electronics

MDPI

Review

A Survey on Applications of Reinforcement Learning in Flying Ad-Hoc Networks

Sifat Rezwan and Wooyeol Choi *

Department of Computer Engineering, Chosun University, Gwangju 61452, Korea; sifat.rezwan@chosun.kr
* Correspondence: wyc@chosun.ac.kr

Abstract: Flying ad-hoc networks (FANET) are one of the most important branches of wireless ad-hoc networks, consisting of multiple unmanned air vehicles (UAVs) performing assigned tasks and communicating with each other. Nowadays FANETs are being used for commercial and civilian applications such as handling traffic congestion, remote data collection, remote sensing, network relaying, and delivering products. However, there are some major challenges, such as adaptive routing protocols, flight trajectory selection, energy limitations, charging, and autonomous deployment that need to be addressed in FANETs. Several researchers have been working for the last few years to resolve these problems. The main obstacles are the high mobility and unpredictable changes in the topology of FANETs. Hence, many researchers have introduced reinforcement learning (RL) algorithms in FANETs to overcome these shortcomings. In this study, we comprehensively surveyed and qualitatively compared the applications of RL in different scenarios of FANETs such as routing protocol, flight trajectory selection, relaying, and charging. We also discuss open research issues that can provide researchers with clear and direct insights for further research.

Keywords: flying Ad-hoc network; reinforcement learning; routing protocol; flight trajectory; unmanned air vehicles

Citation: Rezwan, S.; Choi, W. A Survey on Applications of Reinforcement Learning in Flying Ad-Hoc Networks. *Electronics* **2021**, *10*, 449. https://doi.org/10.3390/electronics10040449

Academic Editor: Nurul I. Sarkar
Received: 20 January 2021
Accepted: 8 February 2021
Published: 11 February 2021

1. Introduction

Flying ad-hoc networks (FANETs) are gaining popularity because of their versatility, easy deployment, high mobility, and low operational cost [1]. FANETs are usually formed by unmanned aerial vehicles (UAVs), which can fly autonomously or can be controlled remotely [2]. UAVs have been used by militaries around the globe since the beginning of surveillance and rescue purposes [3]. Nowadays, with the advancement of technology, UAVs have been extensively used in every domain for sensitive tasks such as traffic monitoring [4], disaster monitoring [5], relay for other ad-hoc networks [6], remote sensing [7], and wildfire monitoring [8]. Multiple UAVs can be used to perform different tasks individually; however, when it comes to FANET, the UAVs must communicate with each other and coordinate accordingly, as shown in Figure 1. FANET is an ad-hoc network of UAVs. Generally, in FANETs small UAVs are used because coordination and collaboration among small UAVs can outperform the large UAVs. Moreover, small UAVs have low acquisition, operational costs, increased scalability, and survivability [9].However, FANET has some major challenges to overcome such as

- Communication: UAVs can move at high speed, which poses difficulties in maintaining communication with other UAVs. In addition, the distance among the nodes is higher than that other ad-hoc networks [10].
- Power constraint: Generally, UAVs carry batteries as a power supply, which is limited to support their operations and flying time. Increasing the capacity of the battery may degrade the performance of the UAVs after a certain point owing to the energy and weight ratio. Therefore, effective battery and charging management is one of the major challenges of FANETs [11].

- Routing protocol: Routing in FANETs is also a challenge owing to the high mobility and power constraints of the UAVs. Many routing protocols have been designed for ad-hoc networks but FANET requires a highly dynamic routing protocol to cope with the dynamic changes in the FANET topology [12].
- Ensuring QoS: There are also some quality of service (QoS) related challenges that should be addressed such as ensuring low latency, determining the trajectory path to provide service, synchronization among UAVs, and protection against jamming attacks.

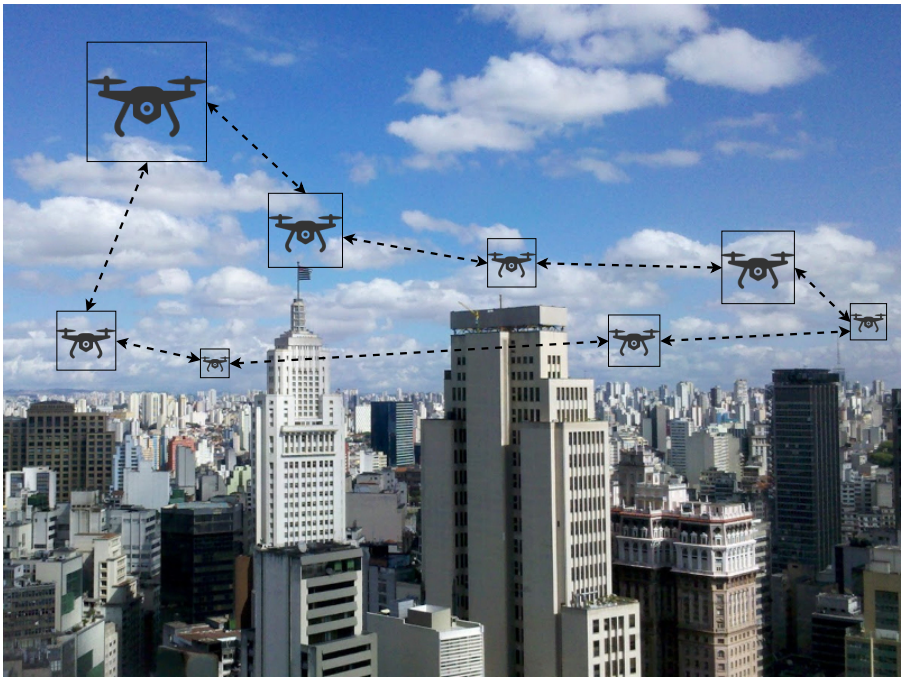

Figure 1. Flying ad-hoc network of UAVs.

Many researchers have been working for the last few years to overcome these challenges. They have been using different techniques related to artificial intelligence (AI) so that the network can autonomously and adaptively learn and overcome the challenge by itself. Reinforcement learning (RL) is one of the most important algorithms that has a significant contribution to the development of AI [13–15]. RL is popular for its trial-and-optimize scheme. RL consists of an agent and an environment in which the agent explores the environment by taking actions and reaches an optimal policy for the system [16]. However, to achieve the optimal policy, the agent must know the entire system, which makes the RL time-consuming and unsuitable for large networks. With the development of computational capability achieved by the GPU, this problem can be addressed by integrating deep neural networks (DNNs) into RL, namely deep reinforcement learning (DRL) [15,17].

Currently, there is no survey discussing the applications of RL in FANETs. This motivates us to deliver the survey with the fundamentals of RL and DRL and a comprehensive literature review on the applications of RL and DRL to address the challenges in FANETs. The major issues include routing protocol, selecting flight trajectory, charging UAVs, anti-jamming, and ensuring the QoS of FANETs.

2. Fundamentals of Deep Reinforcement Learning

In this section, we briefly discuss the internal structure, decision-making process, and convergence process of reinforcement learning (RL) and deep reinforcement learning (DRL).

2.1. Reinforcement Learning

Reinforcement learning is an effective and extensively used tool of AI which learns about the environment by taking different actions and achieves an optimal policy for

operation. The RL consists of two main components: an agent and an environment. The agent explores the environment and decides which action to take using the Markov decision process (MDP) [18].

MDP is a framework for modeling decision-making problems and helping the agent to control the process stochastically [18]. MDP is a useful tool for dynamic programming and RL techniques. Generally, MDP has four parameters represented by the tuple (S, A, p, r), where S is a finite state space, A is a finite action space, p is the transition probability from the present state s to the next state s' after taking action a, and r is the immediate reward given by the environment for action a [19]. As shown in Figure 2, at each time step t, the agent observes its present state s_t in the environment and takes action a_t. Then, the agent receives a reward r_t and the next state s_{t+1} from the environment. The main goal of the agent is to determine a policy π to accumulate the maximum possible reward from the environment. In long term, the agent also tries to maximize the expected discounted total reward defined by $\max[\sum_{t=0}^{T} \delta r_t(s_t, \pi(s_t))]$, where $\delta \in [0, 1]$ is the discount factor. Using the discounted reward, a Bellman equation named the Q-function (1) is formed to take the next action a_t using MDP when the state transition probabilities are known in advance. The Q-function can be expressed as

$$Q(s_t, a_t) = (1 - \alpha) \times Q(s_t, a_t) + \alpha[r + \delta(\max Q(s_{t+1}, a_t))], \tag{1}$$

where α is the learning rate.

