Computer Networks 198 (2021) 108379

Contents lists available at ScienceDirect

Computer Networks

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

# Q-FANET: Improved Q-learning based routing protocol for FANETs

Luis Antonio L.F. da Costa [a,*], Rafael Kunst [b], Edison Pignaton de Freitas [a]

[a] Federal University of Rio Grande do Sul, Brazil
[b] University of Vale do Rio dos Sinos (UNISINOS), Brazil

ARTICLE INFO

*Keywords:*
Flying Ad-Hoc Networks
Routing protocols
Reinforcement learning
Network performance evaluation

ABSTRACT

Flying Ad-Hoc Networks (FANETs) introduce ad-hoc networking into the context of flying nodes, allowing real-time communication between these nodes and ground control stations. Due to the nature of this kind of node, the structure of a FANET is dynamic, changing very often. Since it has applications in military scenarios and other mission-critical systems, an agile and reliable network is essential with robust and efficient routing protocols. Nonetheless, maintaining an acceptable network delay generated by the selection of routes remains a considerable challenge, owing to the nodes' high mobility. This article addresses this problem by proposing a routing scheme based on an improved Q-Learning algorithm to reduce network delay in scenarios with high-mobility, called Q-FANET. This proposal has its performance evaluated and compared with other state-of-the-art methods using the WSNET simulator. The experiments provide evidence that the Q-FANET presents lower delay, a minor increase in packet delivery ratio, and significant lower jitter compared with other reinforcement learning-based routing protocols.

## 1. Introduction

The technological advances in the last decades, especially the development and miniaturization of electronic components, lead to the popularization and the decrease in production costs of Unmanned Air Vehicles (UAV) [1]. Consequently, UAVs became applied in many different military and civilian domains, such as surveillance [2] and monitoring tasks [3], provision of communications networks in natural disasters [4] and conflict regions [5], or as general purpose aerial data collectors [6].

The use of a single UAV is already well understood and even considered ordinary. However, the use of multiple simultaneous UAVs, which can provide a significant advantage over the option of a single UAV, is still a research area with many possibilities. Despite its usefulness, these multiple UAV setup scenarios pose a challenge regarding communication, which is not a trivial task [7]. It is necessary to exchange packets between UAVs and base station(s) in situations with unique characteristics such as nodes that are continually moving in regions with artificial and natural obstacles, in addition to the most varied types of climate conditions in which they operate. Dealing with these difficulties demand the proposal of robust routing protocols, which is critical for the deployment of high level networked services, such as [8].

FANETs (Flying Ad-Hoc Networks) [9], composed of many UAVs, the nodes' high mobility creates a highly dynamic network topology [10]. Such scenario demands n adaptive and autonomous protocol to address this issue, meaning that the protocol for routing in FANETs should be able to discover a stable neighbor to send the data by detecting changes in the environment. In this context, Q-learning is an adaptive reinforcement learning technique that receives input feedback from the environment – contributing to provide a routing design focused on adaptation – and presents a promising approach for a routing protocol scheme [11].

The premise of routing protocols that are based in Q-learning usually takes into account the data provided from the neighboring nodes, not making any assumption about any other network aspect. Most of them work by making the most suitable decision among the neighbors to forward a packet until it reaches the destination. Since multiple UAVs systems require real-time data transmission, a routing protocol must have a low delay to support several applications.

FANETs are highly dynamic and, if parameters from Q-learning, such as the learning rate and the discount factor, are fixed, the efficiency of the selection of the best action declines, making the selected link present a minimum possibility of establishing a connection to a neighbor node. This strategy applies to the majority of the known routing protocols that are based on Q-Learning and may limit their performance. Based on these constraints, this paper proposes a novel Q-Learning based routing protocol called Q-FANET, which addresses the mentioned limitations and combines positive features of existing approaches that use reinforcement learning to create an optimized

* Corresponding author.
*E-mail addresses:* lalfcosta@inf.ufrgs.br (L.A.L.F. da Costa), rafaelkunst@unisinos.br (R. Kunst), edison.pignaton@inf.ufrgs.br (E. Pignaton de Freitas).

https://doi.org/10.1016/j.comnet.2021.108379
Received 21 September 2020; Received in revised form 20 July 2021; Accepted 1 August 2021
Available online 8 August 2021
1389-1286/© 2021 Elsevier B.V. All rights reserved.

routing protocol able to address the tough requirements presented by FANETs. The main contributions reported in this work are:

- **Delay and Jitter decrease:** Without a fixed routing table, Q-Learning can be used with specific rules and mechanisms to choose the optimal routing path based on a low delay constraint;
- **Exploration of last episodes with different weights:** Standard Q-Learning approaches always consider the most recent last episode to update the Q-Values what may lead to imprecise decisions. Therefore, the proposed solution considers a finite amount of last episodes;
- **Enhanced protocol parametrization based on channel conditions:** The transmission quality is an important element that can directly impact the delay of data transmission in a FANET, even when the optimal route is selected. The proposed solution also considers the channel conditions as a new metric to calculate the Q-Values parameter of the proposed approach.

