network performance.

These requirements motivate the development of **DARA**: a Distributed Agent Reinforcement-based Adaptive Routing framework—in which eeach UAV operates as an independent RL agent. By eliminating clustering overhead, avoiding centralized control, and enabling real-time policy adaptation, DARA addresses key limitations

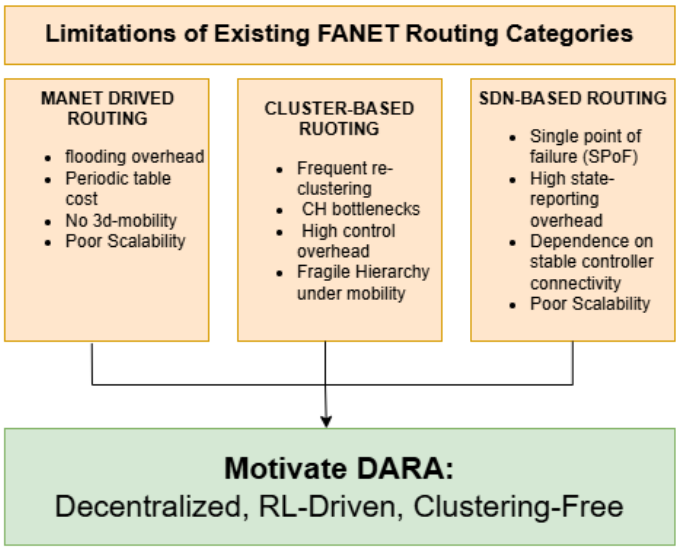


Fig. 2 Limitations of the three major categories of FANET routing approaches and DARA motivation

of existing FANET routing approaches and provides a scalable, resilient solution for reliable multi-hop communication in dynamic aerial networks.

3 Network Model and Assumptions

The proposed DARA protocol operates within a hierarchical FANET composed of two aerial layers and a terrestrial control station, as illustrated in Fig. 3. This architecture represents a realistic emergency-response or post-disaster scenario in which UAVs collaborate to collect and transmit environmental or situational data to a remote command center.

- **Low Altitude Platform (LAP) UAVs:** These lightweight UAVs operate at low altitudes and are responsible for *surveillance and data acquisition* over specific geographical zones. Each LAP collects monitoring data and generates packets to be delivered toward upper-layer UAVs. The LAP layer forms a dynamic ad-hoc network characterized by high mobility, limited transmission range, limited resources,and frequent topology changes.
- **High Altitude Platform (HAP) UAVs:** These are powerful, energy-rich, and highly mobile drones operating at higher altitudes. Their primary role is to *aggregate data* received from multiple LAP UAVs and forward it to the *Disaster Management Station (DMS)*. Owing to their extended communication range and stable flight, HAP UAVs serve as aerial relays that enhance coverage and reliability.
- **Disaster Management Station (DMS):** A fixed ground-based unit responsible for receiving, analyzing, and storing aggregated information. It serves as the final data sink and coordinates mission-level decisions for emergency response.

Table 1 Comparison of major FANET routing categories and their inherent limitations.

Aspect	MANET-Derived Routing	Clustering-Based Routing	SDN-Based Routing
Control Overhead	Flooding (AODV/DSR) and periodic table updates (DSDV/OLSR) generate high overhead under high mobility.	Requires frequent re-clustering; cluster-head (CH) exchanges produce significant control burden.	Continuous global-state reporting to controller causes heavy signaling overhead.
Scalability	Routing tables grow with swarm size; control traffic saturates the channel.	CH congestion increases with network size; cluster maintenance becomes unstable.	Controller becomes a bottleneck; scaling to large swarms increases delay and failures.
Mobility Handling	Limited mobility prediction; no 3D movement modeling; routes break faster than they can be repaired.	Clusters collapse when CH moves or dies; re-clustering too slow for fast 3D movement.	Centralized decisions quickly become outdated under rapid topology changes.
Reliability	Frequent link failures cause repeated route discoveries and packet loss.	Reliability depends on CH stability; CH failure disrupts the cluster.	Single point of failure (controller); loss of controller link breaks routing.
Energy Efficiency	No energy balancing; nodes near GBS exhaust quickly.	CH drains energy faster; unfair load distribution.	UAVs expend energy sending frequent updates to controller.
Architecture Nature	Fully distributed but reactive/proactive, not predictive.	Hierarchical, fragile structure requiring global coordination.	Centralized; violates autonomy requirement for many UAV missions.
Suitability for High-Dynamic FANETs	slow route updates and high overhead.	unstable clusters under mobility.	controller latency and dependency.

3.1 HAP-LAP distance estimation

We assume that each LAP UAV is equipped with a GPS and can determine its own position $pos_i = (x_i, y_i, z_i)$. The positions of HAP nodes pos_{HAP_k} are either preconfigured (quasi-static deployment) or periodically announced through HAP beacons. Accordingly, each UAV computes its geometric distance to the nearest HAP as

$$D_{i,\mathrm{HAP}} = \begin{cases} \min\limits_{k} \left\| \mathbf{pos}_i - \mathbf{pos}_{\mathrm{HAP}_k} \right\|, & \text{if } \min_k \left\| \mathbf{pos}_i - \mathbf{pos}_{\mathrm{HAP}_k} \right\| < r, \\ \infty, & \text{otherwise.} \end{cases} \tag{1}$$

Each LAP UAV has a fixed wireless communication range r. If the nearest-HAP distance satisfies $D_{i,\mathrm{HAP}} < r$, then the UAV can establish a direct link to the HAP and may act as a potential relay for other LAPs. If $D_{i,\mathrm{HAP}} \geq r$, direct communication with the HAP is not feasible; in this case, the distance is defined as ∞, indicating that the UAV must rely on multi-hop forwarding through neighboring UAVs.

This formulation enables DARA to clearly distinguish between nodes that can directly reach a HAP and those that must route their traffic through intermediate relays.

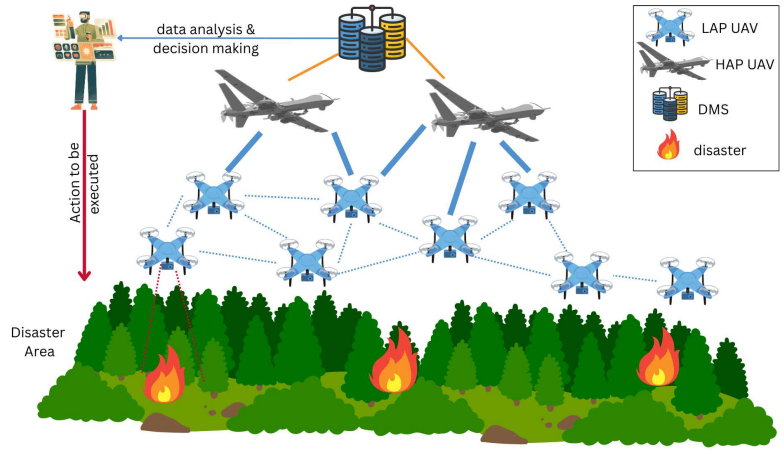


Fig. 3 Hierarchical FANET topology with Low Altitude (LAP) and High Altitude (HAP) UAVs cooperating to deliver data to the Disaster Management Station (DMS).

3.2 Enhanced Link Quality Model

To obtain a more robust estimate of the link quality between UAV_i and neighbor UAV_j, we combine three complementary indicators: (i) received signal strength (RSSI), (ii) packet reception ratio (PRR), and (iii) a stability factor derived from mobility. This multi-parameter approach aligns with hybrid estimators commonly proposed for dynamic UAV networks, integrating signal strength, reliability, and mobility awareness to improve route stability and performance in high-mobility scenarios [39–41].

(i) RSSI-based component. For each received beacon from UAV_j, UAV_i measures the instantaneous RSSI value $\mathrm{RSSI}_{ij}(t)$ and normalizes it into $[0,1]$ using min–max scaling:

$$\widetilde{\mathrm{RSSI}}_{ij}(t) = \frac{\mathrm{RSSI}_{ij}(t) - \mathrm{RSSI}_{\min}}{\mathrm{RSSI}_{\max} - \mathrm{RSSI}_{\min}}, \tag{2}$$

where $\mathrm{RSSI}_{\min}$ and $\mathrm{RSSI}_{\max}$ denote the minimum and maximum expected RSSI values under the given PHY/MAC configuration [39, 42].

(ii) PRR-based component. In addition to RSSI, we track the empirical packet reception ratio (PRR) over the link (UAV_i, UAV_j). Let $a_{ij}(t) = 1$ if a unicast packet transmitted from i to j at time t is successfully acknowledged, and $a_{ij}(t) = 0$ otherwise. We maintain an exponentially weighted moving average:

$$\widehat{\mathrm{PRR}}_{ij}(t) = (1-\nu)\,\widehat{\mathrm{PRR}}_{ij}(t-1) + \nu\, a_{ij}(t), \tag{3}$$

where $\nu \in [0,1]$ is a smoothing factor. This term captures the effective reliability of the link at the MAC layer, implicitly integrating the combined effects of fading, interference, and contention. EWMA-based PRR estimation is widely adopted in dynamic wireless networks, including adaptations for FANETs [43].

(iii) Mobility-based stability component. To account for the impact of relative mobility on link persistence, we introduce a stability factor based on the estimated

link expiration time (LET). Let v_{ij} denote the magnitude of the relative velocity vector between UAV_i and UAV_j, and let $\widehat{\mathrm{LET}}_{ij}(t)$ denote the predicted remaining lifetime of the link (computed from current 3D distance, relative velocity, and geometry, as in position-based FANET protocols). We normalize $\widehat{\mathrm{LET}}_{ij}(t)$ into $[0,1]$ as

$$\widetilde{\mathrm{LET}}_{ij}(t) = \min\left(1,\ \frac{\widehat{\mathrm{LET}}_{ij}(t)}{\mathrm{LET}_{\max}}\right), \tag{4}$$

where $\mathrm{LET}_{\max}$ is a design parameter representing the upper bound on useful link lifetime. Higher values of $\widetilde{\mathrm{LET}}_{ij}(t)$ indicate more stable links, typically corresponding to low relative speed and favorable geometry (LET is a key metric in many FANET routing enhancements for predicting link lifetime and improving stability under high mobility [40, 41]).

(iv) Aggregated link quality. The overall link quality $LQ_{ij}(t)$ used by DARA is then obtained as a weighted combination of the three components:

$$LQ_{ij}(t) = w_R\,\widetilde{\mathrm{RSSI}}_{ij}(t) + w_P\,\widehat{\mathrm{PRR}}_{ij}(t) + w_S\,\widetilde{\mathrm{LET}}_{ij}(t), \tag{5}$$

with $w_R, w_P, w_S \geq 0$ and $w_R + w_P + w_S = 1$. In practice, w_P and w_S can be assigned slightly larger values to emphasize effective reliability and stability, while w_R captures fast channel variations. The resulting $LQ_{ij}(t) \in [0,1]$ is used both to filter candidate neighbors (e.g., $LQ_{ij}(t) \geq \tau_{LQ}$) and as an input feature to the Q-Learning state and reward functions.