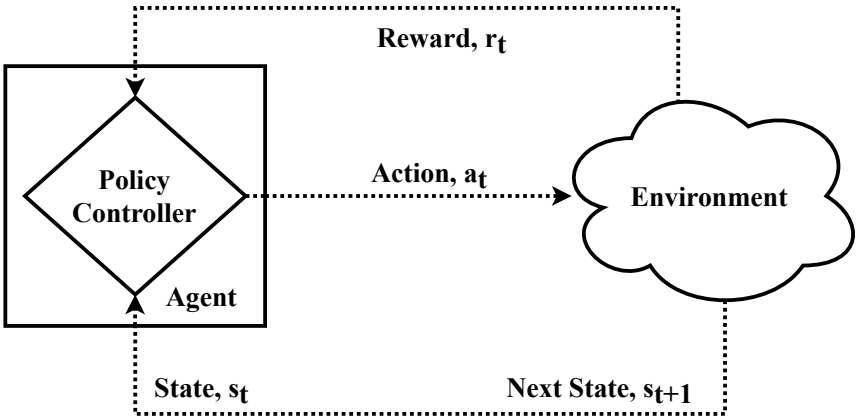

Figure 2. The agent-environment in Markov decision process.

RL with a Q-function is also known as Q-learning. Initially, the agent explores every state of the environment taking different actions and forms a Q-table using the Q-function for each state-action pair. Then, the agent starts exploiting the environment by taking actions with the maximum Q-value from the Q-table. This policy is known as the ϵ-greedy policy, where the agent starts exploring or exploiting the environment depending on the value of the probability ϵ. An illustration of Q-learning is presented in Algorithm 1.

Algorithm 1 The Q-learning Algorithm.

Require: $Q(S, A) = 0$.
Ensure: Initialize α, δ, ϵ. **for** $t = 1, 2, \ldots, T$ **do**
  Choose an action a_t for present state s_t based on the value of ϵ.
  Obtain an immediate reward r_t and next state s_{t+1}.
  Update $Q(S, A)$ via Markov decision process (1).
  $s_t \leftarrow s_{t+1}$
end
Optimal policy, $\pi(s) = \arg\max Q(S, A)$

2.2. Deep Reinforcement Learning

The Q-learning algorithm is efficient in terms of its comparatively small action and state space. However, the system becomes more complicated for large action and state space. In this situation, the Q-learning algorithm may not be able to achieve an optimal policy owing to the complex and large Q-table. To overcome this problem, researchers replaced the Q-table with a deep neural network (DNN) and named it deep Q-learning (DQL) [15]. DQL is a deep reinforcement learning (DRL) that works with Q-values similar to Q-learning, except for the Q-table part as shown in Figure 3.

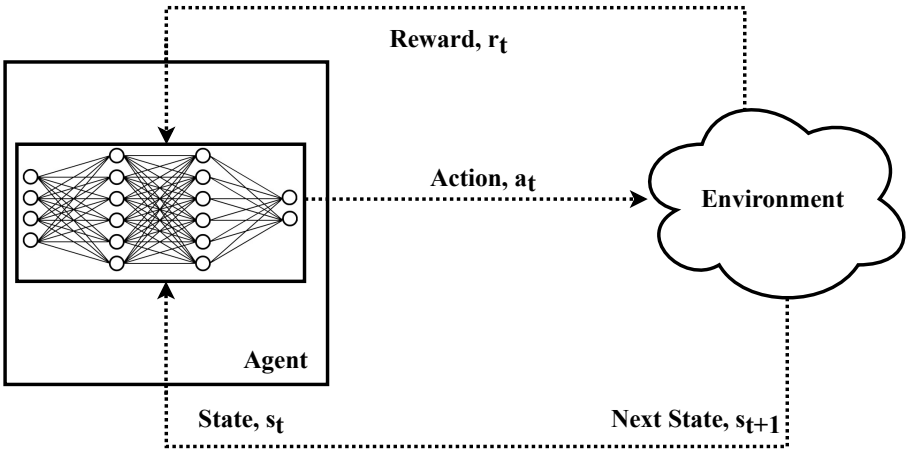

Figure 3. Simple deep Q-learning.

The main goal of the DNN is to skip manual calculations each time by learning from the data. A DNN is a computational nonlinear model like the structure of the human brain, which can learn and perform tasks such as decision-making, prediction, classification, and visualization [20]. It is composed of neurons arranged in multiple layers. It typically has one input layer, two hidden layers, and on output layer, interconnected as depicted in Figure 4 [21]. The input layer accepts the inputs with the input neurons and sends them to the hidden layers. The hidden layer then sends the data to the output layer. Every neuron has a weighted input, an activation function, and an output. The activation function determines the output depending on the input of the neuron [22]. It acts as a trigger that depends on the weighted input.

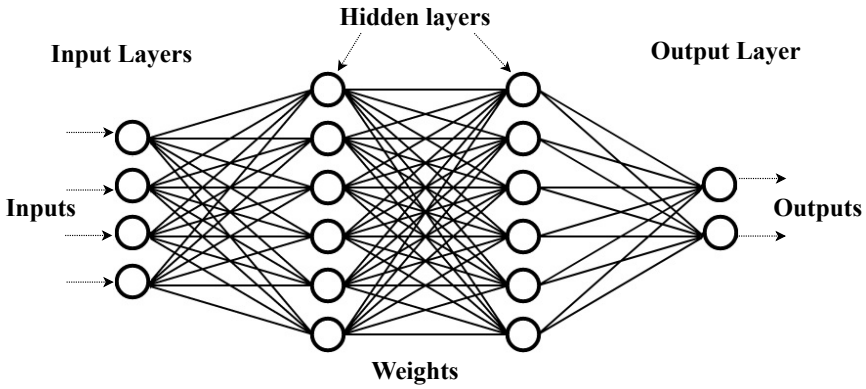

Figure 4. Deep neural network.

During the training phase, the weighted values of the inputs of the neurons are updated based on the outputs of the output layer using backpropagation by the agent. The agent takes the output of the policy DNN and compares it with a target DNN model and calculates error [23]. Then the agent updates the policy DNN using backpropagation. This process is generally referred to as optimization with gradient descent. After a certain time, the agent updates the target DNN using policy DNN. For a more stable

convergence of the optimal policy, experience replay memory (ERM) is introduced into the DQL framework [24,25]. The agent takes different actions and saves the present states, obtained rewards, next states, and actions taken in ERM [24,25]. Then, the agent takes a mini-batch of data from the ERM and trains the policy DNN. Figure 5 and Algorithm 2 illustrate the framework and flow of the DQN better [26]. Thus, the agent can make decisions efficiently and in a timely manner using the learned DNN.

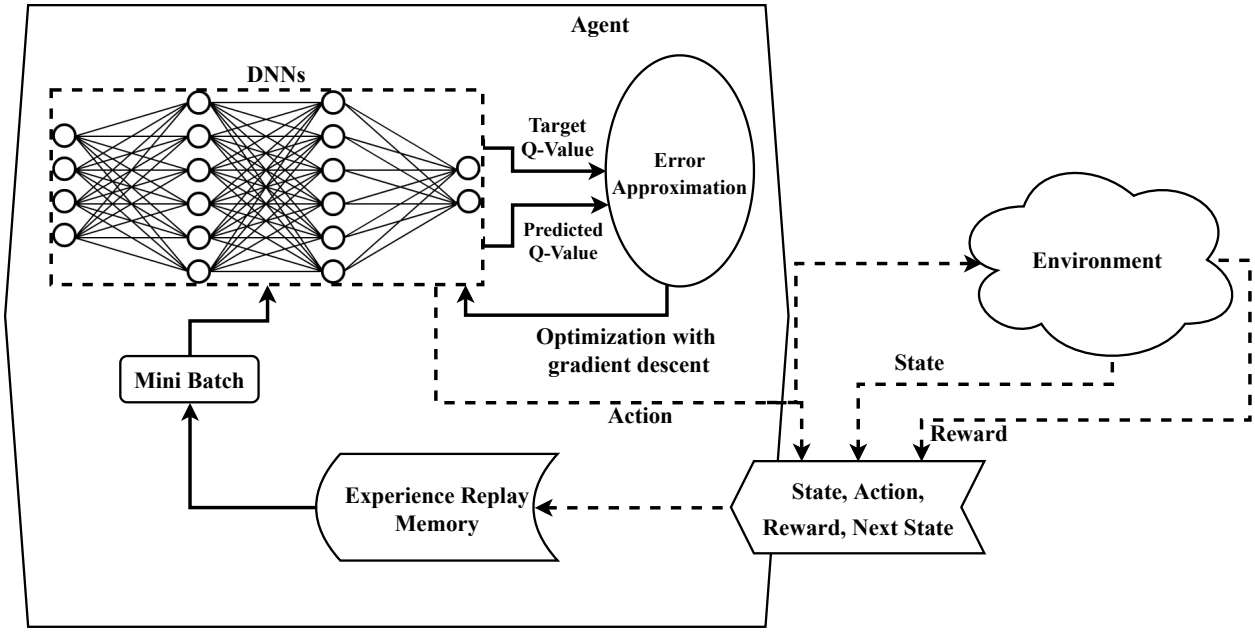

Figure 5. DQL framework.

