mathematics

MDPI

Review

Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review

Jan Lansky [1], Saqib Ali [2], Amir Masoud Rahmani [3,*], Mohammad Sadegh Yousefpoor [4], Efat Yousefpoor [4], Faheem Khan [5,*] and Mehdi Hosseinzadeh [6,7]

[1] Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, 101 00 Prague, Czech Republic
[2] Department of Information Systems, College of Economics and Political Science, Sultan Qaboos University, Al Khoudh, Muscat P.C.123, Oman
[3] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Douliou 64002, Taiwan
[4] Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful 5716963896, Iran
[5] Department of Computer Engineering, Gachon University, Seongnam 13120, Korea
[6] Mental Health Research Center, Psychosocial Health Research Institute, Iran University of Medical Sciences, Tehran 1449614535, Iran
[7] Computer Science, University of Human Development, Sulaymaniyah 0778-6, Iraq
* Correspondence: rahmania@yuntech.edu.tw (A.M.R.); faheem@gachon.ac.kr (F.K.)

Abstract: In recent years, flying ad hoc networks have attracted the attention of many researchers in industry and universities due to easy deployment, proper operational costs, and diverse applications. Designing an efficient routing protocol is challenging due to unique characteristics of these networks such as very fast motion of nodes, frequent changes of topology, and low density. Routing protocols determine how to provide communications between drones in a wireless ad hoc network. Today, reinforcement learning (RL) provides powerful solutions to solve the existing problems in the routing protocols, and designs autonomous, adaptive, and self-learning routing protocols. The main purpose of these routing protocols is to ensure a stable routing solution with low delay and minimum energy consumption. In this paper, the reinforcement learning-based routing methods in FANET are surveyed and studied. Initially, reinforcement learning, the Markov decision process (MDP), and reinforcement learning algorithms are briefly described. Then, flying ad hoc networks, various types of drones, and their applications, are introduced. Furthermore, the routing process and its challenges are briefly explained in FANET. Then, a classification of reinforcement learning-based routing protocols is suggested for the flying ad hoc networks. This classification categorizes routing protocols based on the learning algorithm, the routing algorithm, and the data dissemination process. Finally, we present the existing opportunities and challenges in this field to provide a detailed and accurate view for researchers to be aware of the future research directions in order to improve the existing reinforcement learning-based routing algorithms.

Keywords: flying ad hoc networks (FANET); reinforcement learning (RL); routing; artificial intelligence (AI); unmanned ariel vehicles (UAVs)

MSC: 68-02

Citation: Lansky, J.; Ali, S.; Rahmani, A.M.; Yousefpoor, M.S.; Yousefpoor, E.; Khan, F.; Hosseinzadeh, M. Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review. *Mathematics* **2022**, *10*, 3017. https://doi.org/10.3390/math10163017

Academic Editor: Shih-Wei Lin

Received: 13 July 2022
Accepted: 9 August 2022
Published: 22 August 2022

1. Introduction

In the last decade, unmanned aerial vehicles (UAVs) are widely used in various applications and services. When drones or UAVs are organized as connected groups in an ad hoc form, they can perform complex tasks and form a flying ad hoc network (FANET). This network is a subset of vehicular ad hoc network (VANET) and mobile ad hoc network (MANET) [1,2]. They have common features such as mobile nodes, wireless media, decentralized control, and multi-hop communications. However, a FANET has unique features such as the very fast movement of nodes, the frequent changes in topology, and low-density network. In Figure 1, a flying ad hoc network is shown. This network has different applications in military and civilian areas. Military applications include public protection, search and rescue, reconnaissance, border monitoring, independent tracking, fire fighting, internal security, wind estimation, remote sensing, traffic monitoring, and relaying networks [3]. Furthermore, drones have many commercial applications in civilian areas, such as film-making, agricultural monitoring, Internet shipping, transportation, and architecture and infrastructure monitoring [4,5].

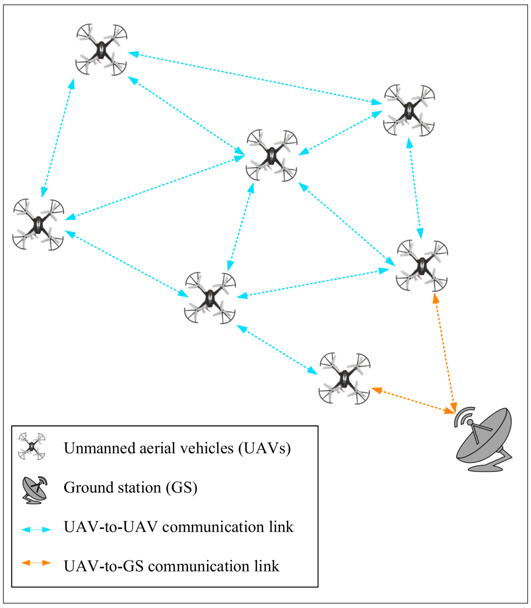
Unmanned aerial vehicles (UAVs)
Ground station (GS)
UAV-to-UAV communication link
UAV-to-GS communication link

Figure 1. Flying ad hoc network.

Given specific features of FANETs, such as high dynamic topology, the rapid motion of nodes, and frequent link failure, designing an efficient routing protocol is a very important research challenge in these networks. In general, in the routing process, we answer the question *"How do UAVs send data packets from source to destination?"*. In this process, if there is no direct connection between the source and destination, the data packets must be transferred by intermediate nodes (that play the router role) to the destination [6,7]. According to this definition, the routing path represents a series of hops (i.e., intermediate nodes) that relay data packets. A routing protocol is responsible for building such a route between the source and destination. Additionally, these protocols should manage the link failure by finding appropriate alternative routes. In FANETs, there are many challenges that can affect the routing process. For example, routing protocols must solve problems related to intermittent links, frequent changes in topology, network partitioning, and node movement [8,9]. Furthermore, they must consider some constraints such as energy, computing power, and delay. In addition, these protocols must be free-loop, self-repairing, and scalable. Note that UAVs move in a three-dimensional space, which affects the quality of communication links [10,11]. This is a major challenge when designing routing protocol in FANETs. In recent years, many researchers try to improve the performance of routing protocols in FANETs. Despite many efforts in this area, it is very difficult to design a routing protocol that guarantees efficient communication in these networks [12,13].

Machine learning (ML) techniques can be used for solving various challenges, such as routing in FANETs. In general, machine learning is divided into three groups: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the designed model must explore dependencies between the training data (labeled data) to predict the correct answer to the requested problem (unlabeled data). In fact, this technique trains a model based on initial data to predict the label of new data. Unsupervised learning also tries to discover the patterns in data samples. In this learning process, the algorithm uses an unlabeled dataset, whereas reinforcement learning (RL) can learn the environment without any initial data samples. RL is similar to the human learning approach, meaning that this learning process does not require large datasets and the learning agent is not aware of the data label. Furthermore, reinforcement learning learns through interactions with the environment without the need for any dataset. The FANET is a very dynamic and complex network. For this reason, supervised and unsupervised learning techniques cannot find appropriate responses for routing in the FANET environment. For example, when failing each link or breaking each path, supervised and unsupervised learning methods must first have a dataset related to the failed path to find a new response (new paths) through learning this dataset. Reinforcement learning is suitable for FANET because it can control the dynamic and real-time environment in a desirable manner. RL can constantly learn new information about FANETs and communication links between nodes. Note that it is difficult to model the routing process in FANETs. To model this process, supervised and unsupervised learning methods must first execute a simulation process to produce a dataset because there is no dataset to train the model; then, they use this dataset to train the model. RL is the only machine learning technique which can learn the routing process without the need for a dataset. The reinforcement learning algorithms use the trial-and-error technique to learn a proper routing model in the network environment. This method can reduce the complexity of supervised and unsupervised learning methods to simulate and solve this problem in FANETs.

Today, reinforcement learning (RL) techniques have been used in flying ad hoc networks to solve challenges related to the routing issue. RL is a branch of artificial intelligence (AI), which allows machines to become intelligent without human intervention and learn based on previous experiences [14,15]. Reinforcement learning increases efficiency and reliability, and reduces computational costs compared to other AI techniques. In this process, the agent interacts with the dynamic environment to find its ideal behavior based on the reward-penalty feedback received from the environment [16,17]. In FANETs, reinforcement learning allows drones to decide on various network operations, especially routing. In reinforcement learning, the agent should understand the environment by collecting data from the environment to find the best action for achieving a specific goal, such as creating a route with a maximum packet delivery rate (PDR) [17–19].

In fact, reinforcement learning algorithms have potential to improve routing protocols in FANETs. Therefore, it is necessary to study the RL applications in flying ad hoc networks. There are several works in this area. However, the number of review papers is not sufficient and further studies should be carried out. For example, ref. [20] has presented a comprehensive and useful review paper about artificial intelligence-based routing protocols in flying ad hoc networks. In [21], authors have investigated machine learning applications in various fields of FANETs including routing, flight trajectory selection, relay, and recharge. In [22], authors have reviewed and studied Q-learning-based position-aware routing methods. In [23], authors have examined the application of machine learning techniques to improve UAV communications. Finally, in [24], issues and challenges related to FANETs such as mobility models, communications, architecture, and applications have been studied. Our research shows that the number of review papers in the field of RL-based routing protocols in FANETs is very low. This issue proves the need for more research in this field, to familiarize researchers with future research directions and challenges in this field, and to find a suitable view of how to design a RL-based routing method in FANETs.

In this paper, a detailed classification of RL-based routing methods has been presented. According to this classification, routing protocols are categorized based on a RL algorithm (traditional reinforcement learning and deep reinforcement learning), routing algorithm (routing path, network topology, data delivery method, and routing process), and the data dissemination process (unicast, multicast, broadcast, and geocast). Then, the state-of-the-art RL-based routing methods are studied and reviewed, and their advantages and disadvantages are expressed. The contributions of this paper are presented as follows:

- This paper proposes a detailed classification of reinforcement learning-based routing methods in FANETs. This classification includes three main parts: learning algorithm, routing algorithm, and the data dissemination process.
- This paper introduces and compares the state-of-the-art RL-based routing methods in FANETs according to the suggested classification. Furthermore, it studies and reviews the most important advantages and disadvantages of each scheme.
- Finally, this paper expresses some challenges and open issues related to RL-based routing methods in FANETs and presents future research directions.

The rest of our paper is as follows: Section 2 presents the related works in this area. Section 3 describes fundamentals of reinforcement learning (RL) in summary. Section 4 introduces flying ad hoc networks, communications, unmanned aerial vehicles (UAVs), and their applications, and focuses particularly on the routing process and its challenges in FANETs. In Section 5, we present a detailed classification for reinforcement learning-based routing algorithms in FANETs. Section 6 reviews the state-of-the-art RL-based routing schemes in FANETs. Section 7 discusses RL-based routing schemes generally. Section 8 presents the most important challenges and open issues in the RL-based routing schemes. Finally, Section 9 concludes this paper.

2. Related Works

Our reviews show that there are few review papers which survey routing issues, specifically RL-based routing methods in FANETs. We studied some related papers in this field, as follows:

In [20], authors studied artificial intelligence-based routing protocols in flying ad hoc networks. In this survey, the applications of artificial intelligence (AI) are studied in different areas of FANETs. This paper pursues two goals: (1) investigating the features of FANET, UAV technology, networking protocols, and UAV swarms; (2) studying the routing protocols designed in these networks by emphasizing the AI application in this area. In [20], authors suggested the classification of routing methods in FANETs. It includes centralized and distributed routing, deterministic and probabilistic routing, and static and dynamic routing. Then, they categorized dynamic routing methods into five classes: position-based (geographic, location-based, and opportunistic), proactive, reactive, hybrid, and AI-based (topology predictive and self-adaptive learning-based). Then, different routing methods have been investigated based on this categorization, and the challenges and issues in this area have been expressed. It is a comprehensive and useful review paper that is recommended to researchers in this field. However, they do not emphasize the AI techniques and their structures in these methods.

In [21], the authors presented reinforcement learning (RL) and deep reinforcement learning (DRL), and studied their applications in FANETs. They claim that this paper is the first review paper on RL applications in FANETs. In general, they focused on the RL applications in five important areas, including routing protocols, flight trajectory, protection against jamming, relaying, and charging UAVs. Then, they studied RL-based methods in the five areas and expressed their advantages and disadvantages. However, this paper does not review the details of the RL algorithms used in these methods.

In [22], the authors studied and evaluated the Q-learning-based position-aware routing protocols in FANETs. Initially, they introduced flying ad hoc networks and their features, and described all mobility models available for FANETs by focusing on their applications. Then, they introduced a Q-learning algorithm and its application for designing routing protocols in FANETs. Next, Q-learning-based routing protocols were investigated and their advantages and disadvantages were expressed. Finally, these methods were compared with each other in terms of key features, performance, and implementation. However, the most important disadvantage of this paper is that it focuses only on a Q-learning algorithm and ignores other reinforcement learning algorithms.

In [23], the authors examined machine learning (ML) and artificial intelligence (AI) applications for UAV networks. They studied various communication issues, from the physics layer, channel modeling, and resource management, to flight trajectory and caching; in particular, they emphasized security and safety issues, and provided solutions based on learning techniques in this area. However, this paper does not focus on ML-based routing protocols in FANETs.

In [24], the authors presented a comprehensive review to examine issues related to FANETs, including their features, architecture, and communication. Then, the authors examined various mobility models such as random-based, time-based, path-based, group-based, and topology-based mobility models for these networks. Finally, in [24], a detailed classification of routing protocols was provided in flying ad hoc networks. In this classification, routing methods were divided into various groups, including delay-tolerant network (deterministic, stochastic, social network), position-based (single-path and multi-path), heterogeneous, energy-aware, swarm-based, cluster-based (probabilistic and deterministic), topology-based (static, hybrid, reactive, proactive), and secure routing protocols. Then, routing protocols were investigated based on the suggested classification in this paper.

In [25], the authors reviewed various routing protocols in vehicular ad hoc networks. In this paper, routing methods were divided into four categories: unicast-based routing, multicast-based routing, geocast-based routing, and broadcast-based routing. It is a very comprehensive and applicable review paper in VANETs. However, this paper does not consider the reinforcement learning-based routing methods.

In [26], the authors studied various issues, including architecture, application, and different routing protocols in flying ad hoc networks. This paper discusses routing methods for highly dynamic networks. However, the authors have not mentioned an important category of routing methods: namely, reinforcement learning-based methods. This paper focuses specifically on security challenges in FANETs.

In [13], the authors provided a comprehensive review paper on the routing protocols in FANETs. They described issues such as mobility models and UAV applications. Furthermore, different routing protocols were compared in terms of performance scales. The most important weakness of this paper is the focus on a special type of routing protocol; namely, position-based routing protocols.

In [27], the authors analyzed the various types of routing protocols in flying ad hoc networks. They evaluated and compared these methods from different aspects. The authors studied these protocols in terms of network conditions and application needs. However, this paper is not comprehensive. It does not consider other routing methods; namely, hierarchical routing, probabilistic routing, and reinforcement learning-based routing.

In [28], the authors evaluated various issues of FANETs such as architecture, characteristics, routing protocols, and challenges in this area. They divided the routing methods into three groups: deterministic routing, stochastic routing, and social network-based routing. Finally, these routing methods were compared in terms of features and performance. However, this paper does not address reinforcement learning-based routing approaches in FANETs.

3. Fundamentals of Reinforcement Learning (RL)

In this section, fundamentals of reinforcement learning (RL), including RL process, Markov decision process (MDP), and RL techniques, are briefly described.

3.1. *Reinforcement Learning*

It is known as a powerful tool for learning optimal policy by interacting between an agent and the environment. In reinforcement learning, the agent employs a trial-and-error technique to increase the reward obtained from the environment [16,17]. In each step, it obtains its state (s_t) from the environment and chooses an action a_t. Next, the environment determines the reward r_t and the new state s_{t+1} based on the selected action. If this action is good, the environment has positive feedback, that is, the agent receives a positive reward. Otherwise, it has negative feedback. The agent continues this process until it maximizes the expected discounted feedback for any state. More precisely, a RL-based system includes four components:

- **Policy:** This component includes a set of stimulus-action rules that map each state of the surrounding environment to a list of the allowed actions. In the simplest case, a lookup table can be used to implement this policy. However, searching in this table requires a high computational cost [18,19].
- **Reward:** This component is a function such as a random function, which depends on state and the action selected by the agent. In fact, it indicates the response of the environment with regard to the selected action. After taking the action, the environment changes its state and produces a reward. The purpose of the agent is to increase the sum of rewards obtained from the interaction with the environment because the reward reflects the main mechanism for modifying the policy. Now, if the agent finds out that the selected action leads to a small reward, it changes its selection and chooses another action when there is a similar condition in the future. This helps the agent to explore different probabilities [19,29].
- **Value function:** This component is also called the value-action function. Although, the reward represents whether the current selected action is suitable. However, the agent must follow solutions that make more profit in the middle and long term. The value of a certain state represents the sum of received rewards by passing from that state. Thus, actions are selected by searching for the maximum value and not the highest reward. However, it is difficult to calculate the value compared to the reward because the reward is immediately received from the environment. In contrast, the agent must estimate the value by searching for previous interactions with the environment. In all RL algorithms, it is a very challenging issue to estimate the value function [29].
- **Model:** This component describes the performance of the environment. This model estimates the future state and immediate reward according to a specific state and the selected action. Based on this view, two reinforcement learning methods can be defined: model-based and free-model. Model-based methods design a model for solving RL problems. However, free-model methods learn the optimal policy based on trial and error [18,29].

In the following, important challenges in reinforcement learning algorithms are described:

- **Trade-off between exploration and exploitation:** Exploration and exploitation are two important concepts in reinforcement learning. In exploration, the agent searches for unknown actions to obtain new knowledge. In exploitation, the agent utilizes the existing knowledge and uses the explored actions to produce high feedback. In the learning process, it is necessary to make a trade-off between exploration and exploitation. The agent can exploit its existing knowledge to achieve the suitable value and can explore new actions to increase its knowledge and obtain more valuable rewards in the future. As a result, the agent should not focus only on exploration or exploitation and must experience various actions to gradually obtain the actions with the highest value.
- **Uncertainty:** Another challenge for RL is that the agent may face uncertainty when interacting with the environment and updating its state and reward, whereas the purpose of RL is to learn a policy that leads to a high value over time.

3.2. Markov Decision Process

Reinforcement learning is known as an experience-based method. This means that the agent experiences various actions in a trial-and-error manner to improve its future choices. This problem can be formulated as a Markov decision process (MDP) [17,18]. The important definitions of RL are presented as follows:

Definition 1. *MDP includes a tuple (S, A, P, R, γ):*

- *S indicates the state space.*
- *A indicates the action space.*
- *R is the reward function, which is defined as $R = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$.*
- *P is defined as the state transition probability $P = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$.*
- *γ indicates the discount factor, where $\gamma \in [0, 1]$.*

In MDP, the next state depends only on the current state and does not depend on the previous states.

$$\mathbb{P}[S_{t+1} - S_t] = \mathbb{P}[S_{t+1} - S_1, ..., S_t]. \tag{1}$$

In finite MDP, there are a limited state set, a finite action set, and a dynamic environment. Furthermore, the probability of each next state-reward pair such as (s', r) based on the current state-action pair (s, a) can be formulated as follows:

$$p(s', r|s, a) \doteq \Pr\{S_{t+1} = s, R_{t+1} = r|S_t = s, A_t = a\}. \tag{2}$$

In the learning process, the agent is responsible for maximizing $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$, so that G_t is the sum of the rewards obtained from the learning process and T reflects the last time step. If there is an episodic task, meaning that this task includes a final state, the mentioned function is used for calculating G_t. However, if there is a continuous task, meaning that this task has no final state, i.e., $T = \infty$, we can not use the mentioned function.

Definition 2. *G_t means the sum of discounted feedback received from the environment in the long term. It is calculated according to Equation (3):*

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{3}$$

where, G_t is the long-term feedback. $\gamma^k \in [0, 1]$ represents the discount factor and R_{t+k+1} indicates the reward obtained from environment at the moment $t + k + 1$.

Definition 3. *In RL, policy defines the results of a state and a certain action. When the agent is in a particular state, it must select its next action based on this policy π, which is a probability distribution on the actions performed in the given states.*

$$\pi(a|s) \doteq \mathbb{P}[A_t = a|S_t = s] \tag{4}$$

where, π is the policy, which determines the probability of performing the action a in the state s. Based on the policy π and the feedback G_t, two value functions with regard to the expected feedback can be obtained.

