Vehicular Communications 53 (2025) 100905

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom

QRCF: A new Q-learning-based routing approach using a smart cylindrical filtering system in flying ad hoc networks

Amir Masoud Rahmani [a,1], Amir Haider [b,1], Monji Mohamed Zaidi [c,d], Abed Alanazi [e], Shtwai Alsubai [e], Abdullah Alqahtani [e], Mohammad Sadegh Yousefpoor [f], Efat Yousefpoor [f], Mehdi Hosseinzadeh [g,h,*]

[a] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan
[b] Department of Artificial intelligence and Robotics, Sejong University, Seoul, South Korea
[c] Department of Electrical Engineering, College of Engineering, King Khalid University, Abha, 61421, Saudi Arabia
[d] Center for Engineering and Technology Innovations, King Khalid University, Abha 61421, Saudi Arabia
[e] Department of Computer Science, College of Computer Engineering and Sciences in Al-Kharj, Prince Sattam bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia
[f] Center of Research and Strategic Studies, Lebanese French University, Kurdistan Region, Iraq
[g] School of Computer Science, Duy Tan University, Da Nang, Viet Nam
[h] Jadara Research Center, Jadara University, Irbid 21110, Jordan

ARTICLE INFO

Keywords:
Unmanned aerial vehicle (UAV)
Flying ad hoc network (FANET)
Reinforcement learning (RL)
Routing
Artificial Intelligence (AI)

ABSTRACT

To ensure reliable data transmission in flying ad hoc networks (FANETs), efficient routing protocols are necessary to establish communication paths in FANETs. Recently, reinforcement learning (RL), particularly Q-learning, has become a promising approach for overcoming challenges faced by traditional routing protocols due to its capacity for autonomous adaptation and self-learning. This study presents a Q-learning-based routing strategy, enhanced by an innovative cylindrical filtering technique, named QRCF in FANETs. In QRCF, the dissemination interval of hello packets is adaptively adjusted based on the connection status of nearby UAVs. Then, this routing process leverages Q-learning to discover reliable and stable routes, using a state set refined by the cylindrical filtering technique to accelerate the search for the optimal path in the network. Afterward, the reward value is computed using metrics such as relative speed, connection time, residual energy, and movement path. Finally, QRCF is deployed in the network simulator 2 (NS2), and its performance is evaluated against three routing schemes, QRF, QFAN, and QTAR. These evaluations are presented based on the number of UAVs and their speed. In general, when changing the number of nodes, QRCF improves energy usage (about 5.01%), data delivery ratio (approximately 1.20%), delay (17.71%), and network longevity (about 3.21%). However, it has a higher overhead (approximately 10.91%) than QRF. Moreover, when changing the speed of UAVs in the network, QRCF improves energy usage (about 4.94%), data delivery ratio (approximately 2.36%), delay (about 17.5%), and network lifetime (approximately 8.75%). However, it increases routing overhead (approximately 15.47%) in comparison with QRF.

1. Introduction

The need for humans to access environments that are difficult or impossible to reach has led to the development of unmanned aerial vehicles (UAVs), which are also known as automatic or remotely controlled flying vehicles [1,2]. Initially, UAV applications were typically conducted using a single large drone, which was responsible for completing specific missions. Monitoring and surveillance have emerged as the most common applications for UAVs [3–5]. However, relying on a single UAV poses significant challenges in large-scale operations due to its inher-

* Corresponding author at: School of Computer Science, Duy Tan University, Da Nang, Viet Nam.
E-mail addresses: rahmania@yuntech.edu.tw (A.M. Rahmani), amirhaider@sejong.ac.kr (A. Haider), amzaydi@kku.edu.sa (M.M. Zaidi), ad.alanazi@psau.edu.sa (A. Alanazi), sa.alsubai@psau.edu.sa (S. Alsubai), aq.alqahtani@psau.edu.sa (A. Alqahtani), mohammad.sadegh@lfu.edu.krd (M.S. Yousefpoor), efat.yousefpoor@lfu.edu.krd (E. Yousefpoor), mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.vehcom.2025.100905
Received 25 April 2024; Received in revised form 9 February 2025; Accepted 4 March 2025
Available online 4 April 2025

ent limitations and vulnerability [6,7]. In response to these challenges, researchers have successfully proposed collaborative approaches that led to the development of multi-UAV systems. This innovation has garnered considerable interest from scientists and experts in the field [8,9]. When multiple UAVs collaboratively complete a specific task, they create a flying ad hoc network (FANET), which is a subset of mobile ad hoc networks (MANETs) and vehicular ad hoc networks (VANETs) [10,11].

The most significant differences between FANET and other types of ad hoc networks are limited energy resources, rapid movement of UAVs, and frequent topological changes. However, advancements in technologies such as electronics and robotics have significantly enhanced UAV capabilities, particularly in flexibility, agility, payload capacity, and sensing power [12,13]. Recently, FANETs have found success in both military and civilian parts. These networks include numerous applications, like search and rescue operations, traffic monitoring in smart cities, agricultural monitoring, wildlife tracking, border security, and forest fire monitoring [14,15]. To effectively conduct a wide range of missions using UAVs within FANETs, it is essential to design efficient routing protocols that accommodate these unique characteristics. Such protocols are critical for ensuring the timely dispatch of collected data to the ground control station (GCS) [16,17].

In recent years, various routing protocols have been developed to enhance the performance of FANETs. These protocols significantly influence communication among UAVs in the network. Traditional routing protocols, such as topology-based methods, rely on routing tables to manage data packet delivery along the shortest path, which is constructed hop-by-hop [18,19]. The constructed path must be efficient, as each hop represents an intermediate node. However, because UAVs in FANETs move at high speeds, these traditional methods can create additional overhead when they need to restart the route discovery process [20,21]. As a result, these protocols must be adapted for the three-dimensional environment in which FANETs operate. In contrast, location-based routing methods utilize geographic information obtained from GPS or other localization services. In this approach, a source UAV dispatches data packets to a destination UAV based on its location and that of any intermediate nodes [22,23].

However, location-based methods have limitations since they depend solely on position information. If a source UAV cannot find an appropriate next-hop node that is closer to the destination node, it may face delays during the routing process [24,25]. Additionally, due to their high speed and movement patterns, UAVs can quickly exit each other's communication ranges, and links between them can be severed unexpectedly [26,27]. This leads to increased packet loss in FANET [28,29]. Thus, these protocols need to estimate the geographical locations of all UAVs so that they can consistently identify optimal next-hop nodes toward their intended destinations [30,31]. Since maintaining effective communication links between UAVs and enhancing stability in connection management pose ongoing challenges for designing effective routing algorithms in FANETs, researchers are actively exploring solutions tailored specifically for this context [32–34].

A growing number of studies are exploring the use of machine learning (ML) algorithms to tackle routing challenges and enhance the flexibility of routing approaches in FANETs [35,36]. Among these techniques is reinforcement learning (RL), one of the most prominent ML approaches, for improving the performance of routing algorithms in FANETs. In RL-based routing protocols, the agent explores the network by selecting various actions to find the most efficient routing strategy [37–39]. In this regard, the agent must decide on the best route based on several criteria derived from the local information collected by UAVs to lower the packet loss rate and increase communication stability in FANET. However, to develop an optimal routing solution, the agent must have access to global network information [40,41].

This paper presents a Q-learning-based routing strategy, enhanced by an innovative cylindrical filtering technique, called QRCF for FANETs. QRCF includes two key phases: the adaptive neighbor discovery phase and the Q-learning-based routing phase. In the first phase, the interval for propagating hello messages is adaptively adjusted based on the connection status of neighboring UAVs. This ensures that communication remains efficient and responsive to changes in network dynamics. The second phase involves the Q-learning-based routing process, which employs a filtered state set generated by a cylindrical filtering algorithm. This enhances the Q-learning process, accelerating the identification of optimal routing paths. The main contributions introduced in this proposed method include:

- In QRCF, an adaptive neighbor discovery process is carried out. This process helps to construct a local network topology and facilitates routing between different UAVs. Each hello message in QRCF has specified content, and the propagation interval of these messages is adjusted based on the connection status of neighboring UAVs. This ensures that QRCF is compatible with FANET by enabling timely updates to neighbor tables and reducing routing overhead.
- In QRCF, a Q-learning-based routing mechanism is introduced to find reliable and stable routes in FANET. This distributed learning model utilizes a filtered state set that is dependent on a cylindrical filtering algorithm. By utilizing a smaller state space, this approach enhances the routing phase when determining optimal paths.
- In QRCF, reward values for taking specific actions are determined by four environmental components: relative speed, connection time, residual energy, and movement path.
- In QRCF, the learning parameters associated with the Q-learning-based routing process (i.e. learning rate and discount factor) are dynamically calculated for each UAV based on its unique conditions. This characteristic improves adaptability within FANET and enhances overall efficiency; for instance, the learning rate is calculated according to connection time while the discount factor reflects stability among neighboring nodes over successive time intervals.
- To prevent route failures and packet loss caused by events such as routing holes or failed links, QRCF implements a prevention mechanism that aims to mitigate these issues. When a UAV encounters local optima during its route selection process, it provides feedback to its previous-hop UAV; as a result, any action leading to this scenario receives minimal rewards in future evaluations.

The organization of this study is as follows: Section 2 reviews relevant literature. In Section 3, essential concepts, including an overview of Q-learning, are explained. Section 4 details the network configurations used. Section 5 outlines the proposed methodology. In Section 6, the evaluation outcomes are noted. Finally, Section 7 includes the conclusion of the paper.

2. Related works

In [42], a smart filtering technique-based Q-learning-based routing strategy called QRF is presented in FANETs. QRF employs Q-learning to improve data transfer and enhance network efficiency. QRF designs a filtering algorithm to lower the size of the state set. This idea speeds up the learning process of the routing algorithm when converging toward the optimal solution (i.e. the most efficient path from source to destination). Moreover, QRF adapts learning parameters to better handle the dynamic environment of FANETs. The evaluations performed in this paper reveal that QRF evenly distributes energy consumption and increases network longevity. In addition, the smart filtering algorithm causes high routing overhead but succeeds in reducing delay in data transmission.

In [43], a Q-learning-based clustering approach called QSCR is offered for FANETs. In this scheme, hello messages are exchanged periodically and are responsible for finding the local network topology. The period of these hello messages varies in each cluster, and cluster leaders calculate the updated hello time based on the similarity of velocities of UAVs. In the clustering algorithm, the selection of cluster leaders is dependent on a merit value. It is obtained from criteria such as remaining

energy, centrality, connection degree, velocity similarity, and connection duration. Afterward, the central server is responsible for adjusting clustering weights based on a centralized Q-learning algorithm. Finally, QSCR carries out a greedy forwarding process to dispatch data packets to the destination. The study's results show that QSCR increases the speed of the clustering process, but it does not achieve suitable cluster stability. Moreover, QSCR succeeds in improving energy usage and increasing network lifetime and packet delivery rate.

In [44], a fuzzy trust-based secure routing technique (FTSR) is suggested for FANETs. This scheme can improve the security level of communication between UAVs in FANET. Two trust systems are suggested in FTSR: local trust and route trust. The first is a distributed local trust process, which searches trustworthy neighboring UAVs and separates malicious nodes. Hence, the path discovery phase is carried out by trusted UAVs. As a result, a lower number of fake paths is built in FANET. The second mechanism, path trust, targets malicious UAVs that were not detected in the local trust system. It evaluates the security level of the generated path using a fuzzy security system to identify secure paths in the network. Evaluations obtained from the simulation process show that FTSR detects malicious nodes accurately and improves network security.

In [45], a greedy perimeter stateless routing scheme called GPSR+ is offered for FANETs. A position prediction strategy in GPSR+ is offered to estimate the next position of each UAV. This strategy is applied to update the hello broadcast period in each UAV. As a result, GPSR+ gets better adaptability to FANET. Additionally, GPSR+ incorporates a spherical removal strategy when selecting the next-hop node. This strategy identifies a set of potential UAV candidates to serve as the next-hop node and subsequently selects the most suitable one during the routing process.

In [46], a Q-learning-based clustering routing approach called ICRA is introduced for FANETs. This approach is responsible for designing a clustering module, setting clustering weights based on Q-learning, and finding the best routes in FANETs. In the first step, ICRA allocates a utility value for each UAV to select UAVs with the higher utility values as CHs. Then, it dynamically adjusts the weights related to the utility function using reinforcement learning to improve network stability and increase network lifetime. This Q-learning-based cluster adjustment strategy makes an optimal and efficient clustering process, which is proportional to network conditions. In the last step, ICRA utilizes inter-cluster forwarding nodes in the routing phase to reduce delay and improve data delivery ratio.

In [47], a fuzzy optimized link state routing method (OLSR+) is presented in FANETs. In OLSR+, a novel approach is proposed to obtain the connection time of communication links between UAVs. In this approach, different metrics such as relative velocity, link quality, movement angle, and distance are considered. In OLSR+, multipoint relays (MPR) are chosen using a fuzzy system whose inputs are connection time, neighbor degree, and remaining energy. Lastly, OLSR+ builds communication paths based on remaining energy, connection time, and hop counts. The evaluations performed in OLSR+ show its successful performance in comparison with existing methods.