Algorithm 2 The Deep Q-learning Algorithm.

Require: Initialize policy and target DQL network with random w and w', respectively.
Require: Initialize experience replay memory (ERM).
Require: Initialize ϵ.
for $t = 1, 2, \ldots, T$ **do**
Select an action a_t for present state s_t based on probability ϵ.
Observe the immediate reward r_t and next state s_{t+1}.
Insert (s_t, a_t, r_t, s_{t+1}) in ERM.
Create a mini-batch with random sample of (s_t, a_t, r_t, s_{t+1}) from ERM.
Optimize the weights w of the policy DNN with gradient descent via MDP.
$w' \leftarrow w$ after certain number of time steps.
end

3. Fundamentals of FANET

In this section, we briefly discuss the architectural design and characteristics of FANET. We also compare the FANET with other ad-hoc networks such as vehicular ad-hoc networks (VANETs), robot ad-hoc networks (RANETs), ship ad-hoc networks (SANETs), smartphone ad hoc networks (SPANs), and wireless sensor networks (WSNs). Finally, we discuss the optimal FANET design that researchers are trying to achieve.

3.1. FANET Architecture

The architecture of FANET is similar to MANET as it is a subset of MANET. FANET contains multiple manned or unmanned aerial vehicles and ground gateway units (GGUs) communicating with each other in an ad-hoc manner [9,27]. There are different types of topologies in FANET, such as:

- **Centralized topology:** An example of centralized topology is shown in Figure 6, where all UAVs are communicating with a GGU directly to transmit data to the control center. In this topology, UAVs also communicate with each other via the GGU [28]. This topology is more fault-tolerant but requires higher bandwidth, causes high latency, and constrains high-speed UAV mobility. Furthermore, putting up GGUs for multiple UAV groups is not economically feasible.

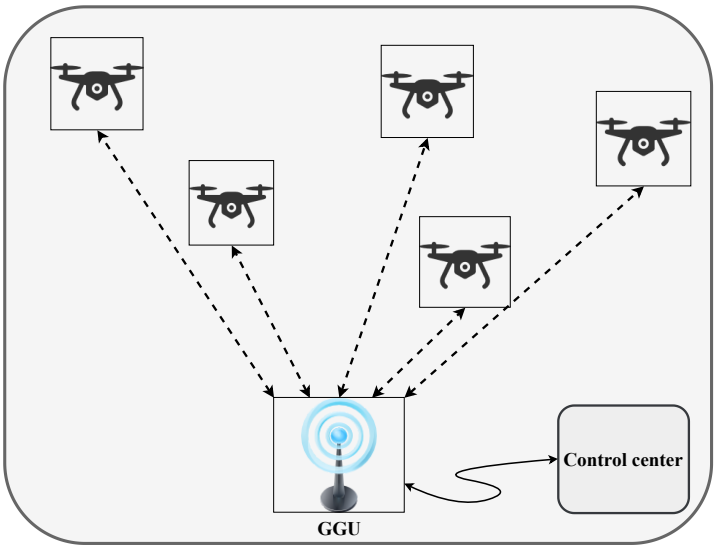

Figure 6. Centralized topology of FANET.

- **Decentralized topology:** In this topology, UAVs can communicate with each other as well as with the GGUs as shown in Figure 7 [9]. This topology provides the UAVs more flexibility for mobility and requires less bandwidth but increases the power consumption owing to the large overheads.

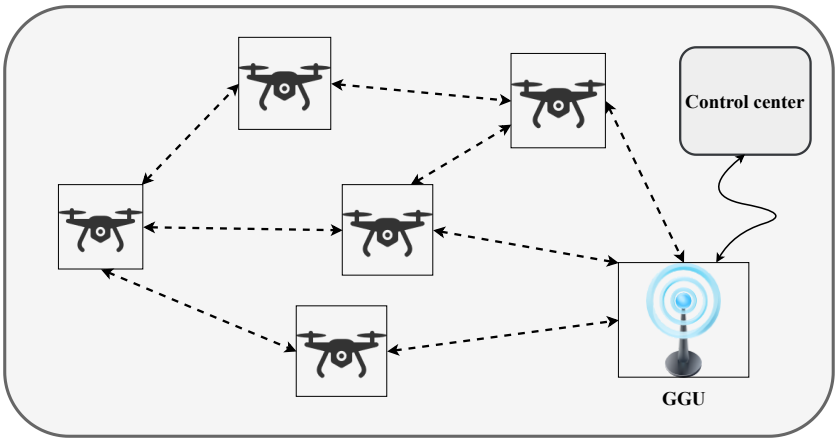

Figure 7. Decentralized topology of FANET.

- **Multigroup topology:** In this topology, UAVs are divided into multiple clusters, and each group contains a cluster head (CH), which is responsible for communicating with the GGU and other groups of UAVs as shown in Figure 8 [29]. With this topology, a large number of UAVs can be covered. However, it increases the probability of a single-point failure problem.

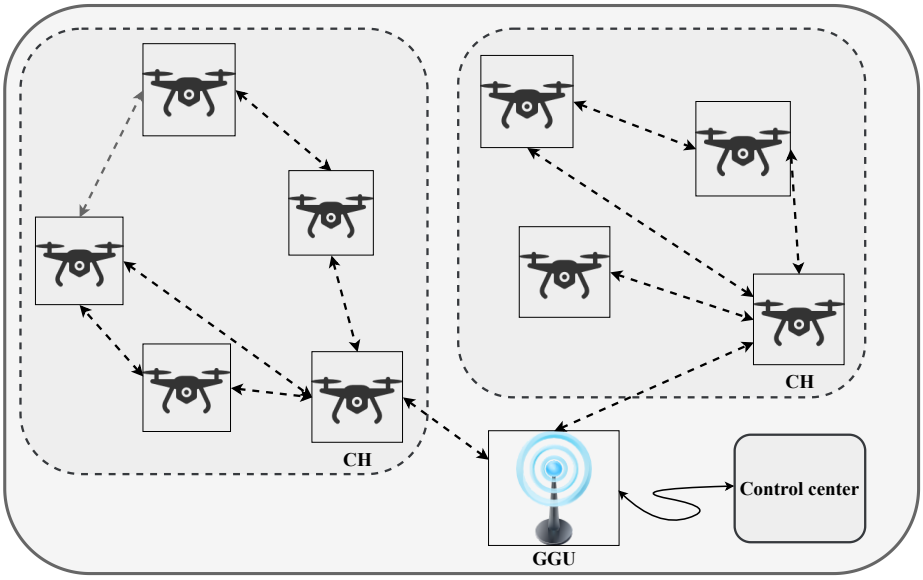

Figure 8. Multigroup topology of FANET.

3.2. Characteristics of FANET

FANETs have some unique characteristics that make them different from other ad-hoc networks. Some of the major characteristics are given as follows:

- **Node mobility and model:** There are different types of aerial vehicles which can move at an average speed of 6–500 km/h [9]. Thus, node mobility is the most important distinguishable factor which makes the FANET different from other ad-hoc networks. Furthermore, node mobility results in several challenges in communication designing. In FANET, UAVs can move freely at any direction and speed, depending on the task on its own. By contrast, other ad-hoc networks have regular, low, predefined, and controlled mobility [27]. Moreover, high mobility in FANET results in frequent changes in network topology compared to other ad-hoc networks.
- **Node density:** In wireless ad-hoc networks, node density is a crucial parameter for selecting data routing path. In FANET, node density mostly depends on the type of UAV, objective, UAV speed, and communication range. As UAVs can be speedy and have a long communication range, the number of UAV per unit area can decrease [30]. In other ad-hoc networks, such as VANETs, SANETs, WSNs, and SPANs, the node density is high compared to FANET [31].
- **Localization:** In ad-hoc network, global positing system (GPS) is widely used to locate the nodes. However, owing to the high speed mobility, FANETs use low latency GPS system to locate the UAVs such as network-based positioning [32], height-based positioning [33], differential GPS (DGPS) [34], and assisted GPS (AGPS) [35]. Moreover, localization is a major factor in flight trajectory and routing path selection.
- **Radio propagation:** When it comes to radio propagation model, FANETs have a great advantage of line-of-sight (LoS) over other ad-hoc networks. In FANET, UAVs can have a clear LoS among them due to their free mobility in the air. By contrast, in other ad-hoc networks, there is little or no LoS between the source and the destination owing to the geographical structure of the terrain.
- **Energy Constraint:** Energy limitation is one of the major design issues in ad-hoc networks. In FANET, it depends on the size of the UAV. Most of the large UAVs are not power-sensitive, whereas energy limitation is a concern for mini-UAVs [9]. In other ad-hoc networks, it varies from type to type as shown in Table 1.