Definition 4. *State-value function $(v_\pi(s))$ represents the expected feedback when the agent is in the state s and follows the policy π.*

$$v_\pi \doteq \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right]. \tag{5}$$

Definition 5. *Action-value function $(q_\pi(s, a))$ indicates the expected feedback in the state s when the agent chooses the action a and follows the policy π.*

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right]. \tag{6}$$

Both value functions are calculated based on the Bellman equation:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_\pi(s')\right], \ \forall \ s \in S. \tag{7}$$

The RL algorithm converges when the agent finds the optimal policy π^* for all available policies in a certain state. The optimal policy π^* is used to calculate the optimal state-value function and the optimal action-value function.

Definition 6. *Optimal state-value function ($v_*(s)$) is equal to the maximum state-value function in all policies.*

$$v_*(s) = \max_\pi v_\pi(s), \ \forall \ s \in S. \tag{8}$$

Definition 7. *Optimal action-value function ($q_*(s,a)$) is the maximum action-value function in all policies.*

$$q_*(s,a) = \max_\pi q_\pi(s,a), \ \forall \ s \in S. \tag{9}$$

Refer to [29] for more details about reinforcement learning.

3.3. Reinforcement Learning Techniques

In this section, the most important RL methods, including dynamic programming (DP) and deep reinforcement learning (DRL) and their features, are presented [14,30]. These schemes are briefly presented in Table 1.

- Dynamic programming (DP) assumes that there is a complete model of the environment, such as the Markov decision process (MDP). DP consists of a set of solutions that are used to compute the optimal policy according to this model.
- Monte Carlo (MC) approaches are known as the free-model RL techniques, meaning that they do not need to know all the features of an environment. These approaches interact with the environment to achieve experiences. MC methods solve the reinforcement learning problem by averaging sample returns. They are episodic. As a result, MC assumes that the experience is divided into episodes. At the end step, all episodes will be finished no matter what action is selected. Note that the agent can only change values and policies at the end of an episode. Therefore, MC is an incremental episode-by-episode method.
- Q-learning is one of the most important RL algorithms. In this algorithm, the agent tries to learn its optimal actions and store all the state-action pairs and their corresponding values in a Q-table. This table includes two inputs, state and action, and one output called Q-value. In Q-learning, the purpose is to maximize the Q-value.
- State–action–reward–state–action (SARSA), similar to Q-learning, tries to learn MDP. However, SARSA, dissimilar to Q-learning, is an on-policy RL technique that chooses its actions by following the existing policy and changing Q-values in a Q-table. In contrast, an off-policy RL method such as Q-learning does not pursue this policy and selects its actions using a greedy method to maximize the Q-values.
- Deep reinforcement learning (DRL) uses deep learning to improve reinforcement learning and solve complex and difficult issues. Deep learning helps RL agents to become more intelligent, and improves their ability to optimize policies. Compared to other machine learning techniques, RL does not need any dataset. In DRL, the agent interacts with the environment to produce its dataset. Next, DRL uses this dataset to train a deep network.

Table 1. Comparision of reinforcement learning algorithms [15].

Algorithm	Advantage	Disadvantages
Dynamic programming (DP)	Acceptable convergence speed	Considering a complete model of the environment, high computational complexity
Monte Carlo (MC)	A free-model reinforcement learning method	High return variance, low convergence speed, trapping in local optimum
Q-learning	A free-model, off-policy, and forward reinforcement learning	Lack of generalization, inability to predict the optimal value for unseen states
State–action–reward–state–action (SARSA)	A free-model, on-policy, and forward reinforcement learning	Lack of generalization, inability to predict the optimal value for unseen states
Deep reinforcement learning (DRL)	Suitable for solving problems with high dimensions, the ability to estimate unseen states, the ability to generalization	Being unstable model, making rapid changes in the policy with little change in Q-value

4. Flying Ad Hoc Networks

Recent advances in wireless technologies, easy access to radio interfaces, and other equipment such as positioning systems, sensors, and microcomputers, lead to the production of smart and small flying vehicles, especially unmanned aerial vehicles (UAVs) that form a new network called a flying ad hoc network (FANET) [28,31]. In general, a flying ad hoc network consists of a group of UAVs that cooperate with each other and communicate without any infrastructure to perform a specific task without human intervention [32,33]. In FANETs, all UAVs can establish UAV-to-UAV communication and only a small number of UAVs can communicate with the ground station (GS) [34,35]. As a result, UAVs do not require complex hardware. When breaking communication links between UAVs, the connection between the ground station and ad hoc network is always active [36,37]. However, these networks face several challenges:

- **Connectivity:** In this network, UAVs have high mobility and low density, which cause the failure of communications between UAVs and affect network connectivity. These features lead to unstable links and high delay in the data transmission process [38].
- **Battery:** The biggest challenge in these networks is energy consumption because small UAVs use small batteries to supply their required energy for real-time data processing, communications, and flight [38].
- **Computational and storage capacity:** Flying nodes have limited resources in terms of storage and processing power. It is another challenge in FANETs that must be considered when designing a suitable protocol for sending data packets by UAVs to the ground station [38].
- **Delay:** Multi-hop communications are suitable for ad hoc networks such as FANETs, which are free-infrastructure, to guarantee end-to-end connectivity. However, this increases delay in the data transmission process. As a result, providing real-time services on these networks is a serious challenge [38].
- **Interference management:** UAVs are connected to each other through wireless communication. Due to the limited bandwidth capacity in this communication model and dynamic topology in FANETs, interference management is difficult and complex [38].

In FANETs, there are three types of communication between UAVs:

- **UAV-to-UAV communication (U2U):** As shown in Figure 2, UAVs communicate with each other using U2U communication in a multi-hop manner. This improves their communication range and increases the data rate. This communication is used when a UAV wants to send its data packet to another UAV or ground station beyond its communication radius [39].
- **UAV-to-GS communication GS (U2G):** As shown in Figure 2, UAVs communicate directly with GS through U2G communication when it is in their communication range.

In this communication, GS provides the necessary services to flying nodes, and UAVs send important data to the ground station.

- **Hybrid communication:** This is a combination of U2U and U2G communications and helps UAVs to send their data to GS in a single-hop or multi-hop manner using intermediate nodes [39].

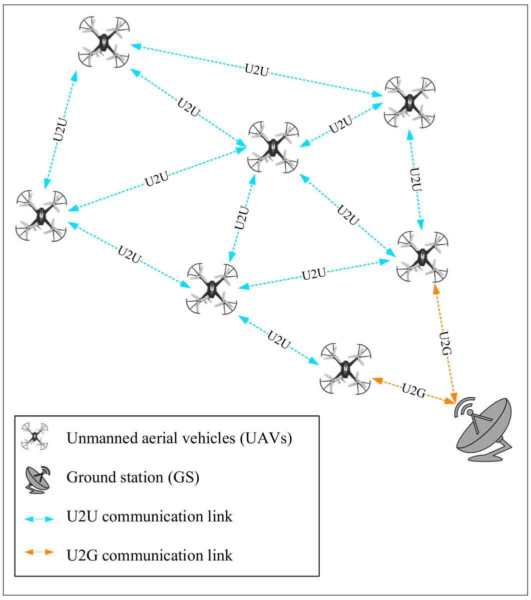

Figure 2. Various types of communication in the flying ad hoc network.

4.1. Unmanned Aerial Vehicles (UAVs)

Progress in communication technologies, sensors, and electronic equipment has facilitated the manufacture of unmanned aerial vehicles (UAVs). These aerial systems, also known as drones, can fly automatically and pilot without human intervention. A UAV includes various equipment such as a power system, control system, different sensors, and communication module [40,41]. These components are shown in Figure 3.

- **Power system:** This system is responsible for supplying the energy needed for data processing, sending and receiving data, flying, and especially controlling rotors through one or more batteries [42,43].
- **Control system:** This component is responsible for managing flight operations such as changing the flight height by rotating rotors based on the defined commands [42,43].
- **Sensor:** This module is responsible for sensing the environment and sending the collected data to the control system [42,43].
- **Communication module:** This module is responsible for receiving and sending information through radio signals or WiFi [42,43].

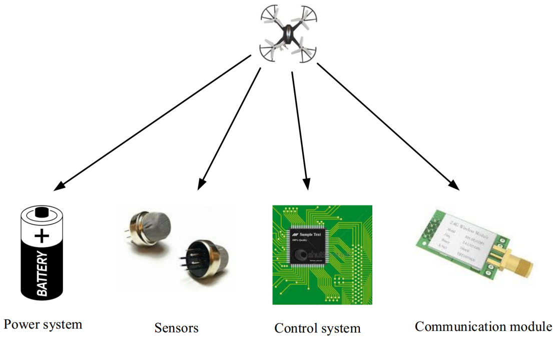

Figure 3. Components of a UAV.

Today, various types of drones have been designed and manufactured. They have unique features and are suitable for specific applications. For example, drones can be divided into two categories based on configuration:

- **Rotary-wing UAVs (RW-UAVs):** These drones can be fixed in the air and perform vertical take-off and landing. As a result, they are more stable and more suitable for indoor areas. However, these drones have higher energy restrictions, slower speeds, and a low capacity compared to fixed-wing UAVs. These features affect their flight time because this time depends on various factors such as path plan, speed, weight, and energy source. RW-UAV is represented in Figure 4a [44,45].
- **Fixed-wing UAVs (FW-UAVs):** These drones have a longer flight duration, higher flight speed, and aerodynamic design compared to rotary-wing UAVs. These drones can be used for aerial surveillance. They include one body and two wings and are made in different sizes (small and large). However, their weight is more than rotary-wing UAVs. They are similar to traditional aircraft, and cannot be fixed in the air. Therefore, these drones are not suitable for fixed applications. FW-UAVs are displayed in Figure 4b [44,45].

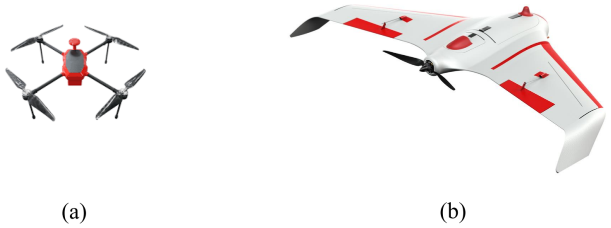
(a) (b)

Figure 4. Types of UAVs (**a**) rotary-wing UAV (**b**) fixed-wing UAV.

Moreover, UAVs are divided into two groups based on autonomy level: remote control-based and fully automatic.

- **Remote control-based UAV:** These drones are piloted directly by a pilot in line of sight (LoS) or based on feedback received from UAV sensors [45].
- **Fully automatic UAV:** They perform the flight operations completely independently and without human intervention and can complete their mission when faced with unforeseen operational and environmental conditions [45].

In addition, drones are categorized into two classes based on operating height: low-altitude and high-altitude.

- **Low-altitude UAVs:** These drones are almost small, lightweight, and cheap. One feature of these drones is that they can easily deploy in an area and fly at a low flight altitude (from 10 meters to a few kilometers). Their speed is very high. However, they suffer from energy restrictions, and therefore, have a short flight duration. Due to fast and easy deployment, these drones are suitable for time-sensitive applications and can be used as an aerial station to provide a high data rate and wireless services at high-risk areas, sports events, and festivals [46].
- **High-altitude UAVs:** These drones are almost heavy and large and perform their flight operations at a high altitude (more than 10 km). Compared to low-altitude drones, high-altitude UAVs have a longer flight duration and can cover a wide area. However, they are more expensive than low-altitude drones and their deployment is more difficult. They are suitable for applications, which need a longer flight duration and more area coverage. For example, internet broadcasting, remote sensing, and navigation [46].

Additionally, drones can be divided into two groups based on size: small and large.

- **Large UAVs:** These drones are commonly used in single-UAV systems to carry out some specific missions. They are expensive. Furthermore, their maintenance and repair costs are very high because their structure is complex.

- **Small UAVs:** These drones can be used in multi-UAV systems and swarms, and are very useful for civilian applications because they have many advantages, including suitable price and less maintenance and repair costs compared to large UAVs. Furthermore, they have simpler structures and include lightweight equipment, such as a cheap body, small batteries, lightweight radio module, and microprocessor. However, their ability is less than larger UAVs. These drones have a short flight time and low coverage range because they are limited in terms of weight and payload. These restrictions can be a serious issue in important missions, for example, search and rescue scenarios.

4.2. Applications

Flying ad hoc networks are applied in various areas. In the following, we explain these applications, which are also shown in Figure 5.

- **Search and rescue operations:** In this application, flying ad hoc networks can act as the first defensive line during natural disasters due to their fast and easy deployment capability. Furthermore, drones play the role of human relief forces in high-risk areas and pursue specific goals such as finding the precise location of survivors or victims.
- **Wildfire monitoring:** Flying ad hoc networks can be used to monitor temperature, diagnosis, and prevent fire in forests.
- **Traffic monitoring:** Highway traffic monitoring is one of the FANET applications. Drones can easily perform this monitoring task to detect gridlock and report traffic management data. This is a viable and economical option. Moreover, these networks can achieve different real-time security solutions to provide security on roads and trains [43].
- **Reconnaissance:** In aerial surveillance applications, drones fly statically to identify a particular area without human intervention. During surveillance operations, drones collect images of the desired goals and sites in a wide area. This information is quickly processed and sent to a smart control station. When drones oversee a particular target or area, they periodically patrol the target to inspect and monitor their security goals. For example, the border police can identify illegal border crossing through a flying ad hoc network [43].
- **Agricultural monitoring:** This application is known as precision agriculture. It includes all information technology-based solutions and methods that monitor the health of agricultural products. This application can be upgraded by FANETs to overcome the existing problems in this area. In this case, drones collect information on the quality of agricultural products, growth, and chemical fertilizers in a short period of time and analyze them based on precision scales and criteria [42].
- **Remote sensing:** Recently, flying ad hoc networks are used with other networks such as wireless sensor networks (WSN). It includes the use of drones equipped with sensors and other equipment. These drones automatically fly throughout the area to obtain information about the desired environment.
- **Relaying networks:** In this application, drones act as aerial relays to send information collected by ground nodes to base stations efficiently and securely; for example, sending data produced by ground nodes in wireless sensor networks and vehicular ad hoc networks. UAVs are also used to increase the communication range of ground relay nodes.

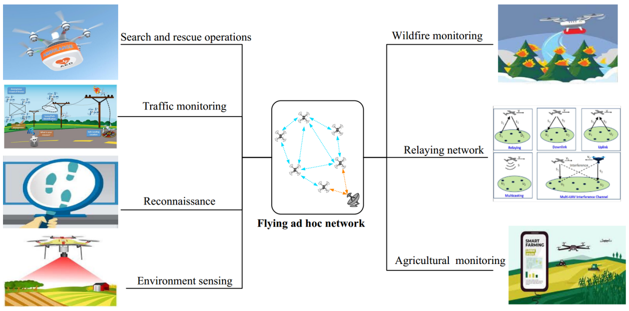

Figure 5. FANET applications.

4.3. Routing

Routing means sending information to a certain destination on the network. Given specific features of FANETs, it is a very challenging issue to design an efficient routing protocol in these networks. In the routing process, we answer the question *"How do UAVs send data packets from source to destination?"*. In this process, if there is no direct connection between the source and destination, the data packets must be relayed by intermediate nodes to reach the destination. It is known as a multi-hop routing [5]. According to this definition, the routing path is a series of hops (i.e., intermediate relay nodes) that are responsible for relaying packets. The task of routing protocols is to build such a route between the source and destination. Furthermore, these protocols should manage path failure by finding appropriate alternative routes. The routing process depends on the optimal path selection. However, it is always difficult to choose suitable criteria for deciding on the best route because an incorrect path selection leads to weak network performance. In flying ad hoc networks, there are many challenges that affect the routing process. These challenges are rooted in very dynamic topology, network partitioning, rapid movement of nodes, and frequent disconnections [7]. When designing routing protocols in FANETs, researchers must consider the application and quality of service (QoS) requirements, energy limitations of UAVs, load balancing, link stability, addition and removal of UAVs, and their mobility characteristics. Note that the drone movement is carried out in a three-dimensional space, which affects the quality of links between nodes. It is a major challenge when designing routing protocols. In addition, these protocols should efficiently consume network resources, especially energy, and consider mechanisms to prevent routing loops. Furthermore, researchers must take into account scalability. In terms of efficiency, the routing process should have low routing overhead, high reliability, low packet loss, and acceptable delay. In general, the following points should be taken into account when designing the routing protocols:

- **Limited resources:** One of the main challenges in small-sized drones is resource restrictions, including energy, processing power, storage capacity, communication radius, and bandwidth. These restrictions affect the routing protocols. For example, a small communication radius proves that routing protocols should be designed in a multi-hop manner to send data packets to the destination node by assisting intermediate nodes. Limited storage capacity and processing power also indicate that the routing protocol should be optimal and lightweight. Additionally, limited storage capacity and bandwidth can affect the size of packets exchanged on the network. Constrained energy also states that intermediate relay nodes should be carefully selected [13].
- **Dynamic topology:** FANETs have a highly dynamic topology, which is rooted in the failure of drones due to hardware malfunction, battery discharge, environmental conditions, and the mobility of UAVs. Thus, wireless links between flying nodes must be constantly re-configured. As a result, the routing protocols should be sufficiently flexible to adapt to the dynamic network topology [47].
- **Scalability:** Flying ad hoc networks have various sizes. This means that they are different in terms of the number of nodes and covered geographical area.

Therefore, scalability should be considered when selecting relay nodes (next-hop nodes) on the routing path [13].

- **Partitioning and void areas:** Another important challenge is that the routing process may be faced with network partitioning and void areas in the network because the FANET is a low-density network. The network partitioning means that one or more network parts cannot connect with other network parts, and the nodes in these parts cannot communicate with the nodes in other network parts. Furthermore, the void area means that one part of the network environment is disconnected, meaning that this area is not covered by flying nodes because there is no node in that part that connects to the outside nodes [47].
- **Delay:** Delay is the time required to transmit a data packet from the source to destination. When designing a routing algorithm, delay should be considered because real-time applications, such as monitoring, are sensitive to delay. In these applications, a high delay in the data transmission process can lead to unpleasant results [28].
- **Packet delivery rate (PDR):** It is equal to the ratio of the number of data packets delivered to the destination to the total number of packets sent by the source. Obviously, routing protocols need a higher packet delivery rate. If routing protocols are weakly designed, and the formed paths include routing loops, this has a negative effect on PDR [48].
- **Adaptability:** This means that routing protocols must quickly react to the network dynamics. For example, if a routing path is broken due to the link failure or discharging battery of nodes, the routing protocol should quickly find the alternative route.
- **Load balancing:** Routing protocols must evenly distribute their operational load, including energy consumption, calculations, and communications in the network, so that no route does not consume resources faster than other routes [28].
- **Routing loops:** The routing process should be free-loop to achieve a successful packet delivery rate.
- **Routing overhead:** Routing protocols must have low routing overhead, meaning that drones can communicate with each other with the least overhead in an efficient routing protocol.
- **Communication stability:** High mobility of nodes and different environmental conditions such as climate changes can lead to the disconnection of communication links. Therefore, a routing technique must guarantee communication stability in the network.
- **Bandwidth:** In applications such as aerial imaging, it is very important to consider bandwidth because there are restrictions such as communication channel capacity, drone speed, and sensitivity of wireless links relative to error.

5. Proposed Classification

In this section, we present a detailed classification of reinforcement learning-based routing algorithms in flying ad hoc networks. This classification consists of three groups:

- Based on the reinforcement learning algorithm;
- Based on the routing algorithm;
- Based on the data dissemination process.

Figure 6 displays the proposed classification.

5.1. Classification of RL-Based Routing Protocols Based on Learning Algorithm

Reinforcement learning algorithms can solve challenges and issues related to the routing process, which are mentioned in Section 4.3. These algorithms use intelligent flying nodes that observe and collect information from the network environment to make an optimal policy for deciding on the best routes in the network. In the proposed categorization, the routing methods are divided into two categories based on the learning algorithm: reinforcement learning-based routing and deep reinforcement learning-based routing. In the following, we explain the most important characteristics of the two groups. Moreover, a comparison between these two schemes is presented in Table 2.

Table 2. Comparison of RL-routing methods and DRL-routing schemes.

Routing Scheme		Convergence Speed	Computational Cost	Generalization	Learning Speed	Scalability	State and Action Spaces	Implementation	Fault-Tolerance
RL-based	**Single-agent**	Low	Low	No	Low	Low	Small	Simple	No
	Multi-agent	High	Very high	No	High	Medium	Medium	Complex	Yes
	Model-based	High	Very high	No	Low	Low	Small	Complex	No
	Free-model	Medium	Low	No	Medium	Medium	Small	Simple	No
DRL-based	**Single-agent**	High	High	Yes	High	High	Large	Complex	No
	Multi-agent	Very high	Very high	Yes	Very high	High	Large	Complex	Yes
	Model-based	High	Very high	Yes	Medium	High	Large	Complex	No
	Free-model	High	High	Yes	High	High	Large	Complex	No

5.1.1. Traditional Reinforcement Learning-Based Routing Method

In this routing protocol, the agent learns the network environment without any initial knowledge and finds the suitable path between the source and destination. This learning system is shown in Figure 7. Compared to deep learning reinforcement-based routing methods, these routing protocols are simpler and have less computational complexity. Therefore, they are easier to implement. Note that traditional reinforcement learning algorithms are suitable for discovering the best routing policy in small-scale FANETs because the dimensions of the state and action spaces are controllable, and the learning algorithm has an acceptable learning speed. However, if the flying ad hoc network is large-scale, traditional reinforcement learning algorithms cannot perform well to find the best routing policy in the network because the state and action spaces are large, and the learning algorithm has a slow convergence speed [49].