The remainder of this paper is organized as follows. Section 2 reviews essential background concepts on routing protocols for FANETs, as well as the characteristics of Reinforcement Machine Learning and the relevant literature in this area. The proposed Q-FANET architecture is presented in Section 3. The simulation scenario, performed experiments, and obtained results are presented and discussed in 4. Finally, concluding the paper, Section 5 presents final remarks and directions for future investigations.

## 2. Background and related works

This section presents background aspects on the routing protocols for FANETs and the Reinforcement Learning paradigm, including the favored technique of Q-Learning. The section also discusses relevant related work in the area.

### 2.1. Routing protocols for FANETs

The existing routing protocols used in Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs) are not entirely suitable to be directly applied in UAVs networks, as they must adapt to the higher degree of mobility that characterizes FANETs and the consequently more frequent changes in the topology [12]. According to the literature, one can organize the routing protocols used in FANETs into two categories: single-hop routing [13] and multi-hop routing [14].

For the single-hop routing protocols, a static routing table defines the transmission paths, being computed and loaded before the start of the UAV nodes' operation and cannot be changed. In the multi-hop routing protocols, packets are forwarded hop by hop towards the destination. The selection of the proper hop node is the core issue of the route discovery. Usually, one can classify these protocols into two categories: topology-based and position-based routing [15]. Furthermore, the first category consists of three specific types of protocols: proactive protocols, reactive protocols, and hybrid protocols.

#### 2.1.1. Static protocols

Lightweight and designed for fixed topologies, these protocols are not fault-tolerant, since, in case of failure, it is mandatory to wait until the end of the operation to update the routing table, which makes them not suitable for dynamic environments. Examples of these protocols are **Load-carry-and-deliver (LCAD)** [16] and **Data Centric Routing (DCR)** [17].

#### 2.1.2. Proactive protocols

These protocols record and store the routing information in each UAV belonging to the network, with each node updating its routing table to meet changes in the network topology. Therefore, the routing paths can be chosen to send packets with minimum waiting time [18]. Although highly used due to its characteristics of serving high-mobility network scenarios, this type of protocol presents several disadvantages, such as the number of control packets necessary for the route establishment, increasing communication overhead. Examples of proactive protocols include **Destination Sequenced Distance Vector (DSDV)** [19] and **Optimized Link State Routing Protocol (OLSR)** [20].

#### 2.1.3. Reactive protocols

This class of routing protocols presents low overhead since they create routing information only when there is a communication between two nodes. However, the overhead reduction comes at the cost of increasing the end-to-end delay, due to the processing time required to establish a path [21]. Examples of reactive protocols include **Dynamic Source Routing (DSR)** [22] and **Ad-hoc On-demand Distance Vector (AODV)** [23].

#### 2.1.4. Hybrid protocols

Representing a combination of proactive and reactive routing protocols, these protocols are used to overcome the limitations of each isolate approach, i.e., time demanded to find routes and control messages overhead [24]. Examples of hybrid protocols include **Zone Routing Protocol (ZRP)** and **Temporarily Ordered Routing Algorithm (TORA)** [25].

#### 2.1.5. Position-based protocols

They overcome the limitations of proactive and reactive protocols, specifically with the static routing tables and the establishment of the route before the transmission of each packet, correspondingly [26]. Examples of position-based protocols include **Greedy Perimeter Stateless Routing (GPSR)** [27].

#### 2.1.6. Hierarchical protocols

The last class of routing protocols explores cluster-based approaches to perform route discovery. Examples of hierarchical protocols include **Mobility prediction clustering (MPC)** [28] and **Clustering Algorithm of UAV Networking** [29].

### 2.2. Reinforcement learning

Reinforcement Learning (RL) is another crucial paradigm of the learning process in Artificial Intelligence [30]. A simple analogy that is possible to imagine is a person that does not know the flavor of specific food and tries it for the first time. This individual may identify the food as something good or bad, and this acquired knowledge may apply to decide next time if this individual should eat or do not eat that food. In the context of Computer Science, RL applies to algorithms with some knowledge about the task that they should perform and can use it to make better choices to complete the task. As shown in [31], ML can deal with several challenges involving the communication in FANETs, as well as improving different design and functional aspects such as channel modeling, resource management, positioning, and security. The main components of an RL algorithm are the agent, the environment, the state, the action, and the reward.