Table 1 summarizes the differences among different ad-hoc networks [36–40].

Table 1. Comparative analysis of different ad-hoc networks.

Characteristics	FANETs	VANETs	SANETs	RANETs	SPANs	WSNs
Node mobility	Random	Regular	Predefined	Controlled	Regular	Static or regular
Node Speed	Upto 500 km/h	Upto 150 km/h	Upto 130 km/h	Upto 25 km/h	Upto 6 km/h	Upto 8 km/h
Node Density	Low	High	Medium	Low	Medium	Varies with application
Localization	DGPS/AGPS/Net/Height	GPS	GPS	GPS	GPS	GPS
Radio propagation	In air and high LoS	On ground and low LoS	On water and high LoS	On ground and low LoS	On ground and low LoS	On ground and low LoS
Energy Limitation	Depends on the UAV	Low	Low	High	High	High

3.3. Optimal FANET Design

Many researchers are trying to establish an optimal solution for FANET, which is more adaptable to any situation and more scalable to any extend. We discuss some optimal conditions that many researchers are trying to achieve. Moreover, we discuss the advantages of using RL over conventional methods in FANET.

As discussed earlier, FANETs have unpredictable nature owing to their high mobility and speed. The flying routes may vary from UAV to UAV in a multi-UAV system, depending on the operation requirements. More UAVs can join an ongoing operation to complete the task faster. UAVs also may fail owing to any technical problems or any environmental issues. There are so many variables in FANET environment that needs to be addressed. Thus, the optimal design should be more adaptive, super-fast, highly scalable, energy-efficient, more stable, and highly secure.

To achieve these features, there is no alternative to RL owing to its self-learning capability and energy efficiency. The conventional methods of selecting routing paths and flying trajectories are energy inefficient and slow. Moreover, these are not self-learning methods. To make the design solutions more adaptive and scalable, UAVs should learn to make their own decision based on the current situation. To establish self-learning design solutions, researchers have started using RL. Furthermore, many other problems, such as autonomous charging, jamming protection, relaying, localization, and fault handling, can be addressed using RL.

4. Applications of RL in FANET

In this section, we discuss the challenges of FANET that researchers solved with RL or DRL and how they implemented RL or DRL in FANET in detail. We focus on the main challenges of the FANET like routing protocol, flight trajectory selection, protection against jamming, and other challenges such as charging and relaying, as shown in Figure 9.

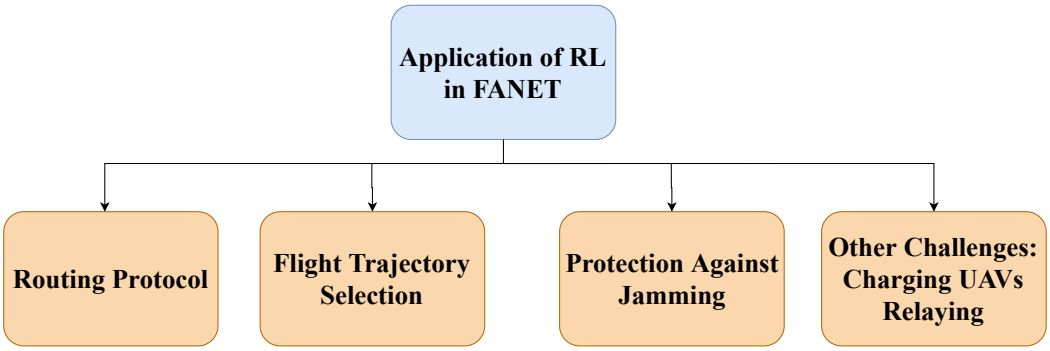

Figure 9. A taxonomy of the applications of RL in FANET.

4.1. Routing Protocol

We discuss the basics of the routing protocol, the RL-based approaches for solving routing protocol problems such as energy consumption, end-to-end delay, and path stability and we present a comparative analysis among them. The routing protocol specifies how one node communicates with other nodes in a wireless ad-hoc network [12]. Figure 10 illustrates two possible routing paths from source to destination in multi-UAV FANET. The main goal of the routing protocol is to direct the traffic toward the destination regardless of the node mobility [41]. There are no dedicated routing protocols currently available for FANETs [41]. FANET still uses conventional routing protocols used in mobile ad-hoc network (MANET) an VANET. There are different types of conventional routing protocols [42], given as follows:

- **Proactive routing:** Like wired network routing, all nodes in an ad-hoc network maintain a route table consisting of routes to other nodes. Whenever a node transmits data, a route table is used to determine the route to the destination. The route table continues to be updated to maintain the change in topology. This type of routing

protocol is unsuitable for FANETs owing to the frequent high-speed mobility of the nodes [43].

- **Reactive routing:** Whenever a node initiates communication, this type of routing protocol starts discovering routes to the destination. Predefined routing tables were not maintained in this protocol. These types of routing protocols are known as on-demand routing protocols. The main drawbacks of this protocol in terms of FANETs are poor stability, high delay, high energy consumption, and low security [44].
- **Hybrid routing:** This is a combination of and a trade-off between proactive and reactive routing protocols. In this protocol, nodes maintain a route table consisting of routers to their neighbors and start route discovery whenever the nodes try to communicate the nodes beyond their neighbors. [45].
- **Others:** In addition to the conventional routing protocols, different types of routing protocols, such as energy-based routing, heterogeneous-based routing, swarm-based routing, and hierarchical routing, etc., have been established by different researchers in the past [27,46].

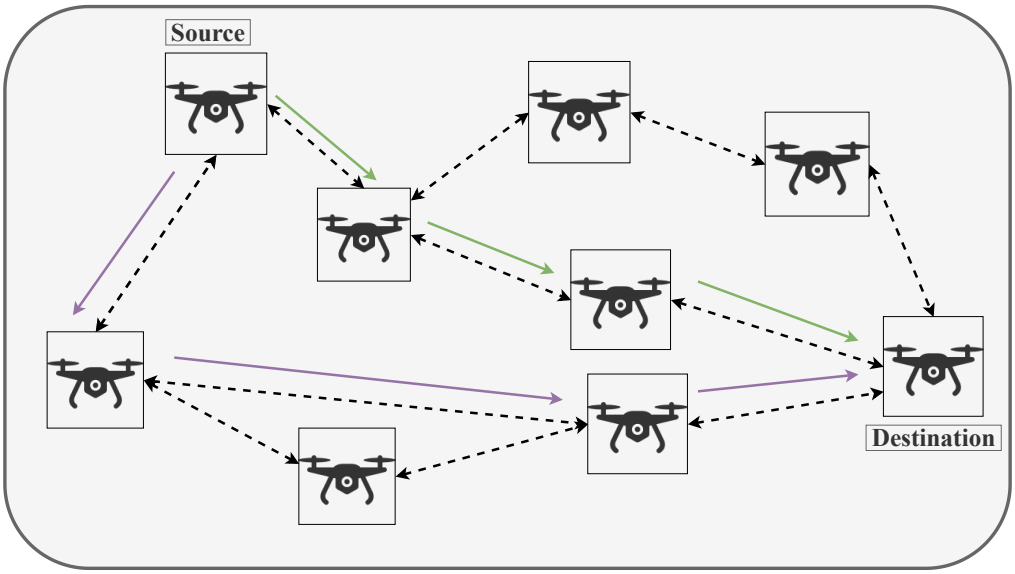

Figure 10. Multihop routing in FANET.

Owing to the complex flying environment and high mobility, UAV nodes are unpredictable [47]. Hence, conventional protocols of VANETs and MANETs cannot cope with changes in the network in real time. Therefore, many researchers have attempted to develop a self-learning, highly reliable, adaptive, and autonomous routing protocol using reinforcement learning (RL) [48]. The main purpose of using RL in FANET routing is to ensure fast and stable routing with minimum energy consumption.

4.1.1. QMR

Liu et al. [12] proposed a Q-learning-based multiobjective optimization routing protocol (QMR) where end-to-end delay and energy consumption are optimized simultaneously. They also dynamically changed the Q-learning parameters such as learning rate, discount factor, and ϵ-value for exploration and exploitation. QMR consists of routing neighbor discovery, Q-learning algorithm, routing decision, and penalty mechanism. Initially, the QMR collects the geographic locations of their neighbors using a global positioning system (GPS) and sends HELLO packets to start the route discovery process. Each HELLO packet contains the node's geo-location, energy, mobility model, queuing delay, and discount factor. Nodes start to maintain and update their neighbor table upon receiving the HELLO packets. A neighbor table contains the arrival time, learning rate, MAC delay, and Q-value along with the information of the HELLO packet [12].