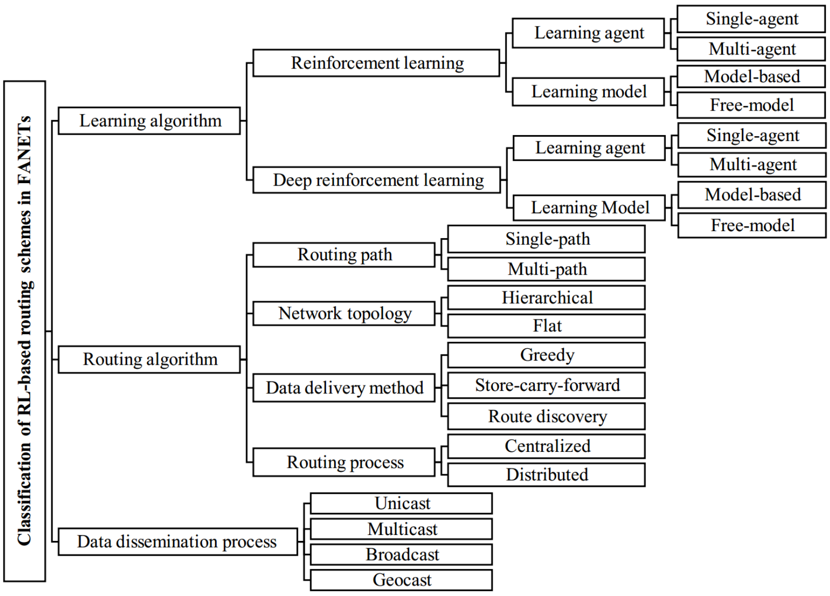

Figure 6. Classification of reinforcement learning-based routing protocols.

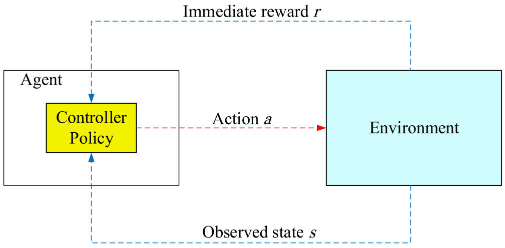

Figure 7. Traditional reinforcement learning system.

According to the proposed classification in this paper, the routing protocols are divided into two categories based on learning agent: single-agent routing and multi-agent routing.

Single-Agent Routing

In these routing methods, an agent alone learns the best behavior or the best route between the source and destination through interactions with the environment [50,51]. A single-agent reinforcement learning system is represented in Figure 8. The performance of these routing methods is not suitable for large-scale networks with large state and action spaces because the agent needs a long time to find an optimal response, meaning that their convergence speed is slow. In some cases, the agent may never find an optimal policy for the network. However, compared to multi-agent routing schemes, these methods are easier implemented because their computational complexity is lower than multi-agent routing approaches [52].

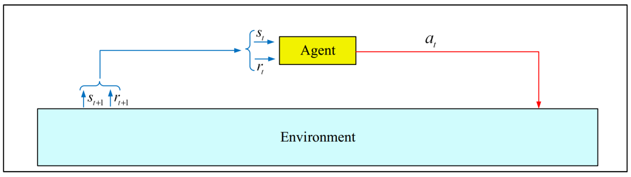

Figure 8. Single-agent learning system.

Multi-Agent Routing

In these routing methods, a group of agents (for example, UAVs) tries to learn the optimal behavior through interactions with the network to find the best route between the source and destination. A multi-agent reinforcement learning system is shown in Figure 9. An important challenge in these routing methods is how to coordinate and cooperate among agents because the behavior of agents can affect the dynamics of the network environment [50]. In these routing methods, if the agents can communicate with each other and exchange their experiences, the routing calculations are performed parallel by the agents, and the learning ability of the multi-agent system is greatly improved. These routing protocols are fault-tolerant, meaning that if one or more agents fail in the network for any reason, other agents can perform their tasks to prevent an abnormal network performance [51,52]. Furthermore, these protocols are suitable for networks with large state and action spaces because they have more learning ability. However, they have more computational complexity than single-agent methods.

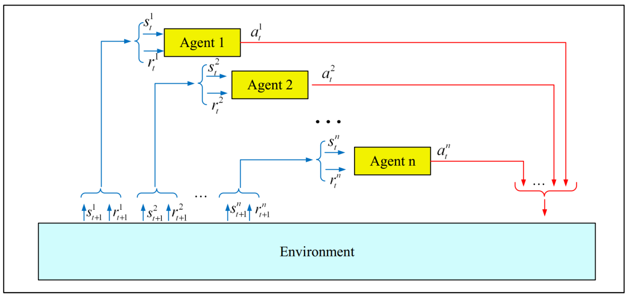

Figure 9. Multi-agent learning system.

In addition, according to the suggested classification in this paper, RL-based routing protocols are categorized into two classes based on the learning model: model-based and free model.

Model-Based Routing

In these routing methods, the task of the agent is to construct a model based on experiences obtained from the environment. This model is used for estimating the value function. A model-based learning system is shown in Figure 10. Compared to free-model routing methods, model-based routing schemes are data-efficient, meaning that they need less interaction with the environment to learn an accurate estimation of the value function [53,54]. Another feature of these methods is their flexibility against the sudden changes in the network. However, the computational complexity of these methods is very high and is not suitable for time-sensitive applications. Note that the performance of these routing methods is acceptable when the learning agents have sufficient computational resources. However, this is extremely challenging for large-scale networks, which have large state and action spaces.

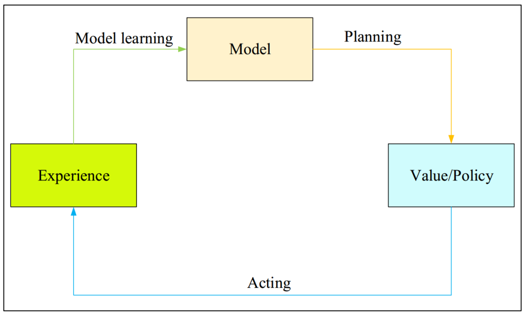

Figure 10. Model-based learning system.

Free-Model Routing

In these routing protocols, the task of the agent is to estimate the value function directly based on knowledge obtained from the environment and does not create any model of the network environment. Figure 11 shows a free-model learning system. Compared to model-based learning methods, these routing methods have appropriate computational complexity and are suitable for large and time-sensitive applications [53,54]. However, these routing methods must perform more interactions with the environment to obtain more experiences for finding an optimal response and have less flexibility relative to the sudden changes made to the network.

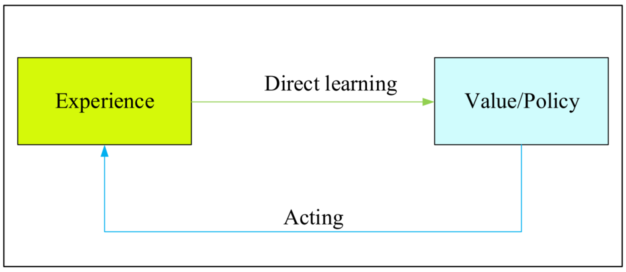

Figure 11. Free-model learning system.

5.1.2. Deep Reinforcement Learning-Based Routing Method

These routing protocols utilize deep reinforcement learning (DRL) in the routing process. This technique integrates deep learning (DL) with reinforcement learning (RL). This learning system is shown in Figure 12. These routing schemes can solve complex problems in FANETs. When the size of network is large, the value estimation and the optimal policy calculation are not simple. Therefore, a proper solution is to use a deep network to approximate these parameters. The deep network makes a high-intelligent agent and increases its ability to find the optimal policy. This routing protocol is a good choice for finding routing paths in large-scale networks because their learning speed is very high [49]. Note that, similar to the RL-based routing methods, the DRL-based routing approaches are also divided into two categories, single-agent and multi-agent, based on the learning agent. Additionally, these methods are categorized based on the learning model into two groups, model-based and free model, which were explained in Section 5.1.1.

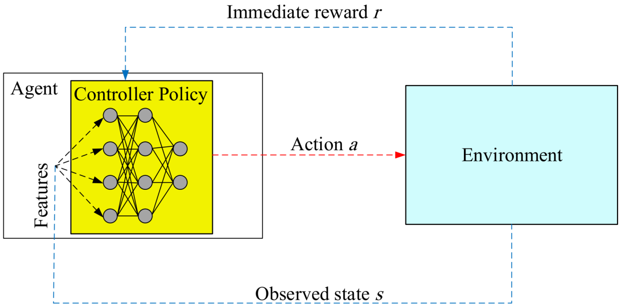

Figure 12. Deep reinforcement learning system.

5.2. Classification of RL-Based Routing Protocols Based on Routing Algorithm

In the proposed classification, reinforcement learning-based routing methods can be examined based on the routing algorithm in four aspects:

- Routing path;
- Network topology;
- Data delivery method;
- Routing process.

5.2.1. Routing Path

In this section, the routing methods are categorized into two classes according to the routing path: single-path routing and multi-path routing. Table 3 compares these two methods.

- **Single-path routing:** It means that only one route is formed between the source and destination. Single-path routing is shown in Figure 13. Compared to multi-path routing, this routing method can easily manage routing tables in each UAV. However, single-path routing is not fault-tolerant, meaning that there is no alternative path for sending data packets when failing the routing path. This increases packet loss in the network.

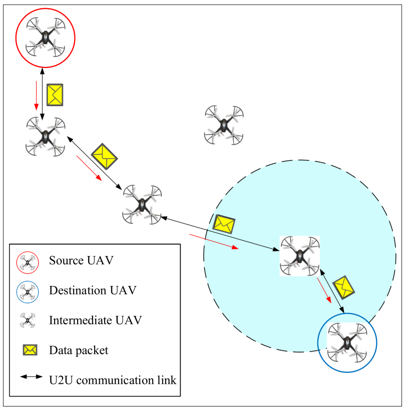

Figure 13. Single-path routing.

- **Multi-path routing:** This routing technique creates several routes between the source and destination [55]. Multi-path routing is shown in Figure 14. In this case, it is more difficult to maintain routing tables in UAVs because a UAV may act as intermediate nodes in two or more different paths. This routing method is fault-tolerant, meaning that if one path fails, it is easy to detect and replace this failed path. However, the configuration of this routing scheme is more difficult than the single-path routing approaches because the least errors cause routing loops in the network.

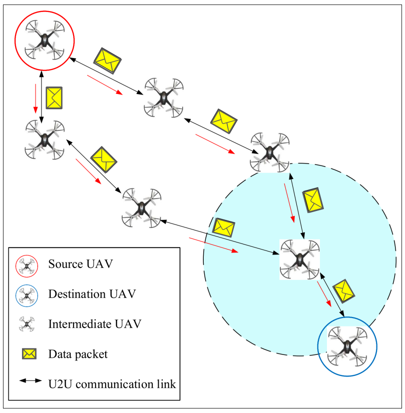

Figure 14. Multi-path routing.

Table 3. Comparison of single and multiple paths.

Routing Scheme	Routing Table Management	Packet Loss	Fault-Tolerance	Network Congestion	Routing Loop
Single-path	Simple	High	No	High	Low
Multi-path	Complex	Low	Yes	Low	High

5.2.2. Network Topology

In this section, routing methods are divided into two groups based on the network topology: hierarchical and flat. These two techniques indicate how to execute the routing process in the network. In the following section, we have explained these approaches. Furthermore, Table 4 has made a comparison between these routing algorithms.

- **Hierarchical routing:** In this routing technique, UAVs are divided into several hierarchical levels shown in Figure 15. At each level, UAVs can be directly connected to each other. Furthermore, they are connected to a node called the parent node (at their upper level) to communicate with other UAVs at the upper level. The parent node is responsible for managing its children (UAVs at the lower level) and sending their data to the UAVs at the upper level. In this method, determining the different roles for nodes leads to the efficient consumption of the network resources in the route calculation process and reduces routing overhead. This method is scalable. However, there are important research challenges, including the management of different roles and the selection of parent nodes, especially if the UAVs have high mobility in the network. These challenges should be considered in these methods.

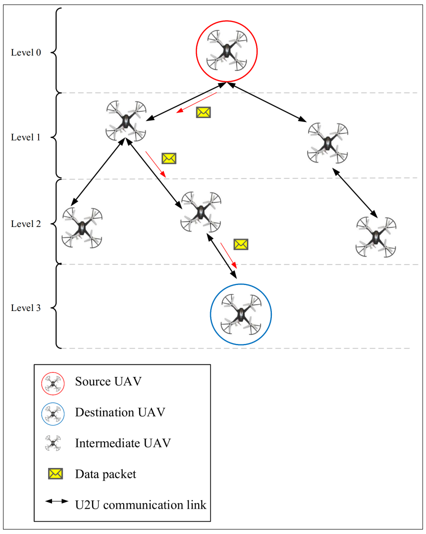

Figure 15. Hierarchical routing scheme.

- **Flat routing:** In a flat routing scheme, all UAVs play a similar role in the network, meaning that, dissimilar to hierarchical methods, it does not define any different roles such as parent node and cluster head node in the network to manage the routing process. The flat routing scheme is shown in Figure 16. In this approach, each UAV executes a simple routing algorithm and makes its routing decisions based on its status and neighbors on the network. These routing methods suffer from low scalability and high routing overhead.

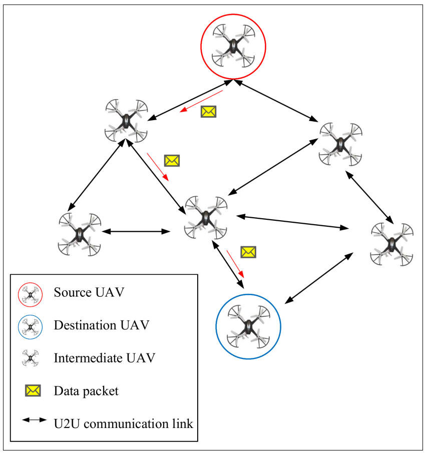

Figure 16. Flat routing scheme.

Table 4. Comparison of hierarchical and flat routing methods.

Routing Scheme	Management of Node Roles	Scalability	Routing Overhead	Network Congestion	Energy Consumption	Network Lifetime
Hierarchical	Difficult	High	Low	Low	Low	High
Flat	Simple	Low	High	High	High	Low

5.2.3. Data Delivery Method

In this section, the routing schemes are divided into three categories based on data delivery method: greedy, store-carry-forward, and route discovery. Table 5 presents a comparison between different data delivery methods. In the following, we describe these techniques in summary:

Table 5. Comparison of data delivery methods.

Data Delivery Scheme	Bandwidth Consumption	Scalability	Routing Overhead	Network Density	Delay in the Data Transmission Process	Delay in the Route Discovery	Packet Loss	Local Optimum	Broadcast Storm
Greedy	Low	High	Low	High	Low	Low	High	Yes	No
Store-carry-forward	Very low	High	Very low	Low	Very High	Very high	Very low	No	No
Route discovery	High	Medium	High	High and low	Medium	High	Medium	Low	Yes

- **Greedy:** The purpose of this routing technique is to reduce the number of hops in the created path between the source UAV and destination UAV. The main principle in the greedy method is that the closest UAV to the destination is geographically selected as a next-hop node, and this process continues until the packet reaches the destination. Figure 17 shows this process. The performance of this method is desirable when the network density is high. However, trapping in a local optimum is the most important weakness of this method. In this case, the data transmission process is stopped at the nearest node to the destination because there is no node closer to the destination. As a result, a path recovery technique is used to find an alternative path and guarantee the reliability of this method.

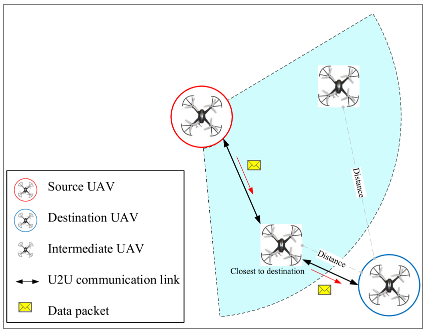

Figure 17. Greedy routing technique.

- **Store-carry-forward:** This routing technique is efficient when the network is periodically connected and the source node fails to find an intermediate node to send its data packets. In this case, the source UAV must carry this data packet until it finds a suitable relay or destination node. This process is presented in Figure 18. According to this figure, the source UAV has no intermediate node around itself to send data packets. Therefore, it carries its data until it meets the destination UAV. This method is beneficial in low-density networks such as FANETs. In addition, this routing method has low routing overhead and is scalable. However, it is not suitable for real-time applications because it increases delay in the data transmission process.

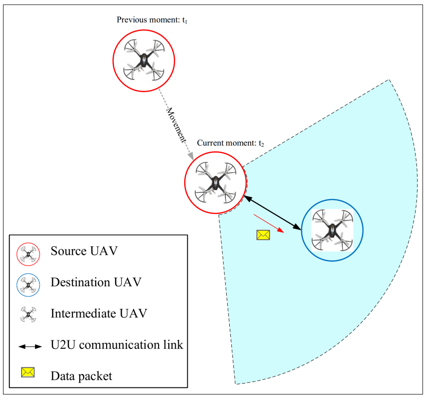

Figure 18. Store-carry-forward technique.

- **Route discovery:** When the source UAV does not know the geographic position of the destination UAV, the route discovery technique is suitable. In this case, UAVs use the flooding technique and broadcast the route request (RREQ) packets to discover all possible paths to the destination UAV. After receiving the RREQ packet, the destination node is responsible for selecting a suitable path among the discovered paths based on certain criteria. Finally, this route is used to transfer the data packet between source and destination. This process is shown in Figure 19. This routing technique is highly regarded by researchers due to its simplicity. However, it has a high routing overhead due to flooding messages. This issue can lead to a broadcast storm in some cases. This greatly increases bandwidth consumption.

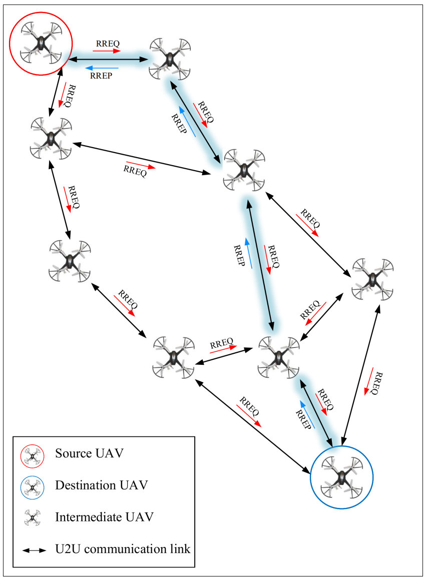

Figure 19. Route discovery technique.

5.2.4. Routing Process

In this section, the routing methods are divided into two categories based on the routing process: centralized and distributed. These methods are compared in Table 6. In the following section, we explain these techniques in summary.

- **Centralized routing:** In this routing method, a central server manages the routing process. This process is shown in Figure 20. This scheme assumes that the central server has global knowledge of the entire network. The central server is responsible for managing all UAVs and calculating the optimal routing paths on the network. The most important advantage of these methods is that the central server can fully control the entire network and obtains optimal routes at the lowest computational cost. However, this routing technique has disadvantages, such as server maintenance cost, lack of fault-tolerance, single point of failure, high delay, and high routing overhead. In highly dynamic networks such as FANET, it is difficult or even impossible to obtain complete knowledge of the network by the central server. For this reason, these methods are not successful in FANETs, and are not scalable.

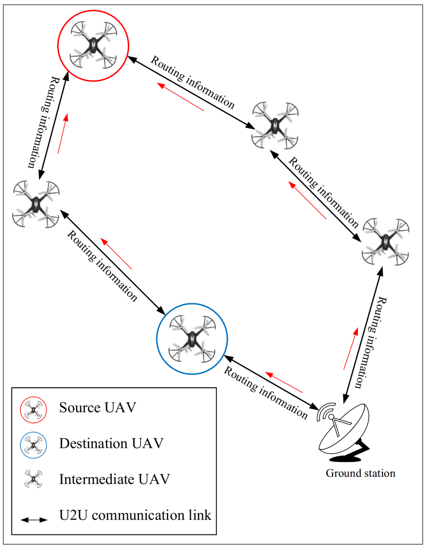

Figure 20. Centralized routing scheme.

