Unlocking the Power of Reinforcement Learning: Investigating Optimal Q-Learning Parameters for Routing in Flying Ad Hoc Networks

Mariem Bousaid, Safa Kaabi, Amine Dhraief, Khalil Drira

Unlocking the Power of Reinforcement Learning: Investigating Optimal Q-Learning Parameters for Routing in Flying Ad Hoc Networks

Mariem Bousaid *ENSI*, *University of Manouba*
La Manouba, Tunisia
mariem.bousaid@ensi-uma.tn

Safa Kaabi[1], Amine Dhraief[2] *ESEN*, *University of Manouba*
La Manouba, Tunisia
[1]safa.kaabi@esen.tn, [2]amine.dhraief@esen.tn

Khalil Drira *LAAS-CNRS, University of Toulouse, France*
khalil@laas.fr

April 12, 2024

Abstract

The routing challenges in Flying Ad Hoc Networks (FANETs), characterized by high-speed Unmanned Aerial Vehicles (UAVs), limited UAV battery life, intermittent links, network partitioning, and dynamic topologies, have led to the development of specialized routing protocols based on Reinforcement Learning (RL). In this context, the Q-Learning algorithm is the most commonly used RL algorithm. It relies on two primary hyperparameters: the learning rate and discount factor. The protocol's efficiency hinges on the selection of these parameters. To tackle this challenge, numerous adaptive Q-Learning routing protocols introduce novel functions to dynamically adjust the learning parameters. Therefore, this paper delves into an examination of these parameters and introduces a novel taxonomy categorizing them into three distinct classes: linear function-based adjustment, exponential function-based adjustment, and grid search-based adjustment. This paper highlights that the prevailing adjustment function for the learning rate follows a decreasing exponential pattern, while the discount factor adheres to a linear function. This equilibrium facilitates swift adaptation to changes while ensuring a stable transition between short-term and long-term rewards. Such balance is essential for efficient and effective routing in FANETs.

Index Terms: FANET, Q-Learning, Learning rate, Discount factor, Adaptive learning, Routing protocol, Topology changes.

1 Introduction

Unmanned Aerial Vehicles (UAVs), also known as drones, are objects that autonomously fly using onboard computer systems and sensors or can be remotely controlled by an operator [1]. They are becoming increasingly popular due to their versatility and ability to perform dangerous or challenging tasks for humans. According to the Federal Aviation Agency (FAA), there will be a significant increase in the number of UAVs in the coming years.

UAVs cooperate through an ad-hoc network to form a Flying Ad-Hoc Network (FANET) and collaborate to route data among themselves. Each UAV collects data and transmits it within the swarm until it reaches the destination, which is mostly the base station. Due to the highly dynamic nature of UAV network topology, routing in such networks presents a significant challenge. The relative positions of UAVs change rapidly, often resulting in intermittent links. Furthermore, in many UAV deployment scenarios, UAV density may be very low, leading to frequent network partitions.

Due to the unique characteristics of FANETs, conventional routing protocols designed for Mobile Ad-Hoc Networks (MANETs) and Vehicular Ad-Hoc Networks (VANETs) cannot be applied to them. To address this issue, two different routing approaches have been proposed in the literature: (i) topology-based routing and (ii) position-based routing. In topology-based routing, UAVs build their routing tables based on their link status. In position-based routing, UAVs share their localization to build their routing tables. Machine learning algorithms enable interactive decision-making in wireless networks by learning from data and past experiences. Reinforcement Learning (RL) refers to a machine learning algorithm that not only uses existing data but also acquires new data through exploration of the environment to achieve real-time dynamic optimization. This characteristic makes RL particularly suitable for finding paths in the context of FANETs. Recently, RL has gained widespread use in designing routing protocols for FANETs. While node locations play a fundamental role in routing decisions, especially in scenarios with high node speeds like FANETs, most RL-based routing protocols are position-based. These protocols use the physical locations of nodes, typically obtained through GPS or other localization methods, as input features for the RL agent. In the literature, the most commonly used RL algorithm for routing is Q-Learning (QL) due to its simplicity and low computational cost.