After route discovery, QMR selects a neighbor to forward the data packet using Q-learning. The Q-learning algorithm considers energy consumption, link stability, one-hop delay, and neighbor relationships to select the next hop for data forwarding. The learning rate of the algorithm is an exponential adaptive function that depends on the one-hop delay. The discount factor varies with the velocity of the neighbor. For faster neighbors, the discount factor is low, and vice versa. Moreover, the trade-off between the exploration and exploitation depends on the actual velocity of the data packet traveling over a link, link quality, and neighbor relationship [12].

By incorporating all the variables, the source node computes k-weighted Q-values and forms a Q-table, where k represents the link quality and neighbor relationship. Then, the source node selects the link with the maximum k-weighted Q-value to forward the data and obtains maximum reward [12]. If there is no neighbor with a nonzero k-weighted Q-value, then the source node receives the minimum reward for all neighbors, updates the neighbor table, and searches for new neighbors using route discovery [12].

4.1.2. RLSRP with PPMAC

Reinforcement learning based self-learning routing protocol (RLSRP) with position-prediction-based directional MAC (PPMAC) is a hybrid communication protocol proposed in [49] wherein PPMAC resolves the directional deafness problem with directional antennas and RLSRP provides the routing path using RL.

In [49], Zheng et al. predicted the positions of other nodes, controlled the communication and data transmission using the PPMAC scheme. The authors used self-learning RL to determine the shortest route with the shortest delay from the source to the destination. The partially observable Markov decision process (POMDP) is incorporated with the proposed RL algorithm, where the end-to-end data transmission delay is provided as a reward. Similar to QMR, RLSRP maintains a neighbor table to keep track of the changes in the network topology. The learning parameters, such as discount factor and learning rate, are fixed. Moreover, RLSRP uses a greedy policy and selects the route with the maximum value function, where the end-to-end delay is minimum.

4.1.3. Multiobjective Routing Protocol

Yang et al. [50] proposed a Q-learning-based fuzzy logic for multiobjective routing protocol. The source node determines the routing path using the proposed algorithm while considering the transmission rate, residual energy, energy drain rate, hop count, and successful packet delivery time. A fuzzy system is used to identify reliable links, and Q-learning supports the fuzzy system by providing a reward on the path [50]. The algorithm not only considers the single-link performance but also the whole path performance using two types of Q-values from two Q-learning algorithms. After obtaining the Q-values for the single links and the entire path, the fuzzy logic evaluates the Q-values and determines the optimal path for routing. Moreover, the learning parameters, such as the discount factor and learning rate, are fixed for the Q-learning algorithm.

Similarly, in [51], He et al. determined the routing path using a fuzzy logic-based RL algorithm, but they considered delay, stability, and bandwidth efficiency factors. Figure 11 summarizes the applications of RL in the routing protocol via block diagrams. Moreover, a comparative analysis of the aforementioned protocols is presented in Table 2.

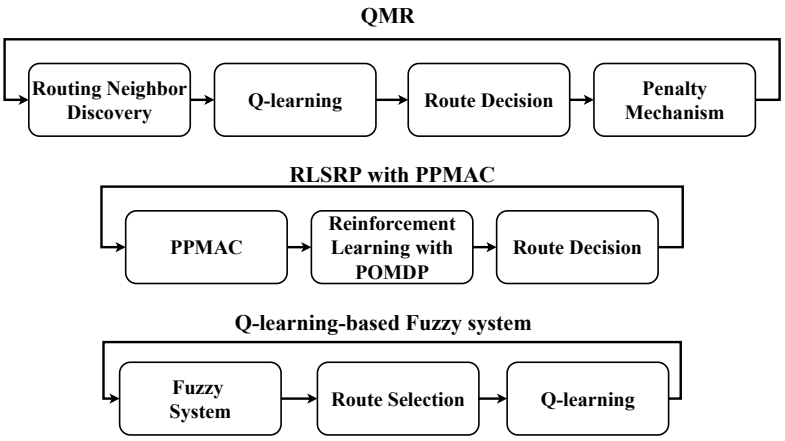

Figure 11. Application of RL in Routing Protocols of FANET.

Table 2. Comparative analysis of the routing protocols based on RL in FANET.

Routing Protocol	Algorithm	Advantages	Limitations
QMR [12]	*Q*-learning with dynamic learning rate, discount factor, and adaptive mechanism of exploration and exploitation	**1.** Multiple objectives, such as end-to-end delay, energy consumption are considered. **2.** Dynamic and adaptive *Q*-learning parameters, such as learning rate, and discount factor based on nodes' velocity and link stability. **3.** An adaptive mechanism is used for balancing exploration and exploitation. **4.** A penalty mechanism is used to combat "neighbor unavailability" problem.	**1.** Re-establishing communication is uncertain if a node gets lost. **2.** Whole route stability is not considered. **3.** Computational energy consumption is not considered.
RLSRP with PPMAC [49]	Reinforcement learning with partially observable Markov decision process (POMDP)	**1.** The positions of nodes are predictable. **2.** Antenna direction can be changed towards the routing direction. **3.** Partially observable Markov decision process (POMDP) is used. **4.** Broadcasting is used for re-establishing the communication with other nodes.	**1.** Fixed RL parameters. **2.** Only end-to-end delay is considered for route selection. **3.** There is no adaptive mechanism for balancing exploration and exploitation. **4.** Computational energy consumption is not considered.
Multiobjective Routing Protocol [50,51]	*Q*-learning-based fuzzy logic	**1.** Multiple factors, such as the transmission rate, residual energy, energy drain rate, hop count, and successful packet delivery time are considered. **2.** Both single and whole route performances are considered. **3.** Two *Q*-values are used from two *Q*-learning algorithm. **4.** Fuzzy logic is used to select the optimal route.	**1.** There is no adaptive mechanism for balancing exploration and exploitation. **2.** Fixed RL parameters. **3.** There is no mechanism to remedy the "neighbor unavailability" problem. **4.** Computational energy consumption is not considered.

4.2. Flight Trajectory Selection

We discuss the basics of UAV flight trajectory and RL-based approaches for solving problems of flight trajectory selection such as energy consumption[52], data fetching, QoS, quality of experience (QoE), coverage [53,54], and obstacles[55]. In addition, we present a comparative analysis.

As the existence of FANETs comes from the flying nodes, selecting the flying trajectory is a crucial factor in autonomous flying scenarios. There are various usages of FANETs, where flying trajectory selection plays a vital role. Using FANETs as portable interconnected aerial base stations (BSd) is one of the major commercial and civilian applications of FANETs. Because UAV base stations (UBSs) can be easily deployed to handle temporary traffic congestion, provide emergency coverage in disaster areas, to ensure the QoS, or collect data from remote internet of things (IoT) devices regardless of terrestrial territory as shown in Figure 12 [14]. Moreover, UAVs can also be used to deliver products at the doorstep of people.

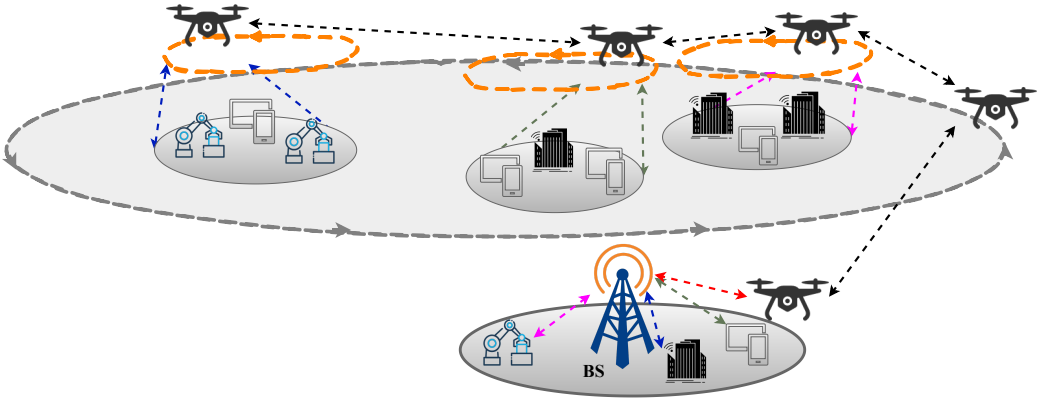

Figure 12. UBS flight trajectory in FANET.

Owing to the complex flying environment, limited data memory, limited power support, user mobility, and various QoS requirements, many researchers have proposed different trajectory designs that incorporate RL. The main reason for using RL is to obtain an optimal solution for the aforementioned challenges. The applications of RL in flight trajectory selection are summarized below and a comparative analysis is presented in Table 3.