The agent learns over time to behave in an optimized manner in an environment by interacting continuously with it. During its learning course, the agent experiences various scenarios in the environment, which are called states. While in a particular state, the agent may choose from a set of allowable actions and, depending on the result of each action, it receives rewards or penalties. Overtime, the agent learns

ways to increase these rewards with the goal of behaving optimally at any given state.

Q-learning is a basic form of RL that makes use of Q-values to improve the learning agent's behavior iteratively. The algorithm defines values for states and actions. $Q(S, A)$ represents an estimation of how good it is to perform action $A$ at a given state $S$. In Q-Learning, an agent starts from a given state and performs several transitions from its current state to the next one. Every transition happens due to an action considering the environment with which the agent is interacting. In each step of the transition, the agent performs an action, receives a reward, then transitions to another state until it reaches the goal. This situation is called the completion of an episode. The algorithm estimates $Q(S, A)$ using a specific rule that calculates the value of Q at every interaction of an agent with the environment, expressed by (1):

$$Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma max Q(S', A') - Q(S, A)) \quad (1)$$

$S$ represents the current state of the agent, $A$ is the current action chosen in accordance to a specific policy, $S'$ represents the next state that the agent is going to transition, $A'$ is the next best action to pick using the current Q-value estimation, and $R$ is the reward received from the current action. Other important parameters of this update function are:

- **Discounting factor for future rewards** ($\gamma$): a value set between 0 and 1. Common Q-learning approaches consider future rewards less valuable than current ones, therefore they must be discounted;
- **Learning rate** ($\alpha$): step length taken to perform the update of the estimation $Q(S, A)$;
- $\epsilon$-**greedy policy:** a simple method to select actions using the current Q-value estimations. The probability of selecting the action with the highest Q-value is given by (1-$\epsilon$) while selecting a random action is ($\epsilon$).

### 2.3. Related work

Routing protocols based on Q-Learning methods are promising to deal with the dynamic changes in FANETs. Q-Grid [32] is a protocol designed for VANETs that makes use of macroscopic (optimal next-hop grid by querying Q-value table learned offline) and microscopic aspects (specific vehicle in the optimal next-hop grid) to perform the routing decision, dividing the region into different grids. With this approach, Q-Grid calculates the Q-values of various movements between neighboring grids for a specific destination using the Q-learning algorithm. The performed simulations have shown that Q-Grid presents advantages compared to other existing position-based routing protocols.

Q-learning based Adaptive Routing model (Q-LAR) [33] detects the level of mobility at each node of the network and proposes a metric, entitled Q metric, which accounts for both the static and dynamic routing metrics to respond to topology changes. Simulation results show that Q-LAR is more effective than the standard OLSR protocol. Q-Learning based geographic routing (Q-Geo) [34] proposes a system to minimize the network overhead in high mobility scenarios. The authors compare the performance of Q-Geo with other approaches utilizing the NS-3 simulator, with the results showing that Q-Geo presents a higher packet delivery ratio and also a lower network overhead than the compared solutions.

Q-Fuzzy [35] uses fuzzy logic considering parameters as transmission rate, energy state, and flight status between neighbor UAVs to determine the optimal routing path to the destination. The algorithm updates these parameters dynamically using an RL method. The results show that, compared with distance vector routing based on Q-Values, Q-Fuzzy can maintain low hop count and energy consumption and extend the network lifetime. Q-Learning Multi-Objective Routing (QMR) [36] is a routing protocol that uses adaptive parameters (as the learning rate and the mechanism of exploration) combined with link conditions and specific constraints to provide low delay and low energy consumption. The performed simulations compared QMR with Q-Geo, and showed that the proposed routing scheme presents a higher packet arrival ratio, lower delay, and energy consumption.

In the context of MANETs, [37] propose a Q-learning based CSMA/MAS protocol. In this method, every node in the network is able to be synchronized and then attend in a round-robin way without have to deal with contention collisions. At the network layer, the approach performs several modifications to Q-Geo and Q-Grid. The results have shown that this transmission protocol approach provides a higher packet arrival ratio and lower end-to-end delay than the existing transmission protocols. QNGPSR [38] is a routing protocol which is inspired on the GPSR protocol for the ad-hoc network. It aims at reducing the network delay by using reinforcement learning to perform the next-hop selection. Results show that QNGPSR provides a higher packet delivery ratio and a lower end-to-end delay when compared to the performance of GPSR. In the context of cognitive sensor networks, Q-Noise+ [39] proposes three improvements to algorithms which are based on reinforcement learning and used used for dynamic spectrum allocation. Simulation results show that Q-Noise+ allows better quality in the allocation of channels (up to 6 dB), and also presents 4% higher efficiency compared to the standard Q-Learning.