In this paper, we provide a comprehensive review of existing Q-Learning-based routing protocols for FANETs. We explore the relationship between Q-Learning and routing in FANETs, emphasizing that the Q-Learning algorithm relies on two main hyperparameters: the learning rate and the discount factor. The efficiency of the algorithm hinges on the careful selection of these parameters. Our primary focus is on the challenge of adapting Q-Learning hyperparameters. We delve into a detailed review of adaptive Q-Learning-based routing protocols and discuss the formulation of these parameters. Additionally, we categorize adaptive Q-Learning-based routing protocols based on the types of functions used to adjust the learning parameters. We propose a novel

taxonomy categorizing the learning parameters adjustments into three classes: linear function-based adjustment, exponential function-based adjustment, and grid search-based adjustment.

The rest of this paper is organized as follows. Section 2 explains the fundamentals of the Q-Learning algorithm. Then, in section 3, we discuss the challenges of the Q-Learning-based routing protocols designed for FANETs and techniques proposed to address them. Section 4 is dedicated to the investigation of adaptive Q-Learning parameters and categorization of adaptive Q-Learning-based routing protocols. Finally, section 5 concludes the paper.

2 Fundamentals of Q-Learning

In this section, we detail the Q-Learning algorithm and its challenges.

The Q-Learning algorithm is a principal technique in RL that has been widely used in various applications [2]. It does not require *a priori* knowledge of the environment's dynamics. It is based on the concept of a Q-value, which represents the expected utility of taking a particular action in a given state. Eq. (1) presents the Bellman equation for the Q-value [3]. The Bellman equation provides the framework for updating Q-values, which is a key step in the process of learning an optimal policy. This optimal policy enables the agent to make well-informed decisions within the environment, aimed at maximizing cumulative rewards.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{1}$$

$Q(S_t, A_t)$ is the Q-value of the current state S_t when action A_t is selected at time t. $\max_a Q(S_{t+1}, a)$ represents the maximum Q-value among all possible actions a that can be taken in the next state S_{t+1}. This value estimates the potential future rewards that the agent can obtain from the next state. R_{t+1} is the immediate reward received by the agent at time $t+1$ after taking action A_t in state S_t. α is the learning rate, and γ is the discount factor. Both α and γ are referred to as learning parameters, and their values typically range between 0 and 1.

α represents the extent to which an agent updates its Q-values based on new information. It determines the weight assigned to new observations in comparison to existing Q-values. When α is set to a higher value, the agent tends to prioritize new information over previous knowledge. This means that the agent will rapidly adjust its Q-values based on recent experiences. A higher α is advantageous in dynamic environments where rapid learning is necessary. On the contrary, when α is set to a lower value, the agent places greater emphasis on its existing Q-values and exhibits a more conservative approach in updating them with new information. This can be useful in stable environments where the agent needs to gradually refine its policy and avoid making rapid changes

based on limited observations. Choosing an appropriate α depends on the environment. It often requires experimentation and tuning to find the optimal balance between exploration and exploitation of the agent's knowledge.

γ represents the rate at which future rewards are discounted. It determines the importance given to immediate rewards compared to future rewards when updating the Q-values. When the γ value is close to 1, the agent places high importance on long-term rewards. This means that the agent considers not only immediate rewards but also potential future rewards. As a result, the agent makes decisions that maximize cumulative returns over time. We conclude that, a higher γ value is suitable for scenarios where delayed rewards are significant, such as in tasks with long-term planning, or where the consequences of actions are observed over multiple steps. Conversely, when the γ value is close to 0, the agent prioritizes immediate rewards over future rewards. This means that it focuses on short-term gains rather than long-term planning. This makes the agent myopic in decision-making [3]. We conclude that, a lower γ value is appropriate in high dynamic environments. Choosing an optimal γ value depends on the targeted application. The trade-off between short-term gains and long-term planning requires careful consideration of the task dynamics and the importance of immediate and future rewards.