4.2.1. Q-SQUARE

Q-SQUARE is a Q-learning-based UAV flight planning algorithm that improves the quality of experience (QoE) of video users proposed in [52]. A macro BS is considered to consist of several user clusters that require video streaming. Multiple UBSs are hovering over multiple clusters with prefetched or on-demand data depending on the QoE demand of the clusters without interfering with each other. The flying path is determined by the Q-learning algorithm, where the location of the cluster with a high QoE requirement, residual energy of the UBS, and flying time are considered. Paths to multiple recharge points are also considered for recharging the UBSs. While hovering over the cluster if the energy level approaches a certain threshold, the UBS will fly to the charging point to charge and comeback. UBS flies back to the macro BS if more video data need to be fetched. Here, the agent UBS follows the ϵ-greedy policy to determine the flight trajectory [52].

4.2.2. Decentralized Trajectory Design

A scenario is considered in [56], where multiple UAVs are performing real-time sensing and sending tasks. The main motive is to determine the decentralized flight trajectories using the opponent modeling Q-learning algorithm to transmit data efficiently using the sense-and-send protocol. The opponent modeling Q-learning algorithm is a

multiagent Q-learning algorithm, in which explicit models of the other agents are learned as stationary distributions. Using these distributions, agents take action from joint state-action pairs in each cycle [56]. Moreover, two performance-enhancing methods, action set reduction, and model-based rewarding, are introduced in the opponent modeling Q-learning algorithm to achieve a high convergence speed.

Similarly, in [57], a decentralized DRL algorithm was used to navigate multiple UBSs to provide data services to a set of ground users. Liu et al. formulated an optimization problem using POMDP and used actor-critic-based distributed control method to fly the UBSs energy-efficiently.

Table 3. Comparative analysis of the selection of flight trajectory based on RL in FANET.

Type	Algorithm	Advantages	Limitations
Q-SQUARE [52]	Single-agent Q-learning	**1.** Energy limitation is considered. **2.** Autonomous visit to charging stations is incorporated.	**1.** Fixed RL parameters. **2.** Single UAV is considered for learning.
Decentralized Trajectory Design [56,57]	Decentralized RL/DRL	**1.** POMDP is used to determine the trajectory path. **2** Energy limitation is considered. **3.** Actor-critic method is used for distributed control.	**1.** Fixed altitude setting. **2.** Autonomous visit to charging stations is not considered. **3.** User mobility is not considered.
Joint Trajectory design and Power Control [58]	Multi-agent Q-learning	**1.** Low complexity and fast convergence due to individual agent training. **2.** User location is predictable, and action is possible accordingly.	**1.** Fixed policy for other agents is considered while training an agent. **2.** Autonomous visit to charging stations is not considered. **3.** Fixed RL parameters. **4.** Energy limitation is not considered.
Multi-UAV Deployment and Movement Design [59–61]	Q-learning /Double Q-learning	**1.** User mobility is considered. **2.** Multiple UAVs are incorporated together by dividing the users in clusters. **3.** 3D deployment scenario is considered. **4.** ϵ-greedy policy is used.	**1.** User mobility is constrained within the cluster. **2.** Energy limitation is not considered. **3.** Autonomous visit to charging stations is not considered. **4.** Mobility of UAVs is constrained within 7 directions.
Trajectory Optimization for UBS [53,55]	Q-learning	**1.** ϵ-greedy policy is used. **2.** Obstacles are considered during flying towards destination. **3.** Safety check is incorporated within the system. **4.** Energy limitation is considered in terms of flying time.	**1.** Fixed altitude setting. **2.** Autonomous visit to charging stations is not considered. **3.** Mobility of UAVs is constrained within 4 directions. **4.** User mobility is not considered.

4.2.3. Joint Trajectory Design and Power Control

In [58], Liu et al. predicted the movement of the users and determined trajectories toward the users to deliver data with minimum power. The authors also predict the future positions of the users using an echo state network (ESN) and determine the trajectory in advance using multiagent Q-learning algorithm. To achieve fast convergence and reduce complexity, the authors trained one agent at a time while maintaining a fixed policy for

other agents. Moreover, the agents use a greedy policy to achieve optimal solutions for joint trajectory design and power allocation.

4.2.4. Multi-UAV Deployment and Movement Design

Multiple UAVs are deployed in a 3D space to serve mobile users in [59]. The Q-learning algorithm is used to solve the NP-hard problem [60] of 3D deployment and movement toward the users considering users' mobility. The main goal is to maximize the sum mean opinion score (MOS) of the users while maintaining the QoE.

Liu et al. [59] proposed a three-step solution in which they used the k-means algorithm to cluster the users, and then trained the UAV agents using a Q-learning algorithm to find its optimal 3D positioning with respect to the mobile users. Finally, they also used a Q-learning algorithm to determine the flying trajectory toward the moving users. However, there is a huge scope for implementing deep Q-learning to overcome constraints such as intercluster users' mobility and UAVs flying in all possible directions. Ghanavi et al. also adopted a similar kind of approach to maintain QoS in [61]. However, a double Q-learning approach was used instead of simple Q-learning in [62] for similar 3D scenarios and achieved a 14.1% gain in user satisfaction compared to simple Q-learning.

4.2.5. Trajectory Optimization for UBS

Bayerlein et al. [55] optimized the trajectory of a UBS using Q-learning to maximize the sum-rate for multiple users. The authors considered a scenario in which a UBS agent is flying at a fixed altitude to serve multiple ground users. A cuboid obstacle was also considered in this scenario. The UBS selects the flying trajectory toward the users while avoiding the obstacles using both table-based and neural network (NN) based Q-learning. Finally, the authors compared the results of table-based and NN-based Q-learning approaches, where NN-based Q-learning is more efficient and scalable.

A similar approach was taken in [53], where Klaine et al. used UBSs to provide emergency radio coverage in disaster areas. The main goal of the approach was to provide an efficient emergency network while maximizing coverage, sum-rate, and avoiding obstacles and interference.

4.3. Other Scenarios

There are other usages and challenges of FANETs, such as charging UAVs, using UAVs as network relay, using UAVs to give protection against jamming, that were solved by some researchers using RL. The applications of RL in these scenarios are summarized in Table 4.

Table 4. Summary of other scenarios based on RL in FANET.

Type	Algorithm	Characteristics	Goal
Relaying	Deep Q-learning	Multiple UAVs are positioned in dynamic UAV swarming applications by using the replay-buffer-based DQN learning algorithm which can keep track of the network topology changes [63].	Achieve the optimal communication among swarming nodes.
Protection against jamming	Federated Q-learning	Mowla et al. developed a cognitive jamming detection technique using priority-based federated learning in [64]. Then, the authors developed an adaptive model-free jamming defense mechanism based on federated Q-learning with spatial retreat strategy in [65] for FANETs.	Jamming protection for other networks.
Charging UAVs	Deep Q-learning	The mobile charging scheduling problem is interpreted as an auction problem where each UAV bids its own valuation and then the charging station schedules drones based on it in terms of revenue optimality. The charging auction enables efficient scheduling by learning the bids distribution using DQL [11].	Scheduling UAVs for charging.

5. Open Research Issues

This section discusses and highlights future possible research issues based on the analysis performed in the previous section. We summarize and compare multiple applications of RL in routing protocols and flight trajectory selection. Moreover, we summarize the applications of RL in other issues of FANET. In designing the routing protocol or selecting the flight trajectory, multiple researchers have implemented RL and attempted to solve different issues. However, there are still some open research issues in FANET that are not addressed by any studies. The open research issues are summarized below:

- **Energy constraint**: UAVs carry batteries as the main power source to support all the functionalities, such as flying, communication, and computation. However, the capacity of the batteries is insufficient for long-term deployment. Many researchers used solar energy for on-board energy harvesting and used RL to optimize the energy consumption. Unfortunately, these solutions are not sufficient for long flights. This opens a key research issue, where UAVs can harvest power wireless from nearby roadside units or base stations or power beacons for communicational and computational functionalities utilizing RL. Another way to solve the energy issue is that UAV has to exploit DRL to visit charging stations while other UAVs fill up the void.
- **3D deployment and movement**: Many studies have been carried out regarding deployment and movement. However, most of the researchers have made some significant assumptions, such as constraining UAV and user mobility [59] or reducing action-state space [56], in multi-UAV scenario. Consequently, 3D deployment and movement design considering all the constraints is still an open research issue of FANET. Furthermore, it is also important for cooperative communication of other networks, where UAVs act as relays.
- **Routing issue**: A few works have been done on routing protocols utilizing RL for FANET. Routing protocol is crucial for FANETs due to their high node mobility, low node density, and 3D node movement. There are still scopes of improvements, such as handling no neighbor problem, multiflow transmission, directional antenna problem, and scalability issues, utilizing RL. Moreover, the scope of extending the routing protocols of VANETs and MANETs for FANET using RL is still an open research issue.
- **Interference management**: Recently, UAVs are using WiFi for communicating with each other. However, interference can occur when working areas of two different FANETs with different targets overlap. Furthermore, UBSs can interfere with each others' UAV to ground communication owing to their high moving speed. These scenarios are still open challenges, where RL can be utilized.
- **Fault handling**: Fault occurrence is widespread in any network. Fault handling is crucial in FANET to avoid interruption. However, there are no existing RL-based solutions that can handle any fault, such as UAV hardware problems, equipped component problems, and communication failure due to any software issues. Thus, fault handling using RL needs to be deeply explored.
- **Security issue**: Many RL-based strategies were developed in the past to prevent jamming and cyber attacks for MANET and VANET [66]. However, there are few RL-based solutions available for FANET security. If even all the aforementioned issues were solved, communication in FANET can still be interrupted due to a security breach. Consequently, RL-based security solutions require an in-depth investigation.