Table 1 presents a summary of the main aspects and gaps of the works found in the literature, highlighting the contribution of this present one. The proposal here presented considers the techniques behind QMR and Q-Noise+ providing an improved solution with optimal channel selection to provide low delay and low jitter in the challenging environment of UAV-networks, using a finite number of last episodes to calculate the Q-Values.

## 3. Q-FANET protocol

This section introduces and describes Q-FANET, an improved Q-learning based routing protocol for FANETs. This solution takes into account two Q-Learning methods: Q-Noise+ and QMR. In Q-FANET, nodes use a reinforcement learning algorithm without knowing the entire network topology to perform optimal routing decisions focusing in delivering low-delay service.

### 3.1. Q-FANET design overview

Q-FANET proposes changes and improvements in several components of QMR and other techniques suitable with Q-Learning to propose a novel approach to deal with FANET routing. The proposal consists of two major modules, the Neighbor Discovery and the Routing Decision. Fig. 1 presents an overview of Q-FANET's architecture. The elements $C1$ and $C2$ are connectors that serve only as visual tools, not representing any important component of the algorithm's architecture.

An important assumption is that this proposal considers FANETs composed of low to medium speed UAVs, such as multi-rotors and aerostats [40,41], flying with speeds under 20 m/s, such as the 3DR Iris+ quadcopter, a mini-UAV manufactured by 3-D Robotics [42]. Considering these types of UAVs, it is reasonable to assume that the proposed RL algorithm successfully converges to useful solutions.

### 3.2. Routing neighbor discovery

The Neighboring Discovery module is a control structure used to maintain the routing information updated. Q-FANET updates the location of the nodes regularly. The updating frequency is one of the parameters of the proposal, and its default value is 100 ms. In cases where a node does not inform its location within a specific expiration time of 300 ms, its neighbors remove the specific route from their routing tables.

This updating process in Q-FANET relies on the exchange of *HELLO* packets. In this case, a given network node broadcasts these packets