3 Q-Learning-based routing protocols and challenges

Q-Learning-based routing protocols in FANETs aim to enhance routing decisions among UAVs. Each node in the network must determine efficient paths for transmitting data while sensing and adapting to environmental changes. Several works have considered the agent as the UAV, others as the data packet. Agents update their action strategies through the reward earned after performing a particular action. So, they do not need to know the whole network when making routing decisions. Agent routing relies on a neighbor table called Q-Table, that stores local information about neighboring nodes, such as position, speed, direction, and energy. This table takes as input the states and actions of the agent, then the Q-value for each stored state-action pair is calculated. A second table is also necessary to store the rewards that will be used when calculating the Q-value.

While effective in many situations, the main limitation of Q-Learning is its slow convergence speed. In the following section, we will discuss three main challenges that arise as a consequence: (i) the curse of dimensionality, (ii) the exploration-exploitation trade-off, and (iii) the optimization of Q-Learning parameters.

3.0.1 The curse of dimensionality

Due to the unique characteristics of FANETs, such as the high mobility of UAVs, the number of states and actions will be high. Similarly, in large-scale

networks, the number of states and actions is high. As the state space increases, the dimensions of the Q-Table also increase. Consequently, the time and memory requirements to store and update Q-values increase exponentially. Hence, the performance of Q-Learning degrades. To mitigate this issue, one possible solution is to reduce the size of the Q-Table [4] [5]. For this, Q-FANET [5] and QMR [4] narrow down the neighborhood set. To filter the neighbors' set, a velocity requirement condition must be checked before proceeding to the decision-making module based on the Q-value. This process involves evaluating the actual velocities of a data packet as it traverses links from the current node to its neighbors. By maintaining a shorter list of potential neighbors selected based on their actual velocities, the data packet can reach its destination with reduced delay. This approach not only improves convergence speed but also minimizes transmission delays. Researchers propose also the use of deep reinforcement learning methods to enhance convergence speed, such as DQN-VR [6] and TQNGPSR [7] [8].

3.0.2 The exploration-exploitation trade-off

Q-Learning is grounded in two primary strategies: exploration and exploitation, as outlined in Watkins' work [2]. In exploration, the agent deliberately selects a random action instead of the one it believes to be the best. This strategy encourages the agent to explore new actions and states, promoting a broader exploration of the environment. On the other hand, exploitation involves choosing actions with the highest Q-values based on the agent's current knowledge or learned policy. Balancing between exploration and exploitation is a fundamental aspect of Q-Learning. Excessive exploration may make it challenging to retain potentially superior actions, while excessive exploitation can hinder the discovery of new and better actions that have not been previously explored. In the context of FANETs, achieving an optimal balance between exploration and exploitation is not only regulated by traditional methods like the ϵ-greedy strategy [3], but also requires adaptation according to the network's changing conditions. Therefore, an adaptive mechanism for exploration and exploitation is needed, one that dynamically adjusts this balance in response to the network's evolving state. For example, in QRIFC [9], the authors introduce a mechanism based on the relationship between the normalized average link duration (NALD), packet travel time (PTT), and packet travel speed (PTS).

3.0.3 The optimization of Q-Learning parameters

Optimizing the values of α and γ based on the degree of topology change is a challenging task in FANETs. Setting an appropriate α can significantly impact the convergence of the Q-Learning algorithm. The higher α is, the faster the Q-value is updated. The more unstable the link between nodes, the faster the Q-values should be updated. Conversely, if α is set too low, the Q-values may converge very slowly. Many existing Q-learning-based routing protocols use fixed α values, which is not suitable for FANETs. Adapting α can lead to faster

and more stable convergence.