In [48], a Q-learning-based routing scheme called QFAN is presented in FANETs. QFAN identifies communication paths based on Q-learning. In QFAN, the search area is restricted based on a filtering parameter, which is dependent on movement angle, remaining energy, connection quality, and distance. Lastly, QFAN presents a route maintenance process to find and rebuild the paths that are likely to be broken. The evaluation results performed in this paper illustrate the superior performance of QFAN compared to other schemes.

In [49], a Q-learning-based topology-aware routing strategy (QTAR) is offered in FANETs. In QTAR, the routing process is modified by using the data of two-hop surrounding UAVs obtained from the exchange of hello messages when discovering local network topology. QTAR decides on the most suitable path based on several metrics such as location, energy, delay, and velocity. Additionally, QTAR designs an adaptive Q-learning algorithm whose learning scales are dynamically adjusted according to network conditions. Additionally, QTAR determines the link duration by analyzing the direct connections between each node and its neighboring nodes. This value is then used to automatically calculate both the hello interval and the connection duration. The evaluation results show the appropriate and useful performance of QTAR in comparison with existing routing methods.

Table 1 illustrates the strengths and weaknesses of the existing works.

3. Basic concepts

QRCF relies on reinforcement learning (RL), especially Q-learning, to manage the routing process within FANET.

3.1. Q-learning

To assess each action in a state, one of the most widely used free-model reinforcement learning methods is Q-learning. This algorithm can interact with the environment to solve the problems caused by transitions and potential rewards of an action. Q is a function similar to $Q'\left(s_t, a_t\right)$, whose learning algorithm is Q-learning. To update the Q-value, Q-learning allows an agent, which is in an interactive environment, to select actions that maximize rewards. The agent tries to modify its action strategy according to the rewards obtained from the environment. This agent chooses an action in the next state (s_{t+1}) based on the previous reward. These steps are continuously repeated in several steps so that the agent learns well the environment and converges to an optimal response. Equation (1) is used to calculate and update the Q-value in Q-learning after each iteration [52,53].

$$Q\left(s_t, a_t\right) \leftarrow (1-\alpha) Q\left(s_t, a_t\right) + \alpha\left(R_t + \gamma \max Q'\left(s_{t+1}, a_{t+1}\right)\right) \tag{1}$$

here, s_{t+1}, s_t, and a_t are the new state, the previous state, and the selected action, respectively. R_t indicates the reward received by the agent. $0 < \alpha \leq 1$ is the learning rate, and $0 < \gamma \leq 1$ represents the discount factor so that $\max Q'\left(s_{t+1}, a_{t+1}\right)$ is equal to the maximum value obtained from the state s_{t+1}. $Q\left(s_t, a_t\right)$ is equivalent to the Q-value, which allows the agent to take an action in the state s_t and get the reward R_t, immediately.

4. System settings

This section describes hypotheses related to the network. See Fig. 1, which displays the network graph $G = (U, L)$ so that the set of vertices (U) and the set of edges (L) include UAVs (i.e. U_i, $i = 1, 2, ..., N$) and all communication links between these UAVs, respectively. Each UAV, such as U_i, utilizes equipment such as different sensors, powerful microprocessors, localization systems, wireless communication interfaces, and cameras. The localization system enables U_i to obtain its three-dimensional position $\left(x_i^t, y_i^t, z_i^t\right)$ at any moment. In the network, U_i spans a spherical region with a radius of R. Due to the support of long-range communication, the resistance to interference, and the coverage of a broad area, the communication standard IEEE 802.11n is applied in the MAC layer to communicate between UAVs. In QRCF, communication links are divided into two types, namely UAV-to-UAV or UAV-to-ground control station (GCS). The UAV-to-UAV communication is used to communicate between each UAV and its adjacent UAVs. Additionally, the UAV-to-GCS communication is applied to form a direct connection between each UAV and GCS when they are close to each other. Therefore, at any moment, only a few UAVs can be connected directly to GCS. If the distance between a UAV and GCS is high, the communication between this UAV and GCS is a combination of UAV-to-UAV and UAV-to-GCS communications.

Table 1
Benefits and drawbacks of various methods.

Approach	Strengths	Weaknesses
QRF [42]	Making improvements on the quality of service (QoS) needs, such as data delivery rate, latency, energy usage, and data reliability, converging to the optimal path quickly due to controlling the size of the state set in the routing process, getting better adaptability to FANET	Being vulnerable to communication overhead and scalability
QSCR [43]	Making improvements on the QoS needs, such as energy usage, packet delivery rate, and network lifetime, having high scalability, reducing the number of isolated clusters in the network, getting better adaptability to FANET	Being vulnerable to delay, communication overhead, and low convergence speed, high time complexity
FTSR [44]	Detecting malicious nodes quickly and accurately, making improvements on the QoS needs, such as packet delivery rate and network security	Being vulnerable to delay and scalability, not getting a suitable adaptability to FANET
GPSR+ [45]	Making improvements on the QoS needs, such as data delivery ratio, routing overhead, network lifetime, and energy usage, using stable communication links, presenting a position prediction system, and getting better adaptability to FANET	Being vulnerable to delay, dealing with the local optimum issue
ICRA [46]	Making improvements on the QoS needs, such as data delivery ratio, delay, energy usage, and cluster construction time, reducing the number of isolated clusters, high scalability, getting better adaptability to FANET	Not having dynamic learning parameters (α and γ), low convergence rate, high time complexity
OLSR+ [47]	Making improvements on the QoS needs, such as delay, network lifetime, data delivery ratio, throughput, and energy usage, and getting better adaptability to FANET	Being vulnerable to scalability and routing overhead
QFAN [48]	Making improvements on the QoS needs, such as delay, data delivery rate, energy usage, and network lifetime	Being vulnerable to communication overhead and scalability, having constant learning parameters (α and γ)
QTAR [49]	Making improvements on the QoS needs, such as data loss ratio, energy usage, delay, and throughput, increasing the stability of communication paths, getting better adaptability to FANET	Low convergence rate and low scalability

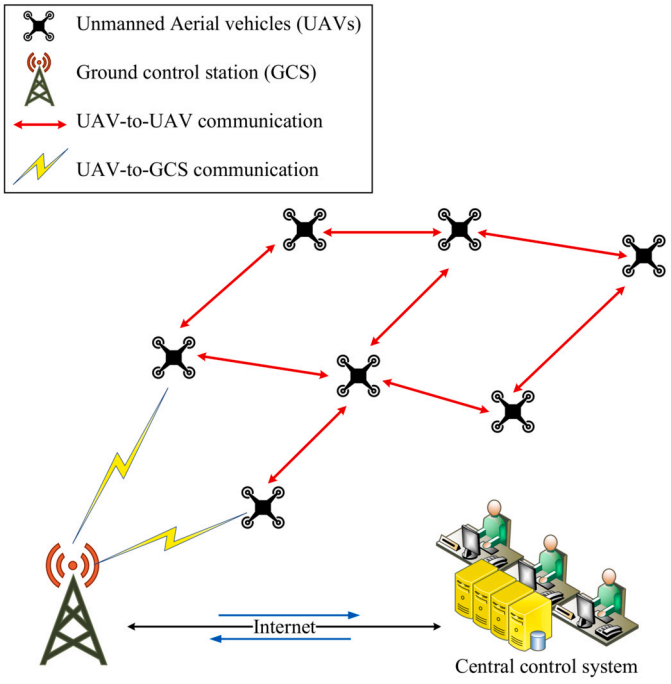

Fig. 1. Network model in QRCF.

5. Proposed scheme

Here, a **Q**-learning-based **r**outing strategy, enhanced by an innovative **c**ylindrical **f**iltering technique, (QRCF) is described in FANETs. In QRCF, two steps are presented: adaptive neighbor discovery phase, and Q-learning-based routing phase. Fig. 2 displays a schematic design of QRCF. Table 2 also states the symbols in this paper.

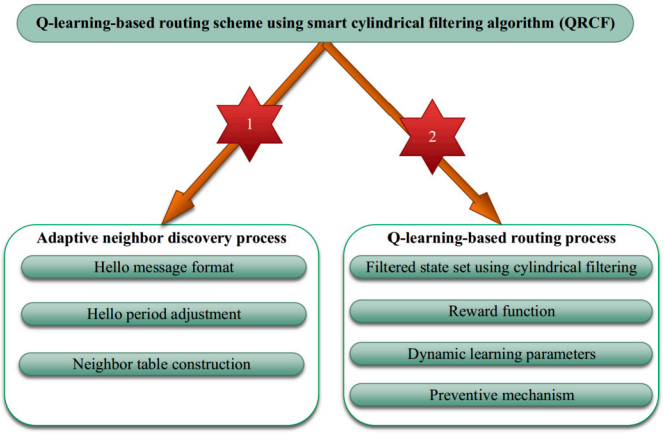

Fig. 2. Schematic design of QRCF.

5.1. Adaptive neighbor discovery process

In this section, each UAV tries to be aware of the local network topology and find its surrounding UAVs by propagating hello messages in FANET. Algorithm 1 states the pseudo-code of the adaptive neighbor discovery phase. Hence, this phase consists of three steps:

- Definition of hello message format
- Determination of the hello propagation interval
- Construction of the neighbor table

Table 2
Symbols used in QRCF.

Symbol	Description
$G=(U,L)$	Network graph
U	Set of vertices in the graph G
L	Set of edges in the graph G
U_i	i-th UAV
N	Total number of all UAVs
(x_i^t, y_i^t, z_i^t)	Position of i-th UAV at the moment t
R	Communication radius of UAVs
GCS	Ground control station
$Hello_i$	Hello message related to U_i
HT_i^t	Propagation period of $Hello_i$
HT_L	Lower boundary of HT_i^t
HT_H	Upper boundary of HT_i^t
ID_{Hello_i}	Identifier of $Hello_i$
ID_i	Identifier of i-th UAV
$(V_i^t, \theta_i^t, \varphi_i^t)$	Velocity vector of i-th UAV at the moment t
E_i^t	Energy level of i-th UAV at the moment t
$V_{\min}$	Lowest velocity of UAVs in the network
$V_{\max}$	Highest velocity of UAVs in the network
CT_{ij}	Connection time between U_i and U_j
α_1	First configuration weight in CT_{ij}
α_2	Second configuration weight in CT_{ij}
d_{ij}^t	Euclidean distance between U_i and U_j at the moment t
MD_{ij}	Difference between the movement directions of U_i and U_j
$e_{prediction}$	Position prediction error of U_i
$(x_i^{t+1}, y_i^{t+1}, z_i^{t+1})$	Real spatial coordinates of U_i at the moment $t+1$
$(\tilde{x}_i^{t+1}, \tilde{y}_i^{t+1}, \tilde{z}_i^{t+1})$	Predicted spatial coordinates of U_i at the moment $t+1$
$NTable_i$	Neighbor table of U_i
VT_j	Validity time related to the entry of U_j
α_{ij}	Learning rate related to the link between U_i and U_j
γ_j	Discount factor corresponding to U_j
FS_i^{Set}	Filtered state set of U_i
(x_P, y_P, z_P)	Cartesian coordinates of the point P
(ρ_P, Θ_P, z_P)	Cylindrical coordinates of the point P
α	Angle between the line segment of $U_i - U_D$ and z-axis
V_{ij}	Relative velocity of U_j with regard to U_i
E_j^t	Energy level of U_j
λ_j	Movement direction of U_j with regard to U_i
R_t	Reward function in the Q-learning-based routing process
$R_{\max}$	Maximum reward
$R_{\min}$	Minimum reward

5.1.1. Hello message format

In the adaptive neighbor discovery process, each UAV, such as U_i, utilizes the exchange of hello messages to facilitate the routing process. Thus, the content of each hello message is very important in this process because this content indicates the information collected by U_i from its adjacent nodes in the network. Equation (2) specifies the structure of the hello message in QRCF.

$$Hello_i = \left\langle ID_{Hello_i} \middle\| ID_i \middle\| (x_i^t, y_i^t, z_i^t) \middle\| (V_i^t, \theta_i^t, \varphi_i^t) \middle\| E_i^t \middle\| HT_i^t \right\rangle \tag{2}$$

$Hello_i$ indicates the hello message sent by U_i, and HT_i^t is its broadcast period. In QRCF, this time is adaptively determined using Equation (8) and is continuously updated based on the connection status of neighboring UAVs. Furthermore, ID_{Hello_i} indicates the identifier related to this hello message. Note that each hello message has a unique ID to identify duplicated hello messages. Additionally, ID_i, (x_i^t, y_i^t, z_i^t), $(V_i^t, \theta_i^t, \varphi_i^t)$, and E_i^t mean identifier, position, movement information, and the current energy level of U_i, respectively.