- **Distributed routing:** In this routing method, UAVs share their information with their neighboring UAVs to obtain local knowledge from the network. Then, each UAV participates in the routing process and decides on routing paths based on this limited knowledge. This process is shown in Figure 21. This technique is scalable and flexible because UAVs can quickly and locally react to any issue related to the network dynamics. Therefore, these methods are more suitable for real-time applications. Due to relying on local information, distributed routing methods may form sub-optimal routes and distribute loads using an unbalanced manner in the network. In addition, these methods have more computational overhead compared to centralized routing methods.

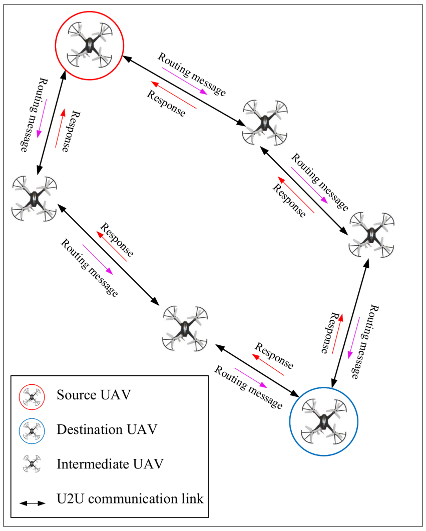

Figure 21. Distributed routing scheme.

Table 6. Comparison of centralized and distributed routing schemes.

Routing Scheme	Scalability	Routing Overhead	Computational Overhead	Single Point of Failure	Adapting with Dynamic Topology	Fault-Tolerance
Centralized	Low	High	Low	Yes	No	No
Distributed	High	Low	High	No	Yes	Yes

5.3. Classification of RL-Based Routing Protocols Based on the Data Dissemination Process

In this section, the routing methods are divided into four classes based on the data dissemination process: unicast, multicast, broadcast, and geocast. Table 7 compares these methods with each other. In the following section, we describe these four categories in summary.

- **Unicast-based routing:** This routing technique uses point-to-point communication, meaning there is only one source and one destination. In unicast-based routing, UAVs need to know the precise location of themselves and the destination. Thus, they must use a localization system such as a global positioning system (GPS). Figure 22 displays this routing technique. Note that the data traffic in FANETs, similar to other wireless networks, has a broadcast nature. Therefore, a unicast-based routing technique is not compatible with the nature of these networks. These routing methods suffer from problems such as high communication overhead compared to the multicast technique, high delay in the route discovery process, and high bandwidth consumption, and show poor performance in dynamic topology networks [56–58].

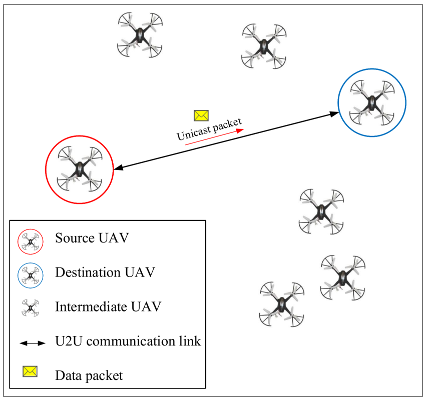

Figure 22. Unicast-based routing.

- **Multicast-based routing:** Multicast means that data packets are disseminated for a group of UAVs on the network. This method is suitable for applications with limited energy and bandwidth. Multicast-based routing must define multiple multicast groups. A multicast group includes a set of UAVs. Therefore, if a UAV wants to receive a multicast message, it must become a member of a multicast group because when the source UAV sends a multicast message to the multicast group, all group members receive this message. This process is shown in Figure 23. Multicast-based routing protocols use tree-like or mesh-like structures to transfer multicast data from the source to a group of destination nodes. The main weakness of this routing technique is that it must constantly reconstruct the routing tree when changing the network topology [59,60]. This is very challenging in FANETs.

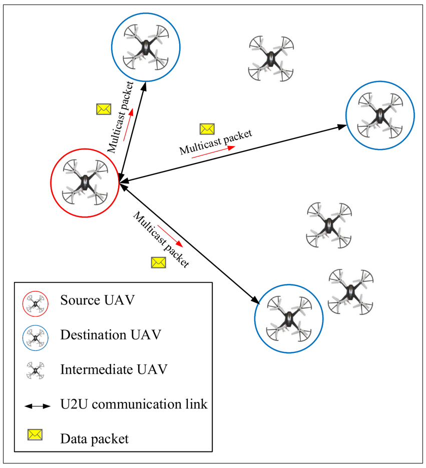

Figure 23. Multicast-based routing.

- **Broadcast-based routing:** In this technique, UAVs flood routing messages in the whole network. This process is represented in Figure 24. The flooding strategy can increase the reception probability of the routing message by the destination, but it consumes a lot of bandwidth. This strategy does not need the spatial information of UAVs in the network and is implemented easily. This method is useful for low-density networks. However, it has a weak performance in dense networks and can cause communication overhead, network congestion, and the broadcast storm problem. The most important disadvantage of this technique is high energy and bandwidth consumption. In addition, this process has a lot of redundancy [61–63].

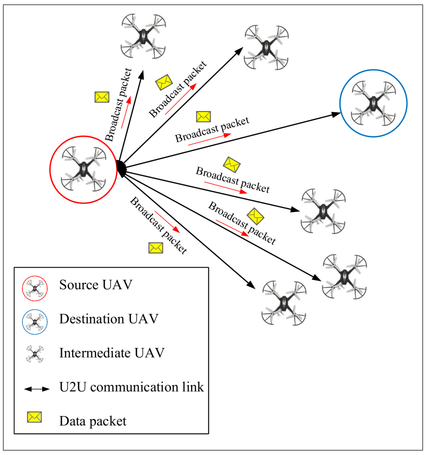

Figure 24. Broadcast-based routing.

Table 7. Comparison of various data dissemination methods.

Routing Scheme	Location Information	Routing Overhead	Delay in the Route Discovery	Implementation	Network Density	Bandwidth Consumption	Energy Consumption	Broadcast Storm
Unicast-based	Yes	High	High	Complex	Low or high	High	High	No
Multicast-based	Yes	Low	Low	Complex	Low or high	Low	Low	No
Broadcast-based	No	High	Low	Simple	Low	Very high	Very high	Yes
Geocast-based	Yes	Low	Low	Complex	Low or high	Low	Low	No

- **Geocast-based routing:** It is a type of multicast technique. The purpose of this routing technique is to send data packets from the source to all UAVs in a particular geographic area. This process is shown in Figure 25. In this method, the geographic area is inserted into each geocast packet. Then, the geocast packet is delivered to the UAVs in that area. In this method, a geocast group includes a set of nodes in a particular geographic area. To determine the members of this group, each node must know its geographical location on the network. Therefore, they need a positioning system [64–66].

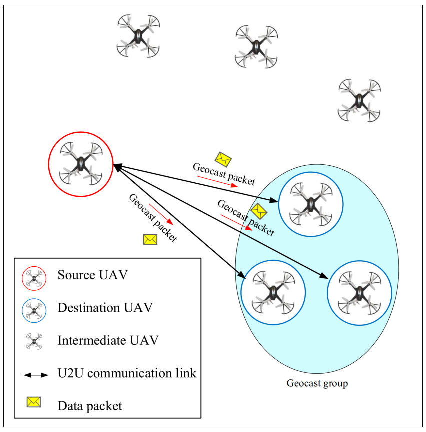

Figure 25. Geocast-based routing.

6. Investigating Reinforcement Learning-Based Routing Protocols

In this section, we review the state-of-the-art RL-based routing methods based on the suggested categorization in this paper.

6.1. DQN-VR

Khan et al. [67] have presented a deep Q-network-based vertical routing (DQN-VR) in the 5G flying ad hoc networks. This method combines distributed and centralized routing techniques. In DQN-VR, 5G technology supports flying ad hoc networks to improve scalability and stability and balance the load distribution in the network. 5G is a new-generation wireless network, which has important features such as Internet connectivity with higher download and upload speeds, wider coverage, and more stability. In 5G, the network is divided into three different levels, including macro-plane, pico-plane, and femto-plane. It includes a central controller (CC) for managing global information, and distributed controllers (DCs) for managing local information. In DQN-VR, CC is responsible for managing, collecting, and processing global data such as remaining energy because these data are less dynamic and expire at a longer duration compared to local information. Therefore, this data can be updated at longer time intervals. DCs are also responsible for managing, collecting, and processing local data such as spatial and movement information of UAVs. These data are more dynamic and expire quickly in a short time. As a result, they must be updated at shorter time intervals. This information is periodically exchanged through beacon messages by UAVs. The beacon message includes spatial information, neighborhood degree, and UAV movement. DQN-VR consists of two main phases: vertical clustering and vertical routing. In the first phase, DCs are responsible for implementing the vertical clustering process to form the clusters at each network level to improve cluster stability and network scalability and reduce the end-to-end delay. The purpose of the clustering process is to group UAVs based on their nature and behavior in clusters. Each cluster consists of a cluster head node (CH), cluster member nodes (CMs), cluster gateway (CG), and vertical cluster gateway (VCG). The cluster head node is responsible for managing cluster operations such as routing, intra-cluster communications, and inter-cluster connections. Cluster member nodes are directly connected to their CH and form intra-cluster communications. The cluster gateway is responsible for creating connections between clusters at the same network level. Furthermore, the vertical cluster gateway is responsible for communicating between different clusters at different network levels. DC uses the movement pattern of UAV and its neighbors and its transmission range to calculate its degree. Then, it chooses nodes with a higher degree as CH. Then, DC determines the cluster member nodes based on the distance between each UAV and CH node so that UAVs are connected to the nearest CH. Among the CMs, nodes with the least intermediate nodes between two clusters are selected as the cluster gateway. Moreover, the vertical cluster gateway represents a cluster member node with the highest link expiration time. In the second phase, CC is responsible for executing a vertical DQN-based routing to determine different routes at the different network levels to create intra-level and inter-level paths. In this process, CC plays the agent role and the network is considered as the environment. The state set represents a two-dimensional array including movement information and the residual energy of the nodes. The action set also indicates the selection of the next-hop node towards the destination. Note that the next hop can be CH, CM, CG, VCG, and BS. The reward function is calculated based on the successful packet transmission rate and the congestion level in the nodes. Note that the DQN parameters, including the learning rate and the discount factor, are also experimentally tested and considered as fixed values. Figure 26 shows the learning process in DQN-VR. Furthermore, Table 8 illustrates the most important advantages and disadvantages of this routing scheme.

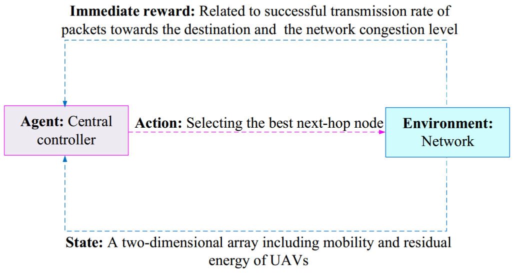
Immediate reward: Related to successful transmission rate of packets towards the destination and the network congestion level
Agent: Central controller
Action: Selecting the best next-hop node
Environment: Network
State: A two-dimensional array including mobility and residual energy of UAVs

Figure 26. Learning process in DQN-VR.

Table 8. The most important advantages and disadvantages of DQN-VR.

Scheme	Advantage	Disadvantages
DQN-VR [67]	Designing a clustering process, reducing communication overhead, reducing end-to-end delay, managing network congestion, high scalability, utilizing both distributed and centralized routing techniques, utilizing deep reinforcement learning algorithm in the routing process, improving the learning rate.	Not designing the adaptive broadcast mechanism for controlling beacon messages in the network, the ambiguity of how to update local and global information in CC and DCs, the ambiguity of how to calculate the reward function including congestion level and data transmission rate, ignoring the CH rotation process, not determining how to manage different network levels and how to change the level of UAVs in the network, not determining the optimal number of clusters and CHs at each level, fixing the DQN parameters.

6.2. QTAR

Arafat and Moh in [68] have suggested a Q-learning-based topology-aware routing method (QTAR) for flying ad hoc networks. QTAR tries to discover the best route between the source and destination using two-hop neighbors' information such as neighboring position, delay, speed, and energy. This method balances the load on the network because the energy level of nodes is used in the learning process. In addition, QTAR prevents routing loops because it uses the two-hop neighbors' information to prevent blind transmission. Moreover, reducing the number of hops in the created path is one of the key goals of two-hop neighboring information. Initially, each UAV obtains its location on the network using GPS and shares this information through the hello message with its two-hop and single-hop neighboring nodes. This message includes node position, link information, energy, speed, mobility model, and queuing delay. According to this message, a neighborhood table is formed in each node. It includes spatial information, energy level, mobility model, speed, and queuing delay. In QTAR, each UAV dynamically adjusts the hello broadcast period and the link holding time based on the shortest lifetime of links between itself and its neighboring nodes to adapt itself to the dynamic network topology. The link lifetime is calculated based on the distance between a UAV and its neighbors and their relative velocities. In QTAR, the Q-learning algorithm is responsible for learning the routing process in a distributed manner. In this process, each packet plays the agent role and neighboring nodes are considered as a state space. To manage the size of the state space, QTAR only puts nodes that are closer to the destination compared to the current node in the state set. This improves the convergence speed of this routing protocol. In this learning issue, the selection of the next-hop node is considered an action. Finally, the reward function is calculated based on three parameters: energy, delay, and speed. The learning process in QTAR is shown in Figure 27. Note that QTAR adjusts learning parameters, including learning rate and the reward factor dynamically, because if these parameters have constant values in dynamic networks, the selected action is not accurate. In this method, the learning rate is adjusted based on the delay information obtained from two-hop neighbors and the reward factor is calculated based on the speed and distance changes between each UAV and its neighbors. Additionally, QTAR considers a penalty mechanism to prevent routing holes. According to this mechanism, if the next-hop node is trapped in the routing holes or the previous-hop node does not receive the acknowledgment message from the next-hop node, it reduces the reward value, corresponding to the next-hop node to prevent the selection of this node in the routing process. Table 9 summarizes the most important advantages and disadvantages of this routing scheme.

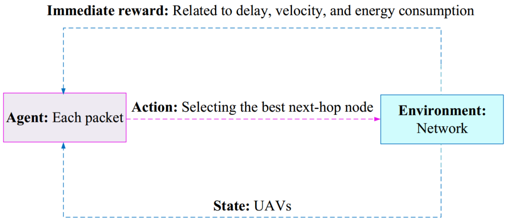
Immediate reward: Related to delay, velocity, and energy consumption
Agent: Each packet
Action: Selecting the best next-hop node
Environment: Network
State: UAVs

Figure 27. Learning process in QTAR.

Table 9. The most important advantages and disadvantages of QTAR.

Scheme	Advantage	Disadvantages
QTAR [68]	Designing a distributed routing process, controlling the size of state space, improving convergence rate, high scalability, adaptive adjustment of the hello broadcast period, designing a penalty mechanism to avoid falling into the routing holes, adjusting the learning parameters, including learning rate and discount factor dynamically, balancing energy consumption in the network, improving network lifetime, preventing the blind path problem by adjusting the link holding time based on the link lifetime.	High communication overhead, slow convergence speed in large-scale networks despite trying to limit the state space, taking into account a flat network topology and ignoring the clustering process.

6.3. TQNGPSR

Chen et al. [69] have proposed a traffic-aware Q-network geographic routing scheme based on greedy perimeter stateless routing (GPSR) called TQNGPSR for flying ad hoc networks. This routing protocol introduces a traffic balancing strategy, which utilizes congestion information of neighboring nodes to evaluate the wireless link quality. The best route between the source and destination is selected based on this evaluation to reduce delay and packet loss in the data transmission process. The Q-network algorithm is responsible for comparing the quality of the candidate links and selecting a link with the largest Q-value. Each node calculates the congestion level based on the buffer queue length received from its neighbors. It is proportional to the queuing delay. When the current node sends a data packet to the next-hop node. It returns an ACK message to the previous-hop node. The ACK message includes the buffer length information. In addition, each node periodically exchanges a hello message including the queue length information and spatial coordinates of the neighboring node, and updates its neighborhood table immediately after receiving any hello message from its neighbors. In the routing process, Q-values are first calculated regardless of congestion information. Next, the congestion penalty process is designed to update Q-values based on the buffer queue information. For this purpose, the reward function is recalculated based on the buffer queue information. If the buffer queue of a UAV is almost full, the node receives a high penalty in the congestion penalty process, and its corresponding Q-value will be reduced to lower the selection chance of this node as the next-hop node in the future. In TQNGPSR, each packet maintains a visit list (VL), which includes nodes visited by the packet. Therefore, when a UAV receives a packet, the node uses VL to know what UAVs were met. As a result, it sends the packet to a node, which is far from the visited area to prevent a local optimum. For this purpose, each UAV calculates the angle between its own and the neighboring nodes in VL and obtains the minimum angle. Given this angle, the Q-network can estimate Q-values corresponding to other neighbors. Therefore, the current node selects the next-hop node among nodes whose angles are larger than the available nodes. In the learning process, each data packet plays the agent role, and the state represents the node that holds the data packet. Additionally, the action indicates the selection of a neighboring node as the next-hop node. Figure 28 shows the learning process in TQNGPSR. In this method, the learning parameters, including the learning rate and the discount factor, are empirically selected. Table 10 presents the most important advantages and disadvantages of this routing scheme in summary.

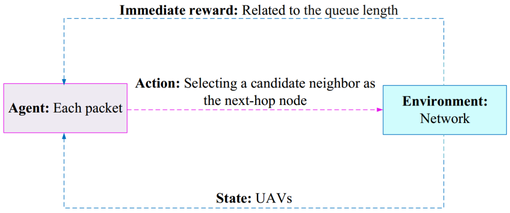
Immediate reward: Related to the queue length
Action: Selecting a candidate neighbor as the next-hop node
Agent: Each packet
Environment: Network
State: UAVs

Figure 28. Learning process in TQNGPSR.

Table 10. The most important advantages and disadvantages of TQNGPSR.

Scheme	Advantage	Disadvantages
TQNGPSR [69]	Designing a distributed routing process, utilizing deep reinforcement learning algorithm, high scalability, appropriate convergence speed, designing a traffic balancing strategy, preventing network congestion, reducing delay and packet loss in the routing process, designing a mechanism for avoiding to trap in a local optimum.	High routing overhead, not managing the size of the state space, not designing a mechanism to adjust the hello broadcast interval, considering a flat network topology, and ignoring the clustering process, not considering energy consumption in the routing process, considering fixed learning parameters.

6.4. QMR

Liu et al. [70] have presented the Q-learning multi-objective optimization routing protocol (QMR) for FANETs. In the routing process, QMR attempts to reduce delay and energy consumption in the data transmission process between the source and destination. In order to balance energy consumption, QMR considers the energy factor in the routing process. In this method, UAVs periodically exchange their information, including geographic location, residual energy, mobility model, queuing delay, and discount factor through hello messages. QMR also adjusts an adaptive hello broadcast interval based on node speed. In the learning process, the entire network is considered as the environment and each packet plays an agent role. The state set contains UAVs, and the action represents a decision to send the packet from the current node to a neighboring node. Figure 29 represents the learning process in this protocol. In QMR, the learning parameters are dynamically determined based on the network conditions. The learning rate is calculated based on one-hop delay, and the discount factor is characterized based on the movement of neighboring nodes in two consecutive time intervals. In addition, QMR has presented a new exploration and exploitation mechanism to balance exploration and exploitation based on the speed specified for the data packet. According to this mechanism, when selecting the next-hop node, the allowed speed for sending the packet from the current node to the next-hop node is calculated, and nodes that meet the allowed speed can be selected as the next-hop node. This idea filters the state space, reduces its size, and improves the convergence speed of the Q-learning algorithm. As a result, this reduces the end-to-end delay when sending the data packet to the destination. After calculating Q-values, a weight coefficient is computed based on the two parameters, including the link quality and the intimacy of the neighboring node and the current node. This coefficient is used to select the next-hop node among nodes, which have the highest weighted Q-value and meet the allowed packet speed. If the candidate node set is empty and there is no neighboring node, which meets the allowed packet speed, then the current node selects the next-hop node from the neighboring nodes, which have a speed greater than zero. In this case, it selects the neighboring node with the largest speed. Otherwise, if there is no node, which has a speed greater than zero, QMR sets up a penalty mechanism. The purpose of this mechanism is to prevent routing holes. According to this mechanism, if the next-hop node is encountered by a routing hole or does not send an ACK message to the previous-hop node, the previous-hop node reduces its reward to lower the selection chance of this node as the next-hop node in the future. Table 11 expresses the most important advantages and disadvantages of this routing scheme in summary.