For the second parameter of Q-Learning, γ, a high value promotes long-term rewards more and indicates that the future Q-values are stable. However, a low γ value gives more weight to immediate rewards, leading the agent to focus on maximizing short-term gains and indicates unstable Q-values. Adjusting γ allows the agent to balance immediate and future rewards based on the characteristics of the environment. In FANETs, it is important to tailor γ to the mobility of neighboring nodes in adjacent time slots to reliably select a neighbor for packet transmission. Adapting γ leads to improved agent performance and convergence speed. In this rest of the paper, we will focus on the strategies proposed to adapt the learning parameters in order to overcome these two aforementioned challenges. 4.

4 Adaptive Q-Learning parameters for routing in FANETs

In FANETs, when Q-Learning parameters are fixed [5] [10] [11], the accuracy of action selection declines, and the selected link may have low probability to connect to a neighbor node. A possible solution is to adjust α and γ in accordance to the topology change. Several works propose the use of adaptive approaches for learning parameters. In this section, we describe the learning parameters formulas proposed in recent works and discuss them. We also categorize them into three classes according to the function type: (i) `Linear function-based adjustment`, (ii) `Exponential function-based adjustment` and (iii) `Grid search-based adjustment`.

In the `Linear function-based adjustment` class, the learning parameter is expressed in the form of a first-order polynom. This category provides an intuitive method that simplifies both the calculation process and the interpretation of results.

In the `Exponential function-based adjustment` class, the learning parameter can be represented in two forms (i) an exponential decreasing function: e^{-x} or (ii) complementary exponential decreasing function: $1-e^{-x}$, with x being a positive value. The first form, e^{-x}, yields a value between 0 and 1, indicating an increase or decrease in the parameter value depending on x. As the value of x increases, e^{-x} approaches 0, which decreases the learning parameter value. Conversely, as the value of x decreases, e^{-x} approaches 1, which increases the learning parameter value. Thus, the value of the learning parameter is influenced by the behavior of the exponential function with respect to the variable x. The second form is $1 - e^{-x}$, offering an alternative representation with similar characteristics. Both forms provide valuable insights into the adjustment process, taking into consideration the dynamic behavior of the environment, which is modeled by the variable x. This method is commonly used for adjusting the learning rate.

We refer to the last class as `Grid Search`. It consists on defining a set of

learning parameter values and evaluating the performance of Q-Learning for each value. Therefore, a search is conducted over a range of these values to find the best parameter value according to a predefined condition. In the following, we will provide a more detailed examination of how each learning parameter is adjusted for each protocol, based on the mentioned classification.

4.1 Dynamic learning rate adjustment

The learning rate α, also referred to as the step size, controls the speed of updating Q-values. In dynamic network topologies where changes occur rapidly, a larger learning rate is required to prioritize new information. Consequently, several studies have focused on updating the learning rate.

Most studies have predominantly focused on employing an exponential decreasing function-based adjustment method for the adaptive learning rate α. The first form of this method is represented as e^{-x}. As x increases, e^{-x} approaches 0. Cui et al. applied this form in the routing protocol TARRAQ [12], where x denotes the predicted residual link duration, denoted as $\hat{T}_{i,j}^{rf}$. This duration is calculated based on the predicted status (position and velocity) of neighboring nodes. When the connection duration increases, the link stability also improves. In such cases, older information remains valid, making the exploitation of existing knowledge more valuable than exploring new knowledge. Moreover, the value of α gradually decreases and approaches 0, which aligns well with the requirements of a static and less dynamic environment. However, in dynamic environments characterized by rapid network topology changes due to high mobility, leading to frequent link interruptions and shorter connection durations, the value of x decreases. Consequently, the value of α increases and approaches 1. A high α value prioritizes recent information and encourages exploration, which is desirable in such an environment.

The other form of exponential decreasing function-based adjustment is $1 - e^{-x}$. As x increases, the expression $1 - e^{-x}$ approaches 1, and as x decreases, $1 - e^{-x}$ tends to 0. This form has been employed in several works, including QMR [4], QRIFC [9], and QTAR [13], where x represents the delay.