6. Conclusions

In this study, the latest applications of RL in FANETs have been exhaustively reviewed in terms of major features and characteristics and qualitatively compared with each other. However, RL can be computationally expensive, but the outcome from using RL is promising in terms of providing better performance in terms of major performance parameters such as energy consumption, flight time, communication delay, QoS, QoE, and network lifetime. The comparative analysis of different applications of RL in different scenarios

of FANETs presented in this study can be effectively used for choosing and improving flight paths, routing protocols, charging, relaying, etc. We also discuss the RL-based open research issues of FANETs that need to be explored. Finally, it can be concluded that adaptive RL parameters and a balance between exploration and exploitation strategies help RL to converge more rapidly while overcoming the challenges of FANETs.

Author Contributions: Conceptualization, S.R. and W.C.; methodology, W.C.; software, S.R.; validation, S.R. and W.C.; formal analysis, S.R. and W.C.; investigation, S.R. and W.C.; resources, W.C.; data curation, S.R. and W.C.; writing—original draft preparation, S.R.; writing—review and editing, W.C.; visualization, S.R. and W.C.; supervision, W.C.; project administration, W.C.; funding acquisition, W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2019R1F1A1046687) and by the research fund from Chosun University, 2020.

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

References

1. Wang, J.; Jiang, C.; Han, Z.; Ren, Y.; Maunder, R.G.; Hanzo, L. Taking Drones to the Next Level: Cooperative Distributed Unmanned-Aerial-Vehicular Networks for Small and Mini Drones. *IEEE Veh. Technol. Mag.* **2017**, *12*, 73–82. doi:10.1109/MVT.2016.2645481.
2. Batista da Silva, L.C.; Bernardo, R.M.; de Oliveira, H.A.; Rosa, P.F.F. Multi-UAV agent-based coordination for persistent surveillance with dynamic priorities. In Proceedings of the 2017 International Conference on Military Technologies (ICMT), Brno, Czech Republic, 31 May–2 June 2017; pp. 765–771. doi:10.1109/MILTECHS.2017.7988859.
3. Alshbatat, A.I.; Dong, L. Cross layer design for mobile Ad-Hoc Unmanned Aerial Vehicle communication networks. In Proceedings of the 2010 International Conference on Networking, Sensing and Control (ICNSC), Chicago, IL, USA, 10–12 April 2010; pp. 331–336. doi:10.1109/ICNSC.2010.5461502.
4. Semsch, E.; Jakob, M.; Pavlicek, D.; Pechoucek, M. Autonomous UAV Surveillance in Complex Urban Environments. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Milan, Italy, 15–19 September 2009; Volume 2, pp. 82–85. doi:10.1109/WI-IAT.2009.132.
5. Maza, I.; Caballero, F.; Capitan, J.; Martinez-de Dios, J.R.; Ollero, A. Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications. *J. Intell. Robot. Syst.* **2011**, *61*, 563–585. doi:10.1007/s10846-010-9497-5.
6. de Freitas, E.P.; Heimfarth, T.; Netto, I.F.; Lino, C.E.; Pereira, C.E.; Ferreira, A.M.; Wagner, F.R.; Larsson, T. UAV relay network to support WSN connectivity. In Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems, Moscow, Russia, 18–20 October 2010; pp. 309–314. doi:10.1109/ICUMT.2010.5676621.
7. Xiang, H.; Tian, L. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosyst. Eng.* **2011**, *108*, 174–190. doi:10.1016/j.biosystemseng.2010.11.010.
8. Barrado, C.; Messeguer, R.; Lopez, J.; Pastor, E.; Santamaria, E.; Royo, P. Wildfire monitoring using a mixed air-ground mobile network. *IEEE Pervasive Comput.* **2010**, *9*, 24–32. doi:10.1109/MPRV.2010.54.
9. Bekmezci, I.; Sahingoz, O.; Temel, Ş. Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Netw.* **2013**, *11*, 1254–1270. doi:10.1016/j.adhoc.2012.12.004.
10. Mukherjee, A.; Keshary, V.; Pandya, K.; Dey, N.; Satapathy, S. Flying Ad-hoc Networks : A Comprehensive Survey. *Inf. Decis. Sci.* **2018**, *701*, 569–580.
11. Shin, M.; Kim, J.; Levorato, M. Auction-Based Charging Scheduling With Deep Learning Framework for Multi-Drone Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4235–4248. doi:10.1109/TVT.2019.2903144.
12. Liu, J.; Wang, Q.; He, C.; Jaffres-Runser, K.; Xu, Y.; Li, Z.; Xu, Y.J. QMR: Q-learning based Multi-objective optimization Routing protocol for Flying Ad Hoc Networks. *Comput. Commun.* **2019**, *150*. doi:10.1016/j.comcom.2019.11.011.
13. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.; Kim, D.I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. doi:10.1109/COMST.2019.2916583.
14. 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. doi:10.3390/s19235170.
15. Xiong, Z.; Zhang, Y.; Niyato, D.; Deng, R.; Wang, P.; Wang, L. Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges. *IEEE Veh. Technol. Mag.* **2019**, *14*, 44–52. doi:10.1109/MVT.2019.2903655.
16. Forster, A. Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey. In Proceedings of the 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Melbourne, Australia, 3–6 December 2007; pp. 365–370. doi:10.1109/ISSNIP.2007.4496871.
17. Qian, Y.; Wu, J.; Wang, R.; Zhu, F.; Zhang, W. Survey on Reinforcement Learning Applications in Communication Networks. *J. Commun. Inf. Netw.* **2019**, *4*, 30–39. doi:10.23919/JCIN.2019.8917870.