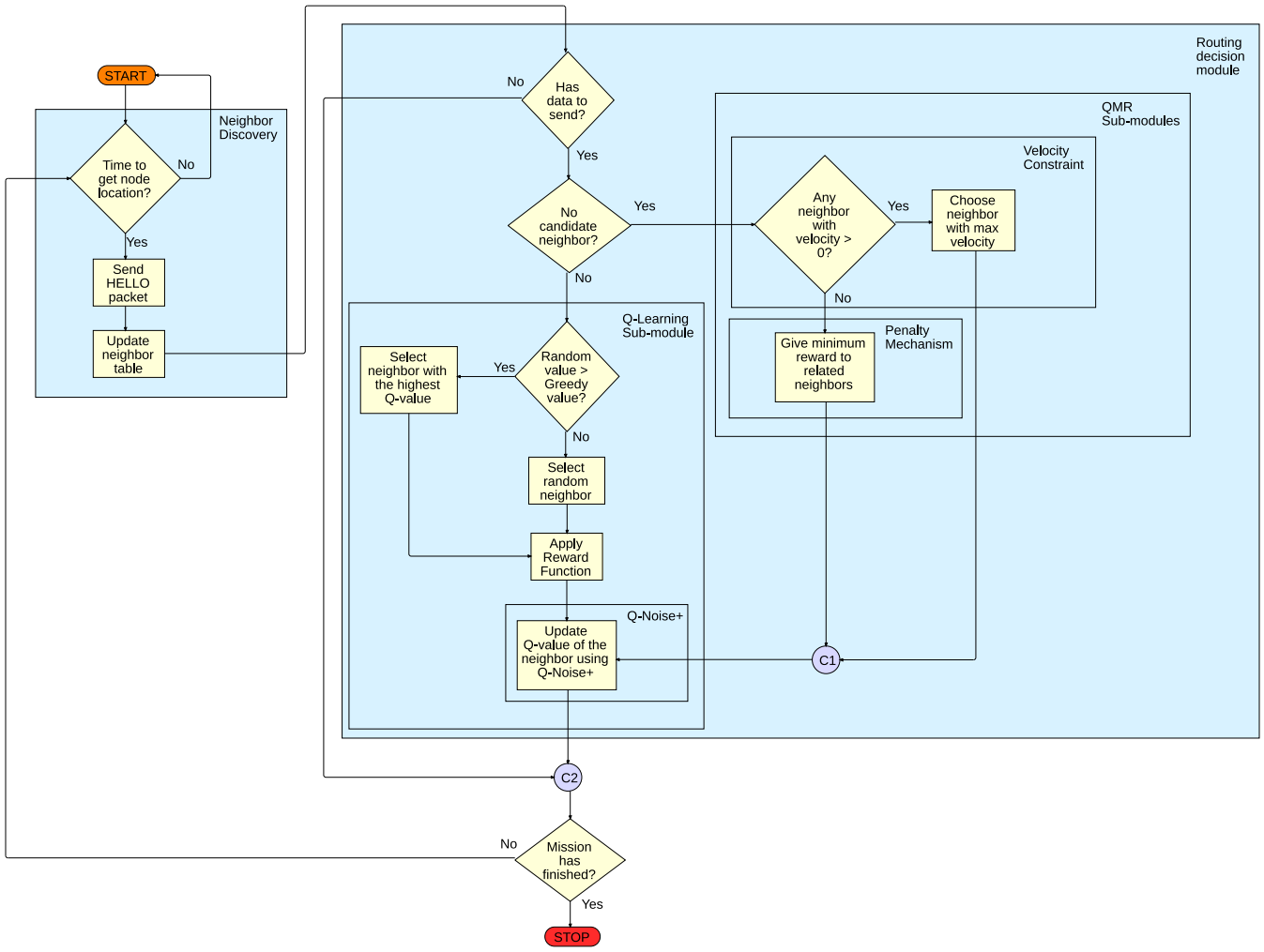



**Fig. 1.** Q-FANET flowchart exposing its internal modules.

aiming to discover its neighbors. This packet exchange happens periodically and carries the following information: node's geographic location, energy level, mobility model, queuing delay, learning rate, and Q-value. When a node receives *HELLO* packets, the node will use the packet information to establish and maintain its neighbor table.

The idea behind this module is to keep the network ready for transmission at any time. Therefore, its functionalities are always running, despite the existence of a current transmission session. Whenever a transmission is necessary, the Neighbor Discovery module communicates with the Routing Decision, the other module of Q-FANET, to provide information regarding the best routes.

### 3.3. Routing decision module

The Routing Decision Model receives information about the available routes and selects the one for a given node to transmit data. To do that, it counts with two sub-modules: (I) QMR and (II) Q-Learning that consider the background of existing approaches but modify them to improve the routing response. The flow of the model will proceed to these two sub-modules in a scenario where the list of possible neighbor candidates is not empty. This situation corresponds to the nonexistence of the routing hole problem [43], i.e, all the neighbors of a node are distant than the distance from this node to the destination.

#### 3.3.1. Q-learning sub-module

The standard Q-Learning algorithm applies a reward-based approach that considers two main aspects: (I) the successful transmission rate in the last episode and (II) the sum of the success rates of all past episodes. However, except for the most recent episode, the remaining episodes generally possess the same weight in the decision process. The problem is that this approach may lead to imprecise decisions, especially when considering scenarios where a high number of episodes is necessary, such as in military, surveillance, and rescue missions, where the dynamism of the situation implies high mobility of the nodes.

Taking the mobility problem into consideration, an extension to the Q-Learning, called Q-Learning+, was proposed by [39]. This extension of the algorithm considers a finite amount of past episodes, defined by a lookback value ($l$). In this approach, the newer the episode, the higher its weight will be. Consequently, the Q-value at time $t+1$ is obtained as shown in (2), where $w_i$ states for the weight of the last $l$ instants of time and $r_i$ represents the calculated reward based on $l+1$ actions. Also, Q-Learning+ evaluates each of its actions based on the total sum of all of its future rewards, therefore the discount factor ($\gamma$) is set to 1 and is omitted from the equation, since its usage does not change the calculated Q-value.

$$Q_{t+1}(a_t) = (1-\alpha)\sum_{i=1}^{l}[w_{t-i}r_{t-i}](a_t) + \alpha r_t(a_t) \tag{2}$$

Although Q-Learning+ improves the original approach's efficiency, it only considers the number of successful transmissions to perform its decision, ignoring the channel's propagation conditions. In order to also consider the channel conditions to perform routing-related decisions, another algorithm called Q-Noise+ is available. This algorithm takes