In QMR [4], Liu et al. introduced an adjusted α that measures the quality of each link based on one-hop delay. The adaptive α corresponding to a link (i, j) is expressed in (2).

$$\alpha_{i,j} = \begin{cases} 1 - e^{-\varepsilon_{i,j}} & \sigma_{i,j} \neq 0 \\ 0.3 & \sigma_{i,j} = 0 \end{cases} \tag{2}$$

Where $\varepsilon_{i,j}$ is the normalized one-hop delay from node i to node j.

A link with a low one-hop delay is considered relatively stable. However, links with significant delays are considered unstable so they require rapid updates. To address this, the expression of α assigns a higher value based on the increased delay.

In QRIFC [9] , α is introduced through (3). m_{ij} and σ_{Uij} are respectively the mean and variance of the $PTT_{U_{ij}}$. $PTT_{U_{ij}}$ is the packet travel time from node i to node j.

$$\alpha_{U_{i,j}} = \begin{cases} 1 - e^{-\frac{\left\| PTT_{U_{ij}} - m_{ij} \right\|}{\sigma_{U_{ij}}}} & \sigma_{U_{ij}} \neq 0 \\ 0.3 & \sigma_{U_{ij}} = 0 \end{cases} \tag{3}$$

In this context, x represents the normalized packet travel time, which reaches a minimal value when the link is stable. In such cases, α approaches 0. However, as x increases, it indicates a dynamic environment with unstable links, so we need more exploration. Consequently, α increases and approaches 1.

In QTAR [13], α is introduced by (4). x denoting $\xi_{U_{i,f\to m}}$, which is the two-hop normalized delay.

$$\alpha_{U_{i,f\to m}} = \begin{cases} 1 - e^{-\xi_{U_{i,f\to m}}}, & \varphi_{U_{i,f\to m}} \neq 0 \\ 0.3, & \varphi_{U_{i,f\to m}} = 0. \end{cases} \tag{4}$$

Similar to QMR and QRIFC, x decreases in a dynamic environment and increases in the case of a static or less dynamic environment.

On the other hand, few works propose using the linear function-based adjustment method, such as SAIQL [14]. The adaptive learning rate is denoted by η_2 and its expression takes the form of ax, where $a = \eta \cdot k$ and $x = \frac{T_{\text{est}}}{T_{\max}}$, shown in (5).

$$\eta_2 = \frac{T_{\text{est}}}{T_{\max}} \cdot \eta \cdot k \tag{5}$$

η and k refer respectively to a basic fixed learning rate and a predefined parameter to be tuned by the experiments for optimal performance. T_{est} is the estimate of the average delivery time and $T_{\max}$ is the estimate of the maximum average delivery time. When the average delivery time increases, it indicates that the link stability declines. However, as x increases, link stability decreases. Consequently, η_2 increases to prioritize exploration and new information.

In summary, we conclude that the most common form for the learning rate follows a decreasing exponential pattern. It decreases because it is considered that learning is more focused on exploration in the earlier stages of the routing decision process, making the learning rate intuitively more critical at these stages. Additionally, many formulas that calculate α are related to the packet transmission delay to a neighbor node. Some formulas also make adjustments based on the predicted residual link duration. In practice, high mobility can lead to sudden disconnections between a node and its neighbors, resulting in short link durations. In such cases, when the connection duration is short, the exponential function yields a value close to 1, whereas it approaches 0 for significant connection durations.

4.2 Dynamic discount factor adjustment

The faster the network topology changes, the smaller discount factor γ should be to reflect unstable future expectations. Most of works have mainly focused on using the linear function-based adjustment.