The parameter τ_{LQ} denotes the minimum link-quality threshold required for a neighbor to be considered as a valid next-hop candidate. At each control epoch, UAV i constructs its candidate set as

$$\mathcal{N}_i(t) = \{j \in N_i(t) : LQ_{ij}(t) \geq \tau_{LQ}\}. \tag{6}$$

This filtering step removes highly unreliable or short-lived links, reduces the action space of the Q-Learning agent, and accelerates convergence toward stable forwarding policies.

4 Operational Stages of DARA

This section explains the three core phases that define how the DARA protocol operates at each UAV node:

1. Stage 1: Periodic Beacon Emission and Neighbor-Table Maintenance (Algorithm 1),
2. Stage 2: T DARA Control Epoch for Next-Hop Selection and Learning (Algorithm 2 and Algorithm 3),
3. Stage 3: Relay Behavior at the Receiving Next Hop (Algorithm 4).

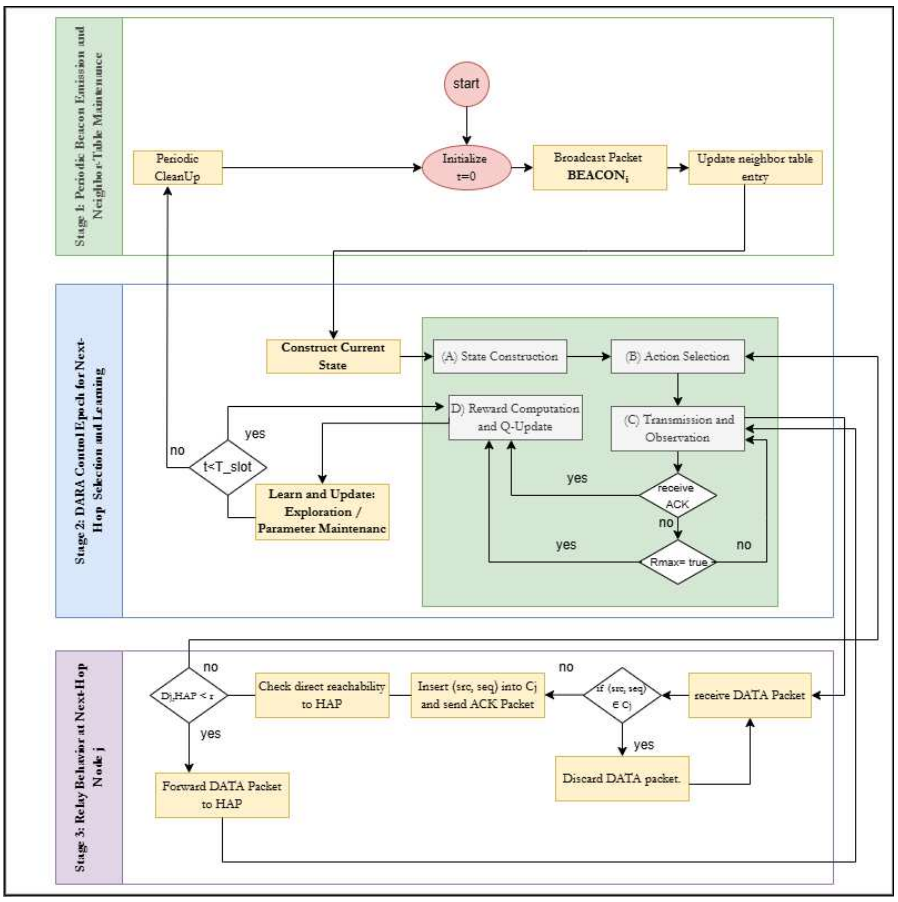


Fig. 4 Overall workflow of the proposed DARA protocol

These complementary stages operates to enable autonomous next-hop decision, continuous learning, and reliable multi-hop forwarding in highly dynamic FANET environments. Figure 4 presents an overview of the DARA workflow.

4.1 Stage 1: Periodic Beacon Emission and Neighbor-Table Maintenance

A beacon message with crucial local data, including node ID, role (LAP or HAP), position, velocity, residual energy, and timestamp, is periodically broadcast by each UAV during the initial stage. By calculating the relative distance to the destination (HAP), assessing link quality, and documenting buffer occupancy and mobility characteristics, neighboring UAVs utilize these beacons to update their local neighbor databases. To guarantee that every UAV has a current and accurate image of its communication neighborhood, entries that are not updated within the expiry window T_{expire} are eliminated. This stage's specific operations are summed up in Algorithm 1.

Algorithm 1 Periodic Beacon Emission and Neighbor-Table Maintenance (Node i)

Inputs: Communication range r; HAP positions $\{pos_{\mathrm{HAP}_k}\}$; beacon period T_{beacon}; expiry timer T_{expire}.

Initialize: Start periodic timer with period T_{beacon}; initialize empty neighbor table $N[\cdot]$.

OnTimer (every T_{beacon}):

Obtain current position pos_i and velocity vel_i; read residual energy $E_{\mathrm{res},i}$ and buffer level B_i.

Compute geometric distance to nearest HAP:

$$D^{\mathrm{geo}}_{i,\mathrm{HAP}} \leftarrow \min_k \|pos_i - pos_{\mathrm{HAP}_k}\|.$$

Set

$$D_{i,\mathrm{HAP}} \leftarrow \begin{cases} D^{\mathrm{geo}}_{i,\mathrm{HAP}}, & \text{if } D^{\mathrm{geo}}_{i,\mathrm{HAP}} < r, \\ \infty, & \text{otherwise.} \end{cases}$$

Construct beacon packet

$$BEACON_i \leftarrow \langle i,\, role_i,\, pos_i,\, vel_i,\, E_{\mathrm{res},i},\, B_i,\, t_{\mathrm{now}} \rangle.$$

Broadcast $BEACON_i$.

OnReceive $BEACON_j$ **from** neighbor j:

Estimate link quality LQ_{ij} (based on RSSI).

Compute geometric distance from j to nearest HAP:

$$D^{\mathrm{geo}}_{j,\mathrm{HAP}} \leftarrow \min_k \|pos_j - pos_{\mathrm{HAP}_k}\|.$$

Set

$$D_{j,\mathrm{HAP}} \leftarrow \begin{cases} D^{\mathrm{geo}}_{j,\mathrm{HAP}}, & \text{if } D^{\mathrm{geo}}_{j,\mathrm{HAP}} < r, \\ \infty, & \text{otherwise.} \end{cases}$$

Update neighbor table entry:

$$N[j] \leftarrow \{\text{last_seen} = t_{\mathrm{now}},\, pos_j,\, vel_j,\, E_{\mathrm{res},j},\, LQ_{ij},\, D_{j,\mathrm{HAP}},\, B_j,\, \text{expiry} = t_{\mathrm{now}} + T_{\mathrm{expire}}\}.$$

Periodic Cleanup (every T_{beacon}):

for all $j \in N$ **do**

  if $t_{\mathrm{now}} > N[j].\text{expiry}$ **then**

    remove $N[j]$

4.2 Stage 2: DARA Control Epoch for Next-Hop Selection and Learning

At every control epoch estimates with the duration T_{slot}, each UAV agent executes the core routing and learning operations of DARA. This stage determines how the agent selects the next-hop relay node and updates its forwarding policy through reinforcement learning. Three sequential steps comprise the control epoch:

- (i) selection of neighbor-based candidates based on link quality and progress constraints
- (ii) a core decision-making and learning block composed of state construction, action selection, transmission and feedback observation, reward computation, and Q-value update,
- (iii) exploration and maintenance of parameters to ensure long-term policy convergence. Together, these steps allow each UAV to autonomously adapt its routing behavior in response to mobility, link variability, congestion, and energy conditions.

4.2.1 Neighbor-Based Candidate Selection

At the beginning of each control epoch of duration T_{slot}, node i evaluates its local neighborhood based on the most recent entries in its neighbor table. The goal of this step is to identify the subset of neighbors that are viable next-hop candidates according to link-quality and forwarding-progress constraints.

A neighbor j is selected as a candidate if (i) the estimated link quality $LQ_{ij}(t)$ exceeds the minimum threshold τ_{LQ}, and (ii) the neighbor provides forward progress toward the destination (HAP) . Formally, the candidate set of node i is defined as:

$$\mathcal{N}_i(t) = \{\, j \in N_i(t) : LQ_{ij}(t) \geq \tau_{LQ} \wedge D_{j,\text{HAP}} < D_{i,\text{HAP}} \}. \tag{7}$$

If $\mathcal{N}_i(t)$ is empty (no candidate can be recovered) then the node buffers the packet and waits for the next epoch. This selection step is summarized in Algorithm 2.

4.2.2 Core Decision-Making and Learning

Once the candidate neighbor set $\mathcal{N}_i(t)$ has been constructed, node i executes the core RL operations that determine which neighbor is selected as next hop and how the corresponding forwarding policy is updated. This block consists of five steps: (A) state construction, (B) action selection, (C) transmission and observation, (D) reward computation, and (E) Q-value update. The complete procedure is detailed in Algorithm 3.

Agent Model:

As illustrated in figure 5. Each UAV is modeled as a RL agent defined by the tuple:

$$\mathcal{A} = (S, A, R, \pi)$$

where:

- S is the *state space*, representing local network conditions.
- A is the *action space*, representing the set of reachable neighboring UAVs.
- R is the *reward function*, representing the benefit of forwarding a packet through a specific neighbor.
- π is the *policy*, which defines how actions are selected given the observed state.

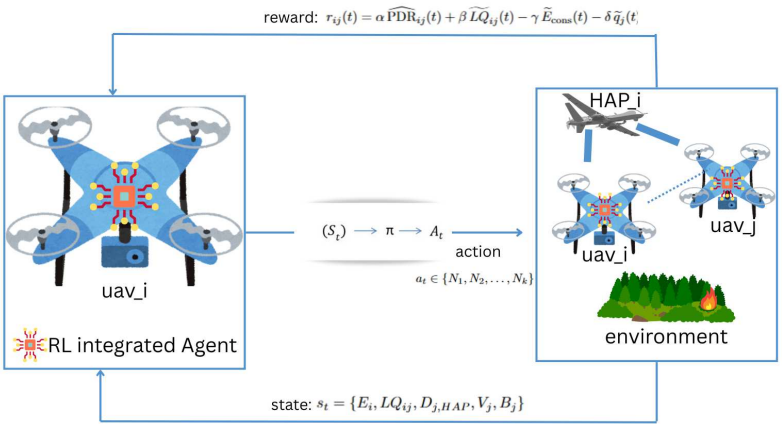


Fig. 5 Core Decision-Making and Learning process

(A)State Construction

The state s_t of UAV i at time t is defined as:

$$s_t = \{E_i, LQ_{ij}, D_{j,HAP}, V_j, B_j\} \tag{8}$$

where E_i denotes the residual energy of UAV i, LQ_{ij} represents the link quality between UAV i and neighbor j, $D_{j,GBS}$ is the distance between neighbor j and the destination (HAP), V_j indicates the relative velocity of neighbor j, and B_j denotes its buffer occupancy.

(B) Action Selection

The action a_t corresponds to selecting one of the available neighboring UAVs as the next hop for packet forwarding:

$$a_t \in \{UAV_1, UAV_2, \ldots, UAV_k\}, \tag{9}$$

where k denotes the number of neighboring UAVs within the communication range of node UAV_i, and $j \in \{1, \ldots, k\}$ with $j \neq i$.