**Table 1**
Related work summary.

| Ref. | Main approach | Discussed research gap |
|---|---|---|
| [32] | Use of macroscopic and microscopic aspects to perform the routing decision, dividing the region into different grids | When the time slot is long, the prediction of Q-Grid by Markov chain may be inaccurate |
| [33] | Q-Learning technique that detects the level of mobility of each node and uses a special metric to account for the static and dynamic route metrics | The algorithm is too much dependent of the metric and could use different approaches, such as energy-aware metrics |
| [34] | A machine-learning-based geographic routing scheme to reduce network overhead in high-mobility scenarios. | Underutilization of network resources and capacity |
| [35] | Q-learning based fuzzy logic in a multi-objective routing algorithm using link and path-related parameters | Method performance is not outstanding when the number of nodes is small because of power consumption |
| [36] | QMR is able to provide low-delay and low-energy service guarantees by adaptively adjusting Q-Learning parameters | No implementation and testing in physical UAV networks |
| [37] | The cross-layer protocol for MANETs uses CSMA/MAS and modified Q-Learning based routing | The experiments do not take into account node failures or the routing hole problem |
| [38] | A Q-Learning network enhanced GPSR routing protocol that reduces end-to-end delay and packet delivery ratio | The method is not optimized for feature extraction and does not take into account the link status |
| [39] | It improves the Q-Learning method for routing protocols in CR sensor networks by considering the channel conditions and the SINR of the channels | It lacks analysis of the trade-off between more precise decisions and network overhead |
| [This work] | A Q-Learning mechanism that combines features of existing approaches with channel conditions analysis | **(Covered Gap)** Combination of QMR and Q-Noise+ features into an approach that provides lower end-to-end delay, jitter and higher packet delivery ratio than other methods in UAV dynamic networks |

into account the quality of the transmission as a secondary metric, calculated considering the Signal-to-Interference-plus-Noise Ratio (SINR) measured in a given channel. This approach tries to avoid selecting an available channel that can be noisy — a situation that might occur when using the Q-Learning+ approach.

Two aspects are considered for the decision taken by Q-Noise+: (I) the learning rate considering the reward obtained in an episode $T$ and (II) a quality status that considers both the SINR level of the channel. These values are parameters to calculate a weighted reward for the most recent episode. Eq. (3) presents details regarding this calculation.

$$Q_{t+1}(a_t) = (1-\alpha)\sum_{i=1}^{l}[w_{t-i}r_{t-i}](a_t) + \alpha r_t(a_t) + (S_w * \eta) \tag{3}$$

In (3), $S_w$ ($0 \leq S_w \leq 1$) represents the weight of the SINR in the calculated reward. $\eta$ corresponds to SINR ranges (see Table I [44]). This weight is set as a parameter that defines the importance given to the quality of transmission. This means that, by increasing the value of $S_w$, it also increases the SINR impact on the Q-value of the channel. On the other hand, the $\eta$ parameters defines a set of SINR values that have been chosen to change the Q-value according to the channel conditions.

**Table 2**
SINR ranges.

| SINR value | $\eta$ |
|---|---|
| $SINR < 15$ dB | 0 |
| 15 dB $\leq SINR <$ 17 dB | 0.25 |
| 17 dB $\leq SINR <$ 20 dB | 0.5 |
| 20 dB $\leq SINR <$ 25 dB | 0.75 |
| $SINR \geq 25$ dB | 1 |

It is expected that, when in a scenario with favorable propagation conditions, the $\eta$ value will be higher, which will increase the Q-value of the channel. Consequently, as the channel conditions become worse, $\eta$ decreases, not changing the Q-value.

Q-FANET also makes use of an $\epsilon$-greedy policy for exploration and exploitation [45]. The exploration consists in searching for unknown actions (i.e. obtain new knowledge). Nevertheless, exploration in excess makes it complicated to maintain some better actions. On the other hand, the exploitation aims to create advantages by exploring actions, which have a chance to generate high rewards. Although, if exploitation is used in excess, it may become difficult to choose some undiscovered potential actions that might be optimal.

With the goal of keeping the trade-off between exploration and exploitation well balanced, the $\epsilon$-greedy policy instructs the Q-Learning Sub-module to explore by choosing a random path with probability $\epsilon$ (usually 10%) and exploit by choosing the option which offers the highest Q-Value.

Q-FANET benefits from these approaches as building blocks of the Q-Learning Sub-module. One crucial adaptation proposed in Q-FANET regards the reward function, which is discussed in the next section.