In QMR [4], Liu et al. have introduced an adjusted discount factor as shown in (6).

$$\gamma_i = 1 - \frac{|N_i(t-1) \bigcup N_i(t)| - |N_i(t-1) \bigcap N_i(t)|}{|N_i(t-1) \bigcup N_i(t)|} \tag{6}$$

here, $\gamma = 1 - x$ where $x = \frac{|N_i(t-1) \bigcup N_i(t)| - |N_i(t-1) \bigcap N_i(t)|}{|N_i(t-1) \bigcup N_i(t)|}$. A node i has two neighbor sets at different time steps: $N_i(t-1)$ and $N_i(t)$ at times $t-1$ and t respectively. The size of the intersection between $N_i(t-1)$ and $N_i(t)$ denotes the stable neighbors. An increase in this intersection indicates more stable links and a reduced change in the network topology. In this case, the value of x decreases and approaches 0. As the mobility of neighbors increases, the Q-values of these neighbors become unstable within adjacent time intervals. Consequently, their significance diminishes. To address this issue, it becomes necessary to minimize the value of γ. This is achieved by allowing the expression of γ to decrease based on the stability of the neighbors.

In QTAR [13] γ is introduced by (7). $\gamma = x$, where $x = \sqrt{\frac{N^l_{u,i}(t) \cap N^l_{u,i}(t-1)}{N^l_{u,i}(t) \cup N^l_{u,i}(t-1)}}$. $N^l_{u,i}(t)$ represents the current one-hop neighbor set of node U_i at time t, and $N^l_{u,i}(t-1)$ is the previous one-hop neighbor set of node U_i at time $(t-1)$.

$$\gamma_{U_i} = \begin{cases} \sqrt{\frac{N^l_{u,i}(t) \cap N^l_{u,i}(t-1)}{N^l_{u,i}(t) \cup N^l_{u,i}(t-1)}} & \text{if } N^l_{u,i}(t) \cup N^l_{u,i}(t-1) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Similar to QMR [4], the size of the intersection between $N^l_{u,i}(t)$ and $N^l_{u,i}(t-1)$ represents stable neighbors. As the size of this intersection increases, the value of x also increases, causing γ to approach 1. A large value of x close to 1 indicates stable links and fewer network topology changes. Conversely, in a dynamic environment characterized by rapid changes in network topology, the value of x decreases, leading γ to approach 0. This process reduces the significance of future Q-values due to the increasing fluctuations in network topology.

In QRIFC [9] γ is denoted $\lambda_{U_{i,j}}$ and it is introduced by (8).

$$\lambda_{U_{i,j}} = \begin{cases} 1 - \frac{\|R_r - d_{U_{ij}}\|}{R_r} & \text{if } 0 \leq d_{U_{ij}} \leq R_r \\ 1 - \frac{d_{U_{ij}}}{R_a} & \text{if } R_r \leq d_{U_{ij}} \leq R_a \end{cases} \tag{8}$$

$d_{U_{ij}}$ is the inter-UAV distance, R_r represents the repulsion range, R_a denotes the attraction range. These two regions form together the transmission range of the UAV. Therefore, in order to adhere to both safety distance requirements

and transmission range boundaries, it is necessary to ensure that the distance between UAVs remains within $R_r \leq d_{U_{ij}} \leq R_a$. $d_{U_{ij}} \leq R_a$ indicates the existence of a direct communication link between the two UAVs. Here, $\gamma = 1 - x$. The value of x alternates between two formulas based on the relative distance. In the first condition, when a neighboring UAV is within the repulsion zone, the risk of collision or interference increases. In such a situation, the link between the two UAVs is considered unstable. Therefore, $x = \frac{\|R_r - d_{U_{ij}}\|}{R_r}$. As the distance between UAVs increases, the UAV is directed toward the attraction zone, and the link between them becomes more stable. Consequently, the value of x decreases, and the value of γ increases, approaching 1. This means that as the UAVs move toward the attractive zone, the link becomes stable, and long-term rewards are prioritized. In the second condition, when the neighboring UAV is already within the attraction zone, there is a risk of losing the connection with the current UAV if the distance between them exceeds R_a. Therefore, the expression for x becomes $x = \frac{d_{U_{ij}}}{R_a}$. As mobility increases, the distance between the current UAV and its neighbor also increases. A larger distance corresponds to a higher value of x. Consequently, as x increases, the link stability decreases, so γ decreases and approaches 0. Conversely, when mobility decreases, the distance reduces, leading to a decrease x, followed by an increase in γ that approaches 1. This adjustment aims to assign more importance to long-term rewards in dynamic environments and reduce their importance in stable environments.