5.1.2. Hello broadcast period

In QRCF, each UAV, such as U_i, has a specific broadcast period (HT_i^t) to carry out the broadcasting process of hello messages in FANET. HT_i^t is determined according to the connection status of adjacent UAVs. In a dynamic network, a dynamic broadcast period guarantees the adaptability of QRCF to this network. Because this dynamic broadcast period timely updates neighbor tables and decreases routing overhead in FANET. To determine HT_i^t, the two upper and lower boundaries (i.e. HT_H and HT_L) are intended so that $HT_L \le HT_i^t \le HT_H$, where HT_H and HT_L are respectively calculated according to the lowest and highest velocities of UAVs (i.e. $V_{\min}$ and $V_{\max}$). To calculate HT_i^t, QRCF applies Equation (3) to obtain a criterion called the connection time (CT_{ij}) of the link between U_i and its adjacent UAV, such as U_j.

$$CT_{ij} = \frac{R - \alpha_1 d_{ij}^t}{\left| V_i - \alpha_2 V_j \right|} \tag{3}$$

Here, R indicates the communication radius of each UAV, $d_{ij}^t = \sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2 + \left(z_i^t - z_j^t\right)^2}$ is the distance between U_i with spatial coordinates (x_i^t, y_i^t, z_i^t) and U_j with spatial coordinates (x_j^t, y_j^t, z_j^t). $\left|V_i - \alpha_2 V_j\right|$ indicates the configured relative speed of U_j with regard to U_i. It is calculated according to Equation (4).

$$\left|V_i - \alpha_2 V_j\right| = \sqrt{\left(V_{x_{i,j}}^{\alpha_2}\right)^2 + \left(V_{y_{i,j}}^{\alpha_2}\right)^2 + \left(V_{z_{i,j}}^{\alpha_2}\right)^2} \tag{4}$$

so that,

$$V_{x_{i,j}}^{\alpha_2} = V_i^t \sin\varphi_i^t \cos\theta_i^t - \alpha_2 V_j^t \sin\varphi_j^t \cos\theta_j^t \tag{5}$$

$$V_{y_{i,j}}^{\alpha_2} = V_i^t \sin\varphi_i^t \sin\theta_i^t - \alpha_2 V_j^t \sin\varphi_j^t \sin\theta_j^t \tag{6}$$

$$V_{z_{i,j}}^{\alpha_2} = V_i^t \cos\varphi_i^t - \alpha_2 V_j^t \cos\varphi_j^t \tag{7}$$

here $(V_i^t, \theta_i^t, \varphi_i^t)$ and $(V_j^t, \theta_j^t, \varphi_j^t)$ are extracted from the neighbor table and indicate the velocity vectors of U_i and U_j, respectively. Additionally, α_1 and α_2 are the configuration coefficients, which are calculated below.

- If movement directions of U_i and U_j are similar (i.e. $MD_{ij} = \sqrt{\left(V_i^t - V_j^t\right)^2 + \left(\theta_i^t - \theta_j^t\right)^2 + \left(\varphi_i^t - \varphi_j^t\right)^2} \le k$, where k is a positive number limited to $[0,1]$), then $\alpha_2 = 1$. Now, if U_i is faster than U_j and moves behind it, or if U_j is faster than U_i and moves behind it, then $\alpha_1 = -1$. Otherwise, if U_i is faster than U_j and moves ahead of it, or if U_j is faster than U_i and moves ahead of it, then $\alpha_1 = 1$.
- If movement directions of U_i and U_j are not similar (i.e. $MD_{ij} > k$), then $\alpha_2 = -1$. Now, if U_i and U_j approach to each other (i.e. $d_{ij}^t - d_{ij}^{t-1} \le 0$), then $\alpha_1 = -1$. Otherwise, $\alpha_1 = 1$.

Finally, HT_i^t is updated using Equation (8).

$$HT_i^{t+1} = \begin{cases} \frac{HT_H + HT_L}{2}, & t = 0 \\ \max\left(HT_i^t \exp\left(-\frac{\overline{CT_{ij}}}{\max\limits_{\forall U_j \in NTable_i} \{CT_{ij}\}} \right), HT_L \right), & e_{prediction} \ge e_{\max} \\ HT_i^t, & e_{\min} < e_{prediction} < e_{\max} \\ \min\left(HT_i^t \left(1 + \exp\left(-\frac{\overline{CT_{ij}}}{\max\limits_{\forall U_j \in NTable_i} \{CT_{ij}\}} \right) \right), HT_H \right), & e_{prediction} \le e_{\min} \end{cases} \tag{8}$$

Here, $\overline{CT_{ij}} = \frac{1}{N_i} \sum\limits_{j \in NTable_i} CT_{ij}$ and $\max\limits_{\forall U_j \in NTable_i} \{CT_{ij}\}$ mean two average and maximum connection times between U_i and its neighbors. $e_{prediction} = \sqrt{\left(x_i^{t+1} - \tilde{x}_i^{t+1}\right)^2 + \left(y_i^{t+1} - \tilde{y}_i^{t+1}\right)^2 + \left(z_i^{t+1} - \tilde{z}_i^{t+1}\right)^2}$ means the position prediction error of U_i so that $(x_i^{t+1}, y_i^{t+1}, z_i^{t+1})$ and $(\tilde{x}_i^{t+1}, \tilde{y}_i^{t+1}, \tilde{z}_i^{t+1})$ are real and predicted spatial coordinates of U_i in the moment $t+1$. As a result:

$$\begin{aligned} \tilde{x}_i^{t+1} &= x_i^t + \Delta t \left(V_i^t \sin\varphi_i^t \cos\theta_i^t \right) \\ \tilde{y}_i^{t+1} &= y_i^t + \Delta t \left(V_i^t \sin\varphi_i^t \sin\theta_i^t \right) \\ \tilde{z}_i^{t+1} &= z_i^t + \Delta t \left(V_i^t \cos\varphi_i^t \right) \end{aligned} \tag{9}$$

Table 3
Structure of $NTable_i$.

Information extracted from hello message				Information about the Q-learning-based routing process			
Identifier	**Position**	**Velocity**	**Energy**	**Discount factor**	**Q-value**	**Learning rate**	**Validity time**
ID_j	$\left(x_j^t, y_j^t, z_j^t\right)$	$\left(V_j^t, \theta_j^t, \varphi_j^t\right)$	E_j^t	γ_j	Q_j	α_{ij}	VT_j

According to the first line of Equation (8), U_i determines the initial value of HT_i^t at the moment $t = 0$ so that $HT_i^{t=0} = \frac{HT_H + HT_L}{2}$. Moreover, according to the second line of Equation (8), if the position prediction error is high (i.e. $e_{prediction} \geq e_{\max}$), HT_i^{t+1} is reduced based on $HT_i^t \exp\left(-\frac{\overline{CT_{ij}}}{\max\limits_{\forall U_j \in NTable_i}\{CT_{ij}\}}\right)$ so that its minimum value is limited to HT_L. In addition, according to the third line of Equation (8), if the position prediction error is acceptable (i.e. $e_{\min} < e_{prediction} < e_{\max}$), HT_i^t will be unchanged. Also, according to the fourth line of Equation (8), if the position prediction error is very low (i.e. $e_{prediction} \leq e_{\min}$), HT_i^{t+1} is increased based on $HT_i^t\left(1 + \exp\left(-\frac{\overline{CT_{ij}}}{\max\limits_{\forall U_j \in NTable_i}\{CT_{ij}\}}\right)\right)$ so that its maximum value is limited to HT_H.

5.1.3. Construction of the neighbor table

In QRCF, each UAV, such as U_i, prepares its hello message according to Equation (2) and inserts its information into this message. Then, U_i dispatches this message to its surrounding UAVs in the network and updates its hello period based on Equation (8) to determine when the hello broadcast process is repeated. Thereafter, U_i obtains hello messages from its adjacent UAVs and extracts their information. Now, U_i has access to the local network topology based on the content of these messages and holds this information in its neighbor table ($NTable_i$). $NTable_i$ must be refreshed constantly to delete its old information and record the new information. If U_i records the information of a new neighbor, such as U_j, in $NTable_i$, it considers a timer named validity time (VT_j) for the entry related to U_j. The initial value of VT_j is equal to the upper boundary of the hello broadcast period (i.e. HT_H). Then, VT_j decreases over time. If U_i gets a new hello message from U_j before expiring its validity time in $NTable_i$, the information in this entry will be updated, and VT_j will be reset. Otherwise, this entry will lose its validity and will be removed from $NTable_i$. Table 3 shows the format of $NTable_i$. The entry related to each neighboring UAV, such as U_j, includes the information received from the hello message and the learning parameters related to the Q-learning-based routing process. These parameters include the learning rate (α_{ij}), the discount factor of this neighboring node (γ_j), and its Q-value. In Section 5.2, QRCF defines these parameters accurately.

5.2. Q-learning-based routing process

QRCF applies the Q-learning algorithm to identify valid and stable routes between UAVs in FANET. This distributed learning process benefits from an attractive feature, namely the use of a filtered state set based on a cylindrical filtering algorithm. This feature speeds up the Q-learning algorithm when finding the best path because QRCF utilizes a small state space. In addition, QRCF sets learning rate and discount factor in Q-learning based on the conditions of the learning environment to improve its adaptability to FANET. In Table 4, learning elements are introduced. Thus, QRCF takes into account two key learning components: the learning environment, which corresponds to FANET, and the agent, which corresponds to the data packet. The state set also includes a set of UAVs enclosed in the cylindrical filtering area, and the action space denotes the packet transfer process from the current UAV to the UAV selected from the filtered state set. Thus, as shown in Fig. 3, if the agent takes the action $U_i - to - U_j$, the data packet will be transferred from U_i to U_j. This process has four steps:

Algorithm 1 Adaptive neighbor discovery process.

Input: U_i: i-th UAV in FANET, where $i = 1, 2, ..., N$
HT_i^t: Hello time interval corresponding to U_i
HT_H: Upper boundary of HT_i^t, adjusted based on $V_{\min}$
HT_L: Lower boundary of HT_i^t, adjusted based on $V_{\max}$
T_{Net}: Simulation time
T_{timer}: A timer that counts the spent time so far.
Output: $NTable_i$: Neighbor table related to U_i
Begin
Initialize $T_{timer} = 0$;
U_i: Adjust the initial value of HT_i^t on $\frac{HT_H + HT_L}{2}$;
repeat
  if $\left(T_{timer} \bmod HT_i^t\right) = 0$ **then**
    U_i: Prepare a hello packet based on Equation (2);
    U_i: Disseminate this hello packet to its surrounding UAVs in FANET;
    U_i: Recalculate HT_i^t based on Equation (8);
  end if
  while there is a new neighbor (like U_j), which sends a new hello packet to U_i **and** U_i does not process this packet so far **do**
    if U_i does not record U_j in $NTable_i$ so far **then**
      U_i: Extract the information of U_j from this hello packet;
      U_i: Add a new entry to $NTable_i$;
      U_i: Record the information of U_j in $NTable_i$;
      U_i: Set $VT_j = HT_H$;
    else
      U_i: Extract the information of U_j from this hello packet;
      U_i: Update the information of U_j in $NTable_i$;
      U_i: Reset $VT_j = HT_H$;
    end if
  end while
  while there is a neighboring node (like U_j) in $NTable_i$, which has not been checked by U_i **do**
    if U_i does not receive a hello message from U_j **and** VT_j equals to zero **then**
      U_i: Remove the entry of U_j from $NTable_i$;
    end if
  end while
  $T_{timer} = T_{timer} + 1$;
until $T_{timer} \leq T_{Net}$
End

- Filtered state set using cylindrical filtering
- Reward function
- Dynamic learning metrics
- Prevention mechanism

5.2.1. Filtered state set using cylindrical filtering

Most Q-learning-based routing schemes, such as [48] and [49], assume that the state set consists of all surrounding UAVs of U_i, which are enclosed in a spherical region with the radius of R, when the data packet is in the current state, like U_i. This state set is shown in Fig. 4. According to this figure, the state set of U_i is $State.set = \{U_1, U_2, ..., U_7\}$. These schemes hardly converge to the optimal response (the most suitable paths between UAVs). Specifically, if the network density is high, these schemes often fail to reach the optimal solution. This challenge is due to the high dependence of Q-learning on the size of the state set. The potential response to this issue is to manage the state set in the Q-learning-based routing schemes. If one of these schemes limits the size of the state set, its convergence speed will be increased when finding the best routes between UAVs. Thus, its efficiency will be guaranteed to solve the routing problem. According to the points mentioned above, QRCF utilizes a filtered state set using smart cylindrical filtering to manage its state set. See this set in Fig. 5. According to this figure, the filtered state set of U_i (FS_i^{Set}) is $FS_i^{Set} = \{U_4, U_5, U_6, U_7\}$.

In QRCF, the process to determine FS_i^{Set} is described using the cylindrical coordinate system, as FS_i^{Set} represents a cylindrical region defined within the communication range of U_i, where both radius and

Table 4
Learning elements in the routing process.

Learning element	Definition
Problem	Finding the best routes between UAVs.
Learning environment	Flying ad hoc network
Learning agent	Data packet
State space	UAVs enclosed in the cylindrical filtering area.
Action space	Transferring the data packet from the current UAV to the next-hop UAV.

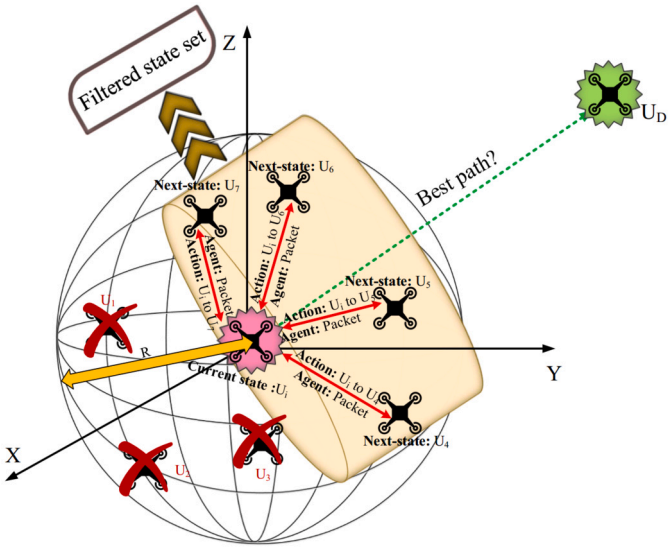

Fig. 3. Learning elements for the routing process in QRCF.