18. Ponsen, M.; Taylor, M.E.; Tuyls, K. Abstraction and Generalization in Reinforcement Learning: A Summary and Framework. In *Adaptive and Learning Agents*; Taylor, M.E., Tuyls, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–32.
19. Tuyls, K.; Nowe, A. Evolutionary game theory and multi-agent reinforcement learning. *Knowl. Eng. Rev.* **2005**, *20*, 63–90. doi:10.1017/S026988890500041X.
20. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *arXiv* **2018**, arXiv:1811.12560.
21. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. doi:10.1016/j.neucom.2016.12.038.
22. Bau, D.; Zhu, J.Y.; Strobelt, H.; Lapedriza, A.; Zhou, B.; Torralba, A. Understanding the role of individual units in a deep neural network. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 30071–30078. doi:10.1073/pnas.1907375117.
23. Li, Y. Deep Reinforcement Learning: An Overview. *arXiv* **2018**, arXiv:1701.07274.
24. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. *arXiv* **2016**, arXiv:1602.01783.
25. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. doi:10.1109/MSP.2017.2743240.
26. Lillicrap, T.; Hunt, J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
27. 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. doi:10.1109/ACCESS.2019.2923840.
28. Bacco, M., Cassarà, P., Colucci, M., Gotta, A., Marchese, M. and Patrone, F. A Survey on Network Architectures and Applications for Nanosat and UAV Swarms. In Proceedings of the 2018 International Conference on Wireless and Satellite Systems; Springer International Publishing; Oxford, United Kingdom, 14–15 September 2017; pp. 75–85.
29. Vanitha, N.; Padmavathi, G. A Comparative Study on Communication Architecture of Unmanned Aerial Vehicles and Security Analysis of False Data Dissemination Attacks. In Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 1–3 March 2018; pp. 1–8. doi:10.1109/ICCTCT.2018.8550873.
30. Van Der Bergh, B.; Chiumento, A.; Pollin, S. LTE in the sky: trading off propagation benefits with interference costs for aerial nodes. *IEEE Communications Magazine* **2016**, *54*, 44–50. doi:10.1109/MCOM.2016.7470934.
31. Chriki, A.; Touati, H.; Snoussi, H.; Kamoun, F. FANET: Communication, Mobility models and Security issues *Computer Networks* **2019**, *163*, 106877. doi:10.1016/j.comnet.2019.106877.
32. Fadlullah, Z.; Takaishi, D.; Nishiyama, H.; Kato, N.; Miura, R. A dynamic trajectory control algorithm for improving the communication throughput and delay in UAV-aided networks. *IEEE Network* **2016**, *30*, 100–105. doi:10.1109/MNET.2016.7389838.
33. Al-Hourani, A.; Al-Hourani, S.; Lardner, S. Optimal LAP Altitude for Maximum Coverage. *IEEE Wireless Communications Letters* **2014**, *3*, 569–572. doi:10.1109/LWC.2014.2342736.
34. Ahn, H.; Won, C. DGPS/IMU integration-based geolocation system: Airborne experimental test results. *Aerospace Science and Technology* **2009**, *13*, 316–324. doi:10.1016/j.ast.2009.06.003.
35. Wong, A.; Woo, T.; Lee, A.; Xiao, X.; Luk, V. An AGPS-based elderly tracking system. In Proceedings of the 2009 First International Conference on Ubiquitous and Future Networks, Hong Kong, China, 7–9 June 2009; pp. 100–105. doi:10.1109/ICUFN.2009.5174293.
36. Mahmood, M.; Seah, M.; Welch, I. Reliability in Wireless Sensor Networks: Survey and Challenges Ahead. *Computer Networks* **2015**, *79*, 166–187. doi:10.1016/j.comnet.2014.12.016.
37. Li, W.; Song, H. ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems* **2016**, *17*, 960–969. doi:10.1109/TITS.2015.2494017.
38. Vandenberghe, W.; Moerman, I.; Demeester, P. Adoption of Vehicular Ad Hoc Networking Protocols by Networked Robots. *Wireless Personal Communications* **2012**, *64*, 489—522. doi:10.1007/s11277-012-0598-2.
39. Al-Zaidi, R.; Woods, J.; Al-Khalidi, M.;Hu, H. Building Novel VHF-Based Wireless Sensor Networks for the Internet of Marine Things. *IEEE Sensors Journal* **2018**, *18*, 2131—2144. doi:10.1109/JSEN.2018.2791487.
40. Mitra, P.; Poellabauer, C. Emergency response in smartphone-based Mobile Ad-Hoc Networks. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 6091–6095. doi:10.1109/ICC.2012.6364839.
41. Yang, H.; Liu, Z. An optimization routing protocol for FANETs. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 120. doi:10.1186/s13638-019-1442-0.
42. Nayyar, A. Flying Adhoc Network (FANETs): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP. In Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 6–7 August 2018; pp. 1–9. doi:10.1109/ICABCD.2018.8465130.
43. Corson, M.S.; Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *RFC* **1999**, *2501*, 1–12. Available online: https://dl.acm.org/doi/pdf/10.17487/RFC2501 (accessed on 15 January 2020).

44. Perkins, C.E.; Royer, E.M. Ad-hoc on-demand distance vector routing. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (Proceedings WMCSA'99), New Orleans, LA, USA, 25–26 February 1999; pp. 90–100. doi:10.1109/MCSA.1999.749281.
45. Ben Haj Frej, M.; Mandalapa Bhoopathy, V.; Ebenezer Amalorpavaraj, S.R.; Bhoopathy, A. Zone Routing Protocol (ZRP)—A Novel Routing Protocol for Vehicular Ad-hoc Networks. In Proceedings of the ASEE-NE 2016, Kingston, RI, USA, 28–30 April 2016.
46. Khan, I.U.; Qureshi, I.M.; Aziz, M.A.; Cheema, T.A.; Shah, S.B.H. Smart IoT Control-Based Nature Inspired Energy Efficient Routing Protocol for Flying Ad Hoc Network (FANET). *IEEE Access* **2020**, *8*, 56371–56378. doi:10.1109/ACCESS.2020.2981531.
47. Dao, N.; Koucheryavy, A.; Paramonov, A. Analysis of Routes in the Network Based on a Swarm of UAVs. In *Information Science and Applications (ICISA) 2016*; Kim, K.J., Joukov, N., Eds.; Springer: Singapore, 2016; pp. 1261–1271.
48. Chettibi, S.; Chikhi, S. A Survey of Reinforcement Learning Based Routing Protocols for Mobile Ad-Hoc Networks. In *Recent Trends in Wireless and Mobile Networks*; Özcan, A., Zizka, J., Nagamalai, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–13.
49. Zheng, Z.; Sangaiah, A.K.; Wang, T. Adaptive Communication Protocols in Flying Ad Hoc Network. *IEEE Commun. Mag.* **2018**, *56*, 136–142. doi:10.1109/MCOM.2017.1700323.
50. Yang, Q.; Jang, S.; Yoo, S.J. Q-Learning-Based Fuzzy Logic for Multi-objective Routing Algorithm in Flying Ad Hoc Networks. *Wirel. Pers. Commun.* **2020**, *113*. doi:10.1007/s11277-020-07181-w.
51. He, C.; Liu, S.; Han, S. A Fuzzy Logic Reinforcement Learning-Based Routing Algorithm For Flying Ad Hoc Networks. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020; pp. 987–991. doi:10.1109/ICNC47757.2020.9049705.
52. Colonnese, S.; Cuomo, F.; Pagliari, G.; Chiaraviglio, L. Q-SQUARE: a Q-learning approach to provide a QoE aware UAV flight path in cellular networks. *Ad Hoc Netw.* **2019**, *91*, 101872. doi:10.1016/j.adhoc.2019.101872.
53. Valente Klaine, P.; Nadas, J.; Souza, R.; Imran, M. Distributed Drone Base Station Positioning for Emergency Cellular Networks Using Reinforcement Learning. *Cogn. Comput.* **2018**, *10*. doi:10.1007/s12559-018-9559-8.
54. Wu, J.; Yu, P.; Feng, L.; Zhou, F.; Li, W.; Qiu, X. 3D Aerial Base Station Position Planning based on Deep Q-Network for Capacity Enhancement. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 8–12 April 2019; pp. 482–487.
55. Bayerlein, H.; De Kerret, P.; Gesbert, D. Trajectory Optimization for Autonomous Flying Base Station via Reinforcement Learning. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5. doi:10.1109/SPAWC.2018.8445768.
56. Hu, J.; Zhang, H.; Song, L. Reinforcement Learning for Decentralized Trajectory Design in Cellular UAV Networks With Sense-and-Send Protocol. *IEEE Internet Things J.* **2019**, *6*, 6177–6189. doi:10.1109/JIOT.2018.2876513.
57. Liu, C.H.; Ma, X.; Gao, X.; Tang, J. Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1274–1285. doi:10.1109/TMC.2019.2908171.
58. Liu, X.; Liu, Y.; Chen, Y.; Hanzo, L. Trajectory Design and Power Control for Multi-UAV Assisted Wireless Networks: A Machine Learning Approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7957–7969. doi:10.1109/TVT.2019.2920284.
59. Liu, X.; Liu, Y.; Chen, Y. Reinforcement Learning in Multiple-UAV Networks: Deployment and Movement Design. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8036–8049. doi:10.1109/TVT.2019.2922849.
60. Wang, W.Y.; Li, J.; He, X. Deep Reinforcement Learning for NLP. In *Tutorial Abstracts, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 19–21. doi:10.18653/v1/P18-5007.
61. Ghanavi, R.; Kalantari, E.; Sabbaghian, M.; Yanikomeroglu, H.; Yongacoglu, A. Efficient 3D aerial base station placement considering users mobility by reinforcement learning. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6. doi:10.1109/WCNC.2018.8377340.
62. Liu, X.; Chen, M.; Yin, C. Optimized Trajectory Design in UAV Based Cellular Networks for 3D Users: A Double Q-Learning Approach. *J. Commun. Inf. Netw.* **2019**, *4*, 24–32. doi:10.23919/JCIN.2019.8916643.
63. Koushik, A.M.; Hu, F.; Kumar, S. Deep Q -Learning-Based Node Positioning for Throughput-Optimal Communications in Dynamic UAV Swarm Network. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 554–566. doi:10.1109/TCCN.2019.2907520.
64. Mowla, N.I.; Tran, N.H.; Doh, I.; Chae, K. Federated Learning-Based Cognitive Detection of Jamming Attack in Flying Ad-Hoc Network. *IEEE Access* **2020**, *8*, 4338–4350. doi:10.1109/ACCESS.2019.2962873.
65. Mowla, N.I.; Tran, N.H.; Doh, I.; Chae, K. AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET. *J. Commun. Netw.* **2020**, *22*, 244–258. doi:10.1109/JCN.2020.000015.
66. Bekmezci, İ.; Şentürk, E.; Türker, T. Security issues in flying Ad-hoc Networks (FANETs). *J. Aeronaut. Space Technol.* **2016**, *9*, 13–21.