### 3.3.2. Reward function

In Q-FANET, a data structure called R-Table (Reward Table) is proposed to store reward cells. The initial value of the reward cell values is zero. After each forwarded data from node $i$ to node $j$, the R-table values are updated according to the logic expressed in (4):

$$R(s,a) = \begin{cases} r_{max} = 100, \text{if link}(i,j) \text{ leads to destination} \\ r_{min} = -100, \text{if link}(i,j) \text{ is local minimum} \\ r = 50, \text{otherwise} \end{cases} \tag{4}$$

where $s$ represents a state, and $a$ represents an action taken by the agent, which in the network scenario, corresponds to a node. Therefore, when a packet is at node $i$ the current state associated with this packet is $s_i$. An action $a_{i,j}$ represents the forwarding of the packet from node $i$ to neighbor node $j$, using link $(i,j)$, changing the state $s_i$ to $s_j$. For each change of state, the reward function $R(s,a)$ is applied.

The maximum reward value $r_{max}$ will be applied to a link $(i,j)$ when the next-hop $j$ is the destination node. On the other hand, the minimum reward value $r_{min}$ will be used when the node $i$ is defined as a local minimum, meaning that all its neighbors are set farther away from the destination than itself. In any other situation, Q-FANET provides a reward of 50. For example, this situation occurs when node $j$ is a relay node in the path to the destination.

This function structure is based on the binary reward function approach [46] and the values for the rewards were empirically chosen taking into account that they must clearly represent a difference between the maximum and the minimum reward, i.e., when the hop is leads to the destination node, to a local minimum or it is a general link.

### 3.3.3. QMR sub-module

The QMR sub-module is responsible for the penalty mechanism and it controls the constraint regarding the nodes' velocity to support the best decision. This sub-module is going to be used in scenarios where

the list of possible candidate neighbors is empty, i.e., a routing hole problem happened in the network.

*Penalty Mechanism* :

The occurrence of routing holes increases the delay of transmitting a data packet. Q-FANET proposes a modification to the penalty mechanism of QMR, aiming to reduce the existence of routing holes. This mechanism applies to the following cases:

- **Routing hole:** when a node $j$ discovers that all of its neighbors are further than itself from the destination, then it sends a feedback to the previous node $i$.
- **Not-ACK:** when a node $i$ does not gets an ACK packet from next-hop node $j$, meaning that node $j$ may be in a failure state.

In both scenarios, the action taken by the penalty mechanism will be that node $i$ will give the $r_m in$ for the link $i, j$ and update the corresponding Q-Value of the link. Nevertheless, even if there is a case where both a routing hole and a Not-ACK happen in the network, the penalty mechanism will have the same behavior it would have in each separated case.

*Velocity Constraint* :

A velocity constraint is necessary to obtain the minimum delay between the hops. Q-FANET adapts this constraint by simplifying the one defined in QMR. In Q-FANET, the velocity constraint of link $i, j$ is defined in (5), observing (6):

$$Velocity_{(i,j)} = \frac{d_{(i,D)} - d_{(j,D)}}{delay_{(i,j)}} \tag{5}$$

$$\begin{cases} Velocity_{(i,j)} < 0, \text{ if } d_{(j,D)} > d_{(i,D)} \\ Velocity_{(i,j)} > 0, \text{ if } d_{(j,D)} < d_{(i,D)} \end{cases} \tag{6}$$

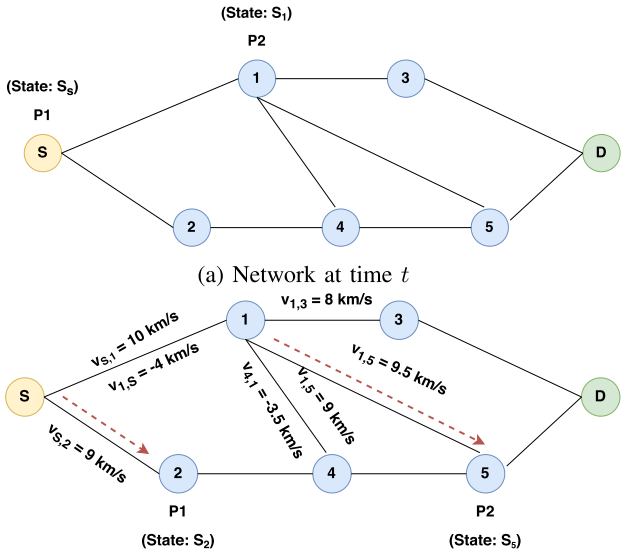
where $d$ represents the distance in meters, and $D$ stands for the final destination node. Therefore $d_{(i,D)}$ describes the distance between nodes $i$ and $D$, and $d_{(j,D)}$ represents the distance between nodes $j$ and $D$.