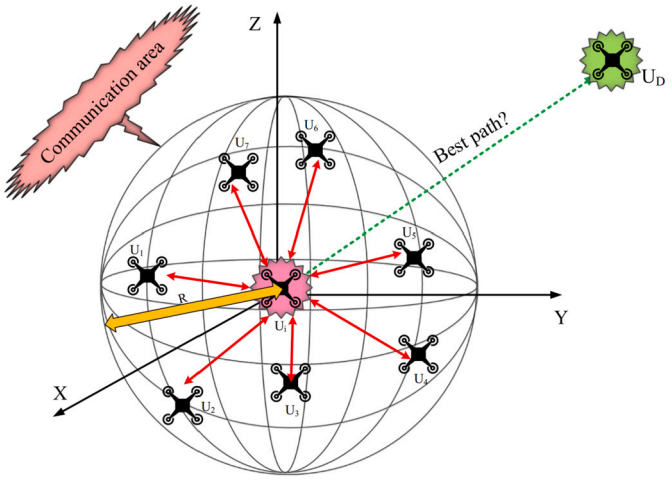

Fig. 4. The initial state set in Q-learning-based routing schemes.

height are R. Suppose that the Cartesian and cylindrical coordinates of the point P are (x_P, y_P, z_P) and (ρ_P, Θ_P, z_P), respectively. In this case, Equations (10) and (11) show the relationship between these two coordinate systems. See Fig. 6.

$$\begin{aligned} \rho_P &= \sqrt{{x_P}^2 + {y_P}^2} \\ \Theta_P &= \arctan\left(\frac{y_P}{x_P}\right) \\ z_P &= z_P \end{aligned} \tag{10}$$

$$\begin{aligned} x_P &= \rho_P \cos\Theta_P \\ y_P &= \rho_P \sin\Theta_P \\ z_P &= z_P \end{aligned} \tag{11}$$

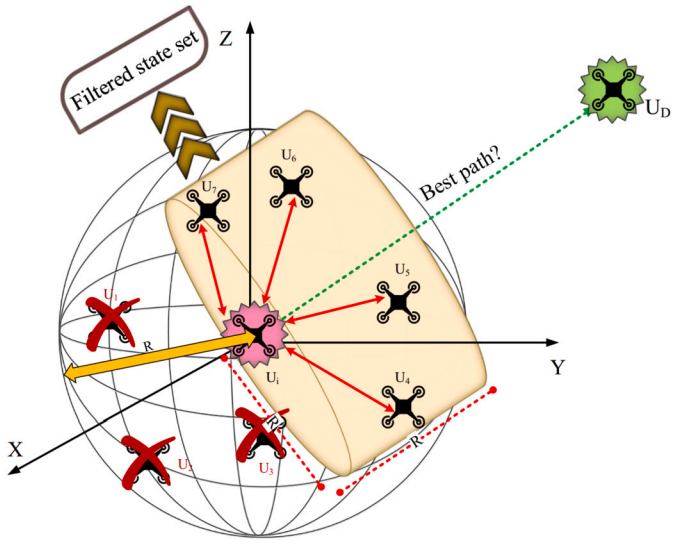

Fig. 5. Filtered state set using cylindrical filtering.

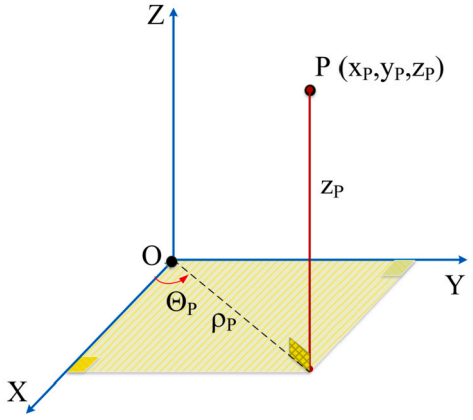

Fig. 6. Comparison of two cylindrical and Cartesian coordinates of the point P.

Now, U_i repeats the following steps for each neighboring UAV in $NTable_i$ to get FS_i^{Set}. Algorithm 2 also presents the pseudo-code related to this process.

1. U_i inserts all members of $NTable_i$ into FS_i^{Set} so that $FS_i^{Set} = NTable_i$.

2. U_i defines a coordinate system whose origin is (x_i^t, y_i^t, z_i^t) to simplify the equations related to the calculation of FS_i^{Set}.

3. U_i extracts a member, such as U_j with spatial coordinates $\left(x_j^t, y_j^t, z_j^t\right)$, from $NTable_i$ and acquires its new coordinates $\left(\dot{x}_j^t, \dot{y}_j^t, \dot{z}_j^t\right)$ in this coordinate system based on Equation (12).

$$\begin{aligned} \dot{x}_j^t &= x_j^t - x_i^t \\ \dot{y}_j^t &= y_j^t - y_i^t \\ \dot{z}_j^t &= z_j^t - z_i^t \end{aligned} \tag{12}$$

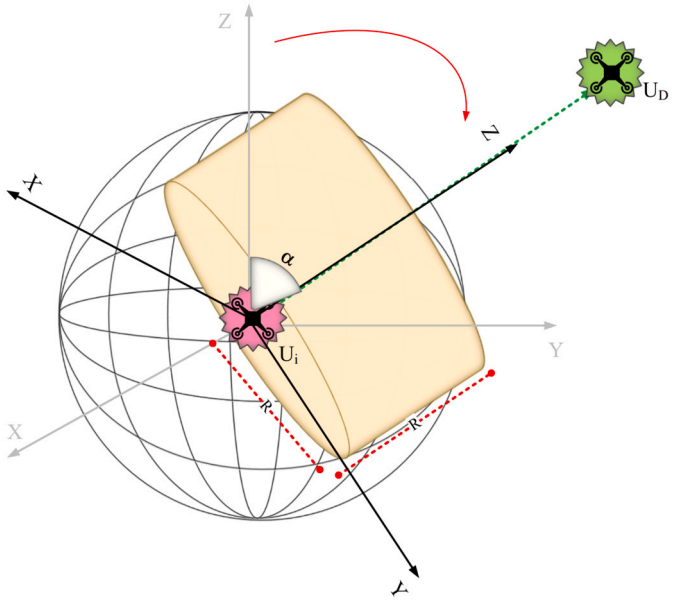

Fig. 7. Rotation of the coordinate system based on the angle α.

4. According to Fig. 7, U_i rotates the coordinate axes based on the angle α (i.e. the angle between the line segment $U_i - U_D$ and the z-axis). In this case, the z-axis is placed on the line segment $U_i - U_D$. Hence, α is computed based on Equation (13).

$$\alpha = \arccos\left(\frac{{z_d}^t - {z_i}^t}{\sqrt{\left({x_d}^t - {x_i}^t\right)^2 + \left({y_d}^t - {y_i}^t\right)^2 + \left({z_d}^t - {z_i}^t\right)^2}}\right), \quad 0 \le \alpha \le 2\pi \tag{13}$$

The rotation matrix is presented in Equation (14).

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

5. U_i gets the new coordinates of U_j i.e. $\left(\widehat{x}_j^t, \widehat{y}_j^t, \widehat{z}_j^t\right)$ in this coordinate system based on Equation (15).

$$\begin{bmatrix} \widehat{x}_j^t \\ \widehat{y}_j^t \\ \widehat{z}_j^t \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_j^t \\ \dot{y}_j^t \\ \dot{z}_j^t \end{bmatrix} \tag{15}$$

6. U_i determines the cylindrical filtering area based on Equation (16).

$$0 \le \rho \le R, 0 \le \Theta \le 2\pi, 0 \le z \le R \tag{16}$$

7. U_i changes the Cartesian coordinates of its neighboring node i.e. $\left(\widehat{x}_j^t, \widehat{y}_j^t, \widehat{z}_j^t\right)$ to the cylindrical coordinates based on Equation (17).

$$\begin{aligned} \rho_j^t &= \sqrt{{\widehat{x}_j^t}^2 + {\widehat{y}_j^t}^2} \\ \Theta_j^t &= \arctan\left(\frac{\widehat{y}_j^t}{\widehat{x}_j^t}\right) \\ z_j^t &= \widehat{z}_j^t \end{aligned} \tag{17}$$

8. If $\left(\rho_j^t, \Theta_j^t, \widehat{z}_j^t\right)$ meets Equation (18), U_j will remain as a member in FS_i^{Set}. Otherwise, it will be removed from this set.

$$0 \le \rho_j^t \le R, 0 \le \Theta_j^t \le 2\pi, 0 \le \widehat{z}_j^t \le R \tag{18}$$

Algorithm 2 Filtered state set using cylindrical filtering.

Input: U_i: Current state in the routing strategy based on Q-learning.
$\left(x_i^t, y_i^t, z_i^t\right)$: Geographic position of U_i
$NTable_i$: Neighbor table related to U_i
Output: FS_i^{Set}: Filtered state set
Begin
1: **U_i:** Initialize its filtered state set as $FS_i^{Set} = NTable_i$;
2: **U_i:** Define a new coordinate system whose origin is equal to $\left(x_i^t, y_i^t, z_i^t\right)$;
3: **while** there is a neighboring node (like U_j) in $NTable_i$, which has not been checked by U_i **do**
4: **U_i:** Extract U_j from $NTable_i$;
5: **U_i:** Get the new spatial coordinates of U_j in this new coordinate system using Equation (12);
6: **U_i:** Calculate the angle α between the z-axis and the line segment $U_i - U_D$ based on Equation (13);
7: **U_i:** Obtain the rotation matrix based on Equation (14);
8: **U_i:** Perform a rotation on the coordinate axes based on the angle α;
9: **U_i:** Calculate the rotated coordinates of U_j based on Equation (15);
10: **U_i:** Get the filtering area using Equation (16);
11: **U_i:** Obtain the cylindrical coordinates of U_j using Equation (17);
12: **if** cylindrical coordinates of U_j meets Equation (18) **then**
13: **U_i:** Store U_j into FS_i^{Set};
14: **else**
15: **U_i:** Remove U_j from FS_i^{Set};
16: **end if**
17: **end while**
18: **U_i:** Return FS_i^{Set};
End

5.2.2. Reward function

After determining the filtered state space and selecting a suitable action, such as $U_i - to - U_j$, by the agent, the next step is to estimate the reward value corresponding to this action and assign it to the learning agent. Here, QRCF calculates the reward function based on four components, namely relative speed, connection time, residual energy, and movement path.

- **Relative Speed (V_{ij}):** This component means the relative velocity of U_j with regard to U_i. It seeks to construct a communication path, which includes intermediate UAVs with a relatively similar velocity. In this case, this path is valid and stable for a longer time. If $V_{ij} = 0$, the communication link between U_i and U_j is very stable and has a longer lifespan because the two UAVs will not be removed from each other's communication ranges. Thus, the packet loss rate will be severely low. V_{ij} is obtained using Equation (19).

$$\left|\overrightarrow{V_{ij}}\right| = \sqrt{\left(V_{x_{i,j}}^t\right)^2 + \left(V_{y_{i,j}}^t\right)^2 + \left(V_{z_{i,j}}^t\right)^2} \tag{19}$$

so that,

$$V_{x_{i,j}}^t = V_i^t \sin\varphi_i^t \cos\theta_i^t - V_j^t \sin\varphi_j^t \cos\theta_j^t \tag{20}$$

$$V_{y_{i,j}}^t = V_i^t \sin\varphi_i^t \sin\theta_i^t - V_j^t \sin\varphi_j^t \sin\theta_j^t \tag{21}$$

$$V_{z_{i,j}}^t = V_i^t \cos\varphi_i^t - V_j^t \cos\varphi_j^t \tag{22}$$

so that $\left(V_i^t, \theta_i^t, \varphi_i^t\right)$ and $\left(V_j^t, \theta_j^t, \varphi_j^t\right)$ indicate the two velocity vectors related to U_i and U_j, respectively. They are extracted from $NTable_i$. In QRCF, the normalized value of V_{ij} is calculated using Equation (23). After the normalization process, the value of this component will be limited to $[0, 1]$. Since the reward function contains four components with different units. Hence, the normalization process equalizes the effect of these components on the reward function.

$$\left|\overrightarrow{V}_{ij}^{norm}\right| = \frac{\left|\overrightarrow{V}_{ij}\right| - \min\limits_{\forall U_j \in FS_i^{Set}} \left\{\left|\overrightarrow{V}_{ij}\right|\right\}}{\max\limits_{\forall U_j \in FS_i^{Set}} \left\{\left|\overrightarrow{V}_{ij}\right|\right\} - \min\limits_{\forall U_j \in FS_i^{Set}} \left\{\left|\overrightarrow{V}_{ij}\right|\right\}} \tag{23}$$

here $\max_{\forall U_j \in FS_i^{Set}} \left\{ \left| \overrightarrow{V}_{ij} \right| \right\}$ and $\min_{\forall U_j \in FS_i^{Set}} \left\{ \left| \overrightarrow{V}_{ij} \right| \right\}$ mean the maximum and minimum relative velocities of UAVs in FS_i^{Set} with regard to U_i, respectively.