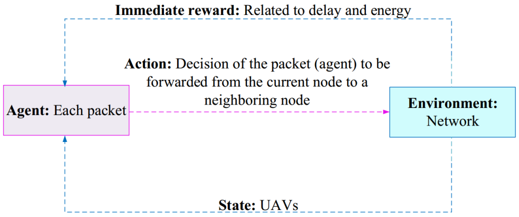
Immediate reward: Related to delay and energy
Action: Decision of the packet (agent) to be forwarded from the current node to a neighboring node
Agent: Each packet
Environment: Network
State: UAVs

Figure 29. Learning process in QMR.

Table 11. The most important advantages and disadvantages of QMR.

Scheme	Advantage	Disadvantages
QMR [70]	Designing a distributed routing process, high scalability, managing the size of state space, appropriate convergence speed, adaptive adjustment of hello broadcast interval, designing a penalty mechanism to prevent routing holes, balancing energy consumption in the network, improving network lifetime, designing a new exploration and exploitation mechanism, reducing delay and energy consumption in the data transmission process, adjusting learning parameters dynamically.	High communication overhead, increasing the size of the state space in large-scale networks and reducing the convergence speed, taking into account a flat network topology and ignoring the clustering process, not considering the mobility pattern of the nodes in the routing process.

6.5. QGeo

Jung et al. [71] have introduced the Q-learning-based geographic routing method (QGeo) for flying ad hoc networks. The purpose of this method is to control routing overhead and improve the packet delivery rate in highly dynamic networks. In QGeo, the best route between the source and destination is discovered in a distributed manner without the need for global information. In the first step, each UAV periodically exchanges hello messages with its neighbors to share information such as the location of node, the current Q-value, and the link condition. The hello broadcast interval is adaptively adjusted based on node speed. As a result, QGeo is compatible with the dynamic network environment. According to the messages received from neighboring nodes, each UAV forms a neighborhood table to record neighbors' information in this table. Then, a Q-learning-based routing algorithm is designed to decide on routing paths. In this scheme, each UAV is defined as a state in the state space. Additionally, any action indicates the transition from the current node to a neighboring node. The learning process in QGeo is shown in Figure 30. This method introduces a new concept called the packet travel speed, which guarantees the reliable and fast data transmission process in the network. In this routing algorithm, the reward function is calculated based on the packet speed. Note that QGeo dynamically adjusts the discount factor based on the distance and speed of UAVs in the network. However, the learning rate has a constant value. Table 12 briefly describes the most important advantages and disadvantages of QGeo.

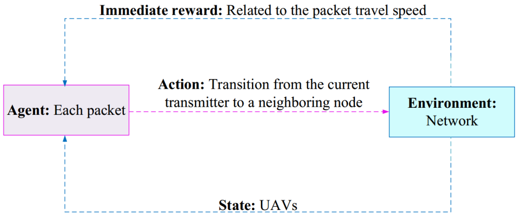
Immediate reward: Related to the packet travel speed
Action: Transition from the current transmitter to a neighboring node
Agent: Each packet
Environment: Network
State: UAVs

Figure 30. Learning process in QGeo.

Table 12. The most important advantages and disadvantages of QGeo.

Scheme	Advantage	Disadvantages
QGeo [71]	Designing a distributed routing process, adaptive adjustment of hello broadcast interval based on UAV speed, adjusting learning parameters dynamically.	Low scalability, high communication overhead, low convergence speed, enlarging the size of state space in large-scale networks, considering a flat network topology and ignoring clustering process, lack of attention to energy and the mobility pattern of nodes in the routing process, not balancing energy consumption in the network, reducing network lifetime, and not solving the routing hole problem.

6.6. QSRP

Lim and Ko in [72] have designed a Q-learning-based stepwise routing protocol (QSRP) for flying ad hoc networks. This method assumes that the network includes a central controller (CC) and a number of UAVs. First, CC executes a neighbor discovery process to calculate the link stability and the minimum number of hops to the destination node. Then, it uses a Q-learning algorithm to find paths with a minimum number of hops and high link quality. In the stepwise neighbor discovery process, the central controller broadcasts a discovery message on the network. This packet includes the ID of the central controller and hop count. Each UAV that receives this packet for the first time, increases the hop count to one unit and rebroadcasts this message on the network. In addition, it generates an ACK message including its ID, spatial information, speed, and the number of hops to CC, and sends this message to its parent node on the network. When the parent node receives the ACK message, it forms a neighborhood table to store this information. Then, it unicasts this message to its parent node until this message reaches CC. Note that the broadcast period of the discovery packet is adaptively adjusted based on speed and location of the nodes. After receiving ACK messages, CC creates a table that contains information about all network nodes. As a result, the central controller can obtain global knowledge of the whole network. According to this information, it calculates a link stability scale based on node speed and the quality of the link between the nodes. In QSRP, the link quality is achieved based on the packet delivery rate and the transmission time. Then, the central controller executes a Q-learning algorithm to find the best route between the network nodes. In this process, each node is considered as the state, and the action indicates the selection of the route between the nodes. Furthermore, the reward function is defined based on two parameters, including the number of hops and link stability to guarantee the transmission quality and reduce the number of re-transmission in the network. In QSRP, learning parameters, including learning rate and the discount factor, have constant values. Figure 31 shows this learning process in QSRP. Moreover, Table 13 presents the most important advantages and disadvantages of QSRP.

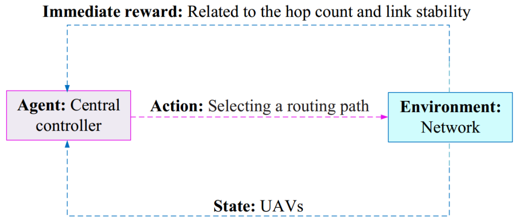
Immediate reward: Related to the hop count and link stability
Agent: Central controller
Action: Selecting a routing path
Environment: Network
State: UAVs

Figure 31. Learning process in QSRP.

Table 13. The most important advantages and disadvantages of QSRP.

Scheme	Advantage	Disadvantages
QSRP [72]	Adaptative adjustment of the discovery packet broadcast period based on the location and speed of UAVs, route stability	Designing a centralized routing method, low scalability, high communication overhead, low convergence speed, enlarging the size of state space in large-scale networks, considering a flat network topology and ignoring clustering process, lack of attention to the energy parameter, not balancing energy consumption in the network, reducing network lifetime, failure to solve the routing hole problem, considering fixed learning parameters, not performing enough tests

6.7. QLGR

Qiu et al. [73] have offered a Q-learning geographic routing method (QLGR) for flying ad hoc networks. This method uses a multi-agent reinforcement learning technique in the routing process. QLGR reduces packet loss and routing overhead in the network. Initially, UAVs periodically share their information with each other by exchanging hello messages. This message includes spatial information, sequence number, message length, Q-value, maximum queue length, and occupied queue length. In QLGR, each UAV acts as a smart agent, and the network is regarded as the environment. Furthermore, the state of each node indicates the state space at any moment, and the action space is a set of neighbors of the current node. After taking an action, the agent receives two types of feedback (i.e., local reward (LR) and global reward (GR)) from the environment. The local reward is calculated based on two parameters, including load capacity (obtained from queue length) and link quality (obtained from successful packet delivery rate). This reward only evaluates the fitness of the next-hop node, and cannot guarantee whether the next-hop node can send the data packet to the destination. In this case, the purpose of the routing process (i.e., transferring data to the destination or the next-hop node closer to the destination) is guaranteed by the global reward (GR). This learning process is shown in Figure 32. Note that QLGR considers learning parameters, including learning rate and the discount factor, as constant values. In this learning process, each node maintains a Q-table. This table only stores information about active neighboring nodes to save memory. Q-values in this table are updated after receiving each hello message. In the Q-table, the Q-value is considered a weight coefficient for choosing the best route. When a node wants to send data packets to the destination node, it should select the best next-hop node for sending information to the destination. For this purpose, it computes a score for each neighboring node based on its distance to the destination and multiplies this score by the Q-value corresponding to that node. Then, the node uses the softmax policy for choosing the next-hop node. If there is no node to select as the next-hop node, QLGR uses a path recovery strategy similar to GPSR to find the next-hop node to prevent routing holes. Moreover, QLGR determines the hello broadcast interval adaptively. It defines this issue as an MDP problem to select the best hello broadcast interval based on the change degree of neighboring nodes and the number of packets in the buffer queue. Moreover, Table 14 describes the most important advantages and disadvantages of QLGR in summary.

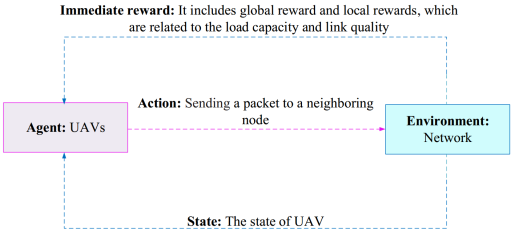
Immediate reward: It includes global reward and local rewards, which are related to the load capacity and link quality
Action: Sending a packet to a neighboring node
Agent: UAVs
Environment: Network
State: The state of UAV

Figure 32. Learning process in QLGR.

Table 14. The most important advantages and disadvantages of QLGR.

Scheme	Advantage	Disadvantages
QLGR [73]	Designing a distributed routing method, adaptive adjustment of hello broadcast interval for reducing routing overhead, creating stable paths, reducing packet loss, considering a multi-agent reinforcement learning technique, improving convergence speed, high scalability, using both local and global rewards, managing Q-table size, solving the routing hole problem by applying a routing recovery mechanism, preventing congestion in the network.	High routing overhead, considering a flat topology for the network and not paying attention to the clustering process, ignoring the energy of nodes in the routing process, the unbalanced distribution of energy consumption and reducing network lifetime, considering constant learning parameters.

6.8. FEQ-Routing-SA

Rovira-Sugranes et al. [74] have proposed an improved version of the Q-routing algorithm called fully-echoed Q-routing with simulated annealing inference (FEQ-routing-SA) for FANETs. In FEQ-routing-SA, the purpose is to dynamically adjust the learning rate in the Q-learning algorithm to be compatible with the frequent changes in topology in FANETs to reduce energy consumption and packet loss. FEQ-routing-SA has major advantages, such as acceptable computational complexity, low routing overhead, and local decision-making about routing paths. In the first step, FEQ-routing-SA proposes a trajectory construction process based on a piece-wise linear mobility model. In this method, a hierarchical generative model is introduced to produce random parameters for each UAV based on its class to deduce its motion profile. Then, each UAV implements the Q-learning-based routing process to find the most appropriate route between source and destination and reduce the energy required for data transmission. This routing procedure allows UAVs to decide on the previous experience and minimize the transmission energy required for sending the packet. In FEQ-routing-SA, no information is exchanged between nodes. This significantly reduces the routing overhead. In this process, each node plays the agent role and the status of a node, including location and energy, and is considered the state space. Furthermore, the reward function is defined based on the transmission energy required for transferring packets from the source to destination. This learning process is shown in Figure 33. Note that this scheme defines two learning rates: the basic learning rate, which is constant, and the extra learning rate estimated based on the delivery time. In this method, the simulated annealing algorithm is used to determine the exploration rate. SA starts at a large exploration rate (high temperature) and learns better decisions by lowering the temperature. In FEQ-routing-SA, the temperature is adjusted according to the speed of the network nodes. In this process, when changing node speed, the temperature reaches its highest level and gradually is cooled during the interval. The fitness function in this method is calculated based on energy consumption changes (or Q-value) in the last ten iterations. Table 15 expresses the most important advantages and disadvantages of FEQ-routing-SA in summary.

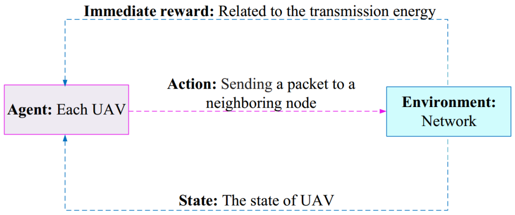
Immediate reward: Related to the transmission energy
Action: Sending a packet to a neighboring node
Agent: Each UAV
Environment: Network
State: The state of UAV

Figure 33. Learning process in FEQ-routing-SA.

Table 15. The most important advantages and disadvantages of FEQ-Routing-SA.

Scheme	Advantage	Disadvantages
FEQ-routing-SA [74]	Designing a distributed routing process, not needing to broadcast hello packets, decreasing routing overhead, improving convergence speed, adjusting dynamically learning rate, providing an efficient exploration-exploitation mechanism based on the SA algorithm, forming routes with the least transmission energy, designing a trajectory creation model, reducing packet loss, managing Q-table size.	Enlarging Q-table size in large-scale networks and decreasing convergence speed, taking into account a flat network topology and ignoring the clustering process, ignoring the movement directions and link quality in the routing process, not solving the routing hole problem, low scalability, not considering a mechanism to prevent congestion in the network.

6.9. Q-FANET

Costa et al. [75] have introduced a Q-learning-based routing method (Q-FANET) for flying ad hoc networks. The purpose of this method is to reduce delay in the route selection process to support the real-time applications. Q-FANET utilizes both QMR and Q-Noise+ routing methods. It consists of two modules: the neighbor discovery and the routing decision. The neighbor discovery process is responsible for updating the routing information. The update process is carried out by exchanging hello messages with neighboring nodes. In Q-FANET, the hello updating frequency is a constant value. This can increase routing overhead because Q-FANET does not consider the network dynamics. Each hello message includes geographical location, energy level, motion model, queuing delay, learning rate, and Q-value. After receiving this message, each UAV applies this information to create and maintain its neighborhood table. In the second step, Q-FANET utilizes two modules—namely, QMR and Q-learning. The Q-learning module uses an improved version of the Q-learning algorithm called Q-learning+ to perform the routing process. Q-learning only uses the latest episode for updating Q-values. This may cause false routing decisions. In contrast, Q-learning+ considers a limited number of the latest episodes for updating Q-values. This improves Q-learning. However, Q-learning+ does not consider channel conditions in the routing process. Thus, Q-FANET modifies the Q-learning+ algorithm and utilizes the channel conditions in the Q-value update process. This idea is inspired by Q-Noise+. In this modification, the transmission quality is evaluated by the signal-to-interference-plus-noise ratio (SINR) and is used in the Q-value update process. In the routing process, each packet plays the agent role, and UAVs are considered as the state space. In addition, sending a packet from the current node to the neighboring node (next-hop node) is defined as the action space. In Figure 34, this learning process is shown. Additionally, Q-FANET utilizes the penalty mechanism presented in QMR to solve the routing hole problem and the route failure issue. According to this mechanism, if a routing hole is created or the next-hop node does not send the ACK message to its previous-hop node, the reward value related to the next-hop node is equal to a minimum reward to prevent the selection of this node for data transfer in the future. Finally, Q-FANET introduces a speed constraint to obtain the least delay in the data transmission process. This constraint is similar to QMR. The most important advantages and disadvantages of Q-FANET are outlined in Table 16.

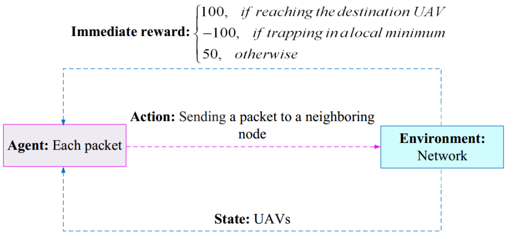
Immediate reward: $\begin{cases} 100, & \textit{if reaching the destination UAV} \\ -100, & \textit{if trapping in a local minimum} \\ 50, & \textit{otherwise} \end{cases}$
Action: Sending a packet to a neighboring node
Agent: Each packet
Environment: Network
State: UAVs

Figure 34. Learning process in Q-FANET.

Table 16. The most important advantages and disadvantages of Q-FANET.

Scheme	Advantage	Disadvantages
Q-FANET [75]	Designing a distributed routing method, reducing delay in the routing process, improving the packet delivery rate, solving the routing hole problem due to the use of penalty mechanism, paying attention to the link quality in the routing process.	Not having a mechanism for controlling the hello updating interval, high routing overhead, not managing Q-table size in large-scale networks and reducing convergence speed, low scalability, considering constant learning parameters, ignoring the energy of UAVs in the routing process, considering a flat topology network and ignoring the clustering process, not applying a mechanism to prevent congestion in the network.

6.10. PPMAC+RLSRP

Zheng et al. [76] have proposed an adaptive communication protocol—namely, the position-prediction-based directional MAC protocol (PPMAC) and the RL-based self-learning routing protocol (RLSRP) for FANETs. This routing method tries to quickly form communication paths and deliver data packets with low delay. PPMAC has three steps: position prediction, communication control, and data transfer. In the first step, UAVs obtain their speed and position using GPS. In PPMAC, each UAV periodically shares a position packet including ID, position information, antenna and status information, and path information with its own single-hop neighboring nodes. Thus, each node knows the position of its neighboring UAVs and can predict their future position. In the second step, the communication control process is performed by exchanging control packets. In the third step, the data are transferred from a flying node to the neighboring node. In the routing process, RLSRP updates the routing policy based on the position of UAVs and designs a reward function based on the transmission delay. RLSRP searches the shortest routes with the least delay. In this routing method, the routing process is defined as a partially observable Markov decision process (POMDP). In RLSRP, each node plays the agent role, and the status of the nodes is considered as the state space. Furthermore, the action indicates the transmission of the packet from the current node to the next-hop node. This learning process is shown in Figure 35. In this routing process, a greedy strategy is used to select the next-hop node, and the neighboring node with the highest value is selected as the next-hop node. In RLSRP, learning parameters, including learning rate and discount factor, have constant values. The most important advantages and disadvantages of PPMAC+RLSRP are outlined in Table 17.

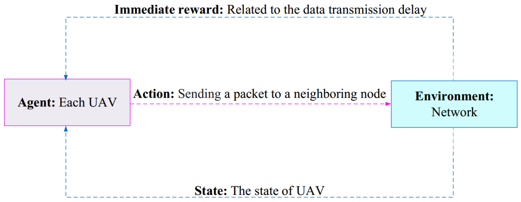
Immediate reward: Related to the data transmission delay
Agent: Each UAV
Action: Sending a packet to a neighboring node
Environment: Network
State: The state of UAV

Figure 35. Learning process in PPMAC+RLSRP.

Table 17. The most important advantages and disadvantages of PPMAC+RLSRP.

Scheme	Advantage	Disadvantages
PPMAC+RLSRP [76]	Designing a distributed routing method, reducing delay in the routing process, improving the packet delivery rate, taking into account the routing problem as a POMDP problem, predicting the accurate location of nodes in the network.	Not providing a mechanism for controlling the hello broadcast interval, high routing overhead, low scalability, enlarging Q-table size in large-scale networks and decreasing in convergence speed, considering constant learning parameters, ignoring the energy of UAVs, and link quality in the routing process, considering a flat network topology and not taking into account the clustering process, not considering a mechanism to prevent congestion in the network, not solving the routing hole problem.

6.11. PARRoT

Sliwa et al. [77] have designed the predictive ad hoc routing fueled by reinforcement learning and trajectory knowledge (PARRoT) for flying ad hoc networks. This routing method can reduce the end-to-end delay in the data transmission process. PARRoT consists of three steps: predicting cross-layer motion, distributing routing messages, and RL-based route maintenance. In the mobility prediction process, a cross-layer technique is used to acquire knowledge about the UAV motion pattern and predict relative mobility between different agents so that each agent can estimate its next position based on the current position. This information is broadcast through routing messages called chirp to obtain local knowledge of the network topology. The broadcast period of these chirp messages is a fixed time interval. As a result, PARRoT is not compatible with the dynamic network topology and may cause a high routing overhead. Chirp message contains location information and a sequel number to prevent routing loops and eliminate repeated messages. After receiving each chirp message, UAV first checks its sequence number to guarantee its freshness. Then, this information is used for updating the Q-table. In the Q-learning-based routing process, UAVs use the local information obtained from chirp messages to achieve a partial view of the network topology. To send a data packet to the destination node, each UAV only evaluates the fitness of its single-hop neighbors for reaching the destination node, and selects a neighbor with the highest Q-value as the next-hop node. PARRoT finds multiple paths to the destination, which improves fault tolerance and accelerates the route recovery process when failing the nodes in a path. In the routing process, each node plays the agent role and the set of UAVs is regarded as the state set. Moreover, the action indicates the selection of a neighboring node as the next hop. The Q-value update process is also performed by exchanging chirp messages. In the RL-based routing process, the learning rate has a constant amount, but the discount factor is dynamically evaluated based on the link expiry time (LET) and the change degree of neighbors at a specific time interval. This learning process is represented in Figure 36. Furthermore, Table 18 lists the most important advantages and disadvantages of PARRoT in summary.