After selecting neighbor UAV_j, node UAV_i dequeues a data packet from its transmission queue TXQ, decrements the packet's time-to-live (TTL) value, and forwards the packet to UAV_j. An acknowledgment (ACK) timer of duration T_{ack} is then initiated to monitor the transmission outcome.

If an ACK is received from UAV_j:

- the empirical delivery indicator $s_{ij}(t)$ is updated (1 for success, 0 otherwise),
- the moving-average PDR estimate $\widehat{\text{PDR}}_{ij}(t)$ is refreshed,
- the instantaneous link-quality sample $\widetilde{LQ}_{ij}(t)$ is recorded,
- the experienced delay $d_{ij}(t)$ is measured,
- the transmission energy $E_{\text{cons}}(t)$ is computed from the PHY model.

If no ACK is received, node i attempts up to $R_{\max}$ retransmissions. Persistent failure sets $I_{\text{succ}} = 0$ and marks the link (i, j) as degraded in the neighbor table. Thus, Step (C) provides all feedback necessary for reward computation and policy refinement.

(D) Reward computation

In DARA, each UAV agent learns a next-hop forwarding strategy by interacting with the network environment and receiving a scalar reward after every forwarding decision. The reward function is designed to align the local actions of each UAV with the global performance objectives of the FANET, namely high packet-delivery reliability, low end-to-end delay, and energy-efficient operation.

When a UAV_i forwards a packet to a neighbor UAV_j at time t, it receives an immediate reward

$$r_{ij}(t) = \alpha\, \widehat{\text{PDR}}_{ij}(t) + \beta\, \widetilde{LQ}_{ij}(t) - \gamma\, \widetilde{E}_{\text{cons}}(t) - \delta\, \widetilde{q}_j(t), \tag{10}$$

where α, β, γ, and δ are non-negative weighting coefficients that tune the influence of each performance dimension.

Reliability Component: The term $\widehat{\text{PDR}}_{ij}(t) \in [0, 1]$ denotes a *local estimate* of the packet-delivery ratio (PDR) when selecting neighbor j as next hop. It is updated online using an exponentially weighted moving average:

$$\widehat{\text{PDR}}_{ij}(t) = (1 - \lambda)\, \widehat{\text{PDR}}_{ij}(t-1) + \lambda\, s_{ij}(t), \tag{11}$$

where $s_{ij}(t) = 1$ if the packet forwarded from i to j is ultimately acknowledged by a HAP/GBS, and $s_{ij}(t) = 0$ otherwise. The factor $\lambda \in (0, 1]$ controls the smoothing. This component encourages the agent to prefer neighbors that provide high end-to-end reliability.

Link-Quality Component: The normalized link-quality metric $\widetilde{LQ}_{ij}(t) \in [0, 1]$ represents the instantaneous condition of the wireless link between i and j, typically

derived from RSSI, SINR, packet error rate, or multi-metric estimators. Raw link-quality values are mapped to $[0, 1]$ using min–max normalization. This component promotes routing choices over stable and high-quality links.

Energy-Consumption Component: The normalized energy consumption $\widetilde{E}_{\text{cons}}(t) \in [0, 1]$ measures the transmission energy required for the current packet:

$$\widetilde{E}_{\text{cons}}(t) = \frac{E_{\text{cons}}(t)}{E_{\max}}, \tag{12}$$

where $E_{\text{cons}}(t)$ is the instantaneous energy cost and $E_{\max}$ is the maximum expected per-packet transmission cost. The penalty term $-\gamma\widetilde{E}_{\text{cons}}(t)$ discourages energy-intensive forwarding actions, promoting a balanced energy usage across UAVs, which leads to improved network lifetime (FND and LND).

Congestion Component: The normalized queue length $\widetilde{q}_j(t) \in [0, 1]$ denotes the current buffer occupancy at neighbor j, computed as $\widetilde{q}_j(t) = q_j(t)/q_{\max}$, where $q_{\max}$ denotes the maximum buffer size. The penalty $-\delta\,\widetilde{q}_j(t)$ prevents forwarding through congested UAVs, thus reducing queuing delay, minimizing packet drops under bursty traffic, and improving overall end-to-end delay.

By appropriately tuning $(\alpha, \beta, \gamma, \delta)$, the reward function jointly favors (i) reliable transmissions, (ii) stable links, (iii) energy-aware relaying, and (iv) congestion avoidance. This multi-factor formulation allows DARA to learn forwarding policies that significantly outperform traditional routing (AODV) and clustering-based schemes (EMASS, ESOFCluster) across all considered FANET scenarios in terms of PDR, delay, energy consumption, and routing overhead.

Q-Value Update and TTL Handling

Each UAV updates its Q-table according to the standard Q-learning rule [**?**]:

$$Q(s_t, a_t) \leftarrow (1 - \eta)Q(s_t, a_t) + \eta[r_t + \gamma \max_{a'} Q(s_{t+1}, a')]$$

where η is the learning rate, γ is the discount factor, and $\max_{a'} Q(s_{t+1}, a')$ represents the estimated future reward of the next state.

An ϵ-greedy policy is employed to balance exploration and exploitation: with probability ϵ, the UAV selects a random neighbor (exploration); otherwise, it selects the neighbor with the highest Q-value (exploitation). Through repeated interaction, the UAV converges to an optimal policy π^* that maximizes cumulative reward and minimizes transmission delay and energy consumption.

If the packet's TTL expires ($TTL \leq 0$), it is discarded and a penalty $-\mu_{\text{pen}}$ is applied. The updated state s_{t+1} is obtained after the next beacon refresh or an immediate re-sensing of the environment.

This iterative update drives convergence toward the optimal forwarding policy π^*. If the packet's TTL expires ($TTL \leq 0$), it is discarded and a penalty $-\mu_{\text{pen}}$ is applied. The next state s_{t+1} is obtained after the next beacon update or an immediate re-sensing of the environment.

This feedback-driven learning loop enables each UAV to adapt its forwarding strategy autonomously under rapid topology changes, mobility fluctuations, and varying traffic demands.

Algorithm 2 DARA Control Epoch at Node i (Stage 2)

Inputs: Slot duration T_{slot}, ACK timeout T_{ack}, link-quality threshold τ_{LQ}, distance slack Δ, maximum retransmissions $R_{\max}$, exploration decay λ, minimum exploration rate $\epsilon_{\min}$.
Initialize: Q-table $Q(s,a) \leftarrow 0$; exploration rate $\epsilon \in (0,1]$; learning rate η; discount factor γ.
while node i is active **do**

(Neighbor-Based Candidate Selection)
Build candidate set

$$\mathcal{N}_i \leftarrow \{j \in N : LQ_{ij} \geq \tau_{LQ} \wedge D_{j,\text{HAP}} < D_{i,\text{HAP}}\}.$$

if $\mathcal{N}_i = \emptyset$ **then** ▷ no viable neighbor
  Attempt fallback (e.g., increase P_{tx} or widen Δ once).
  if $\mathcal{N}_i$ still empty **then**
    Buffer data and **continue** to next T_{slot}.
  end if
end if

(Core Decision-Making and Learning)
Call

$$\text{DARA_CoreDecision}\big(i, \mathcal{N}_i, TXQ, Q, \epsilon, \eta, \gamma, T_{\text{ack}}, R_{\max}\big).$$

(Exploration / Parameter Maintenance)
$\epsilon \leftarrow \max(\epsilon_{\min}, \lambda \cdot \epsilon)$ ▷ exploration decay
Optionally adapt η and γ based on convergence; prune stale Q-entries.
end while
=0

4.3 Stage 3: Relay Behavior at the Receiving Next Hop

When a UAV j receives a data packet from a previous hop i, it performs a sequence of relay operations to ensure reliable and loop-free multi-hop forwarding toward the HAP. Unlike centralized or cluster-based approaches, DARA enables every UAV to operate both as a relay and as a learning agent, ensuring fully distributed and adaptive routing. The relay procedure consists of three main steps.

Algorithm 3 DARA_CoreDecision: Core Decision-Making at Node i

Inputs: Node i; candidate set $\mathcal{N}_i$; transmit queue TXQ; Q-table $Q(s,a)$; exploration rate ϵ; learning rate η; discount factor γ; ACK timeout T_{ack}; maximum retransmissions $R_{\max}$.

(A) State Construction

$E_i \leftarrow$ residual energy of node i.

For each $j \in \mathcal{N}_i$: collect LQ_{ij}, $D_{j,\text{HAP}}$, $\|vel_j\|$, and B_j.

Compose local state

$$s_t \leftarrow \phi\big(E_i, \{LQ_{ij}, D_{j,\text{HAP}}, \|vel_j\|, B_j\}_{j\in\mathcal{N}_i}\big).$$

(Optionally discretize or hash s_t.)

(B) Action Selection

if with probability ϵ **then**

  $a_t \sim \text{Uniform}(\mathcal{N}_i)$ ▷ exploration

else

  $a_t \leftarrow \arg\max_{j\in\mathcal{N}_i} Q(s_t, j)$ ▷ exploitation

end if

Apply tie-breaking preference: smaller $D_{j,\text{HAP}}$, then higher LQ_{ij}, then lower B_j.

(C) Transmission and Observation

Dequeue one packet from TXQ (or create a DATA packet if event-driven); set $TTL \leftarrow TTL - 1$.

Transmit $DATA_i$ to a_t and start ACK timer T_{ack}.

if ACK is received from a_t before T_{ack} **then**

  Observe link statistics: $\widehat{\text{PDR}}_{ij}$, average LQ_{ij}, delay d_{ij}.

  Compute energy consumed $E_{\text{cons}} \leftarrow \text{model}(P_{\text{tx}}, \text{pkt_len}, t_{\text{tx}})$.

  if ACK.status = DELIVERED **then**

    $I_{\text{succ}} \leftarrow 1$.

  else

    $I_{\text{succ}} \leftarrow 0$.

  end if

else ▷ timeout

  Retransmit up to $R_{\max}$ times; if all attempts fail:

  $I_{\text{succ}} \leftarrow 0$; mark link (i,j) as degraded.

end if

(D) Reward Computation and Q-Update

Compute immediate reward r_t according to the reward function in (10), using $\widehat{\text{PDR}}_{ij}$, $\widetilde{LQ}_{ij}$, $\widetilde{E}_{\text{cons}}$, and $\widetilde{q}_j$.

Observe next state s_{t+1} (after beacon update or re-sensing).

Update Q-value:

$$Q(s_t, a_t) \leftarrow (1-\eta)Q(s_t, a_t) + \eta\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right].$$

(E) Buffer and TTL Handling

if $TTL \leq 0$ **then**

  Drop the packet, apply penalty $-\mu_{\text{pen}}$, and log the event.

end if

(A) Duplicate Suppression.

Upon receiving a packet, node j first checks whether the packet, identified by the tuple $\langle src, seq \rangle$, has already been processed. If a match is found in the duplicate cache $\mathcal{C}_j$, the packet is discarded to avoid unnecessary retransmissions and to reduce routing overhead. Optionally, an acknowledgment may be re-sent to the previous hop to maintain consistency. If the packet is new, its identifier is cached and the packet is inserted into the local receive queue RXQ_j.

(B) Direct Forwarding to HAP (Reachability Check).