- **Connection time (CT_{ij}):** This component means the connection time related to the link between U_i and U_j. It seeks to construct a communication path, which includes intermediate UAVs with strong connections. In this case, the constructed path has a longer lifetime. If CT_{ij} is high, U_i and U_j do not get out of each other's communication ranges and experience a less number of link failures. This increases throughput and facilitates the data transfer process. According to Section 5.1.2, CT_{ij} is calculated through Equation (3). Then, it is normalized using Equation (24).

$$CT_{ij}^{norm} = \frac{CT_{ij} - \min_{\forall U_j \in FS_i^{Set}} \{CT_{ij}\}}{\max_{\forall U_j \in FS_i^{Set}} \{CT_{ij}\} - \min_{\forall U_j \in FS_i^{Set}} \{CT_{ij}\}} \tag{24}$$

here $\max_{\forall U_j \in FS_i^{Set}} \{CT_{ij}\}$ and $\min_{\forall U_j \in FS_i^{Set}} \{CT_{ij}\}$ mean the maximum and minimum connection times related to links between UAVs in FS_i^{Set} and U_i.

- **Residual energy (E_j^t):** This component means the energy level of U_j and seeks to construct a communication path, which involves high-energy intermediate UAVs so that this route does not fail due to the lack of enough energy of intermediate UAVs. It is registered in $NTable_i$, and Equation (25) is used to get the normalized value of this component.

$$E_j^{norm} = \frac{E_j^t - \min_{\forall U_j \in FS_i^{Set}} \left\{ E_j^t \right\}}{\max_{\forall U_j \in FS_i^{Set}} \left\{ E_j^t \right\} - \min_{\forall U_j \in FS_i^{Set}} \left\{ E_j^t \right\}} \tag{25}$$

here $\max_{\forall U_j \in FS_i^{Set}} \left\{ E_j^t \right\}$ and $\min_{\forall U_j \in FS_i^{Set}} \left\{ E_j^t \right\}$ mean the maximum and minimum residual energies of UAVs in FS_i^{Set}.

- **Movement path (λ_j):** This component means the movement path of U_j with regard to U_i and seeks to construct the communication path, which involves intermediate UAVs with the movement angle almost close to each other. In this case, these UAVs move at same direction ($\lambda_j = 0$) to decrease the number of hops and delay in this path. For calculating λ_j, consider Fig. 8. According to this figure, a triangle is drawn between three nodes, namely U_i, U_j, U_D, and λ_j shows the angle between $\overrightarrow{A}$ and $\overrightarrow{B}$. This angle is calculated based on Equation (26).

$$\lambda_j = cos^{-1}\left(\frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{|A|\,|B|} \right) \tag{26}$$

so that,

$$\overrightarrow{A} = \underbrace{\left(x_i^t - x_D^t\right)}_{a_1} i + \underbrace{\left(y_i^t - y_D^t\right)}_{a_2} j + \underbrace{\left(z_i^t - z_D^t\right)}_{a_3} k \tag{27}$$

And,

$$\overrightarrow{B} = \underbrace{\left(x_i^t - x_j^t\right)}_{b_1} i + \underbrace{\left(y_i^t - y_j^t\right)}_{b_2} j + \underbrace{\left(z_i^t - z_j^t\right)}_{b_3} k \tag{28}$$

so,

$$\left|\overrightarrow{A}\right| = \sqrt{\left(x_i^t - x_D^t\right)^2 + \left(y_i^t - y_D^t\right)^2 + \left(z_i^t - z_D^t\right)^2} \tag{29}$$

And,

$$\left|\overrightarrow{B}\right| = \sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2 + \left(z_i^t - z_j^t\right)^2} \tag{30}$$

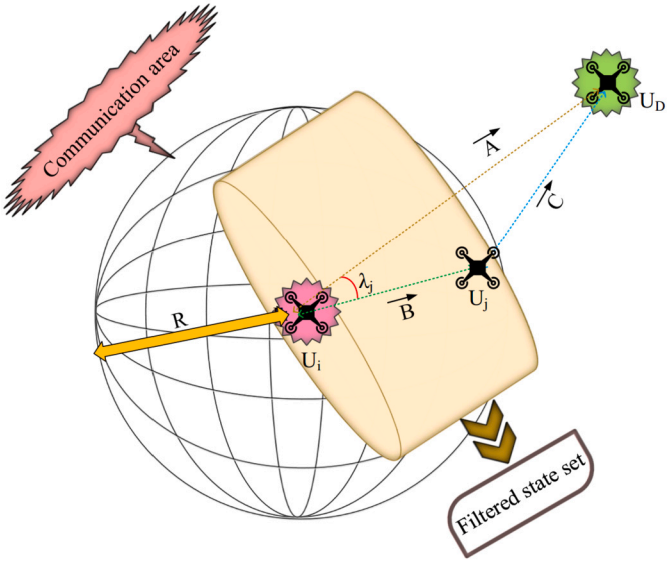

Fig. 8. Movement path of U_j with regard to U_i.

Finally, the normalized value of λ_j is obtained using Equation (31).

$$\lambda_j^{norm} = \frac{\lambda_j - \min_{\forall U_j \in FS_i^{Set}} \{\lambda_j\}}{\max_{\forall U_j \in FS_i^{Set}} \{\lambda_j\} - \min_{\forall U_j \in FS_i^{Set}} \{\lambda_j\}}, \quad 0 \le \lambda_j \le \pi \tag{31}$$

here $\max_{\forall U_j \in FS_i^{Set}} \{\lambda_j\}$ and $\min_{\forall U_j \in FS_i^{Set}} \{\lambda_j\}$ mean the maximum and minimum movement paths of the UAVs in FS_i^{Set} with regard to U_i.

Hence, the reward value (R_t) is calculated based on Equation (32).

$$R_t = \begin{cases} R_{\max}, & U^{t+1} \text{ is destination} \\ R_{\min}, & U^{t+1} \text{ is a local minimum} \\ \omega_1\left(1 - \left|\overrightarrow{V}_{ij}^{norm}\right|\right) + \omega_2 CT_{ij}^{norm} + \omega_3 E_j^{norm} + \omega_4\left(1 - \lambda_j^{norm}\right), & Otherwise \end{cases} \tag{32}$$

so that ω_1, ω_2, ω_3, ω_4 are weight coefficients, and their sum is equal to one ($\sum_{i=1}^{4} \omega_i = 1$). In the first line of this equation, if U_j and U_D are the same, then the environment gives the maximum reward, $R_{\max}$, to the agent when taking the action $U_i - to - U_j$. In the second line of Equation (32), if U_j is local optimal, and all UAVs in FS_i^{Set} have more distance to U_D, then the environment gives the least award, $R_{\min}$, to the learning agent when taking the action $U_i - to - U_j$. Otherwise, according to the third line of this equation, R_t is calculated based on the four components, namely relative speed, connection duration, residual energy, and movement path.

5.2.3. Dynamic learning factors

As stated in Section 5.1.3, in addition to the information about each neighboring node, $NTable_i$ also includes the learning parameters related to Q-learning. These parameters consist of the learning rate (α_{ij}), which pertains to the connection between U_i and U_j, as well as the discount factor for U_j (γ_j). These values are computed dynamically to ensure that each UAV possesses its distinct learning rate and discount factor. In QRCF, this feature improves adaptability to FANET and enhances network performance. In the following, these learning parameters are defined and calculated.

- **Learning rate (α_{ij}):** This element governs the speed at which the Q-learning algorithm learns. It illustrates the effect of new and old information on the learning phase, and its value is limited to [0, 1].

QRCF calculates this learning component based on the connection time between U_i and U_j (i.e. CT_{ij}). CT_{ij} and α_{ij} have an opposite relationship to each other. If the link between U_i and U_j has a long connection time, then this link is more stable. As a result, α_{ij} can approach zero to perform the routing phase with a focus on old information. Otherwise, if the link between U_i and U_j includes a short connection time, this link is unstable. As a result, α_{ij} is close to one so that the routing phase is performed by focusing on new information. For calculating α_{ij}, CT_{ij} is obtained from Equation (3) in Section 5.1.2. Then, it is normalized using Equation (24). Finally, QRCF uses Equation (33) to calculate α_{ij}.

$$\alpha_{ij} = \cosh\left(-CT_{ij}^{norm}\right) + \sinh\left(-CT_{ij}^{norm}\right) \tag{33}$$

- **Discount factor (γ_j):** This component plays a crucial role in balancing exploration and exploitation by adjusting the importance of the reward value during the learning process. γ_j is defined in the interval [0, 1]. In QRCF, U_i aims to select the most stable nearby UAV as the next-hop. To achieve this, U_j computes γ_j based on the stability of its adjacent UAVs in two consecutive periods. If U_j has stable neighbors, γ_j will be closer to one, indicating a stable Q-value. In this case, the learning agent focuses on exploiting its past experiences in the environment. Conversely, if U_j lacks stable neighbors, γ_j approaches zero, signaling an unstable Q-value. Thus, the learning agent the learning agent switches to exploring the environment to gather new experiences. γ_j is determined using Equation (34) [54].

$$\gamma_j = 1 - \frac{\left|N_j(t-1) \cup N_j(t)\right| - \left|N_j(t-1) \cap N_j(t)\right|}{\left|N_j(t-1) \cup N_j(t)\right|} \tag{34}$$

Here $N_j(t-1)$ and $N_j(t)$ are the number of neighbors of U_j in two moments $t-1$ and t, respectively.

5.2.4. Prevention mechanism

The prevention mechanism in QRCF seeks to avoid situations, like routing holes or failed links, which can damage communication routes and lead to packet loss. To avoid these events, QRCF must not be trapped in local optimization. Hence, if the next-hop node (i.e. U_j) is a local optimum, it will dispatch feedback for its previous-hop UAV (i.e. U_i). As a result, the reward value for the action $U_i - to - U_j$ will equal $R_{\min}$, and the Q-table will be refreshed accordingly. As a result, the likelihood of the reselection of this action in the future is reduced, and QRCF is not trapped in the local optimization. In addition, if U_i fails to communicate with its neighboring UAV (i.e. U_j), the link between these UAVs will be cut off. Hence, the reward value corresponding to the action $U_i - to - U_j$ will be equal to $R_{\min}$, and the Q-table will be renewed accordingly. Algorithm 3 describes the pseudo-code related to the routing phase in QRCF.

Algorithm 3 Q-learning-based routing process.

Input: U_i: Current state in the routing technique based on Q-learning.
FS_i^{Set}: Filtered state set
ε, α_{ij}, γ_j, and M: Learning parameters
Output: Q-table: Routing table stored in U_i
Begin
U_i: Select ε randomly in [0, 1];
U_i: Set the initial value of each Q-value in Q-table on zero;
while $episode \leq M$ **do**
  U_i: Calculate FS_i^{Set} using Algorithm 2;
  U_i: Extract the next state (i.e. U_j) from FS_i^{Set} randomly;
  U_i: Compute α_{ij} and γ_j using Equations (33) and (34), respectively;
  for $t = 1$ to N **do**
    U_i: Generate a random number (n_r) in [0, 1];
    if $n_r \leq \varepsilon$ **then**
      U_i: Carry out its action according to the ε-greedy strategy;
    else
      U_i: Select its action from Q-table;
    end if
    if U_j and U_D are the same **then**
      $R_t = R_{\max}$:
    else if U_j is trapped into a local optimization **then**
      $R_t = R_{\min}$;
    else
      $R_t = \omega_1\left(1 - \left|\overline{V}_{ij}^{norm}\right|\right) + \omega_2 CT_{ij}^{norm} + \omega_3 E_j^{norm} + \omega_4\left(1 - \lambda_j^{norm}\right)$:
    end if
    U_i: Extract the next state from FS_i^{Set} randomly;
    Agent: Change its state to the new state;
    Agent: Update Q-table based on the reward value;
    $t = t + 1$;
  end for
  $episode = episode + 1$;
end while
End

6. Simulation results

To assess the efficiency of QRCF, simulations were conducted using the network simulator 2 (NS2) [55]. The outcomes are described in comparison with different routing schemes, namely QRF [42], QFAN [48], and QTAR [49]. These assessments are measured across five criteria: used energy, data delivery ratio, overhead, latency, and network longevity based on the number of UAVs and their speed. In the simulation process, the network environment is defined by the dimensions of $2000 \times 2000 \times 300\, m^3$. It contains 20 to 100 UAVs that are randomly scattered in this environment. These UAVs fly at the speed $10 - 40\ m/s$, and follow the three-dimensional Gauss-Markov mobility model (3D GM). In addition, their primary energy and communication radius are equal to 1000 Jules and 250 meters, respectively. This network adheres to the IEEE 802.11n communication standard, which benefits from features, such as high throughput, high data rate, and excellent quality and long-range communication [50,51]. Note that we repeat the experiments for 30 runs and then present the average results with 95% confidence interval to provide an accurate analysis. See the most important simulation parameters in Table 5. Evaluation scales are defined as follows:

- **Consumed energy:** This refers to the average energy used by each UAV while transmitting data packets to other UAVs within the network. It is a very important criterion for evaluating a routing scheme.
- **Packet delivery rate (PDR):** This scale measures the ratio of packets delivered to the destination to the sent packets. It is calculated using Equation (35).

$$PDR = \frac{\sum_{r=1}^{n_r} PK_r}{\sum_{s=1}^{n_s} PK_s} \times 100 \tag{35}$$

here PK_r, PK_s, n_r, and n_s represent the packets come into the destination node, the packets sent by the source node, the number of the entered packets, and the number of the sent packets, respectively.