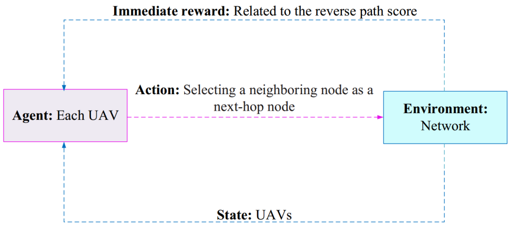
Immediate reward: Related to the reverse path score
Agent: Each UAV
Action: Selecting a neighboring node as a next-hop node
Environment: Network
State: UAVs

Figure 36. Learning process in PARRoT.

Table 18. The most important advantages and disadvantages of PARRoT.

Scheme	Advantage	Disadvantages
PARRoT [77]	Designing a distributed routing method, reducing delay in the data transmission process, improving the packet delivery rate, adjusting the learning parameters dynamically, predicting the location of nodes in the network, preventing routing loops, designing a multi-path routing, improving fault-tolerance.	Not controlling the chirp broadcast interval, high routing overhead, low scalability, enlarging Q-table size in large-scale networks and reducing convergence speed, not considering energy of UAVs and link condition in the routing process, using a flat network topology and ignoring the clustering process, not considering the congestion control mechanism, not solving the routing hole problem.

6.12. QFL-MOR

Yang et al. [78] have introduced a Q-learning-based fuzzy logic for multi-objective routing protocol (QFL-MOR) in FANETs. QFL-MOR uses various link-level factors and path-level parameters to evaluate the routing performance to find an optimal path between the source and destination. Link-level factors include transmission rate (TR), energy status (ES), and flight status (FS). Moreover, the route-level parameters include hop count (HC) and successful packet delivery time (SPDT). In this routing method, each UAV periodically shares its local information, such as speed, position, motion direction, and remaining energy with its neighboring UAVs to calculate link-level factors. Note that ES is calculated based on two parameters, including residual energy and the energy discharge rate, and FS is obtained from the speed and motion direction. In QFL-MOR, the routing process includes three steps. In the first step, each node (beginning from the source node) uses a fuzzy system to evaluate the quality of links between itself and neighboring nodes in terms of TR, ES, and FS to select an appropriate node with the best link quality as the next-hop node. This process continues until the data packet reaches the destination. Thus, a route is discovered between the source and destination at this step. In the second step, Q-learning evaluates the cost of the route created between the source and destination, and updates Q-values corresponding to hop count and successful packet delivery time in the selected path. In the last step, each node (beginning from the source node) uses a fuzzy system to improve the calculated path. Q-values related to the routing path—namely, HC and SPDT—and the link parameters such as TR, ES, and FS are considered as fuzzy inputs for this fuzzy system to improve the constructed route between the source and destination. The second and third steps are repeated to get the best route between the source and destination. In the learning process, the network is considered as an environment. Additionally, UAVs are corresponding to the state space, and the action indicates the selection of a neighboring node as the next-hop node. This learning process is shown in Figure 37. Note that in this process, the Q-learning parameters, including the learning rate and the discount factor, have constant values. Table 19 describes the most important advantages and disadvantages of QFL-MOR in summary.

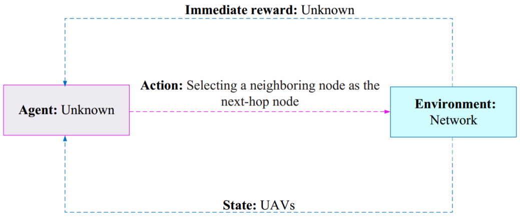
Immediate reward: Unknown
Action: Selecting a neighboring node as the next-hop node
Agent: Unknown
Environment: Network
State: UAVs

Figure 37. Learning process in QFL-MOR.

Table 19. The most important advantages and disadvantages of QFL-MOR.

Scheme	Advantage	Disadvantages
QFL-MOR [78]	Designing a distributed routing method, utilizing local and global parameters to calculate the optimal route, considering energy, transmission rate, and motion pattern of UAVs in the routing process, taking into account delay in the routing process, reducing delay in the data transmission process.	Not providing a mechanism for controlling the hello broadcast interval, high routing overhead, low scalability, enlarging Q-table size in large-scale networks and reducing convergence speed, considering a flat network topology, and ignoring the clustering process, considering constant parameters, not solving the routing hole problem, not performing enough tests.

6.13. 3DQ

Zhang et al. [79] have presented a three-dimensional Q-learning-based routing protocol (3DQ) for flying ad hoc networks. This method combines both greedy and store-carry-forward techniques. In 3DQ, each UAV is equipped with a global navigation satellite system (GNSS) to obtain its spatial information. Moreover, 3DQ considers the Gauss–Markov mobility model for simulating the motion of UAVs in the network. According to this mobility model, 3DQ has introduced a new parameter called the UAV degree towards the ground station (DTGS). This parameter is calculated based on the two parameters, including the communication radius of the ground station and the Euclidean distance between the UAV and GS. 3DQ includes two modules called the link state prediction and routing decision. The first module allows each node to predict the link state of its neighboring nodes based on their three-dimensional motion and packet arrival. In this module, there are two algorithms, called the degree towards the ground station prediction (DTGSP) algorithm and the packet arrival prediction (PAP) algorithm. In the DTGSP algorithm, the least square technique, which forecasts the next position of UAVs in the future, is used to obtain DTGS corresponding to each neighboring node. The PAP algorithm calculates a new parameter called the estimated next packet arrival time (ENPAT) for each neighboring node to evaluate its traffic quality. Then, the routing decision module uses the Q-learning algorithm to produce the best route between the source and destination based on the link status. In this module, the routing process is modeled as an MDP problem. In this issue, each UAV plays the agent role and the state space includes DTGS values related to the neighboring nodes. In addition, the action space is defined as a set of neighboring nodes and the current UAV. In the action space, the set of neighboring nodes is used for the greedy routing mode and the current UAV is used for the store-carry-forward mode. For this reason, 3DQ defines two reward functions. In the greedy mode, the reward function is calculated based on the difference of the DTGS after taking the selected action. In the store-carry-forward mode, the reward function is calculated based on the ratio of packet delay to the maximum delay. In Figure 38, this routing process is shown. The 3DQ updates Q-values in parallel to accelerate the convergence process. In the updating process, each UAV constantly performs an action with all its neighbors to receive a reward based on DTGS difference, ENPAT values of neighboring nodes, delay, and throughput. In this routing method, the learning parameters—namely, the learning rate and the discount factor—have constant values. Table 20 summarizes the most important advantages and disadvantages of 3DQ.

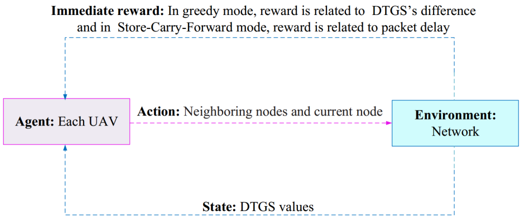
Immediate reward: In greedy mode, reward is related to DTGS's difference and in Store-Carry-Forward mode, reward is related to packet delay
Agent: Each UAV
Action: Neighboring nodes and current node
Environment: Network
State: DTGS values

Figure 38. Learning process in 3DQ.

Table 20. The most important advantages and disadvantages of 3DQ.

Scheme	Advantage	Disadvantages
3DQ [79]	Designing a distributed routing method, combining both greedy and store-carry-forward technique, preventing routing holes, predicting UAV motion and traffic in the network, reducing congestion in the routing process, paying attention to delay in the routing process, improving the packet delivery rate.	Not presenting a solution for sharing the position of nodes, high routing overhead, low scalability, enlarging Q-table size in large-scale networks and lowering convergence speed, considering a flat network topology, ignoring clustering process considering fixed learning parameters, and ignoring the energy of nodes in the routing process.

6.14. ICRA

Guo et al. [80] have presented an intelligent clustering routing approach (ICRA) for flying ad hoc networks. This method increases the stability of the network topology and improves network lifetime by balancing energy consumption in the network. In ICRA, the nodes are equipped with GPS to obtain their position and movement information. They broadcast hello messages periodically on the network to exchange information including location, speed, movement direction, and timestamp with other network nodes, and form a neighborhood table. This information is used in the clustering process. Clustering balances energy consumption in the network. ICRA includes three phases: clustering, clustering strategy adjustment, and routing. In the clustering process, each node calculates a utility parameter based on four utility factors, including residual energy, centrality, speed similarity between a node and its neighbors, and the link holding time. Then, UAVs share their utilities with each other to select a node with the highest utility as the cluster head node. Then, CH nodes broadcast an advertisement message to announce their role. When non-CH nodes receive advertisement messages from different CH nodes, they are connected to a CH node with higher utility and a longer link lifetime. If a cluster member node receives a hello message from a CH node except its own CH, this cluster member node acts as an inter-cluster forwarding node. CH and inter-cluster forwarding nodes are responsible for creating routes between different clusters. When calculating the utility parameter, the weight coefficients corresponding to the four utility factors are characterized by a Q-learning-based clustering strategy adjustment process. This learning process follows a centralized strategy and is implemented by the ground station (GS). In this process, GS plays the agent role and the action space includes the selection of four weight coefficients corresponding to the utility factors. Furthermore, the state space represents four utility factors. The reward function is also calculated based on two parameters, including stability of the cluster structure (i.e., the number of times changing the node roles in the network) and the energy change rate of the nodes. In this process, learning parameters, including learning rate and discount factor, have fixed values. This learning process is represented in Figure 39. In the routing process, when one UAV receives a data packet from another UAV, and is not the destination of this packet, it checks the location of the destination node. If the destination node is its single-hop neighbor, it sends the data packet directly to the destination node. Otherwise, the following three modes should be considered.

- Is the receiver node a CH? If yes, it sends the data packet to the closest CH node or inter-cluster forwarding node to the destination.

- Is the receiver node an inter-cluster forwarding node? If yes, it selects the closest node to the destination among its CH node and its neighboring nodes in other clusters and sends the packet to the selected node.
- Is the receiver node a cluster member node? If yes, it sends this data packet directly to its CH.

Table 21 briefly describes the most important advantages and disadvantages of ICRA.

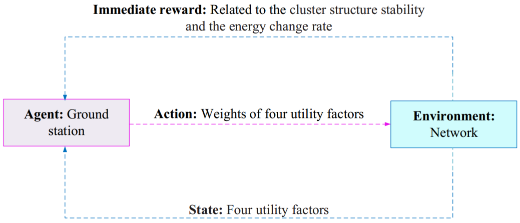
Immediate reward: Related to the cluster structure stability and the energy change rate
Agent: Ground station
Action: Weights of four utility factors
Environment: Network
State: Four utility factors

Figure 39. Learning process in ICRA.

Table 21. The most important advantages and disadvantages of ICRA.

Scheme	Advantage	Disadvantages
ICRA [80]	Designing a distributed routing method, utilizing clustering technique in the network, balancing energy consumption, improving the network lifetime, increasing the stability of the network topology, reducing delay in the data transmission process, improving packet delivery rate, lowering routing overhead in the data transmission process, reducing network congestion in the routing process, managing Q-table size in the network, improving convergence speed, high scalability, considering the consumed energy of UAVs in the clustering process.	Designing a centralized clustering strategy, considering constant learning parameters, not providing a solution to prevent routing holes, considering a constant hello updating time, not determining the optimal number of clusters in the network, considering inadequate parameters to select inter-cluster forwarding nodes.

6.15. TARRAQ

Cui et al. [81] have proposed the topology-aware resilient routing strategy based on adaptive Q-learning (TARRAQ) for flying ad hoc networks. In this method, UAVs periodically exchange hello messages to share their status information with other nodes and form a neighborhood table. The purpose of TARRAQ is to find routes, which reduce delay and energy consumption, and increase the packet delivery rate. TARRAQ tries to accurately obtain topology changes and execute the routing process using a distributed, autonomous, and adaptive manner. To achieve this goal, the queuing theory is used to analyze the dynamic changes in topology. This analysis is used for calculating two parameters, including the neighbor change rate (NCR) and neighbors' change inter-arrival time (NCIT) distribution to describe the dynamic behavior of UAVs. Then, the sensing interval (SI) or the hello broadcast time is achieved based on NCR and NCIT, so that UAVs can adjust SI based on dynamic behavior and performance needs. This technique reduces routing overhead in the neighbor discovery process. In addition, the residual link duration (LD) is estimated based on the Kalman filter (KF) method. It indicates the link expiration time for the neighboring nodes. In TARRAQ, the routing process is modeled by the Markov decision algorithm, and the Q-learning algorithm is used to solve this routing problem. In this learning process, each packet plays the agent role, and the entire network is regarded as the environment. Furthermore, the state space includes all UAVs, and the action space indicates the selection of a neighboring node as the next-hop node. In this process, the reward function is defined based on the quality of the link, energy, and distance between neighbors. This learning process is represented in Figure 40. In this routing method, learning parameters, including learning rate and discount factor, are calculated based on the residual LD. Table 22 briefly expresses the most important advantages and disadvantages of TARRAQ.

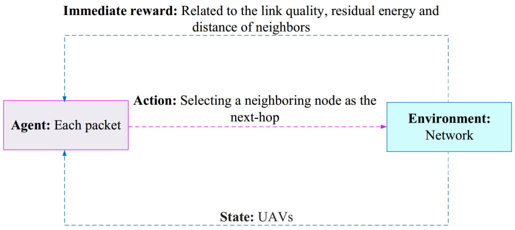
Immediate reward: Related to the link quality, residual energy and distance of neighbors
Action: Selecting a neighboring node as the next-hop
Agent: Each packet
Environment: Network
State: UAVs

Figure 40. Learning process in TARRAQ.

Table 22. The most important advantages and disadvantages of TARRAQ.

Scheme	Advantage	Disadvantages
TARRAQ [81]	Designing a distributed routing method, determining adaptive sensing interval, adjusting learning parameters dynamically, reducing packet loss, paying attention to energy in the routing process, reducing delay in the data transmission process, predicting dynamic behavior of nodes in the network, estimating link lifetime using the Kalman filter.	High routing overhead, low scalability, enlarging Q-table size in large-scale networks and decreasing convergence speed, considering a flat network topology and ignoring the clustering process, using only RWP mobility model for simulating the motion of UAVs and not adapting with other mobility models.

7. Discussion

In this section, RL-based routing methods in FANETs are reviewed and compared in various aspects. The most important features of these routing methods are summarized in Table 23. According to this table, we can deduce that most RL-based routing schemes rely on local knowledge about the network environment. QSRP is the only RL-based routing method which relies on global knowledge obtained by the central controller. This global information includes the number of hops, location, and speed of UAVs in the network. However, the feasibility of this routing scheme is ambiguous, and it is very difficult or even impossible to implement this scheme in FANET because the topology of these networks changes rapidly and the global information obtained from the network will be invalid in a very short period. However, DQN-VR and QFL-MOR use both local and global knowledge in the routing process. In DQN-VR, researchers have presented an attractive idea. They believe that less dynamic information, such as residual energy that requires longer updating intervals, can be collected as global information. In contrast, high dynamic data such as location and speed, which change quickly, can be collected as local information. Then, global and local information are used in the routing process. This idea is more practical than QSRP and can be considered by researchers in the future. In addition, QTAR uses the information of single-hop and two-hop neighbors in its routing process to increase the local view of the agent relative to the network environment. However, this increases routing overhead in the network. Additionally, RL-based routing schemes usually use hello messages for obtaining local information about their neighbors in the network. However, the periodic broadcast of these messages between UAVs imposes a lot of communication overhead in the network and consumes network resources, including energy and bandwidth, and increases congestion in the network. In QTAR, QMR, QGeo, QLGR, QSRP, and TARRAQ, researchers have adjusted the hello broadcast interval with regard to the speed of the network topology changes to manage communication overhead as much as possible. It is essential to pay attention to the energy problem in FANETs, which includes small drones with limited energy sources. DQN-VR, QTAR, QMR, FEQ-routing-SA, QFL-MOR, ICRA, and TARRAQ have attempted to improve energy consumption on the network. Clustering is one useful solution that balances the energy consumption of nodes and improves network lifetime. ICRA and DQN-VR utilize the clustering technique to balance energy consumption in the network because this technique reduces communication overhead and increases network scalability. Scalability is a major challenge when designing RL-based routing methods for large-scale networks because the size of the Q-table depends on the network size. This issue can greatly reduce the speed of the learning algorithm so that achieving an optimal response will be practically impossible. The blind path issue is another challenge when designing RL-based routing protocols. This issue means that the discovered paths in the network have expired before they reach their deadline, while nodes are unaware of this issue and send their data packets through the failed paths. This issue increases packet loss in the network. This problem has been resolved in QTAR, QMR, QLGR, Q-FANET, PARRoT, and TARRAQ.

In Table 24, RL-based routing methods are compared in terms of simulation tools, mobility model, localization service, and network environment. Network simulator version 3 (NS3), MATLAB, and WSNet are the most common simulation tools for simulating routing protocols to evaluate their performance in a virtual network environment under different scenarios. These tools help researchers to analyze the performance of these methods carefully before use in real environments. GPS is the most common positioning service, which is used to find the position and speed of UAVs in FANET. However, this positioning service is highly costly in terms of bandwidth consumption. On the other hand, when routing protocols use a positioning system in their routing process, their performance is dependent on the positioning system, meaning that if the positioning system is not accurate and UAVs cannot calculate their position accurately, the routing protocol is also not accurate. FEQ-routing-SA is the only routing technique that does not use position information in the routing process. Researchers can regard this idea in the future because it reduces routing overhead and prevents congestion in the network. Furthermore, random waypoint (RWP) and Gauss–Markov(GM) are the most common mobility models for simulating drone movement on the network. Another important point is that 3DQ has implemented the movement of UAVs in the network only based on the Gauss–Markov mobility model and relies on this model throughout the routing process. Additionally, TARRAQ has limited drone movement to the RWP mobility model. However, it is not an accurate model for simulating drone movement. When a routing method is designed based on a particular mobility model, this issue makes serious challenges for adapting these routing protocols with other mobility models, as well as the actual motion of UAVs in a real network environment. This weakens the network performance. Another important point is that the FANET is a three-dimensional network environment. However, some RL-based routing methods such as TQNGPSR, QMR, QGeo, QLGR, Q-FANET, OFL-MOR, 3DQ, and ICRA are simulated in a two-dimensional environment, which is not compatible with the FANET environment.

Moreover, Table 25 specifies that RL-based routing methods have been evaluated in terms of some routing criteria. In this evaluation, we consider the most important routing scales, including energy consumption, end-to-end delay, network lifetime, packet delivery rate, throughput, connectivity, and routing overhead. Most researchers evaluate their routing methods in terms of packet delivery rate and end-to-end delay because FANET is a dynamic network that suffers from frequent link failure in the network. This significantly decreases the validity time of the paths formed in the network. Additionally, the path failure causes high packet loss in FANETs compared to other ad hoc networks. Repairing and reconstructing the failed communication routes are also time-consuming. Therefore, the two important purposes for many researchers are (1) reducing delay and (2) increasing the packet delivery rate in the routing protocols designed for FANETs.

Table 23. Comparison of RL-based routing methods.

Scheme	Knowledge	Neighbor Information	Route Discovery Message	Adatptive Adjustment of Hello Broadcast Interval	Routing Loop	Energy Balancing	Scalability	Blind Path Problem
DQN-VR [67]	Local and global	×	Hello	×	×	✓	High	×
QTAR [68]	Local	Single and two-hop neighbors	Hello	✓	✓	✓	High	✓
TQNGPSR [69]	Local	Single-hop neighbors	Hello	×	✓	×	Medium	×
QMR [70]	Local	Single-hop neighbors	Hello	✓	✓	✓	High	✓
QGeo [71]	Local	Single-hop neighbors	Hello	✓	✓	×	Low	×
QSRP [72]	Global	Single-hop neighbors	Discovery and ACK packets	✓	✓	×	Low	×
QLGR [73]	Local	Single-hop neighbors	Hello	✓	✓	×	High	✓
FEQ-routing-SA [74]	Local	Single-hop neighbors	×	×	✓	✓	Low	×
Q-FANET [75]	Local	Single-hop neighbors	Hello	×	✓	×	Low	✓
PPMAC+RLSRP [76]	Local	Single-hop neighbors	Hello	×	×	×	Low	×
PARRoT [77]	Local	Single-hop neighbors	Chirp	×	✓	×	Low	✓
QFL-MOR [78]	Local and global	Single-hop neighbors	Hello	×	×	✓	Low	×
3DQ [79]	Local	Single-hop neighbors	Unknown	×	×	×	Low	×
ICRA [80]	Local	Single-hop neighbors	Hello	×	×	✓	High	×
TARRAQ [81]	Local	Single-hop neighbors	Hello	✓	✓	✓	Low	✓

Table 24. Comparison of RL-based routing protocols in terms of simulation environment and tools.