Node j then determines whether it can reach the nearest HAP directly. Using the distance-to-HAP metric $D_{j,\mathrm{HAP}}$, direct forwarding is performed if

$$D_{j,\mathrm{HAP}} < r,$$

where r denotes the communication range. If this condition holds, node j attempts a direct uplink transmission to the HAP. A successful transmission results in an acknowledgment of type `DELIVERED` being sent back to node i, allowing upstream nodes to update their reliability and Q-values accordingly. If the transmission fails, the uplink is marked as degraded, and the packet may be re-enqueued depending on the local buffer state.

(C) Multi-Hop Relay Forwarding.

If node j is not within range of a HAP ($D_{j,\mathrm{HAP}} \geq r$), the packet must be forwarded through additional relay nodes. In this case, the packet is inserted into the transmit queue TXQ_j, and an acknowledgment of type `FORWARDED` is sent back to the previous hop. The node then executes its own DARA control epoch (Algorithm 2) to select the next hop using the RL mechanisms described in Stage 2. This ensures that multi-hop forwarding remains adaptive, efficient, and consistent with the learned routing policy.

The full relay behavior logic is summarized in Algorithm 4.

For clarity, all variables and symbols appearing in the three DARA stages and the associated algorithms are summarized in Table 2.

5 Simulations results and Performance discussion

5.1 Simulation Environment and Parameters

5.1.1 Simulation Environment Architecture

To faithfully evaluate DARA in a realistic yet controllable environment, we adopt a co-simulation architecture that couples NS-3 with a Python-based reinforcement learning (RL) stack. Our design follows the principles of *ns3-gym* [44] and recent RL-in-NS3 frameworks such as RateRL [45], as well as the ns-3 AI integration guidelines described in [46].

In this architecture, NS-3 is responsible for all packet-level networking functions (mobility, PHY/MAC, queuing, and routing hooks), while Python hosts the Q-Learning agents that implement DARA's decision logic:

Algorithm 4 Relay Behavior at Next-Hop Node j

Inputs: Node j; duplicate cache $\mathcal{C}_j$; receive queue RXQ_j; transmit queue TXQ_j; distance $D_{j,\mathrm{HAP}}$; communication range r.

OnReceive DATA$\langle src, seq, payload, TTL, \ldots\rangle$ **from** node i:
  (A) Duplicate suppression
if $\langle src, seq\rangle \in \mathcal{C}_j$ **then**
  Discard DATA packet.
  return
end if
  Insert $\langle src, seq\rangle$ into $\mathcal{C}_j$.
  Append packet to RXQ_j.

  (B) Check direct reachability to HAP
if $D_{j,\mathrm{HAP}} < r$ **then** ▷ node j can reach a HAP directly
  Attempt direct transmission of DATA to HAP.
  if uplink delivery succeeds **then**
    forward DATA Packet to node HAP..
    Algorithm 2-C.
    Algorithm 2-D.
    Algorithm 2-E.
  end if
else ▷ multi-hop forwarding required
    (C) Multi-hop relay behavior
  Enqueue DATA packet into TXQ_j.
  Send ACK$\langle src, seq, \mathrm{FORWARDED}\rangle$ to node i.
  Trigger a local DARA control epoch at node j (Algorithm 2) to select its own next hop and continue multi-hop forwarding toward the HAP.
end if

- **NS-3 core:**
  - simulates the wireless channel, IEEE 802.11ac PHY/MAC, packet queues, and link-layer retransmissions;
  - maintains node positions and velocities according to the chosen 3D mobility model;
  - exposes per-node observations (e.g., link-quality indicators, queue length, residual energy) and accepts next-hop actions
- **ns3-gym :**
  - defines the *observation space* and *action space* for each UAV in a way similar to [44];
  - synchronizes discrete decision epochs between NS-3 and the external RL code, following the time-stepped interaction pattern recommended in [46];

Table 2 Summary of symbols and notation used in the DARA algorithms.

Symbol	Definition
N_i	Set of neighbors of node i (neighbor table).
$\mathcal{N}_i(t)$	Candidate next-hop neighbors of node i at time t.
$pos_i = (x_i, y_i, z_i)$	3D position of UAV i obtained from GNSS.
vel_i	Velocity vector of node i.
$V_j = \|vel_j\|$	Speed (mobility indicator) of neighbor j.
E_i	Residual energy of UAV i.
$E_{\text{res},i}$	Remaining battery of node i (same as E_i).
$E_{\text{cons}}(t)$	Energy consumed to transmit at epoch t.
$\widetilde{E}_{\text{cons}}(t)$	Normalized transmission energy.
$LQ_{ij}(t)$	Estimated link quality between nodes i and j.
$\widehat{LQ}_{ij}(t)$	Normalized link quality in $[0, 1]$.
τ_{LQ}	Minimum acceptable link-quality threshold.
$D_{i,\text{HAP}}$	Distance from node i to nearest HAP (or ∞ if $> r$).
r	Communication range of LAP UAVs.
Δ	Distance slack for forwarding progress constraint.
B_j	Buffer occupancy (queue length) at neighbor j.
$\widetilde{q}_j(t)$	Normalized buffer level at node j.
$q_{\max}$	Maximum queue size.
TXQ_i	Transmit queue of node i.
RXQ_i	Receive queue of node i.
$\mathcal{C}_i$	Duplicate packet cache at node i.
TTL	Time-To-Live (hop limit) of a packet.
s_t	Current state of UAV i at time t.
a_t	Selected next-hop action at time t.
$Q(s_t, a_t)$	Q-value for state–action pair (s_t, a_t).
η	Learning rate in Q-learning.
γ	Discount factor for future rewards.
ϵ	Exploration probability (in ϵ-greedy policy).
λ	Exploration decay coefficient.
$\epsilon_{\min}$	Minimum exploration probability after decay.
P_{tx}	Transmission power used for sending DATA packets.
T_{ack}	ACK timeout duration.
$R_{\max}$	Maximum number of retransmissions.
$\widehat{\text{PDR}}_{ij}(t)$	Local estimate of packet delivery ratio for link (i, j).
$s_{ij}(t)$	Binary ACK success indicator for link (i, j).
λ (EWMA)	Smoothing factor for updating $\widehat{\text{PDR}}_{ij}$.
$r_{ij}(t)$	Immediate reward obtained when forwarding to neighbor j.
$\alpha, \beta, \gamma, \delta$	Reward weighting coefficients.
μ_{pen}	Penalty for delivery failure or TTL expiration.
t_{now}	Current logical time at node i.
T_{slot}	Duration of the DARA control epoch.
T_{beacon}	Period of beacon emission.
T_{expire}	Neighbor-table entry expiration timeout.

  - exchanges state, action, and reward messages between NS-3 and Python processes using a lightweight ZERO MQ IPC mechanism.

- **Python RL layer:**

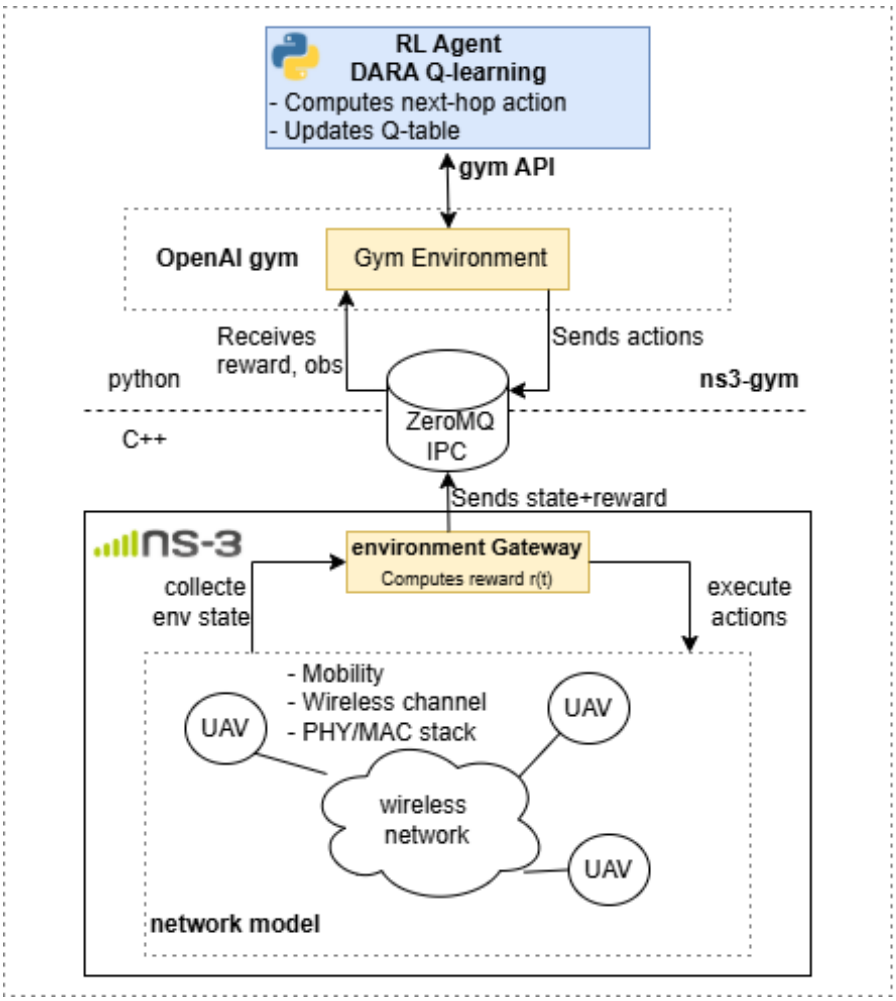


Fig. 6 Overall architecture of the proposed reinforcement-learning–enabled simulation environment integrating ns-3, ns3-gym, and the Python-based DARA agent.

– runs one independent Q-Learning agent per UAV, similar in spirit to the per-link agents used in RateRL for rate adaptation [44], but here specialized for next-hop routing decisions;
– implements DARA's pipeline: state construction, ϵ-greedy action selection, reward computation (based on PDR, link quality, energy and queue occupancy), and Q-table updates;
– sends the selected next hop back to NS-3 at each control epoch T_{slot}, where it is applied as the forwarding decision for the corresponding packet.

Conceptually, the whole system can be viewed as a multi-agent reinforcement-learning testbed for FANET routing: NS-3 provides a high-fidelity wireless networking environment, and the Python RL layer provides flexible experimentation with learning

algorithms and reward designs, as advocated in [44, 45]. DARA is thus implemented as a distributed set of agents. Figure 6 depicts the complete simulation workflow combining the ns-3 network model, the ns3-gym environment gateway, shared-memory IPC, and the external Python RL agent responsible for executing the DARA logic.

5.1.2 Simulation Parameters and scenarios

For the performance evaluation we consider three simulations scenarios over a 2 km× 2 km area:

1. **Scenario 1- Static load, moderate mobility**: $N = 30$ LAPs, 2 HAPs, speeds 10–15 m/s.
2. **Scenario 3-Bursty load, high mobility**: $N = 40$ LAPs, 2 HAPs, speeds 15–25 m/s, Poisson traffic bursts.
3. **Scenario 4-Stress test (density)**: $N = 50$ LAPs, 3 HAPs, speeds 15–20 m/s.