- **Routing overhead (RO):** This metric represents the proportion of control packets to data packets transmitted during the data transfer process. It is obtained using Equation (36).

$$RO = \frac{\sum_{h=1}^{n_{Hello}} Hello_h}{\sum_{s=1}^{n_s} PK_s} \tag{36}$$

where PK_s, $Hello_h$, n_s, n_{Hello} indicate data packets, hello packets, the number of data packets, and the number of hello packets, respectively.

- **Delay:** End-to-end delay (EED) denotes the average duration taken for data to travel from the source node to the destination node. It is obtained through Equation (37).

Table 5
Simulation parameters.

Parameter	Value
Simulation tool	NS2
Compared schemes	QRCF, QRF, QFAN, QTAR
Network dimensions	$2000 \times 2000 \times 300\, m^3$
Number of UAVs in the network	20-100 nodes
Velocity of UAVs	10-40 m/s
Communication radius	250 m
Primary energy	1000 J
Movement model	3D GM
Communication standard	IEEE 802.11n
Antenna	Omni-directional
Traffic model	Constant bit rate (CBR)
Data rate	2 Mbps
Transport protocol	UDP
Carrier frequency	2.4 GHz

$$EED = \frac{\sum_{PK_i \in P = \{PK_1, \ldots, PK_n\}} \left(T_R\left(PK_i\right) - T_S\left(PK_i\right)\right)}{\sum_{r=1}^{n_r} PK_r} \tag{37}$$

where PK_r, PK_i, n_r, and P denote the packets come into the destination node, i-th data packet, the number of packet reached the destination, and the set of all data packets, respectively. Also, $T_R\left(PK_i\right)$ and $T_S\left(PK_i\right)$ mean two recipient and sent time moments related to PK_i, respectively.

- **Network longevity:** It expresses the time period of the routing process from the network bootstrapping time until the first node dies (FND).

6.1. Consumed energy

Fig. 9 shows the energy used by UAVs in various approaches as a function of the number of UAVs. The results reveal that energy consumption in both the two methods, QRCF and QRF, are very close to each other. QRCF lowers the average energy consumption by 5.01%, 22.32%, and 33.78% in comparison with QRF, QFAN, and QTAR, respectively. According to Fig. 9, a direct relationship is displayed between energy consumption and the number of UAVs, where energy use is higher in denser networks compared to sparser ones. In dense networks, the exchange of control messages between UAVs increases, and the neighboring tables stored in UAVs are larger. As a result, the state set in Q-learning will also grow, leading to a decrease in its convergence rate. This causes flying nodes to consume more energy when finding the optimal path. Additionally, Fig. 10 evaluates the energy used by UAVs under various schemes as a function of the speed of UAVs. In this figure, QRCF reduces energy consumption by 4.94%, 26.14%, and 35.85% in comparison with QRF, QFAN, and QTAR, respectively. Clearly, when UAVs fly faster in a FANET, their connections become less stable, and communication paths tend to be disrupted earlier. This requires UAVs to expend more energy on sending additional hello messages, updating their neighbor tables, and recalculating communication routes. Generally, Figs. 9 and 10 indicate that QRCF outperforms other routing schemes in terms of energy efficiency. This has a few reasons. First, QRCF constantly refreshes the hello broadcast period based on the average connection time of links between itself and neighboring UAVs. This dynamic update improves QRCF's adaptability to FANETs and reduces the need for excessive control message exchanges during routing, effectively managing the energy consumption associated with hello message exchanges. However, QRF, QFAN, and QTAR do not have such a mechanism, making it difficult for them to control the volume of hello messages exchanged within the network. Additionally, the Q-learning-based routing algorithm uses a reward function, which contains the energy parameter. Thus, this algorithm seeks to find paths with intermediate nodes that have sufficient energy, thus optimizing energy usage and enhancing network longevity.

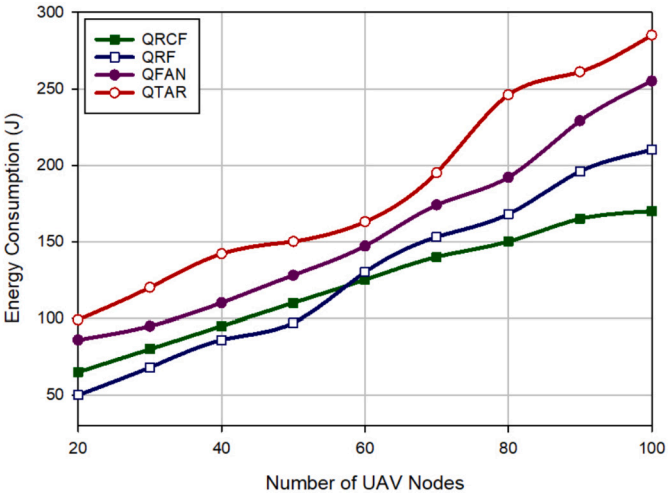

Fig. 9. Energy consumption based on the number of nodes.

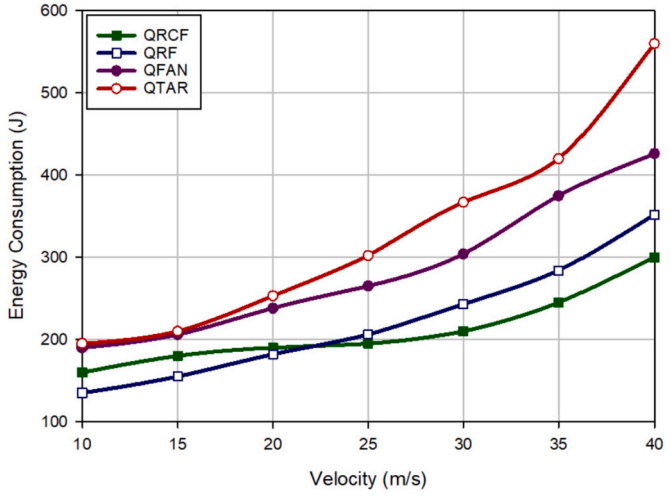

Fig. 10. Energy consumption based on the velocity of UAVs.

6.2. Packet delivery rate

Fig. 11 presents an evaluation of the packet delivery rate across various schemes as a function of the number of UAVs. In this figure, QRCF enhances PDR by 1.20%, 2.31%, and 6.87% in comparison with QRF, QFAN, and QTAR. Clearly, when the number of nodes in FANET is growing, PDR is also ascending in all schemes. This result is because UAVs in dense networks have more stable connections and experience fewer link failures. Hence, the constructed routes will be valid for a longer period, and the number of lost packets caused by the instability of the paths will be reduced. Furthermore, Fig. 12 evaluates PDR under different schemes in relation to UAV speed. As shown in this figure, QRCF enhances PDR by 2.36%, 2.85%, and 7.63% in comparison with QRF, QFAN, and QTAR, respectively. Also, this figure highlights an inverse relationship between PDR and UAV speed. Increasing UAVs' speed leads to less stable communication within the FANET, resulting in shorter route lifetimes and higher chances of packet loss due to disrupted connections. Overall, Figs. 11 and 12 show that QRCF has a higher packet delivery rate than other methods. This performance boost is primarily due to the Q-learning-based routing algorithm's reward function, which incorporates a time connection parameter (CT_{ij}). As a result, QRCF focuses on the stability of communication links in the routing process and chooses

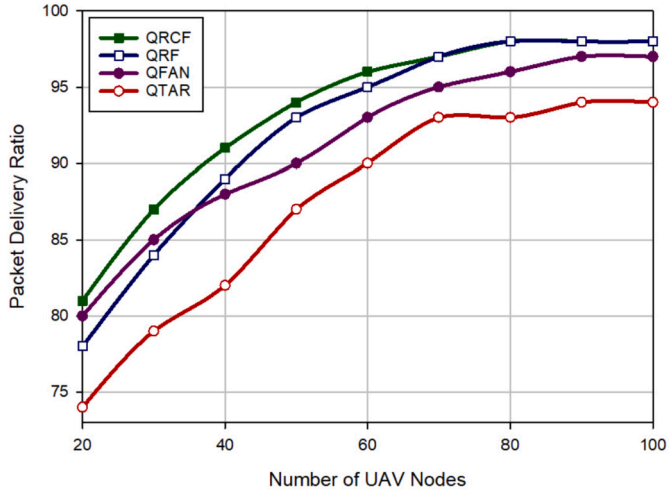

Fig. 11. Packet delivery rate based on the number of UAVs.

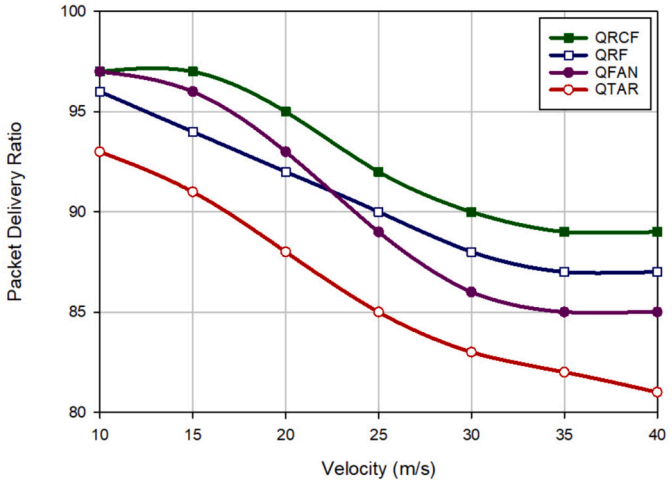

Fig. 12. Packet delivery rate based on different speeds of UAVs.

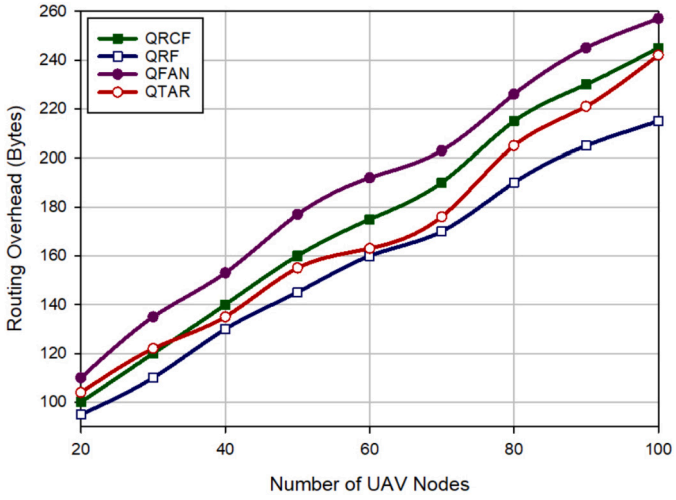

Fig. 13. Communication overhead based on the number of UAVs.

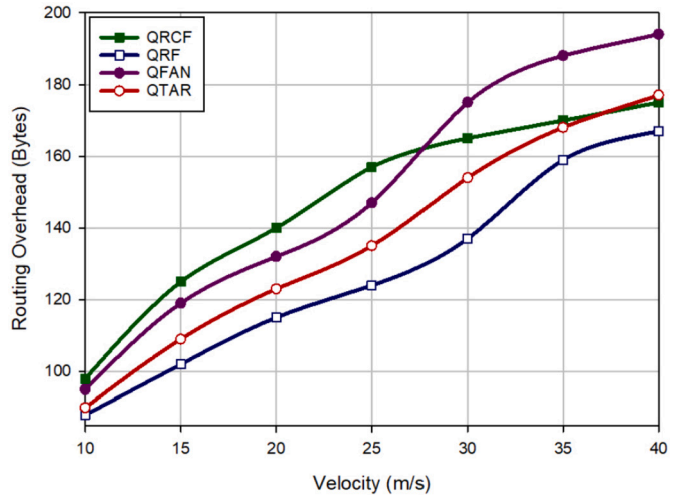

Fig. 14. Communication overhead based on different speeds of UAVs.

more stable connections to send data. This will make an increase in PDR.

6.3. Routing overhead

Fig. 13 illustrates routing overhead across different routing schemes, with respect to the number of nodes. The outcomes demonstrate that the overhead in QRCF is 10.91% and 3.41% more than that in QRF and QTAR, respectively. However, QRCF has about 7.24% less overhead than QFAN. This figure also highlights a clear relationship between routing overhead and the number of UAVs. Obviously, when the number of nodes is growing, these UAVs must exchange more hello messages to know the status of their neighbors in FANET. As a result, the overhead of all routing schemes will be raised when increasing the number of UAVs. In addition, Fig. 14 compares routing overhead in different approaches according to the speed of flying nodes. As shown in this figure, the routing overhead of QRCF is 15.47% and 7.74% more than that of QRF and QTAR, respectively, while it is approximately 1.91% less than that of QFAN. Fig. 14 also shows a direct correlation between routing overhead and UAV's speed. Clearly, as the nodes speed up, the connections between these UAVs are broken faster, and they need to exchange more hello packets with each other so that they can timely update their neighbor's information in the neighbor table and calculate new communication paths accurately. Hence, the routing overhead is greater at the higher velocities of UAVs. Generally, Figs. 13 and 14 show that QRCF operates weaker than QRF and QTAR in terms of routing overhead. This is because the performance of cylindrical filtering in QRCF is weaker than that of spherical filtering in QRF. This leads to more control overhead in QRCF. On the other hand, QFAN exhibits the highest routing overhead, which can be attributed to the additional use of control packets, specifically route request (RREQ) and route reply (RREP) packets.