Scheme	Simulation Tools	Mobility Model	Localization Service	Simulation Environment
DQN-VR [67]	MATLAB and Python	Unknown	Unknown	3D
QTAR [68]	MATLAB	3D Gauss–Markov	GPS	3D
TQNGPSR [69]	Python and SimPy	Aircraft model	Unknown	2D
QMR [70]	WSNet	Random Waypoint	GPS	2D
QGeo [71]	NS3	Gauss–Markov	GPS	2D
QSRP [72]	OPNET	Random Waypoint	Unknown	3D
QLGR [73]	NS3	Gauss–Markov	Unknown	2D
FEQ-routing-SA [74]	Unknown	Unknown	×	Unknown
Q-FANET [75]	WSNet	Random Waypoint	GPS	2D
PPMAC+RLSRP [76]	MATLAB and NS2	Random Waypoint	GPS	3D
PARRoT [77]	OMNeT++	Random Waypoint, distributed dispersion detection, and dynamic cluster hovering	Unknown	3D
QFL-MOR [78]	Unknown	Unknown	Unknown	2D
3DQ [79]	Unknown	Gauss–Markov	GNSS	2D
ICRA [80]	OPNET	Gauss–Markov	GPS	2D
TARRAQ [81]	Monte Carlo	3D Random Waypoint	GPS	3D

Table 25. Comparison of RL-based routing methods in terms of routing parameters.

Scheme	Routing Parameters						
	Energy	Delay	Network Lifetime	PDR	Throughput	Connectivity	Routing Overhead
DQN-VR [67]	✓	×	✓	×	×	✓	×
QTAR [68]	✓	✓	✓	✓	×	×	✓
TQNGPSR [69]	×	✓	×	✓	✓	×	✓
QMR [70]	✓	✓	×	✓	×	×	×
QGeo [71]	×	✓	×	✓	×	×	✓
QSRP [72]	×	✓	×	✓	×	×	×
QLGR [73]	✓	✓	×	✓	✓	×	✓
FEQ-routing-SA [74]	✓	×	×	✓	×	×	✓
Q-FANET [75]	×	✓	×	✓	×	×	×
PPMAC+RLSRP [76]	×	✓	×	✓	×	×	×
PARRoT [77]	×	✓	×	✓	×	×	×
QFL-MOR [78]	✓	✓	×	×	×	×	×
3DQ [79]	×	✓	×	✓	✓	×	×
ICRA [80]	✓	✓	✓	✓	×	×	×
TARRAQ [81]	✓	✓	×	✓	×	×	✓

Table 26 compares RL-based routing methods in terms of different learning components in learning algorithms. According to this table, we can find that Q-learning is the most common reinforcement learning algorithm used to design the routing process in FANETs because this algorithm is simple and has an acceptable computational cost. However, if the state and action spaces are very large, the learning policy will significantly be complex. In this case, the curse of a dimensionality problem occurs. This problem causes a slow convergence speed in the Q-learning algorithm. Thus, this algorithm cannot reach an optimal response at an acceptable time. Many researchers have attempted to solve this problem when designing RL-based routing protocols in FANETs. For example, DQN-VR uses a clustering technique to reduce the state and action spaces in the routing process. It also uses a deep learning technique in the routing process that has a high convergence speed. TQNGPSR also uses a deep reinforcement learning algorithm, which accelerates the convergence speed and increases scalability. QMR and QTAR have also attempted to manage the state space through a suitable solution. In QTAR, only neighboring nodes that are closer to the destination node compared to the current node are inserted into the state space, and other nodes will be filtered. On the other hand, QMR defines a packet velocity constraint. Thus, only neighboring nodes that can meet this speed constraint are inserted into the state space and other neighboring nodes will be filtered. When designing RL-based routing methods, another important point is to determine how to adjust the learning parameters—namely, the learning rate and the discount factor. If these learning parameters have constant values, the selected action may be inaccurate. Therefore, in QTAR, QMR, QGeo, FEQ-routing-SA, PARRoT, and TARRAQ, researchers have adjusted these learning parameters dynamically and based on the network conditions. This has improved the adaptability of the routing methods to the dynamic FANET environment.

Moreover, Table 27 compares RL-based routing methods in terms of learning algorithms. As shown in this table, DQN-VR and TQNGPSR have used a deep reinforcement learning technique in their routing protocols. This learning technique performs complex computational operations in the FANET and has good performance for complex and large-scale networks because its learning speed is very high. Research on deep reinforcement learning algorithms for designing the routing protocols is still in the early steps, and researchers must study further research to solve the challenges related to this learning technique—namely, high computational complexity and its implementation in low-energy nodes. Most research in FANET such as QTAR, QMR, QGeo, QSRP, QLGR, FEQ-routing-SA, Q-FANET, PPMAC+RLSRP, PARRoT, QFL-MOR, 3DQ, ICRA, and TARRAQ use traditional reinforcement learning algorithms to design their routing process. They are easier than DRL-based routing methods and have less computational complexity. Therefore, they are suitable for discovering the best routing path in small FANETs because the size of the state and action spaces are small, and the learning algorithm is converged to the optimal response with acceptable learning speed. However, if the flying ad hoc network is large, these routing methods deal with the curse of the dimensionality problem and cannot present a good performance for finding the best route in the network because their convergence speed decreases sharply. Among these methods, QLGR is the only multi-agent RL-based routing method in the flying ad hoc networks. In this approach, each UAV plays the agent role and tries to learn the best path between the source and destination through interactions with the network. An important challenge in this routing scheme is how to coordinate and cooperate between UAVs to find an optimal response because they are extremely dynamic. QLGR has a faster convergence speed than single-agent routing methods because the routing calculations are done in parallel in different UAVs. This has improved scalability in QLGR. However, its computational complexity is greater than single-agent routing methods.

Table 26. Learning parameters in RL-based routing methods.

Scheme	RL Algorithm	Agent	State Set	Action Set	Reward Function	Learning Rate	Discount Factor
DQN-VR [67]	DQN	Central controller	A 2D array, including mobility and residual energy	Selecting the best neighboring node	Related to the successful transmission rate and network congestion level	Fixed	Fixed
QTAR [68]	Q-learning	Each packet	Neighboring nodes towards destination	Selecting the best next-hop node	Related to delay, energy, and velocity	Based on two-hop delay	Related to the distance and velocity changes between a UAV and its neighbors
TQNGPSR [69]	DQN	Each packet	UAVs	Selecting a candidate neighbor as the next-hop node	Based on the queuing length	Fixed	Fixed
QMR [70]	Q-learning	Each packet	UAVs	Decision of the packet (agent) to be forwarded from the current node to a neighboring node	Based on delay and energy	Based on single-hop delay	Related to the movement of neighbors in two consecutive intervals
QGeo [71]	Q-learning	Each packet	UAVs	Transition from transmitter to neighbor node	Related to packet travel speed	Fixed	Related to distance and the bobility pattern of UAVs
QSRP [72]	Q-learning	Central controller	UAVs	Selecting a routing path	Related to link stability and hop count	Fixed	Fixed
QLGR [73]	Q-learning	UAVs	State of UAVs	Sending a packet to a neighboring node	Related to load capacity and link quality	Fixed	Fixed
FEQ-routing-SA [74]	Q-learning	Each UAV	State of UAVs	Sending a packet to a neighboring node	Related to transmission energy	Based on delivery time	Fixed

Table 26. *Cont.*

Scheme	RL Algorithm	Agent	State Set	Action Set	Reward Function	Learning Rate	Discount Factor
Q-FANET [75]	Q-learning+	Each packet	UAVs	Sending the packet to a neighboring node	100, if reaching the destination; −100, if trapping in a local optimum; 50, otherwise	Fixed	Fixed
PPMAC+RLSRP [76]	Q-learning	Each UAV	The state of UAVs	Sending the packet to a neighboring node	Related to transmission delay	Fixed	Fixed
PARRoT [77]	Q-learning	Each UAV	The state of UAVs	Selecting a neighboring node as the next-hop node	Related to reverse path score	Fixed	Based on the link expiry time (LET) and the change degree of neighbors
QFL-MOR [78]	Q-learning	Unknown	UAVs	Selecting a neighboring node as the next-hop node	Unknown	Fixed	Fixed
3DQ [79]	Q-learning	Each UAV	DTGS corresponding to neighboring nodes	Neighboring nodes and current node	The first function is related to DTGS difference; the second function is calculated based on packet delay	Fixed	Fixed
ICRA [80]	Q-learning	Ground station	Four utility factours	Weights of four utility factors	Related to cluster structure stability and the energy change rate	Fixed	Fixed
TARRAQ [81]	Q-learning	Each packet	UAVs	Selecting the next hop from neighboring nodes	Related to link quality, residual energy, and distance	Related to resigual link duration	Related to resigual link duration

Table 27. Comparison of RL-based routing methods based on learning algorithm.

Scheme	Reinforcement Learning				Deep Reinforcement Learning			
	Single-Agent	**Multi-Agent**	**Model-Based**	**Free-Model**	**Single-Agent**	**Multi-Agent**	**Model-Based**	**Free-Model**
DQN-VR [67]	×	×	×	×	✓	×	×	✓
QTAR [68]	✓	×	×	✓	×	×	×	×
TQNGPSR [69]	×	×	×	×	✓	×	×	✓
QMR [70]	✓	×	×	✓	×	×	×	×
QGeo [71]	✓	×	×	✓	×	×	×	×
QSRP [72]	✓	×	×	✓	×	×	×	×
QLGR [73]	×	✓	×	✓	×	×	×	×
FEQ-routing-SA [74]	✓	×	×	✓	×	×	×	×
Q-FANET [75]	✓	×	×	✓	×	×	×	×
PPMAC+RLSRP [76]	✓	×	×	✓	×	×	×	×
PARRoT [77]	✓	×	×	✓	×	×	×	×
QFL-MOR [78]	✓	×	×	✓	×	×	×	×
3DQ [79]	✓	×	×	✓	×	×	×	×
ICRA [80]	✓	×	×	✓	×	×	×	×
TARRAQ [81]	✓	×	×	✓	×	×	×	×

Table 28 compares different RL-based routing protocols in terms of the routing path. According to this table, almost all RL-based routing methods are single-path. When finding the best route through the RL algorithm, almost all the routing paths between the source and destination are discovered. Then, they are prioritized based on the Q-value. However, in the route selection process, only one route (the path with maximum Q-value) is selected for the data transmission process. The management of the routing table (or Q-table) in a single-path routing method is easier than a multi-path routing scheme. However, it is not fault-tolerant, meaning that if the routing path is disconnected, the data transmission process will be delayed because a new path should be discovered. However, if the routing method selects alternative paths in the route selection process, it reduces packet loss and improves network performance.

Table 28. Comparison of RL-based routing methods by data path.

Scheme	Single-Path	Multi-Path
DQN-VR [67]	✓	×
QTAR [68]	✓	×
TQNGPSR [69]	✓	×
QMR [70]	✓	×
QGeo [71]	✓	×
QSRP [72]	✓	×
QLGR [73]	✓	×
FEQ-routing-SA [74]	✓	×
Q-FANET [75]	✓	×
PPMAC+RLSRP [76]	✓	×
PARRoT [77]	×	✓
QFL-MOR [78]	✓	×
3DQ [79]	✓	×
ICRA [80]	✓	×
TARRAQ [81]	✓	×

In Table 29, different routing methods are categorized based on network topology. According to this table, we can deduce that researchers perform a small number of studies in the field of hierarchical RL-based routing methods, and there are only two hierarchical RL-based routing methods, DQN-VR and ICRA, for FANETs. However, this topology is very suitable for large-scale networks because determining the different roles for nodes efficiently reduces the consumption of network resources in the route calculation process and lowers routing overhead. Researchers should consider this issue in the future to improve network performance. For example, in a clustered network, the RL algorithm is implemented in CHs to find the best path between different CHs, and each cluster is managed by the cluster head node. This reduces the state space and improves the convergence speed. Moreover, in a tree-based network, each parent node is responsible for executing learning operations in its sub-tree to reduce the dimensions of the state space in the learning process. In addition, in a multi-level network, one or more nodes are selected for finding the best route at each network level and managing the relationships between different network levels to improve the routing process. However, the management of different roles in the network and the selection of parent nodes and CHs, especially in highly dynamic networks, are important challenges that should be considered in these methods.

Table 29. Comparison of RL-based routing methods based on network topology.

Scheme	Flat	Hierarchical
DQN-VR [67]	×	✓
QTAR [68]	✓	×
TQNGPSR [69]	✓	×
QMR [70]	✓	×
QGeo [71]	✓	×
QSRP [72]	✓	×
QLGR [73]	✓	×
FEQ-routing-SA [74]	✓	×
Q-FANET [75]	✓	×
PPMAC+RLSRP [76]	✓	×
PARRoT [77]	✓	×
QFL-MOR [78]	✓	×
3DQ [79]	✓	×
ICRA [80]	×	✓
TARRAQ [81]	✓	×

Table 30 compares different routing methods in terms of data delivery techniques. Most RL-based routing methods such as TQNGPSR, QLGR, PPMAC+RLSRP, PARRoT, 3DQ, and ICRA use the greedy technique. However, this data delivery method deals with a major challenge (i.e., routing holes). In this case, the routing algorithm is trapped in the local optimum and cannot find any nodes to reach the destination. Among these routing methods, 3DQ has presented an interesting idea. It combines the greedy technique and the store-carry-forward technique to solve the routing hole challenge in the greedy mode. However, this solution can increase delay in the data transmission process. Usually, the store-carry-forward technique is not alone used to create a route between the source and destination because it causes a high delay in the data transmission process. However, when this scheme is integrated with the greedy method, it can solve the disadvantages of both methods. Furthermore, DQN-VR, QTAR, QMR, QGeo, QSRP, FEQ-routing-SA, Q-FANET, QFL-MOR, and TARRAQ discover the best route between the source and destination based on their desired criteria discussed in Section 6. However, the route discovery technique increases communication overhead, bandwidth consumption, and delay in the routing methods.

In Table 31, the routing methods are compared in terms of the routing process. According to this table, we can find that most RL-based routing methods are distributed. QSRP is the only centralized RL-based routing scheme. In this scheme, the central controller performs the routing process. QSRP assumes that this central server has global knowledge of the entire network and uses this knowledge in the routing process. However, in highly dynamic networks such as FANETs, it is very difficult or even impossible to gain global knowledge of the network by the central agent. For this reason, these methods are not successful in FANETs. Furthermore, they are not scalable. The most important advantage of this method is that the central controller obtains the best route at the lowest computational cost, and manages the routing process. As a result, UAVs do not consume energy for calculating routing paths. However, this technique is not fault-tolerant. On the other hand, obtaining global knowledge requires a lot of routing overhead. DQN-VR and ICRA use both centralized and distributed routing techniques. DQN-VR defines two types of data: global data and local data. Global data such as residual energy are less dynamic. Therefore, their update process is performed at longer time intervals. Moreover, local data, such as the speed and spatial information of UAVs, are more dynamic and expire quickly. Therefore, they must be updated at shorter time intervals. The central controller is responsible for collecting global data and performs the RL-based routing process based on both local and global data. This has improved the performance of this method compared to QSRP. In ICRA, a central controller computes clustering parameters. However, the clustering process is performed by UAVs in the network in a distributed manner.

Table 30. Comparison of RL-based routing methods based on data delivery technique.

Scheme	Greedy	Store-Carry-Forward	Route Discovery
DQN-VR [67]	×	×	✓
QTAR [68]	×	×	✓
TQNGPSR [69]	✓	×	×
QMR [70]	×	×	✓
QGeo [71]	×	×	✓
QSRP [72]	×	×	✓
QLGR [73]	✓	×	×
FEQ-routing-SA [74]	×	×	✓
Q-FANET [75]	×	×	✓
PPMAC+RLSRP [76]	✓	×	×
PARRoT [77]	✓	×	×
QFL-MOR [78]	×	×	✓
3DQ [79]	✓	✓	×
ICRA [80]	✓	×	×
TARRAQ [81]	×	×	✓

Table 31. Comparison of RL-based routing methods based on the routing process.

Scheme	Centralized	Distributed
DQN-VR [67]	✓	✓
QTAR [68]	×	✓
TQNGPSR [69]	×	✓
QMR [70]	×	✓
QGeo [71]	×	✓
QSRP [72]	✓	×
QLGR [73]	×	✓
FEQ-routing-SA [74]	×	✓
Q-FANET [75]	×	✓
PPMAC+RLSRP [76]	×	✓
PARRoT [77]	×	✓
QFL-MOR [78]	×	✓
3DQ [79]	×	✓
ICRA [80]	✓	✓
TARRAQ [81]	×	✓

Finally, Table 32 compares different routing methods in terms of the data dissemination process. Broadcast is the most common process used by almost all routing methods. For example, most RL-based routing methods broadcast hello messages to obtain local information. The most important disadvantage of this technique is high energy and bandwidth consumption. It also imposes a lot of communication overhead on the network. In QTAR, QMR, QGeo, QLGR, QSRP, and TARRAQ, researchers have introduced techniques for adjusting the hello broadcast interval to reduce the routing overhead. However, this issue still requires a lot of research.

Table 32. Comparison of RL-based routing methods based on the data dissemination process.

Scheme	Unicast	Multicast	Broadcast	Geocast
DQN-VR [67]	✓	×	×	×
QTAR [68]	×	×	✓	×
TQNGPSR [69]	✓	×	×	×
QMR [70]	×	×	✓	×
QGeo [71]	×	×	✓	×
QSRP [72]	×	×	✓	×
QLGR [73]	×	×	✓	×
FEQ-routing-SA [74]	✓	×	×	×
Q-FANET [75]	×	×	✓	×
PPMAC+RLSRP [76]	×	×	✓	×
PARRoT [77]	×	×	✓	×
QFL-MOR [78]	×	×	✓	×
3DQ [79]	✓	×	×	×
ICRA [80]	✓	×	✓	×
TARRAQ [81]	×	×	✓	×

8. Challenges and Open Issues

Despite progress in designing the RL-based routing methods for flying ad hoc networks, this subject still deals with various challenges and open issues, which should be addressed. In this section, the most important challenges are presented in this field.

- **Mobility models:** Most RL-based routing protocols use RWP and GM mobility models to simulate the movement of drones in the network. However, these models cannot simulate the actual movement of drones. Therefore, the simulation results may not guarantee the performance of the routing protocols in real conditions. As a result, in future research directions, researchers must consider realistic mobility models such as [82–86] to simulate the movement of UAVs in the network because they are close to real mobility models and can evaluate the performance of the routing protocols under realistic scenarios.
- **Simulation environment:** In many RL-based routing protocols, researchers have implemented drones in a two-dimensional environment. However, this implementation is incompatible with the three-dimensional environment of FANETs and affects the performance of these methods. Therefore, researchers must close the simulation environment to the real conditions to evaluate the performance of these protocols more accurately. As a result, deploying the routing methods in a 3D environment and considering all FANET needs and restrictions are subjects that must be studied in future research directions.
- **Simulation tool:** Most RL-based routing protocols are simulated using simulators such as WSNet, NS3, and MATLAB to evaluate their performance in a virtual environment. They require low cost. However, these tools cannot accurately simulate a real environment, and the simulation results do not usually match the real environment.

Therefore, these methods must be implemented in real environments to analyze their performance in real conditions. However, this is extremely expensive.

- **Localization:** Most RL-based routing methods must obtain location information using a positioning system. Therefore, their performance is dependent on a positioning system. If the positioning system is not accurate and cannot properly measure the position of drones, the network performance will be weakened. GPS is the most common positioning system used in RL-based routing protocols. The accuracy of this positioning system depends on environmental conditions. For example, if UAVs are in an open environment with a good climate, GPS can predict the position of drones with proper accuracy. In contrast, in indoor areas such as tunnels or in inappropriate weather conditions, GPS signals are not properly received. This has a negative effect on network performance. Focusing on free infrastructure localization methods and calculating the position of nodes without the need for GPS is a good solution, which can be considered by researchers in the future.
- **Efficiency:** Reinforcement learning algorithms are useful for designing RL-based routing methods in FANETs when they can solve a serious problem in this area. These algorithms can improve the routing policy and build an optimal path between nodes in the network, but impose a lot of computational costs on UAVs in the network. When designing a RL-based routing method, researchers must determine whether these algorithms help routing protocols to improve network performance and reduce computational cost and energy consumption. In some cases, solving a challenge in FANETs may not need to use reinforcement learning techniques, and existing methods can successfully address this challenge. As a result, proper use of reinforcement learning techniques is a very important issue, which must be considered by researchers.
- **Routing overhead:** In most RL-based routing methods in FANETs, the periodic exchange of control messages is essential to obtain the location and other information of neighboring nodes. However, this increases bandwidth consumption and routing overhead and greatly increases network congestion. Therefore, an essential need is to adaptively adjust the broadcast intervals of these messages based on network dynamics when designing RL-based routing methods.
- **Convergence speed:** It is an important issue in reinforcement learning algorithms. When the size of the state and action spaces are large, the convergence speed of the RL algorithm is greatly reduced. Thus, obtaining an optimal response requires a long time. In this case, the Q-table size also increases sharply. This needs a large storage capacity to store this table. On the other hand, the update process of this table is also associated with high delay, computational costs, and communication overhead. Therefore, reducing the size of the state space by filtering some states based on specific criteria and utilizing clustering techniques in the routing process can be studied and evaluated in the future. Furthermore, the use of deep reinforcement learning techniques is a useful response to deal with this challenge.
- **Trade-off between exploration and exploitation:** The dynamic adjustment of learning parameters, including learning rate and the discount factor, is very important in a RL-based routing algorithm to create a balance between exploration and exploitation. Researchers should consider this issue in the future.