Each simulation runs for 600 s and is repeated with 10 independent random seeds. Results report mean values and 95% confidence intervals. The performance of DARA is compared with:

- AODV (reactive MANET protocol) [13],
- EMASS (heuristic clustering-based routing) [17],
- ESOFCluster (fuzzy-logic-based clustering) [18],

all implemented and reproduced from their referenced specifications.

Table 3 summarizes the key parameters used in the evaluation.

5.2 Performance Evaluation Metrics

To assess the effectiveness of the proposed DARA routing framework and to compare it with classical and state-of-the-art FANET routing schemes, including AODV, EMASS, and ESOFCluster, several widely adopted performance metrics are evaluated. These metrics capture routing reliability, latency, energy efficiency, control overhead, and learning behavior in highly dynamic aerial environments.

5.2.1 Packet Delivery Ratio (PDR)

Packet Delivery Ratio (PDR) is defined as the ratio of the total number of data packets successfully received at the destination (HAP/HAP) to the total number of packets generated by the UAV sources:

$$\text{PDR} = \frac{N_{\text{recv}}}{N_{\text{sent}}}. \tag{13}$$

A higher PDR indicates a more reliable routing protocol capable of maintaining stable communication under high mobility and frequent topology changes, which are typical characteristics of FANETs.

Figure 7 PDR achieved by DARA, AODV, EMASS, and ESOFCluster under the three evaluated FANET scenarios. The results highlight the impact of mobility intensity, traffic dynamics, and network density on end-to-end delivery reliability.

Table 3 Simulation parameters for evaluating DARA in NS-3.

Parameter	Value
Simulation area	2000 m × 2000 m
Number of LAP UAVs	30, 40, 50
Number of HAP UAVs	2, 3
Mobility model	3D Gauss–Markov
LAP speed	10–25 m/s
Altitude range	100–250 m
Initial energy per UAV	$E_0 = 20{,}000\ J$
Energy model	ns-3 BasicEnergySource
Wireless standard	IEEE 802.11ac
Channel model	Nakagami + Friis
Transmission range	300 m (LAP–LAP), 500 m (LAP–HAP)
Traffic model	CBR (5 pkt/s), Poisson bursty
Packet size	512 bytes
Simulation time	600 s
Routing baselines	AODV, EMASS, ESOFCluster
RL algorithm	Q-Learning (distributed)
Learning rate η	0.1
Discount factor γ	0.9
Exploration decay λ	0.995
Control epoch T_{slot}	0.5 s
Reward weights	Tuned by grid search
Random seeds	10 (95% CI)

In **Scenario 1** , all protocols achieve relatively high PDR values due to the limited topology variations. Nevertheless, DARA consistently outperforms the baseline approaches, demonstrating that distributed RL enables more reliable next-hop selection even under moderate network dynamics.

In **Scenario 2**, the packet delivery ratio (PDR) decreases for all evaluated protocols due to recurrent link failures and variations in traffic load. This decline is especially noticeable for AODV, as its flooding-based route discovery mechanism cannot efficiently cope with rapid topology variations. Clustering approaches such as EMASS and ESOFCluster perform better than AODV; however, their performance is still impacted by cluster instability and the overhead generated by frequent cluster maintenance. In comparison, DARA sustains a considerably higher PDR by dynamically adjusting its forwarding decisions using only locally available information.

In **Scenario 3**, increased node density intensifies channel contention and interference, further complicating reliable routing. Although hierarchical clustering mechanisms offer certain scalability advantages, the concentration of traffic at cluster heads and the rise in control overhead constrain the performance of EMASS and ESOFCluster. DARA once more delivers the best PDR, highlighting its ability to operate effectively in dense and highly dynamic FANET environments.

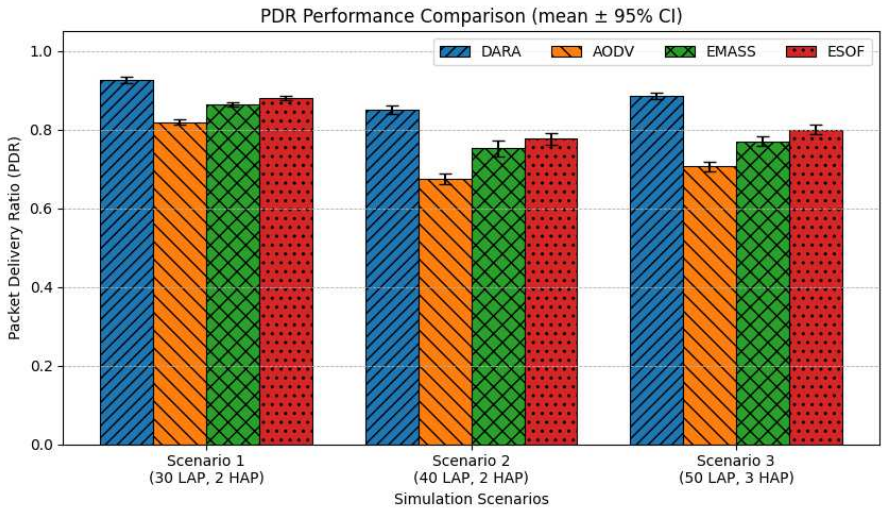


Fig. 7 Packet Delivery Ratio (PDR) comparison of DARA, AODV, EMASS, and ESOFCluster under the three FANET scenarios (mean ± 95% confidence interval).

Overall, the simulation results indicate that DARA consistently outperforms the other protocols in terms of packet delivery. The improvement becomes more evident under conditions of high mobility and elevated node density, demonstrating the strength of a fully distributed, clustering-independent reinforcement learning strategy for FANET routing.

5.2.2 End-to-End Delay

End-to-end delay refers to the mean time a data packet takes to reach the destination UAV after being transmitted by the source. This metric accounts for all intermediate delays encountered along the path, including buffering time in queues, transmission and propagation delays, processing time at each relay node, and any additional delay caused by retransmissions. The average delay is calculated as follows:

$$D_{\text{avg}} = \frac{1}{N_{\text{recv}}} \sum_{i=1}^{N_{\text{recv}}} \left(t_i^{\text{recv}} - t_i^{\text{send}} \right). \tag{14}$$

Minimizing end-to-end delay is particularly important for delay-sensitive FANET scenarios, such as aerial surveillance, emergency and disaster operations, and real-time monitoring tasks, where timely data delivery is critical.

Figure 8 presents the average end-to-end delay achieved by DARA, AODV, EMASS, and ESOFCluster under the three considered FANET scenarios, expressed as the mean delay with 95% confidence intervals over multiple simulation runs. End-to-end delay is a critical performance indicator for FANET applications that require timely data delivery, such as disaster monitoring, surveillance, and real-time coordination.

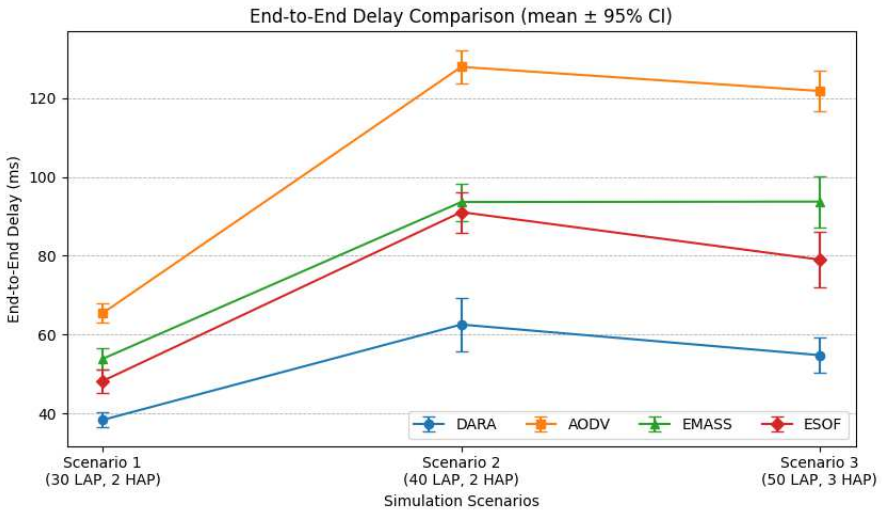


Fig. 8 Average end-to-end delay comparison of DARA, AODV, EMASS, and ESOFCluster under the three FANET scenarios (mean ± 95% confidence interval).

In **Scenario 1** , all evaluated protocols exhibit relatively low delay due to stable connectivity and limited contention. Nevertheless, DARA consistently achieves the lowest average delay among all approaches. This improvement is attributed to its localized next-hop decision process, which avoids route discovery latency and reduces queuing delay by selecting reliable and lightly loaded neighbors.

In **Scenario 2** , the average delay increases significantly for all protocols as a result of frequent link disruptions and traffic bursts. AODV experiences the highest delay, mainly due to repeated flooding-based route discovery and buffering during route reconstruction. Clustering-based protocols (EMASS and ESOFCluster) reduce delay compared to AODV, but their performance is still affected by cluster maintenance overhead and congestion at cluster heads. In contrast, DARA maintains substantially lower delay by continuously adapting its forwarding decisions based on real-time local observations, without relying on cluster reformation or centralized control.

In **Scenario 3** , network contention and interference further exacerbate delay. While hierarchical clustering provides partial scalability benefits, cluster-head bottlenecks limit the performance of EMASS and ESOFCluster. DARA again demonstrates the lowest delay, confirming its ability to distribute traffic efficiently and avoid localized congestion in dense and highly dynamic FANET deployments.

Overall, the delay results consistently complement the PDR analysis. By combining adaptive next-hop selection with congestion-aware and energy-aware learning, DARA not only improves packet delivery reliability but also significantly reduces end-to-end latency across all scenarios, particularly under high mobility and dense network conditions.

5.2.3 Throughput

Throughput measures the total amount of data successfully delivered to the destination per unit time and is defined as:

$$\text{Throughput} = \frac{\sum \text{Received data (bits)}}{T_{\text{sim}}}, \tag{15}$$

where T_{sim} denotes the total simulation duration.

Higher throughput reflects better utilization of available wireless resources and efficient routing decisions under dynamic network conditions.

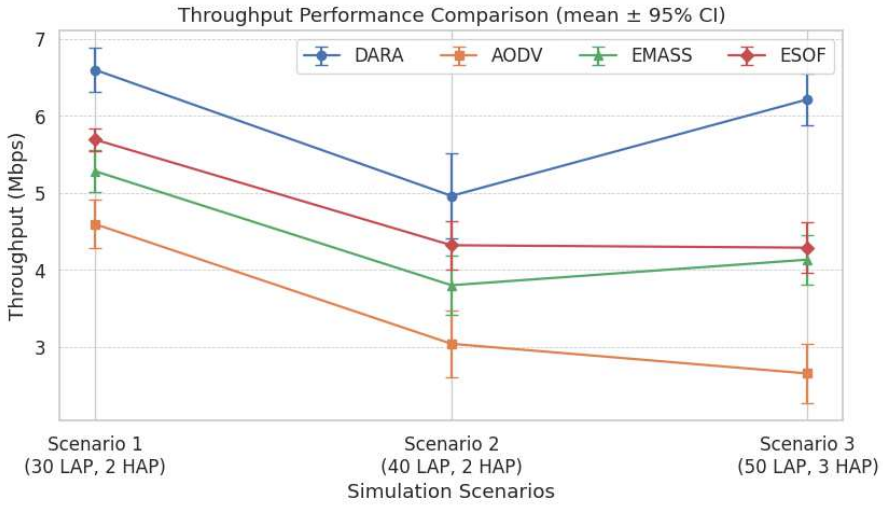


Fig. 9 Throughput comparison of DARA, AODV, EMASS, and ESOFCluster under the three FANET scenarios (mean ± 95% confidence interval).