6.4. Delay

In Fig. 15, delay in different schemes is compared based on the number of flying nodes. As shown in this figure, QRCF decreases delay by 17.71%, 40.60%, and 48.03% in comparison with QRF, QFAN, and QTAR, respectively. Fig. 15 displays a direct relationship between delay and the number of UAVs. As the number of flying nodes increases, the number of control messages exchanged between the UAVs will be greater, and consequently, the size of their neighbor table will be larger. Hence, all four Q-learning-based routing schemes have a larger state space, which speeds up the convergence rate to identify the optimal path, thereby increasing delay during the routing process. Moreover, Fig. 16 compares delay in different approaches based on the speed of UAVs in FANET. According to this figure, QRCF reduces delay by 17.5%, 29.79%, and 43.59% compared to QRF, QFAN, and QTAR, respectively. This figure reveals a direct relationship between delay and UAV's speed. Obviously, UAVs with higher speeds have unstable connections, and re-updating communication routes is essential due to their short lifespan in FANET. This leads to higher delay. In general, QRCF performs bet-

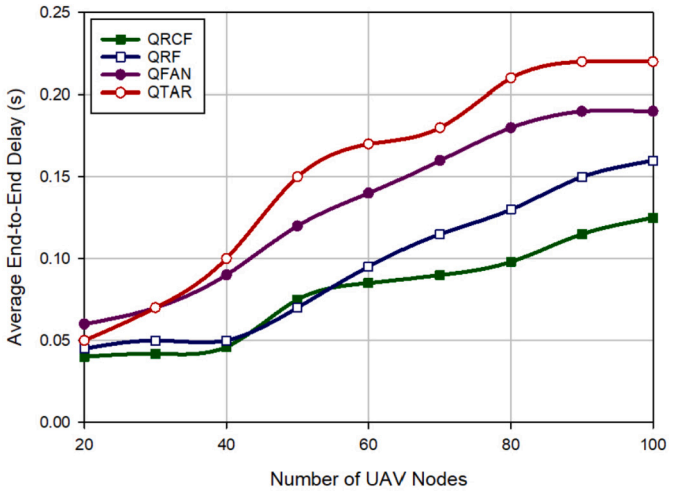

Fig. 15. Delay based on the number of nodes.

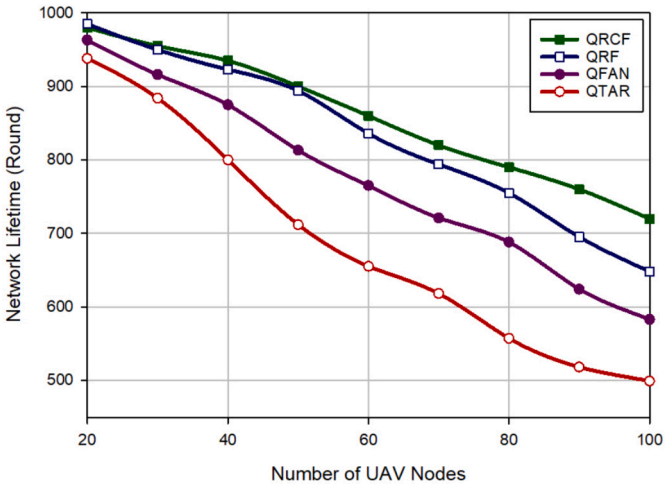

Fig. 17. Network lifetime based on the number of UAVs.

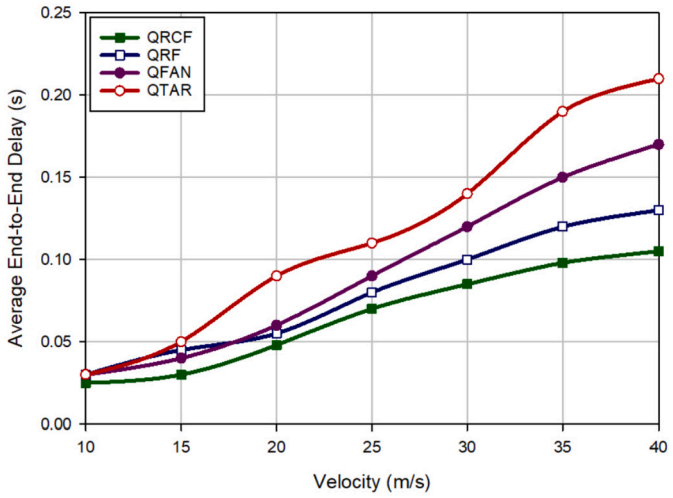

Fig. 16. Delay based on different speeds of UAVs.

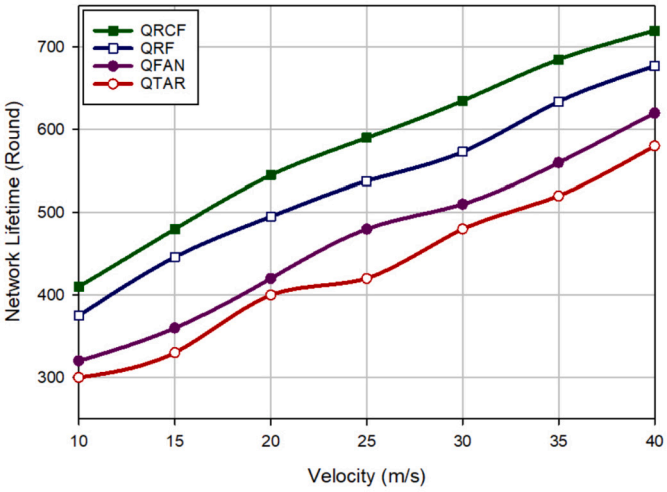

Fig. 18. Network lifetime based on different speeds of UAVs.

ter in terms of delay compared to QRF, QFAN, and QTAR for two main reasons. First, QRCF utilizes a cylindrical filtering technique to refine the state space within the Q-learning algorithm, enabling faster convergence to the optimal routing solution. Secondly, the reward function in QRCF evaluates the next-hop node based on relative velocity and movement path. This ensure that intermediate nodes in the path have similar velocities and movement angles, enhancing the stability of the path. This stability helps minimize the need for frequent recalculations of new paths, thus reducing delay during the routing process.

6.5. Network lifetime

Fig. 17 presents an evaluation of the network lifetime in different schemes, depending on the number of nodes. As shown in this figure, QRCF extends network lifespan by 3.21%, 11.11%, and 24.90% in comparison with QRF, QFAN, and QTAR, respectively. Furthermore, Fig. 17 reveals an inverse relationship between network lifespan and the number of UAVs. As discussed in Section 6.1, the addition of more UAVs leads to higher energy consumption, which in turn shortens the network lifespan. Fig. 18 evaluates the network span for various routing approaches based on the speed of UAVs. Accordingly, QRCF improves network lifetime by 8.75%, 24.31%, and 34.16% compared to QRF, QFAN, and QTAR, respectively. Overall, Figs. 17 and 18 show that QRCF has the best network lifetime compared to other routing approaches. The reasons for this were described in Section 6.1 in detail. The most important reason is that QRCF incorporates the remaining energy of UAVs into the reward function and considers a dynamic interval for broadcasting hello messages in FANET.

7. Conclusion

This study presented a Q-learning-based routing strategy, enhanced by an innovative cylindrical filtering technique, called QRCF in FANETs. Firstly, the propagation interval of hello messages is regularly updated based on the connection duration between each UAV and its neighbors, enhancing the adaptability of QRCF to FANET. The second phase involves a Q-learning-based routing process, where cylindrical filtering is applied to refine the state set. This improves the learning speed when determining the best route. In this phase, a reward function is calculated based on relative velocity, connection time, residual energy, and movement path. Finally, QRCF is implemented in NS2, and its evaluation results are compared with three routing methods, namely QRF, QFAN, and QTAR. These evaluations are based on the number of UAVs and their speed. Overall, when changing the number of nodes, QRCF improves consumed energy by 5.01%, 22.32%, and 33.78%, PDR by 1.20%, 2.31%, and 6.87%, delay by 17.71%, 40.60%, and 48.03%, and network lifespan by 3.12%, 11.11%, and 24.90% compared to QRF, QFAN, and QTAR, respectively. However, QRCF has 10.91% higher routing overhead than QRF. In addition, when changing the speed of UAVs in FANET, QRCF optimizes consumed energy by 4.94%, 26.14%, and

35.85%, PDR by 2.36%, 2.85%, and 7.63%, delay by 17.5%, 29.79%, and 43.59%, and network lifetime by 8.75%, 24.13%, and 34.16% compared to QRF, QFAN, and QTAR. However, the overhead of the proposed scheme is 15.47% more than QRF. In future research directions, the filtering algorithm in QRCF must be optimized to manage the routing overhead in FANET. This optimization can be implemented using fuzzy logic or metaheuristic algorithms. Additionally, the performance of QRCF can be evaluated under diverse mobility patterns and environmental conditions. In the future, QRCF will be improved by considering the effect of external interferences or environmental obstacles on the network performance. In addition, QRCF will review lightweight alternatives to dynamic parameter updates in the future because the adaptive learning parameters enhance adaptability but could increase overhead in resource-constrained scenarios.

CRediT authorship contribution statement

Amir Masoud Rahmani: Writing – original draft, Conceptualization. **Amir Haider:** Writing – original draft, Conceptualization. **Monji Mohamed Zaidi:** Writing – original draft, Data curation. **Abed Alanazi:** Writing – original draft, Data curation. **Shtwai Alsubai:** Writing – original draft, Investigation. **Abdullah Alqahtani:** Writing – original draft, Investigation. **Mohammad Sadegh Yousefpoor:** Writing – original draft, Project administration. **Efat Yousefpoor:** Writing – original draft, Project administration. **Mehdi Hosseinzadeh:** Writing – original draft, Project administration, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/384/45. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2025/R/1446).

Data availability

No data was used for the research described in the article.

References

[1] M. Hosseinzadeh, S. Ali, A.M. Rahmani, J. Lansky, V. Nulicek, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, S.W. Lee, A smart filtering-based adaptive optimized link state routing protocol in flying ad hoc networks for traffic monitoring, J. King Saud Univ, Comput. Inf. Sci. (2024) 102034, https://doi.org/10.1016/j.jksuci.2024.102034.

[2] M. Hosseinzadeh, S. Ali, A.H. Mohammed, J. Lansky, S. Mildeova, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, An energy-aware routing scheme based on a virtual relay tunnel in flying ad hoc networks, Alex. Eng. J. (2024), https://doi.org/10.1016/j.aej.2024.02.006.

[3] A. Saha, A. Kumar, A.K. Sahu, FPV drone with GPS used for surveillance in remote areas, in: 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), IEEE, 2017, November, pp. 62–67.

[4] M. Hosseinzadeh, S. Ali, H.J. Ahmad, F. Alanazi, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, A.M. Rahmani, S.W. Lee, DCFH: a dynamic clustering approach based on fire hawk optimizer in flying ad hoc networks, Veh. Commun. (2024) 100778, https://doi.org/10.1016/j.vehcom.2024.100778.

[5] S.W. Lee, S. Ali, M.S. Yousefpoor, E. Yousefpoor, P. Lalbakhsh, D. Javaheri, A.M. Rahmani, M. Hosseinzadeh, An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (FANETs), IEEE Access 9 (2021) 129977–130005, https://doi.org/10.1109/ACCESS.2021.3111444.

[6] A.I. Ameur, O.S. Oubbati, A. Lakas, A. Rachedi, M.B. Yagoubi, Efficient vehicular data sharing using aerial P2P backbone, IEEE Trans. Intell. Veh. (2024), https://doi.org/10.1109/TIV.2024.3414140.

[7] A. Galletta, J. Taheri, A. Celesti, M. Fazio, M. Villari, Investigating the applicability of nested secret share for drone fleet photo storage, IEEE Trans. Mob. Comput. 23 (4) (2023) 2671–2683, https://doi.org/10.1109/TMC.2023.3263115.

[8] M. Hosseinzadeh, O.H. Ahmed, J. Lansky, S. Mildeova, M.S. Yousefpoor, E. Yousefpoor, J. Yoo, L. Tightiz, A.M. Rahmani, A cluster-tree-based trusted routing algorithm using Grasshopper Optimization Algorithm (GOA) in Wireless Sensor Networks (WSNs), PLoS ONE 18 (9) (2023) e0289173, https://doi.org/10.1371/journal.pone.0289173.

[9] M. Hosseinzadeh, A.H. Mohammed, A.M. Rahmani, A.F. Alenizi, S.M. Zandavi, E. Yousefpoor, O.H. Ahmed, M. Hussain Malik, L. Tightiz, A secure routing approach based on league championship algorithm for wireless body sensor networks in healthcare, PLoS ONE 18 (10) (2023) e0290119, https://doi.org/10.1371/journal.pone.0290119.