9. Conclusions

In this paper, we have focused on reinforcement learning-based routing methods for flying ad hoc networks. Initially, reinforcement learning and the Markov decision process were introduced, and various reinforcement learning methods were summarized and compared with each other. Then, RL-based routing methods were categorized in terms of the learning algorithm, routing algorithm, and data dissemination process. Next, the state-of-the-art RL-based routing methods were studied and reviewed. Finally, the opportunities and challenges in this area were expressed to provide a detailed and accurate view for scholars to know future research directions in the field of RL-based routing algorithms in

FANET. In the future, researchers should focus on clustering-based RL routing algorithms to control routing overhead and improve the convergence speed of the learning algorithm by reducing the action and state spaces. In the future research direction, we should evaluate and compare the performance of reinforcement learning-based routing methods by testing different mobility models and simulating different scenarios. Furthermore, it is very important to focus on deep reinforcement learning algorithms when designing routing methods to improve the convergence speed and solve the curse of the dimensionality problem. The application of reinforcement learning and deep reinforcement learning in other FANET fields should also be studied.

Author Contributions: Conceptualization, M.S.Y. and E.Y.; methodology, M.S.Y., E.Y. and M.H.; validation, A.M.R., J.L. and S.A.; investigation, A.M.R., F.K. and J.L.; resources, A.M.R., F.K. and S.A.; writing—original draft preparation, M.S.Y., E.Y. and M.H.; supervision, M.H.; project administration, A.M.R. and F.K. All authors have read and agreed to the published version of the manuscript.

Funding: The result was created through solving the student project "Security analysis and developing lightweight ciphers and protocols" using objective-oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Authors thank Michal Merta, and Zdeněk Truhlář for their help with the research connected with the topic of the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sharma, V.; Kumar, R. Cooperative frameworks and network models for flying ad hoc networks: A survey. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e3931. [CrossRef]
2. Yousefpoor, M.S.; Barati, H. Dynamic key management algorithms in wireless sensor networks: A survey. *Comput. Commun.* **2019**, *134*, 52–69. [CrossRef]
3. Siddiqi, M.A.; Iwendi, C.; Jaroslava, K.; Anumbe, N. Analysis on security-related concerns of unmanned aerial vehicle: Attacks, limitations, and recommendations. *Math. Biosci. Eng.* **2022**, *19*, 2641–2670. [CrossRef] [PubMed]
4. Yousefpoor, M.S.; Barati, H. DSKMS: A dynamic smart key management system based on fuzzy logic in wireless sensor networks. *Wirel. Netw.* **2020**, *26*, 2515–2535. [CrossRef]
5. Lakew, D.S.; Sa'ad, U.; Dao, N.N.; Na, W.; Cho, S. Routing in flying ad hoc networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1071–1120. [CrossRef]
6. Yousefpoor, M.S.; Yousefpoor, E.; Barati, H.; Barati, A.; Movaghar, A.; Hosseinzadeh, M. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *J. Netw. Comput. Appl.* **2021**, *190*, 103118. [CrossRef]
7. Oubbati, O.S.; Atiquzzaman, M.; Lorenz, P.; Tareque, M.H.; Hossain, M.S. Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives. *IEEE Access* **2019**, *7*, 81057–81105. [CrossRef]
8. Rahmani, A.M.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Naqvi, R.A.; Siddique, K.; Hosseinzadeh, M. An area coverage scheme based on fuzzy logic and shuffled frog-leaping algorithm (sfla) in heterogeneous wireless sensor networks. *Mathematics* **2021**, *9*, 2251. [CrossRef]
9. Xu, M.; Xie, J.; Xia, Y.; Liu, W.; Luo, R.; Hu, S.; Huang, D. Improving traditional routing protocols for flying ad hoc networks: A survey. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 162–166. [CrossRef]
10. Lee, S.W.; Ali, S.; Yousefpoor, M.S.; Yousefpoor, E.; Lalbakhsh, P.; Javaheri, D.; Rahmani, A.M.; Hosseinzadeh, M. An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). *IEEE Access* **2021**, *9*, 129977–130005. [CrossRef]
11. Mukherjee, A.; Keshary, V.; Pandya, K.; Dey, N.; Satapathy, S.C. Flying ad hoc networks: A comprehensive survey. *Inf. Decis. Sci.* **2018**, *701*, 569–580._59. [CrossRef]
12. Rahmani, A.M.; Ali, S.; Yousefpoor, E.; Yousefpoor, M.S.; Javaheri, D.; Lalbakhsh, P.; Ahmed, O.H.; Hosseinzadeh, M.; Lee, S.W. OLSR+: A new routing method based on fuzzy logic in flying ad hoc networks (FANETs). *Veh. Commun.* **2022**, *36*, 100489. [CrossRef]

13. Oubbati, O.S.; Lakas, A.; Zhou, F.; Güneş, M.; Yagoubi, M.B. A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs). *Veh. Commun.* **2017**, *10*, 29–56. [CrossRef]
14. Mohammed, M.; Khan, M.B.; Bashier, E.B.M. *Machine Learning: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
15. Rahmani, A.M.; Yousefpoor, E.; Yousefpoor, M.S.; Mehmood, Z.; Haider, A.; Hosseinzadeh, M.; Ali Naqvi, R. Machine learning (ML) in medicine: Review, applications, and challenges. *Mathematics* **2021**, *9*, 2970. [CrossRef]
16. Uprety, A.; Rawat, D.B. Reinforcement learning for iot security: A comprehensive survey. *IEEE Internet Things J.* **2020**, *8*, 8693–8706. [CrossRef]
17. Padakandla, S. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–25. [CrossRef]
18. Wang, Q.; Zhan, Z. Reinforcement learning model, algorithms and its application. In Proceedings of the 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, China, 19–22 August 2011; pp. 1143–1146. [CrossRef]
19. Al-Rawi, H.A.; Ng, M.A.; Yau, K.L.A. Application of reinforcement learning to routing in distributed wireless networks: a review. *Artif. Intell. Rev.* **2015**, *43*, 381–416. [CrossRef]
20. Rovira-Sugranes, A.; Razi, A.; Afghah, F.; Chakareski, J. A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. *Ad Hoc Netw.* **2022**, *130*, 102790. [CrossRef]
21. Rezwan, S.; Choi, W. A survey on applications of reinforcement learning in flying ad hoc networks. *Electronics* **2021**, *10*, 449. [CrossRef]
22. Alam, M.M.; Moh, S. Survey on Q-Learning-Based Position-Aware Routing Protocols in Flying Ad Hoc Networks. *Electronics* **2022**, *11*, 1099. [CrossRef]
23. Bithas, P.S.; Michailidis, E.T.; Nomikos, N.; Vouyioukas, D.; Kanatas, A.G. A survey on machine-learning techniques for UAV-based communications. *Sensors* **2019**, *19*, 5170. [CrossRef]
24. Srivastava, A.; Prakash, J. Future FANET with application and enabling techniques: Anatomization and sustainability issues. *Comput. Sci. Rev.* **2021**, *39*, 100359. [CrossRef]
25. Suthaputchakun, C.; Sun, Z. Routing protocol in intervehicle communication systems: A survey. *IEEE Commun. Mag.* **2011**, *49*, 150–156. [CrossRef]
26. Maxa, J.A.; Mahmoud, M.S.B.; Larrieu, N. Survey on UAANET routing protocols and network security challenges. *Adhoc Sens. Wirel. Netw.* **2017**, *37*, 231–320.
27. Jiang, J.; Han, G. Routing protocols for unmanned aerial vehicles. *IEEE Commun. Mag.* **2018**, *56*, 58–63. [CrossRef]
28. Arafat, M.Y.; Moh, S. Routing protocols for unmanned aerial vehicle networks: A survey. *IEEE Access* **2019**, *7*, 99694–99720. [CrossRef]
29. Coronato, A.; Naeem, M.; De Pietro, G.; Paragliola, G. Reinforcement learning for intelligent healthcare applications: A survey. *Artif. Intell. Med.* **2020**, *109*, 101964. [CrossRef]
30. Kubat, M. *An Introduction to Machine Learning*; Springer International Publishing: Cham, Switzerland, 2017; Volume 2. [CrossRef]
31. Rahmani, A.M.; Ali, S.; Malik, M.H.; Yousefpoor, E.; Yousefpoor, M.S.; Mousavi, A.; Hosseinzadeh, M. An energy-aware and Q-learning-based area coverage for oil pipeline monitoring systems using sensors and Internet of Things. *Sci. Rep.* **2022**, *12*, 1–17. [CrossRef]
32. Javaheri, D.; Hosseinzadeh, M.; Rahmani, A.M. Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access* **2018**, *6*, 78321–78332. [CrossRef]
33. Nazib, R.A.; Moh, S. Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey. *IEEE Access* **2020**, *8*, 77535–77560. [CrossRef]
34. Yousefpoor, E.; Barati, H.; Barati, A. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer-Peer Netw. Appl.* **2021**, *14*, 1917–1942. [CrossRef]
35. Vijitha Ananthi, J.; Subha Hency Jose, P. A review on various routing protocol designing features for flying ad hoc networks. *Mob. Comput. Sustain. Inform.* **2022**, *68*, 315–325._23. [CrossRef]
36. Azevedo, M.I.B.; Coutinho, C.; Toda, E.M.; Carvalho, T.C.; Jailton, J. Wireless communications challenges to flying ad hoc networks (FANET). *Mob. Comput.* **2020** , *3*. [CrossRef]
37. Wang, J.; Jiang, C. *Flying Ad Hoc Networks: Cooperative Networking and Resource Allocation*; Springer: Berlin/Heidelberg, Germany, 2022. [CrossRef]
38. Noor, F.; Khan, M.A.; Al-Zahrani, A.; Ullah, I.; Al-Dhlan, K.A. A review on communications perspective of flying ad hoc networks: Key enabling wireless technologies, applications, challenges and open research topics. *Drones* **2020**, *4*, 65. [CrossRef]
39. Guillen-Perez, A.; Cano, M.D. Flying ad hoc networks: A new domain for network communications. *Sensors* **2018**, *18*, 3571. [CrossRef] [PubMed]
40. Agrawal, J.; Kapoor, M. A comparative study on geographic-based routing algorithms for flying ad hoc networks. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6253. [CrossRef]
41. Kim, D.Y.; Lee, J.W. Topology construction for flying ad hoc networks (FANETs). In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 18–20 October 2017; pp. 153–157. [CrossRef]

42. Rahman, M.F.F.; Fan, S.; Zhang, Y.; Chen, L. A comparative study on application of unmanned aerial vehicle systems in agriculture. *Agriculture* **2021**, *11*, 22. [CrossRef]
43. Shrestha, R.; Bajracharya, R.; Kim, S. 6G enabled unmanned aerial vehicle traffic management: A perspective. *IEEE Access* **2021**, *9*, 91119–91136. [CrossRef]
44. Liu, T.; Sun, Y.; Wang, C.; Zhang, Y.; Qiu, Z.; Gong, W.; Lei, S.; Tong, X.; Duan, X. Unmanned aerial vehicle and artificial intelligence revolutionizing efficient and precision sustainable forest management. *J. Clean. Prod.* **2021**, *311*, 127546. [CrossRef]
45. Idrissi, M.; Salami, M.; Annaz, F. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms. *J. Intell. Robot. Syst.* **2022**, *104*, 1–33. [CrossRef]
46. Syed, F.; Gupta, S.K.; Hamood Alsamhi, S.; Rashid, M.; Liu, X. A survey on recent optimal techniques for securing unmanned aerial vehicles applications. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4133. [CrossRef]
47. Sang, Q.; Wu, H.; Xing, L.; Xie, P. Review and comparison of emerging routing protocols in flying ad hoc networks. *Symmetry* **2020**, *12*, 971. [CrossRef]
48. Mittal, M.; Iwendi, C. A survey on energy-aware wireless sensor routing protocols. *EAI Endorsed Trans. Energy Web* **2019** , *6*. [CrossRef]
49. Agostinelli, F.; Hocquet, G.; Singh, S.; Baldi, P. From reinforcement learning to deep reinforcement learning: An overview. In *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 298–328. [CrossRef]
50. Althamary, I.; Huang, C.W.; Lin, P. A survey on multi-agent reinforcement learning methods for vehicular networks. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1154–1159. [CrossRef]
51. Canese, L.; Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-agent reinforcement learning: A review of challenges and applications. *Appl. Sci.* **2021**, *11*, 4948. [CrossRef]
52. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2008**, *38*, 156–172. [CrossRef]
53. Drummond, N.; Niv, Y. Model-based decision making and model-free learning. *Curr. Biol.* **2020**, *30*, R860–R865. [CrossRef] [PubMed]
54. Asadi, K. Strengths, Weaknesses, and Combinations of Model-Based and Model-Free Reinforcement Learning. Master's Thesis, Department of Computing Science, University of Alberta, Edmonton, AB, Canada, 2015.
55. Sirajuddin, M.; Rupa, C.; Iwendi, C.; Biamba, C. TBSMR: A trust-based secure multipath routing protocol for enhancing the qos of the mobile ad hoc network. *Secur. Commun. Netw.* **2021**, *2021* . [CrossRef]
56. Bernsen, J.; Manivannan, D. Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification. *Pervasive Mob. Comput.* **2009**, *5*, 1–18. [CrossRef]
57. Panichpapiboon, S.; Pattara-Atikom, W. A review of information dissemination protocols for vehicular ad hoc networks. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 784–798. [CrossRef]
58. Ren, Z.; Guo, W. Unicast routing in mobile ad hoc networks: Present and future directions. In Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, Chengdu, China, 29 August 2003; pp. 340–344. [CrossRef]
59. Biradar, R.C.; Manvi, S.S. Review of multicast routing mechanisms in mobile ad hoc networks. *J. Netw. Comput. Appl.* **2012**, *35*, 221–239. [CrossRef]
60. Guo, S.; Yang, O.W. Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. *Comput. Commun.* **2007**, *30*, 2129–2148. [CrossRef]
61. Khabbazian, M.; Bhargava, V.K. Efficient broadcasting in mobile ad hoc networks. *IEEE Trans. Mob. Comput.* **2008**, *8*, 231–245. [CrossRef]
62. Reina, D.G.; Toral, S.L.; Johnson, P.; Barrero, F. A survey on probabilistic broadcast schemes for wireless ad hoc networks. *Ad Hoc Netw.* **2015**, *25*, 263–292. [CrossRef]
63. Ruiz, P.; Bouvry, P. Survey on broadcast algorithms for mobile ad hoc networks. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–35. [CrossRef]
64. Ko, Y.B.; Vaidya, N.H. Flooding-based geocasting protocols for mobile ad hoc networks. *Mob. Netw. Appl.* **2002**, *7*, 471–480. [CrossRef]
65. Drouhin, F.; Bindel, S. Routing and Data Diffusion in Vehicular Ad Hoc Networks. In *Building Wireless Sensor Networks*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 67–96. [CrossRef]
66. Mohapatra, P.; Li, J.; Gui, C. Multicasting in ad hoc networks. In *Ad Hoc Networks*; Springer: Boston, MA, USA, 2005; pp. 91–122. [CrossRef]
67. Khan, M.F.; Yau, K.L.A.; Ling, M.H.; Imran, M.A.; Chong, Y.W. An Intelligent Cluster-Based Routing Scheme in 5G Flying Ad Hoc Networks. *Appl. Sci.* **2022**, *12*, 3665. [CrossRef]
68. Arafat, M.Y.; Moh, S. A Q-learning-based topology-aware routing protocol for flying ad hoc networks. *IEEE Internet Things J.* **2021**, *9*, 1985–2000. [CrossRef]
69. Chen, Y.N.; Lyu, N.Q.; Song, G.H.; Yang, B.W.; Jiang, X.H. A traffic-aware Q-network enhanced routing protocol based on GPSR for unmanned aerial vehicle ad hoc networks. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1308–1320. [CrossRef]

70. Liu, J.; Wang, Q.; He, C.; Jaffrès-Runser, K.; Xu, Y.; Li, Z.; Xu, Y. QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks. *Comput. Commun.* **2020**, *150*, 304–316. [CrossRef]
71. Jung, W.S.; Yim, J.; Ko, Y.B. QGeo: Q-learning-based geographic ad hoc routing protocol for unmanned robotic networks. *IEEE Commun. Lett.* **2017**, *21*, 2258–2261. [CrossRef]
72. Lim, J.W.; Ko, Y.B. Q-learning based stepwise routing protocol for multi-uav networks. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, 13–16 April 2021; pp. 307–309. [CrossRef]
73. Qiu, X.; Xie, Y.; Wang, Y.; Ye, L.; Yang, Y. QLGR: A Q-learning-based Geographic FANET Routing Algorithm Based on Multi-agent Reinforcement Learning. *KSII Trans. Internet Inf. Syst. (TIIS)* **2021**, *15*, 4244–4274. [CrossRef]
74. Rovira-Sugranes, A.; Afghah, F.; Qu, J.; Razi, A. Fully-echoed q-routing with simulated annealing inference for flying adhoc networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2223–2234. [CrossRef]
75. Da Costa, L.A.L.; Kunst, R.; de Freitas, E.P. Q-FANET: Improved Q-learning based routing protocol for FANETs. *Comput. Netw.* **2021**, *198*, 108379. [CrossRef]
76. Zheng, Z.; Sangaiah, A.K.; Wang, T. Adaptive communication protocols in flying ad hoc network. *IEEE Commun. Mag.* **2018**, *56*, 136–142. [CrossRef]
77. Sliwa, B.; Schüler, C.; Patchou, M.; Wietfeld, C. PARRoT: Predictive ad hoc routing fueled by reinforcement learning and trajectory knowledge. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–7. [CrossRef]
78. Yang, Q.; Jang, S.J.; Yoo, S.J. Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks. *Wirel. Pers. Commun.* **2020**, *113*, 115–138. [CrossRef]
79. Zhang, M.; Dong, C.; Feng, S.; Guan, X.; Chen, H.; Wu, Q. Adaptive 3D routing protocol for flying ad hoc networks based on prediction-driven Q-learning. *China Commun.* **2022**, *19*, 302–317. [CrossRef]
80. Guo, J.; Gao, H.; Liu, Z.; Huang, F.; Zhang, J.; Li, X.; Ma, J. ICRA: An Intelligent Clustering Routing Approach for UAV Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–14. [CrossRef]
81. Cui, Y.; Zhang, Q.; Feng, Z.; Wei, Z.; Shi, C.; Yang, H. Topology-Aware Resilient Routing Protocol for FANETs: An Adaptive Q-Learning Approach. *IEEE Internet Things J.* **2022**. [CrossRef]
82. Zhao, H.; Liu, H.; Leung, Y.W.; Chu, X. Self-adaptive collective motion of swarm robots. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1533–1545. [CrossRef]
83. Xu, W.; Xiang, L.; Zhang, T.; Pan, M.; Han, Z. Cooperative Control of Physical Collision and Transmission Power for UAV Swarm: A Dual-Fields Enabled Approach. *IEEE Internet Things J.* **2021**, *9*, 2390–2403. [CrossRef]
84. Dai, F.; Chen, M.; Wei, X.; Wang, H. Swarm intelligence-inspired autonomous flocking control in UAV networks. *IEEE Access* **2019**, *7*, 61786–61796. [CrossRef]
85. Zhao, H.; Wei, J.; Huang, S.; Zhou, L.; Tang, Q. Regular topology formation based on artificial forces for distributed mobile robotic networks. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2415–2429. [CrossRef]
86. Trotta, A.; Montecchiari, L.; Di Felice, M.; Bononi, L. A GPS-free flocking model for aerial mesh deployments in disaster-recovery scenarios. *IEEE Access* **2020**, *8*, 91558–91573. [CrossRef]