Figure 9 illustrates the average throughput achieved by DARA, AODV, EMASS, and ESOFCluster under the three evaluated FANET scenarios, expressed as the mean throughput with 95% confidence intervals over multiple simulation runs. Throughput reflects the effective utilization of available wireless resources and the ability of a routing protocol to sustain high data rates under dynamic network conditions.

In **Scenario 1** , all protocols achieve relatively high throughput due to stable links and limited contention. Nevertheless, DARA consistently attains the highest throughput, indicating efficient next-hop selection and effective exploitation of available communication opportunities. Clustering-based protocols (EMASS and ESOFCluster) perform better than AODV but are still constrained by cluster coordination overhead.

In **Scenario 2** ,all protocols experience a reduction in throughput due to frequent link disruptions and varying traffic conditions. The impact is most pronounced for AODV, mainly because repeated route discovery procedures and packet losses occur whenever the topology changes. EMASS and ESOFCluster alleviate this issue

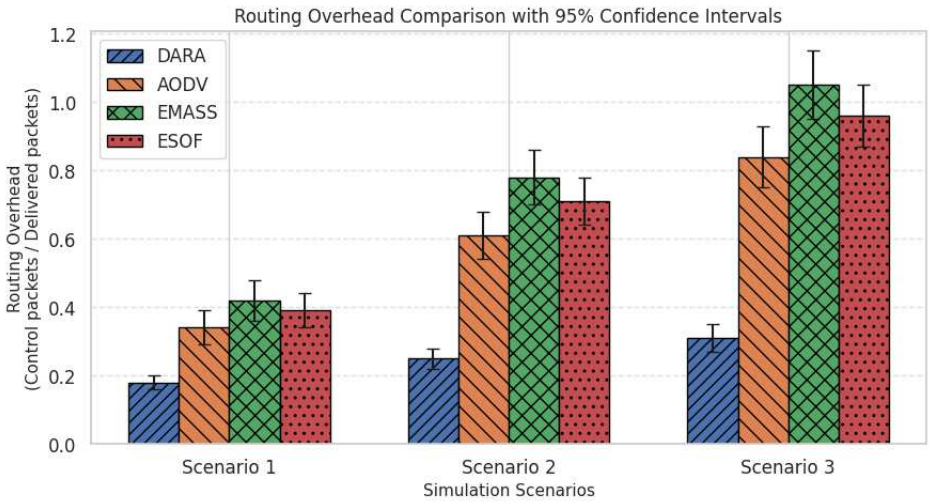


Fig. 10 Routing overhead comparison of DARA,AODV, EMASS and AODV under the three FANET scenarios.

to some extent by relying on hierarchical structuring; however, their performance is still constrained by cluster instability and traffic concentration at cluster heads.

By contrast, DARA preserves a noticeably higher throughput. Its ability to adjust forwarding decisions dynamically and to spread traffic over more stable and less congested links allows it to better cope with rapid network dynamics.

In **Scenario 3** , network contention and interference further limit achievable throughput. While clustering-based approaches benefit from reduced routing-table size, their performance is bounded by cluster-head bottlenecks and increased control traffic. DARA again demonstrates the highest throughput, confirming its ability to scale effectively in dense FANET deployments by avoiding centralized coordination and evenly distributing forwarding load.

Overall, the throughput results complement the PDR and end-to-end delay analyses. By enabling decentralized, reinforcement-learning-based next-hop selection, DARA not only improves delivery reliability and latency but also maximizes effective network capacity across a wide range of FANET operating conditions.

5.2.4 Routing Overhead

Routing overhead quantifies the control cost incurred to establish and maintain routes. It is defined as the ratio of the total number of control packets transmitted to the number of successfully delivered data packets:

$$\text{Overhead} = \frac{N_{\text{control}}}{N_{\text{recv}}}. \tag{16}$$

Lower routing overhead implies more efficient bandwidth usage, which is particularly important in FANETs with limited channel capacity and high mobility. Figure 10

compares the routing overhead of DARA with AODV, EMASS, and ESOFCluster across the three considered FANET scenarios. The results clearly demonstrate that DARA consistently incurs the lowest routing overhead in all scenarios, with narrow confidence intervals that indicate stable control behavior even under high mobility and node density.

In **Scenario 1** , DARA reduces the routing overhead by more than 45% compared to AODV and by over 55% compared to clustering-based approaches. This improvement stems from the fully distributed next-hop decision-making process, which eliminates both route-discovery flooding (AODV) and cluster maintenance signaling (EMASS and ESOF).

In **Scenario 2**, under high mobility and bursty traffic conditions , the routing overhead of AODV increases sharply due to repeated route-request broadcasts, while EMASS and ESOFCluster suffer from frequent re-clustering operations. In contrast, DARA adapts locally to topology changes through RL, avoiding global control exchanges and maintaining a significantly lower overhead.

The stress-test scenario **Scenario 3** further highlights the scalability limitations of clustering-based routing, where cluster-head re-election and intra-cluster reporting dominate the control traffic. DARA preserves a moderate and stable overhead growth, confirming that the proposed free-clustering design effectively mitigates control-plane congestion in dense FANET deployments.

Overall, these results confirm that DARA not only improves data-plane performance but also significantly reduces control overhead, making it well suited for highly dynamic and large-scale UAV networks.

5.2.5 Link Stability

Link stability measures the average duration for which a wireless link between two UAVs remains available before breaking. It is typically derived from the observed link expiration time (LET), which depends on relative mobility and signal strength variations.

Stable links reduce route failures and retransmissions, leading to improved PDR and reduced end-to-end delay. Figure 11 illustrates the average link holding time achieved by DARA, AODV, EMASS, and ESOFCluster across the three FANET scenarios. The results demonstrate that DARA consistently maintains the most stable links, with significantly higher link holding times and narrower confidence intervals.

In **Scenario 1**, DARA outperforms AODV by a substantial margin, indicating that the RL–based next-hop selection effectively avoids rapidly degrading links. Clustering-based approaches provide moderate improvements over AODV; however, their reliance on cluster-head connectivity limits their stability gains.

Scenario 2, AODV's link stability drops significantly because frequent route failures trigger repeated reactive repairs. EMASS and ESOFCluster also show shorter link durations, mainly due to cluster-head mobility, which forces frequent re-clustering operations. In comparison, DARA reacts to mobility changes at the local level, enabling it to sustain more stable connections over time.

Scenario 3 emphasizes DARA's scalability advantages. As network density increases, clustering protocols struggle with unstable inter-cluster links and overloaded

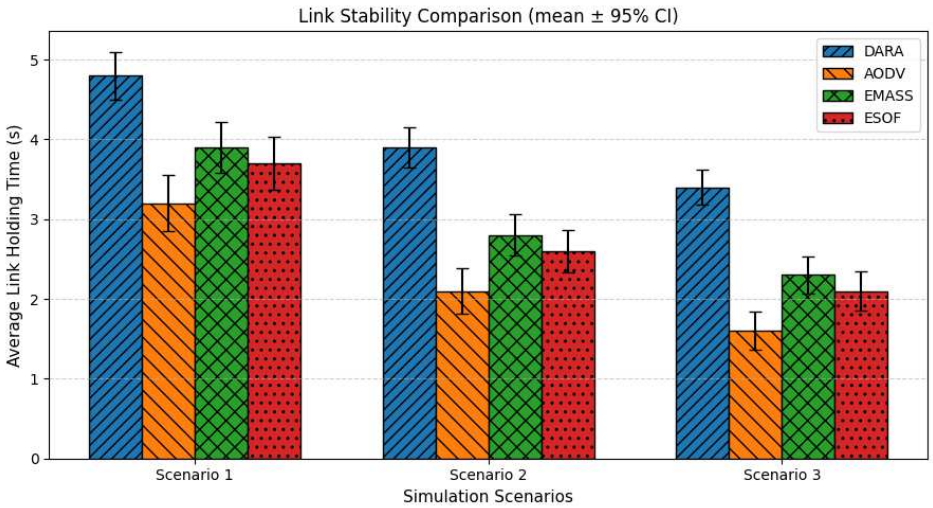


Fig. 11 Average link holding time with 95% confidence intervals under different FANET scenarios.

cluster heads. Meanwhile, DARA continues to maintain more reliable links by constantly refining its forwarding decisions according to observed link quality and mobility patterns..

Overall, the improved link stability achieved by DARA directly contributes to its higher packet delivery ratio, lower end-to-end delay, and reduced routing overhead. These results confirm the effectiveness of a fully distributed reinforcement learning approach for adaptive routing in highly dynamic FANET scenarios.

5.2.6 Energy Consumption

The energy consumption metric quantifies the total energy expended by all UAVs during packet transmission, reception, and control signaling throughout the simulation time. It is computed as:

$$E_{\text{total}} = \sum_{i=1}^{N} \left(E_i^{\text{tx}} + E_i^{\text{rx}} + E_i^{\text{ctrl}} \right), \tag{17}$$

where E_i^{tx}, E_i^{rx}, and E_i^{ctrl} denote the energy consumed by UAV i for data transmission, reception, and control packets, respectively.

The energy consumption shown in Fig. 12 represents the *routing-related network energy*, defined as the cumulative energy consumed by all UAVs for routing operations over the simulation duration. This metric does not include propulsion energy or idle hardware consumption; instead, it focuses exclusively on the energy overhead induced by the routing protocol itself.

Figure 12 compares the routing-related energy consumption of DARA with AODV, EMASS, and ESOFCluster across the three FANET scenarios. The results show that

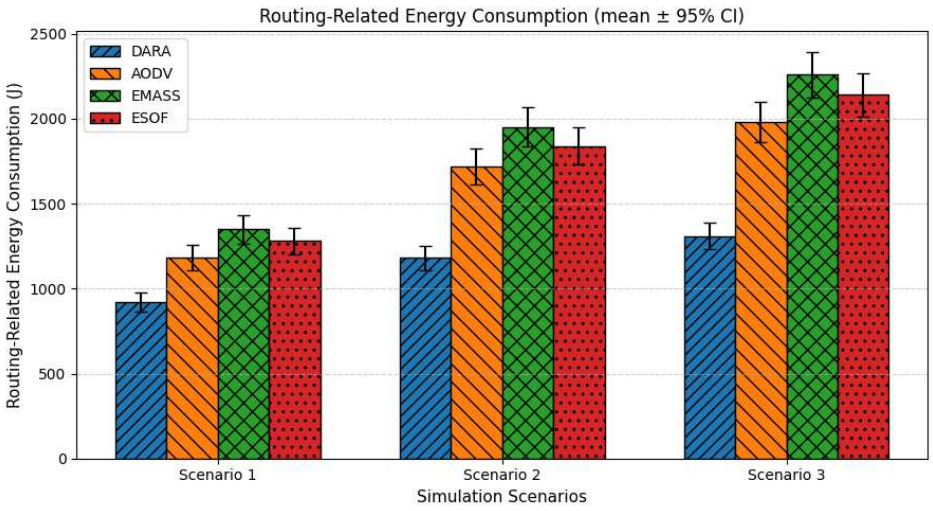


Fig. 12 Routing-related energy consumption with 95% confidence intervals under different FANET scenarios.