[10] J. Lansky, S. Ali, A.M. Rahmani, M.S. Yousefpoor, E. Yousefpoor, F. Khan, M. Hosseinzadeh, Reinforcement learning-based routing protocols in flying ad hoc networks (FANET): a review, Mathematics 10 (16) (2022) 3017, https://doi.org/10.3390/math10163017.

[11] A.M. Rahmani, R.A. Naqvi, E. Yousefpoor, M.S. Yousefpoor, O.H. Ahmed, M. Hosseinzadeh, K. Siddique, A Q-learning and fuzzy logic-based hierarchical routing scheme in the intelligent transportation system for smart cities, Mathematics 10 (22) (2022) 4192, https://doi.org/10.3390/math10224192.

[12] O.M. Alsalami, E. Yousefpoor, M. Hosseinzadeh, J. Lansky, A novel optimized link-state routing scheme with greedy and perimeter forwarding capability in flying ad hoc networks, Mathematics 12 (7) (2024) 1016, https://doi.org/10.3390/math12071016.

[13] J. Lansky, A.M. Rahmani, M. Hosseinzadeh, Reinforcement learning-based routing protocols in vehicular ad hoc networks for intelligent transport system (its): a survey, Mathematics 10 (24) (2022) 4673, https://doi.org/10.3390/math10244673.

[14] A.M. Rahmani, S. Ali, M.H. Malik, E. Yousefpoor, M.S. Yousefpoor, A. Mousavi, F. Khan, M. Hosseinzadeh, An energy-aware and Q-learning-based area coverage for oil pipeline monitoring systems using sensors and Internet of Things, Sci. Rep. 12 (1) (2022) 9638, https://doi.org/10.1038/s41598-022-12181-w.

[15] J. Lansky, A.M. Rahmani, M.H. Malik, E. Yousefpoor, M.S. Yousefpoor, M.U. Khan, M. Hosseinzadeh, An energy-aware routing method using firefly algorithm for flying ad hoc networks, Sci. Rep. 13 (1) (2023) 1323, https://doi.org/10.1038/s41598-023-27567-7.

[16] M. Hosseinzadeh, J. Yoo, S. Ali, J. Lansky, S. Mildeova, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A fuzzy logic-based secure hierarchical routing scheme using firefly algorithm in Internet of Things for healthcare, Sci. Rep. 13 (1) (2023) 11058, https://doi.org/10.1038/s41598-023-38203-9.

[17] M. Hosseinzadeh, J. Yoo, S. Ali, J. Lansky, S. Mildeova, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A cluster-based trusted routing method using fire hawk optimizer (FHO) in wireless sensor networks (WSNs), Sci. Rep. 13 (1) (2023) 13046, https://doi.org/10.1038/s41598-023-40273-8.

[18] M.S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, M. Hosseinzadeh, Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: a comprehensive review, J. Netw. Comput. Appl. 190 (2021) 103118, https://doi.org/10.1016/j.jnca.2021.103118.

[19] D. Wang, J. Zhou, M. Masdari, S.N. Qasem, B.T. Sayed, Security in wireless body area networks via anonymous authentication: comprehensive literature review, scheme classification, and future challenges, Ad Hoc Netw. (2023) 103332, https://doi.org/10.1016/j.adhoc.2023.103332.

[20] J. Xie, Y. Wan, J.H. Kim, S. Fu, K. Namuduri, A survey and analysis of mobility models for airborne networks, IEEE Commun. Surv. Tutor. 16 (3) (2013) 1221–1238, https://doi.org/10.1109/SURV.2013.111313.00138.

[21] M. Tanveer, H. Alasmary, N. Kumar, A. Nayak, Saaf-iod: secure and anonymous authentication framework for the Internet of drones, IEEE Trans. Veh. Technol. (2023), https://doi.org/10.1109/TVT.2023.3306813.

[22] V. Singh, K.P. Sharma, H.K. Verma, ABNT: adaptive beaconing and neighbor timeout for geographical routing in UAV networks, Peer-to-Peer Netw. Appl. 15 (4) (2022) 2079–2100, https://doi.org/10.1007/s12083-022-01341-4.

[23] Z. Guo, Y. Wang, Y. Sun, J. Li, C. Fu, J. Zhong, Resilience analysis of cooperative mission based on spatio–temporal network dynamics for flying ad hoc network, IEEE Trans. Reliab. (2024), https://doi.org/10.1109/TR.2023.3344726.

[24] D. Zhang, H. Ge, T. Zhang, Y.Y. Cui, X. Liu, G. Mao, New multi-hop clustering algorithm for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 20 (4) (2018) 1517–1530, https://doi.org/10.1109/TITS.2018.2853165.

[25] J. Cui, L. Ma, R. Wang, M. Liu, Research and optimization of GPSR routing protocol for vehicular ad-hoc network, China Commun. 19 (10) (2022) 194–206, https://doi.org/10.23919/JCC.2022.00.031.

[26] D.S. Lakew, U. Sa'ad, N.N. Dao, W. Na, S. Cho, Routing in flying ad hoc networks: a comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1071–1120, https://doi.org/10.1109/COMST.2020.2982452.

[27] S. Gangopadhyay, V.K. Jain, A position-based modified OLSR routing protocol for flying ad hoc networks, IEEE Trans. Veh. Technol. (2023), https://doi.org/10.1109/TVT.2023.3265704.

[28] M. Zhang, H. Cheng, P. Yang, C. Dong, H. Zhao, Q. Wu, T.Q. Quek, Adaptive routing design for flying ad hoc networks: a joint prediction approach, IEEE Trans. Veh. Technol. (2023), https://doi.org/10.1109/TVT.2023.3317311.

[29] M. Prakash, S. Neelakandan, B.H. Kim, Reinforcement learning-based multidimensional perception and energy awareness optimized link state routing for flying ad-hoc networks, Mob. Netw. Appl. (2023) 1–19, https://doi.org/10.1007/s11036-023-02255-y.

[30] V. Chandrasekar, V. Shanmugavalli, T.R. Mahesh, R. Shashikumar, N. Borah, V.V. Kumar, S. Guluwadi, Secure malicious node detection in flying ad-hoc networks using enhanced AODV algorithm, Sci. Rep. 14 (1) (2024) 7818, https://doi.org/10.1038/s41598-024-57480-6.

[31] J.V. Ananthi, P.S.H. Jose, Mobility Management UAV-based Grouping routing protocol in flying ad hoc networks for biomedical applications, Int. J. Commun. Syst. 36 (1) (2023) e5362, https://doi.org/10.1002/dac.5362.

[32] K. Messaoudi, A. Baz, O.S. Oubbati, A. Rachedi, T. Bendouma, M. Atiquzzaman, UGV charging stations for UAV-assisted AoI-Aware data collection, IEEE Trans. Cogn. Commun. Netw. (2024), https://doi.org/10.1109/TCCN.2024.3394859.

[33] H. Jin, X. Jin, Y. Zhou, P. Guo, J. Ren, J. Yao, S. Zhang, A survey of energy efficient methods for UAV communication, Veh. Commun. 41 (2023) 100594, https://doi.org/10.1016/j.vehcom.2023.100594.

[34] R. Kumar, B. Sharma, S. Athithan, TBMR: trust based multi-hop routing for secure communication in flying ad-hoc networks, Wirel. Netw. (2023) 1–17, https://doi.org/10.1007/s11276-023-03480-9.

[35] Y. Liu, J. Xie, C. Xing, S. Xie, Topology construction and topology adjustment in flying Ad hoc networks for relay transmission, Comput. Netw. 228 (2023) 109753, https://doi.org/10.1016/j.comnet.2023.109753.

[36] N. Toorchi, W. Lyu, L. He, J. Zhao, I. Rasheed, F. Hu, Deep reinforcement learning enhanced skeleton based pipe routing for high-throughput transmission in flying ad-hoc networks, Comput. Netw. 244 (2024) 110330, https://doi.org/10.1016/j.comnet.2024.110330.

[37] O.S. Oubbati, H. Badis, A. Rachedi, A. Lakas, P. Lorenz, January. Multi-UAV assisted network coverage optimization for rescue operations using reinforcement learning, in: 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), IEEE, 2023, pp. 1003–1008.

[38] A. Jalooli, A. Marefat, Cluster stability-driven optimization for enhanced routing in heterogeneous vehicular networks, Veh. Commun. (2024) 100745, https://doi.org/10.1016/j.vehcom.2024.100745.

[39] A. Mchergui, T. Moulahi, S. Zeadally, Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs), Veh. Commun. 34 (2022) 100403, https://doi.org/10.1016/j.vehcom.2021.100403.

[40] Y. Cui, H. Tian, C. Chen, W. Ni, H. Wu, G. Nie, New geographical routing protocol for three-dimensional flying ad-hoc network based on new effective transmission range, IEEE Trans. Veh. Technol. (2023), https://doi.org/10.1109/TVT.2023.3296082.

[41] M.M. Alam, S. Moh, Q-learning-based routing inspired by adaptive flocking control for collaborative unmanned aerial vehicle swarms, Veh. Commun. 40 (2023) 100572, https://doi.org/10.1016/j.vehcom.2023.100572.

[42] M. Hosseinzadeh, S. Ali, L. Ionescu-Feleaga, B.S. Ionescu, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, A novel Q-learning-based routing scheme using an intelligent filtering algorithm for flying ad hoc networks (FANETs), J. King Saud Univ, Comput. Inf. Sci. 35 (10) (2023) 101817, https://doi.org/10.1016/j.jksuci.2023.101817.

[43] M. Hosseinzadeh, J. Tanveer, A.M. Rahmani, K. Aurangzeb, E. Yousefpoor, M.S. Yousefpoor, A. Darwesh, S.W. Lee, M. Fazlali, A Q-learning-based smart clustering routing method in flying Ad Hoc networks, J. King Saud Univ, Comput. Inf. Sci. 36 (1) (2024) 101894, https://doi.org/10.1016/j.jksuci.2023.101894.

[44] M. Hosseinzadeh, A.H. Mohammed, F.A. Alenizi, M.H. Malik, E. Yousefpoor, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A novel fuzzy trust-based secure routing scheme in flying ad hoc networks, Veh. Commun. 44 (2023) 100665, https://doi.org/10.1016/j.vehcom.2023.100665.

[45] M. Hosseinzadeh, J. Tanveer, L. Ionescu-Feleaga, B.S. Ionescu, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, A greedy perimeter stateless routing method based on a position prediction mechanism for flying ad hoc networks, J. King Saud Univ, Comput. Inf. Sci. 35 (8) (2023) 101712, https://doi.org/10.1016/j.jksuci.2023.101712.

[46] J. Guo, H. Gao, Z. Liu, F. Huang, J. Zhang, X. Li, J. Ma, ICRA: an intelligent clustering routing approach for UAV ad hoc networks, IEEE Trans. Intell. Transp. Syst. 24 (2) (2022) 2447–2460, https://doi.org/10.1109/TITS.2022.3145857.

[47] A.M. Rahmani, S. Ali, E. Yousefpoor, M.S. Yousefpoor, D. Javaheri, P. Lalbakhsh, O.H. Ahmed, M. Hosseinzadeh, S.W. Lee, OLSR+: a new routing method based on fuzzy logic in flying ad-hoc networks (FANETs), Veh. Commun. 36 (2022) 100489, https://doi.org/10.1016/j.vehcom.2022.100489.

[48] J. Lansky, A.M. Rahmani, S.M. Zandavi, V. Chung, E. Yousefpoor, M.S. Yousefpoor, F. Khan, M. Hosseinzadeh, A Q-learning-based routing scheme for smart air quality monitoring system using flying ad hoc networks, Sci. Rep. 12 (1) (2022) 20184, https://doi.org/10.1038/s41598-022-20353-x.

[49] M.Y. Arafat, S. Moh, A Q-learning-based topology-aware routing protocol for flying ad hoc networks, IEEE Internet Things J. 9 (3) (2021) 1985–2000, https://doi.org/10.1109/JIOT.2021.3089759.

[50] W. Stallings, IEEE 802. 11: wireless LANs from a to n, IT Prof. 6 (5) (2004) 32–37, https://doi.org/10.1109/MITP.2004.62.

[51] F. Tramarin, S. Vitturi, M. Luvisotto, A. Zanella, On the use of IEEE 802.11 n for industrial communications, IEEE Trans. Ind. Inform. 12 (5) (2015) 1877–1886, https://doi.org/10.1109/TII.2015.2504872.

[52] A. Srivastava, J. Prakash, Future FANET with application and enabling techniques: anatomization and sustainability issues, Comput. Sci. Rev. 39 (2021) 100359, https://doi.org/10.1016/j.cosrev.2020.100359.

[53] L.A.L. da Costa, R. Kunst, E.P. de Freitas, Q-FANET: improved Q-learning based routing protocol for FANETs, Comput. Netw. 198 (2021) 108379, https://doi.org/10.1016/j.comnet.2021.108379.

[54] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, Y. Xu, QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks, Comput. Commun. 150 (2020) 304–316, https://doi.org/10.1016/j.comcom.2019.11.011.

[55] J. Gomez, E.F. Kfoury, J. Crichigno, G. Srivastava, A survey on network simulators, emulators, and testbeds used for research and education, Comput. Netw. 237 (2023) 110054, https://doi.org/10.1016/j.comnet.2023.110054.