In PARROT [15] γ is introduced by (9) where $\gamma(j)$ is a variable discount factor, γ_0 is a basic constant discount factor, $\Phi_{LET}(i,j)$ represents an estimation of the Link Expiry Time (LET) between i and j which takes into account the results of the mobility prediction process. $\Phi_{Coh}(j)$ is a measure of the neighbor set coherence of message forwarder j based on the difference between two successive neighbor sets.

$$\gamma(j) = \gamma_0.\Phi_{LET}(i,j).\Phi_{Coh}(j) \tag{9}$$

Here, $\gamma = ax$ where, $a = \gamma_0$ and $x = \Phi_{LET}(i,j).\Phi_{Coh}(j)$. In a dynamic environment scenario, the neighbor set of a node becomes unstable, resulting in a decrease in $\Phi_{Coh}(j)$. As a consequence, both x and subsequently the value of γ decrease. Similarly, the high mobility causes a shorter link expiration time, leading to a decrease in $\Phi_{LET}(i,j)$, which, in turn, results in a decrease in x and subsequently the value of γ. In a stable environment, both the link expiration time and the stability of the neighbor set increase, leading to an increase in x and the value of γ, approaches 1.

In the category of exponential decreasing function-based adjustment, we find the TARRAQ protocol [12] where γ is introduced by (10).

$$\gamma_{i,j} = 1 - exp(-\hat{T}^{rl}_{j,k}) \tag{10}$$

γ takes the form of $1 - e^{-x}$, where $x = \hat{T}_{j,k}$. Here, $\hat{T}_{j,k}$ represents the residual link duration. As the node speed increases, the residual link duration between

node j and k could decrease and becomes shorter. This increases the probability of link interruption, which decrease the link stability. Consequently, a smaller value of γ is required. So, with a small value of x, links are unstable, thus γ decreases. Furthermore, the increase in x leads to an increase in link stability and the value of γ. This highlights the dependence of γ and link stability.

In the category of Grid search-based adjustment, we find the Q-GEO protocol [16], in which the authors updated γ using (11). d_{comm} is the communication range and $E[d_{i,j}]$ is the expected neighbor distance.

$$\gamma = \begin{cases} 0.6 & \text{, when } E[d_{i,j}] < d_{\text{comm}} \\ 0.4 & \text{, otherwise} \end{cases} \tag{11}$$

The value of γ alternates between only two values. If the expected neighbor distance is shorter than the communication range, this means that the connection with the neighbor is still valid and the link is stable, leading to an increase in γ to 0.6. Else, the value of γ decreases to 0.4. The idea behind setting the average value of γ at 0.6 is to facilitate the comparison of relative routing schemes [16].

In summary, using a decreasing exponential function for α allows for rapid adaptation when changes occur and slower adaptation when the network stabilizes. Using a linear function for γ means that it changes gradually with network conditions. This is because linear adjustments provide a more stable and predictable transition between prioritizing immediate rewards and future rewards. In dynamic networks, a linear adjustment allows for a continuous adaptation that reflects the changing nature of link stability and network topology without abrupt shifts. As a conclusion, the choice of using a decreasing exponential function for α and a linear one for γ is tailored to the specific requirements of network routing in dynamic environments. It strikes a balance between rapid adaptation to changes (for α) and maintaining a stable transition between short-term and long-term rewards (for γ), which are essential for efficient and effective routing in scenarios like FANETs.