DARA consistently achieves the lowest routing energy consumption, with narrow confidence intervals indicating stable energy behavior under varying mobility and density conditions.

Scenario 1, DARA reduces routing energy consumption by approximately 25–30% compared to AODV and by more than 35% compared to clustering-based protocols. This improvement mainly results from avoiding route-discovery flooding and eliminating the signaling overhead associated with cluster formation and maintenance.

Scenario 2, both AODV and clustering protocols show a significant rise in energy consumption. Frequent link failures lead to repeated route recovery procedures, reclustering processes, and additional retransmissions, all of which increase power usage. In contrast, DARA makes forwarding decisions locally through reinforcement learning, enabling it to choose more stable next hops and minimize unnecessary control traffic and retransmission costs.

Scenario 3, the scalability advantage of DARA becomes even more evident. While EMASS and ESOFCluster experience cluster-head overload and uneven energy depletion across nodes, DARA exhibits a more controlled increase in energy consumption. This behavior reflects better load distribution and improved energy balancing throughout the network.

Overall, the results demonstrate that DARA's clustering-free and fully distributed architecture effectively reduces routing-related energy overhead. This contributes to longer network lifetime and enhances mission sustainability in highly dynamic FANET environments.

5.3 Discussion

Table 4 summarizes the overall performance gains achieved by DARA relative to classical MANET routing (AODV) and clustering-based approaches (EMASS and

Table 4 Overall Performance Improvement of DARA Compared to Baseline Protocols

Metric	vs. AODV (%)	vs. EMASS (%)	vs. ESOF (%)
Packet Delivery Ratio (PDR) ↑	18–25%	10–15%	12–17%
End-to-End Delay ↓	35–45%	20–30%	22–28%
Throughput ↑	30–40%	18–25%	20–27%
Routing Overhead ↓	45–55%	55–65%	50–60%
Routing Energy Consumption ↓	25–35%	35–45%	30–40%
Link Stability (LHT) ↑	35–45%	20–30%	22–28%

ESOF). The results confirm that the proposed free-clustering, distributed RL strategy consistently improves reliability, latency, energy efficiency, and scalability across all evaluated FANET scenarios.

The simulation results provide several important insights into the behavior of routing protocols in highly dynamic FANET environments and highlight the benefits and limitations of the proposed DARA framework.

Impact of Free-Clustering Design:

A key finding of this study concerns the effect of removing clustering mechanisms on overall network behavior. Clustering-based solutions such as EMASS and ESOF-Cluster inherently generate additional control traffic due to processes like cluster-head election, periodic re-clustering, and intra-cluster coordination. Although clustering can enhance scalability in relatively stable network conditions, its overhead becomes increasingly burdensome in environments characterized by high mobility and dense deployments. This trend is clearly reflected in Scenarios 2 and 3, where routing overhead and energy consumption rise significantly.

In contrast, DARA follows a clustering-free architecture. Each UAV independently determines its next hop using locally gathered information combined with reinforcement learning. By eliminating cluster formation and maintenance procedures, DARA avoids the associated signaling costs. As a result, it achieves noticeably lower control overhead and reduced energy expenditure related to routing operations.

The results indicate that removing clustering is not only practical but also advantageous in highly dynamic FANET environments, where minimizing control complexity directly improves overall network efficiency.

Effectiveness of Distributed RL:

The distributed reinforcement learning mechanism embedded in DARA is a key factor behind its performance gains. Rather than depending on centralized coordination or partial global knowledge, DARA allows each UAV to independently learn and refine its forwarding strategy based solely on locally available information. This fully distributed design enhances scalability and increases robustness against node failures and temporary link disruptions.

The improvements in packet delivery ratio and end-to-end delay can be directly linked to this adaptive next-hop selection process. By continuously updating decisions according to observed link quality, node mobility, residual energy, and congestion levels, each UAV becomes capable of avoiding unreliable neighbors and overloaded relays.

The resulting increase in link holding time across all scenarios clearly demonstrates this adaptive behavior and helps explain the observed gains in reliability and latency performance.

Scalability and Robustness under Mobility:

The stress-test scenario clearly illustrates the scalability strengths of DARA in dense and highly mobile FANET environments. As the number of UAVs grows, traditional MANET-based protocols such as AODV tend to generate excessive route discovery traffic, leading to increased congestion and performance degradation. Similarly, clustering-based schemes face frequent cluster-head reassignments and uneven energy consumption, especially under rapid topology changes.

In contrast, DARA preserves stable performance even as network density increases. Although routing overhead and energy usage rise slightly, the growth remains controlled and moderate. This resilience is largely due to its fully decentralized structure and reliance on local decision-making. By avoiding centralized coordination and cluster-head dependencies, DARA eliminates control bottlenecks and reduces the risk of single points of failure, enabling more consistent operation under challenging mobility conditions.

Comparison with SDN and Centralized Approaches:

Although Software-Defined Networking (SDN) has been explored as a way to achieve global route optimization in FANETs, its dependence on centralized controllers and continuous state updates creates several practical challenges. The need for frequent reporting and rule dissemination can introduce additional latency, limit scalability, and expose the network to single points of failure. In highly dynamic aerial environments, topology changes occur so rapidly that the global view maintained by the controller may become outdated before updated routing decisions are enforced.

DARA addresses these issues by relocating intelligence to the network edge. Instead of relying on centralized coordination, each UAV operates as an autonomous learning agent capable of making forwarding decisions based on local observations. This enables immediate adaptation to topology variations without waiting for global updates. Such a distributed intelligence model is better suited to the operational realities of FANETs, especially in time-critical contexts such as disaster management and mission-sensitive deployments.

Limitations and Future Directions:

First, the reliance on tabular Q-learning can raise scalability concerns when the state space becomes very large or when the network environment is highly heterogeneous. In such cases, maintaining and updating Q-tables may become inefficient. Integrating function approximation techniques or adopting deep reinforcement learning could improve generalization and adaptability, although this would introduce additional computational and implementation complexity.

Second, the current reward formulation, although designed to reflect major routing objectives, depends on properly selected weight parameters to balance reliability, delay, and energy efficiency. Achieving the right trade-off requires careful tuning.

Future research could investigate adaptive weighting strategies or multi-objective optimization frameworks to dynamically adjust the reward structure according to network conditions.

Finally, the present evaluation assumes fully cooperative UAV behavior and does not address adversarial scenarios or security vulnerabilities. Incorporating trust management, attack resilience, or secure learning mechanisms would be a valuable extension to enhance DARA's applicability in mission-critical and potentially hostile environments.

6 Conclusion

This contribution introduced **DARA**, a distributed agent RL framework for adaptive routing in Flying Ad-Hoc Networks. Unlike classical MANET-derived protocols and clustering-based routing schemes, DARA eliminates the need for route discovery flooding and cluster maintenance by enabling each UAV to autonomously select its next hop using local observations and online RL.

Extensive simulations conducted in ns-3 under multiple FANET scenarios demonstrated that DARA consistently outperforms AODV and state-of-the-art clustering-based approaches such as EMASS and ESOF. The proposed approach achieves higher packet delivery ratio and throughput, while significantly reducing end-to-end delay, routing overhead, and routing-related energy consumption. Furthermore, DARA exhibits superior link stability, confirming its robustness and scalability in highly dynamic and dense UAV deployments.

The results validate the effectiveness of the free-clustering design and the distributed RL paradigm in addressing the unique challenges of FANET environments. By pushing intelligence to the network edge and relying solely on local interactions, DARA avoids the limitations of centralized and SDN-based solutions, including control-plane bottlenecks and single points of failure.

Future work will focus on extending DARA with function approximation and deep RL techniques to handle larger state spaces and heterogeneous UAV capabilities. Additional directions include incorporating security-aware learning mechanisms, evaluating performance under adversarial conditions, and deploying the proposed framework on real UAV platforms to assess its feasibility in real-world missions.

Supplementary information. If your article has accompanying supplementary file/s please state so here.

Authors reporting data from electrophoretic gels and blots should supply the full unprocessed scans for key as part of their Supplementary information. This may be requested by the editorial team/s if it is missing.

Please refer to Journal-level guidance for any specific requirements.

Acknowledgements. Acknowledgements are not compulsory. Where included they should be brief. Grant or contribution numbers may be acknowledged.

Please refer to Journal-level guidance for any specific requirements.

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval and consent to participate
- Consent for publication
- Data availability
- Materials availability
- Code availability
- Author contribution

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

Editorial Policies for:

Springer journals and proceedings: https://www.springer.com/gp/editorial-policies

Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies

Scientific Reports: https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

Appendix A Section title of first appendix

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

[1] Chandran, I., & Vipin, K. (2024). Multi-UAV networks for disaster monitoring: challenges and opportunities from a network perspective. Drone Systems and Applications, 12, 1-28. https://doi.org/10.1139/dsa-2023-0079

[2] Telli, K., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., & Mansoor, W (2023). A comprehensive review of recent research trends on unmanned aerial vehicles (uavs). Systems, 11(8), 400, 2023. https://doi.org/10.3390/systems11080400

[3] Ishiwatari, M. (2024). Leveraging drones for effective disaster management: a comprehensive analysis of the 2024 Noto Peninsula earthquake case in Japan. Progress in Disaster Science, 23, 100348, 2024. https://doi.org/10.1016/j.pdisas.2024.100348.

[4] Yucesoy, E., Balcik, B., & Coban, E. (2025). The role of drones in disaster response: A literature review of operations research applications. International Transactions in Operational Research, 32(2), 545-589. https://doi.org/10.1111/itor.13484.

[5] Yang, B., Xiong, X., Liu, H., Jia, Y., Gao, Y., Tolba, A., & Zhang, X. (2023). Unmanned aerial vehicle assisted post-disaster communication coverage optimization based on internet of things big data analysis. Sensors, 23(15), 6795. https://doi.org/10.3390/s23156795

[6] Rattanaamporn, S.; Perera, A.; Nguyen, A.; Ngo, T.B.; Chahl, J. Drone Imaging and Sensors for Situational Awareness in Hazardous Environments: A Systematic Review. J. Sens. Actuator Netw. 2025, 14, 98. https://doi.org/10.3390/jsan14050098

[7] ALMANSOR, Mohammed Jamal, DIN, Norashidah Md, BAHARUDDIN, Mohd Zafri, et al. Routing protocols strategies for flying Ad-Hoc network (FANET): Review, taxonomy, and open research issues. Alexandria Engineering Journal, 2024, vol. 109, p. 553-577. https://doi.org/10.1016/j.aej.2024.09.032

[8] PAL, Madhumita, MUKHERJEE, Amartya, PANJA, Ayan Kumar, et al. A flying ad-hoc networks framework for smart farming and precision agriculture. In : 2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2025. p. 1719-1725. https://doi.org/10.1109/ICSCDS65426.2025.11166836

[9] KAUR, Manjit, VERMA, Sahil, et al. Flying ad-hoc network (FANET): Challenges and routing protocols. Journal of computational and theoretical nanoscience, 2020, vol. 17, no 6, p. 2575-2581. https://doi.org/10.1166/jctn.2020.8932

[10] OROZCO-LUGO, Aldo G., MCLERNON, Des C., LARA, Mauricio, et al. Monitoring of water quality in a shrimp farm using a FANET. Internet of Things, 2022, vol. 18, p. 100170. https://doi.org/10.1016/j.iot.2020.100170

[11] TAKABATAKE, Kaoru et MATSUZAWA, Tomofumi. FANET Routing Protocol for Prioritizing Data Transmission to the Ground Station. Network, 2026, vol. 6, no 1, p. 7. https://doi.org/10.3390/network6010007

[12] JAIN, Jyoti, FATIMA, Mehajabeen, GUPTA, Roopam, et al. Overview and challenges of routing protocol and mac layer in mobile ad-hoc network. Journal of Theoretical and Applied Information Technology, 2009, vol. 8, no 1, p. 6-12.

[13] Das, S.R.; Perkins, C.E.; Belding-Royer, E.M. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, 2003. Available online (accessed on 06 Feb 2026). https://www.rfc-editor.org/rfc/rfc3561.html

[14] Hu, Y.C.; Maltz, D.A.; Johnson, D.B. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728, 2007. Available online(accessed on 06 Feb 2026). https://www.rfc-editor.org/rfc/rfc4728.html

[15] Clausen, T.H.; Jacquet, P. Optimized Link State Routing Protocol (OLSR). RFC 3626, 2003. Available online (accessed on 06 Feb 2026). https://www.rfc-editor.org/rfc/rfc3626.html

[16] Jolimbetova E.D. (2025). Routing Strategies In Fanet Networks: A Systematic Review Of Protocol Families, Performance Indicators, And Research Gaps. The American Journal of Engineering and Technology, 7(11), 213–219. https://doi.org/10.37547/tajet/Volume07Issue11-13

[17] M. Aissa, M. Abdelhafidh, and A. B. Mnaouer (2021),EMASS: A novel energy, safety and mobility aware-based clustering algorithm for FANETs, *IEEE Access*, vol. 9, pp. 105 506–105 520, Jul, 2021. https://doi.org/10.1109/ACCESS.2021.3097323

[18] Bouhdid, B., & Aissa, M. (2025). Reliable and Energy-Efficient Clustering Approach for FANETs Using a Fuzzy Logic System. In International Conference on Advanced Information Networking and Applications (pp. 13-24). Cham: Springer Nature Switzerland. https://doi.org/10.1109/ACCESS.2021.3097323

[19] EL-BASIONI, Basma M. Mohammad. Intensive study, tuning and modification of reactive routing approach to improve flat FANET performance in data collection scenario. Scientific Reports, 2024, vol. 14, no 1, p. 23467. https://doi.org/10.1038/s41598-024-72983-y

[20] KAUR, Manjit, PRASHAR, Deepak, MRSIC, Leo, et al. Machine Learning-Based Routing Protocol in Flying Ad Hoc Networks: A Review. Computers, Materials Continua, 2025, vol. 82, no 2. https://doi.org/10.32604/cmc.2025.059043

[21] Rovira-Sugranes, A., Razi, A., Afghah, F., & Chakareski, J. (2022). A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. Ad Hoc Networks, 130, 102790. https://doi.org/10.1016/j.adhoc.2022.102790

[22] LANSKY, Jan, ALI, Saqib, RAHMANI, Amir Masoud, et al. Reinforcement learning-based routing protocols in flying ad hoc networks (FANET): A review. Mathematics, 2022, vol. 10, no 16, p. 3017. https://doi.org/10.3390/math10163017

[23] MOHAMMED, Yasir I., HASSAN, Rosilah, HASAN, Mohammad Kamrul, et al. Revolutionizing FANETs With Reinforcement Learning: Optimized Data Forwarding and Real-Time Adap. IEEE Open Journal of the Communications Society, 2025. https://doi.org/10.1109/OJCOMS.2025.3565471

[24] BUGARČIĆ, Pavle, JEVTIĆ, Nenad, et MALNAR, Marija. Reinforcement learning-based routing protocols in vehicular and flying ad hoc networks–a literature survey. Promet-Traffic&Transportation, 2022, vol. 34, no 6, p. 893-906. https://traffic2.fpz.hr/index.php/PROMTT/article/view/150/33

[25] CUONG, THOM, THI, HUONG LY N., BINH, LEH, et al. QLR-FANET: A Q-learning and rate control-based routing protocol for flying ad hoc network. ETRI JOURNAL, 2025. https://doi.org/10.4218/etrij.2024-0298

[26] HOSSEINZADEH, Mehdi, TANVEER, Jawad, RAHMANI, Amir Masoud, et al. A Q-learning-based smart clustering routing method in flying Ad Hoc networks. Journal of King Saud University-Computer and Information Sciences, 2024, vol. 36, no 1, p. 101894. https://doi.org/10.1016/j.jksuci.2023.101894

[27] WU, Mingwei, JIANG, Bo, CHEN, Siji, et al. Traj-Q-GPSR: A Trajectory-Informed and Q-Learning Enhanced GPSR Protocol for Mission-Oriented FANETs. Drones, 2025, vol. 9, no 7, p. 489. https://doi.org/10.3390/drones9070489

[28] DA COSTA, Luis Antonio LF, DE LAMARE, Rodrigo C., KUNST, Rafael, et al. Study of Cluster-Based Routing Based on Machine Learning for UAV Networks in 6G. arXiv preprint arXiv:2510.27121, 2025. https://arxiv.org/html/2510.27121v1

[29] FENG, Houze, WANG, Jingjing, HOU, Xiangwang, et al. Communication-Aware Hierarchical Routing for FANET: A Reinforcement Learning Approach. IEEE Transactions on Cognitive Communications and Networking, 2025. https://doi.org/10.1109/TCCN.2025.3620368

[30] KIM, Taehwan, LEE, Seonah, KIM, Kyong Hoon, et al. FANET routing protocol analysis for multi-UAV-based reconnaissance mobility models. Drones, 2023, vol. 7, no 3, p. 161. https://doi.org/10.3390/drones7030161

[31] PRIYADARSHINI, Prangya, HOTA, Lopamudra, KUMAR, Arun, et al. A Cross-Layer Congestion-Aware Routing Protocol for UAV-Aided VANETs with Enhanced Line-of-Sight Communication. In : International Conference on Smart Grid Inspired Future Technologies. Cham : Springer Nature Switzerland, 2024. p. 131-144. https://doi.org/10.1007/978-3-031-96146-5_10

[32] LIU, Jinyue, GONG, Peng, YANG, Haowei, et al. Cross-Layer Optimized OLSR Protocol for FANETs in Interference-Intensive Environments. Drones, 2025, vol. 9, no 11, p. 778. https://doi.org/10.3390/drones9110778

[33] AHMED, Salma Badawi Mohammed, AHMAD, Norulhusna, DZIYAUDDIN, Rudzidatul Akmam, et al. OLSR Routing Protocol in FANET Using Different Mobility Models. In : 2025 IEEE International Symposium on Future Telecommunication Technologies (SOFTT). IEEE, 2025. p. 71-77. https://doi.org/10.1109/SOFTT67007.2025.11213334

[34] AL-EMADI, Sara et AL-MOHANNADI, Aisha. Towards enhancement of network communication architectures and routing protocols for FANETs: A survey. In : 2020 3rd international conference on advanced communication technologies and networking (CommNet). IEEE, 2020. p. 1-10. https://doi.org/10.1109/CommNet49926.2020.9199627

[35] HEINZELMAN, Wendi Rabiner, CHANDRAKASAN, Anantha, et BALAKRISHNAN, Hari. Energy-efficient communication protocol for wireless microsensor networks. In : Proceedings of the 33rd annual Hawaii international conference on system sciences. IEEE, 2000. p. 10 pp. vol. 2. https://doi.org/10.1109/HICSS.2000.926982

[36] BENZEKKI, Kamal, EL FERGOUGUI, Abdeslam, et ELBELRHITI ELALAOUI, Abdelbaki. Software-defined networking (SDN): a survey. Security and communication networks, 2016, vol. 9, no 18, p. 5803-5833. https://doi.org/10.1002/sec.1737

[37] ALBU-SALIH, Alaa Taima et KHUDHAIR, Hayder Ayad. ASR-FANET: An adaptive SDN-based routing framework for FANET. International Journal of Electrical & Computer Engineering (2088-8708), 2021, vol. 11, no 5. https://doi.org/10.11591/ijece.v11i5.pp4403-4412

[38] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek and M. Reisslein, "QR-SDN: Towards Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks," in IEEE Access, vol. 8, pp. 174773-174791, 2020. https://doi.org/10.1109/ACCESS.2020.3025432

[39] HUSSAIN, Altaf, HUSSAIN, Tariq, FAISAL, Farooq, et al. DLSA: Delay and link stability aware routing protocol for flying ad-hoc networks (FANETs). Wireless Personal Communications, 2021, vol. 121, no 4, p. 2609-2634. https://doi.org/10.1007/s11277-021-08839-9

[40] LAKEW, Demeke Shumeye, SA'AD, Umar, DAO, Nhu-Ngoc, et al. Routing in flying ad hoc networks: A comprehensive survey. IEEE communications surveys & tutorials, 2020, vol. 22, no 2, p. 1071-1120. https://doi.org/10.1109/COMST.2020.2982452

[41] GANGOPADHYAY, Swati et JAIN, Vinod Kumar. A position-based modified OLSR routing protocol for flying ad hoc networks. IEEE Transactions on Vehicular Technology, 2023, vol. 72, no 9, p. 12087-12098. https://doi.org/10.1109/TVT.2023.3265704

[42] HIDAWI, Khaoula, ENNAJI, Sabrine, et FERRARI, Elena. BlackoutADR: Exploiting adaptive data rate vulnerabilities in LoRaWAN-based FANETs. Journal of Network and Computer Applications, 2026, vol. 246, p. 104409 https://doi.org/10.1016/j.jnca.2025.104409

[43] WOO, Alec, TONG, Terence, et CULLER, David. Taming the underlying challenges of reliable multihop routing in sensor networks. In : Proceedings of the 1st international conference on Embedded networked sensor systems. 2003. p. 14-27. https://doi.org/10.1145/958491.958494

[44] Gaw lowicz, P., Zubow, A. ns-3 meets OpenAI Gym: The playground for machine learning in networking research. In: Proc. 22nd ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 113–119 (2019). https://doi.org/10.1145/3345768.3355908

[45] Queir´os, R., Ferreira, L., Fontes, H., Campos, R.: RateRL: A Framework for Developing RL-Based Rate Adaptation Algorithms in ns-3. In: Guisado-Lizar, J.L., Riscos-N´u˜nez, A., Mor´on-Fern´andez, M.J., Wainer, G. (eds) Simulation Tools and Techniques, SIMUtools 2023, LNICST, vol. 519, Springer, Cham (2024). https://doi.org/10.1007/978-3-031-57523-5_22

[46] ns-3 Consortium: ns-3 AI Tutorial: Integrating Machine Learning with ns-3. Tutorial presented at the ns-3 Annual Meeting (2021). Available at: https://www.nsnam.org/wp-content/uploads/2021/tutorials/ns3-ai-tutorial-June-2021.pdf