5 Conclusion

Applying Q-Learning in FANETs routing protocols has shown both positive and negative outcomes. The positive aspect is that it does not require knowledge of the entire network to make routing decisions. It only needs information about the next-hop. Additionally, it offers a simple and low-computation cost approach. However, this algorithm suffers from slow convergence speed, mainly due to frequent topology changes and the large dimension of the neighboring table. To enhance the convergence speed, many techniques have been proposed, including filtering the neighboring table, adjusting the learning parameters and enhancing exploration-exploitation mechanisms, all while considering network conditions and environmental dynamism. Adjusting the learning parameters is one of the most interesting techniques. Many approaches have been introduced, so we classify them into three classes. Exponential function-based adjustment

represents the most commonly used category for adjusting the learning rate, whereas Linear function-based adjustment is the most commonly used category for adjusting the discount factor. There is also a third category known as Grid search-based adjustment, but its usage is limited because it relies on predefined values for adapting learning parameters, which may not effectively respond to varying degrees of topology changes. As a perspective, It would be interesting to investigate the impact of these different classes of adjustment on a specific mobility model used in FANETs.

References

[1] M. M. Alam and S. Moh, "Survey on q-learning-based position-aware routing protocols in flying ad hoc networks," *Electronics*, vol. 11, no. 7, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/7/1099

[2] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[4] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, "Qmr: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks," *Computer Communications*, vol. 150, pp. 304–316, 2020.

[5] L. A. L. da Costa, R. Kunst, and E. P. de Freitas, "Q-fanet: Improved q-learning based routing protocol for fanets," *Computer Networks*, vol. 198, p. 108379, 2021.

[6] M. F. Khan, K.-L. A. Yau, M. H. Ling, M. A. Imran, and Y.-W. Chong, "An intelligent cluster-based routing scheme in 5g flying ad hoc networks," *Applied Sciences*, vol. 12, no. 7, p. 3665, 2022.

[7] Y.-n. Chen, N.-q. Lyu, G.-h. Song, B.-w. Yang, and X.-h. Jiang, "A traffic-aware q-network enhanced routing protocol based on gpsr for unmanned aerial vehicle ad-hoc networks," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 9, pp. 1308–1320, 2020.

[8] J. Lansky, S. Ali, A. M. Rahmani, M. S. Yousefpoor, E. Yousefpoor, F. Khan, and M. Hosseinzadeh, "Reinforcement learning-based routing protocols in flying ad hoc networks (fanet): A review," *Mathematics*, vol. 10, no. 16, p. 3017, 2022.

[9] M. M. Alam and S. Moh, "Q-learning-based routing inspired by adaptive flocking control for collaborative unmanned aerial vehicle swarms," *Vehicular Communications*, vol. 40, p. 100572, 2023.

[10] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 136–142, 2018.

[11] M. Zhang, C. Dong, S. Feng, X. Guan, H. Chen, and Q. Wu, "Adaptive 3d routing protocol for flying ad hoc networks based on prediction-driven q-learning," *China Communications*, vol. 19, no. 5, pp. 302–317, 2022.

[12] Y. Cui, Q. Zhang, Z. Feng, Z. Wei, C. Shi, and H. Yang, "Topology-aware resilient routing protocol for fanets: An adaptive q-learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 632–18 649, 2022.

[13] M. Y. Arafat and S. Moh, "A q-learning-based topology-aware routing protocol for flying ad hoc networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1985–2000, 2021.

[14] A. Rovira-Sugranes, F. Afghah, J. Qu, and A. Razi, "Fully-echoed q-routing with simulated annealing inference for flying adhoc networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2223–2234, 2021.

[15] B. Sliwa, C. Schüler, M. Patchou, and C. Wietfeld, "Parrot: Predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–7.

[16] W.-S. Jung, J. Yim, and Y.-B. Ko, "Qgeo: Q-learning-based geographic ad hoc routing protocol for unmanned robotic networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2258–2261, 2017.