Also, (5) and (6) show that higher delays will lead to lower velocity constraint values. Moreover, velocities below zero indicate that the distance between node $j$ and the destination is bigger than the one from $i$ to the destination. Q-FANET will obtain this velocity information during the routing neighbor discovery process, where each node will create its routing table.

Since Q-FANET is a reinforcement learning algorithm, one of the concerns regards the possibility of the algorithm not converging. We deal with this potential issue by prioritizing that transmission occurs instead of guaranteeing an optimized link selection. The implementation of this feature occurs within the "Routing Neighbor Discovery" function, which keeps an updated list of nodes' locations within the network. This module works along with the "Route Decision" function in the transmission route selection. "Route Decision" implements a timer to act as a deadline for Q-FANET to converge. Although this timer is a parameter of Q-FANET, in our simulations, we consider a maximum convergence delay of 10 s. We set this value considering the UAVs' velocity and their impact on the network topology. In an unlikely situation in which Q-FANET does not converge, the route decision module selects the last successful calculated route. If there is no successful route available, the algorithm deals with the transmission considering that it is the first transmission, i.e., Q-FANET randomly selects a route to start the reinforcement learning process.

### 3.4. Q-FANET working example

Fig. 2 shows a simple network topology as an example scenario for the application of the proposed Q-FANET. In this network there are the source node (S), destination node (D) and relay nodes (1, 2, 3, 4, 5). In the scenario represented in 2a, it is assumed that at a current time t, there are two data packets in the network, i.e., two agents. Packet 1 (P1) and Packet 2 (P2) are in nodes S and 1, and their states are $S_S$ and $S_1$, respectively. Also, it is possible to determine that the set of neighbors of node S is node 1, node 2 and of node 1 is node S, node 3, node 4, node 5. Node S and node 4 need to select one of their neighbors as the next hop to forward the packet until it reaches the destination node.




(a) Network at time $t$

(b) Network at time $t + 1$

**Fig. 2.** Network topology example.

By using the Velocity Constraint sub-module from QMR, it is possible to determine the velocity of the links described in 2b. Since the velocity of the candidate neighbors must be greater than zero, the set of candidate neighbors is then obtained. Therefore, the set of neighbors of node S and node 1 are node 1, node 2 and node 3, node 4, node 5. Suppose that at time $t$, the Q-value using only the Q-learning+ approach (without considering the channel conditions), SINR and the Q-value when using the Q-Noise+ approach for the links in the network are shown in Table 3. It is important to note that the SINR information was obtained in a time $t-1$, since channel conditions may vary through time.

For Packet 1 (agent 1), although the Q-value of the link (S,1) is bigger than the Q-learning+ Value of the link (S,2), the link (S,2) presented a greater SINR than the link (S,1), therefore the updated Q-value by Q-Noise+ of the link (S,2) will be higher than the value of the link (S,1) and the link (S,2) will be selected to forward the packet. Also, the Q-learning+ Value of the link (S,2) is equal to the initial value of 0.5, which implies that now Packet 1 (agent 1) will explore a new link, meaning that previously undiscovered links might be explored.

For Packet 2 (agent 2), the link (1,5) presents the higher Q-learning+ Value between the possible forwarding neighbor candidates. Even so, notice that this link had at time $t-1$ the highest SINR value amongst all possible links and it was the best channel to forward the packet. Therefore, the updated Q-value by Q-Noise+ of the link (1,5) shows that this is the most suitable link to forward Packet 2. Also, the link (1,5) was the best forwarding link in the past (even considering the channel conditions, with its high SINR). Since Packet 1 (agent 1) is choosing the best link in the past to forward data, Q-FANET is exploiting the knowledge that has been previously learned.

Q-FANET contributions include implementing reinforcement learning in a challenging mobile environment. Even though this work considers UAVs that do not reach very high speeds, such as multi-rotors and aerostats, the mobility raises several challenges. For example, it is important to deal with the convergence uncertainty problem. Dealing with this problem demanded the proposal of a non-convergence detection

**Table 3**
Network link information.

| Link | (S,1) | (S,2) | (1,3) | (1,4) | (1,5) |
|---|---|---|---|---|---|
| Q-learning+ value | 0.63 | 0.5 | 0.52 | 0.61 | 0.7 |
| SINR | 16.2 dB | 17.5 dB | 17.8 dB | 15.5 dB | 20.3 dB |
| Q-Noise+ value | 0.8 | 0.85 | 0.87 | 0.78 | 1.22 |

**Table 4**
Simulation parameters setup.

| Parameters | Settings |
|---|---|
| Area size | 500 m × 500 m |
| Number of nodes | 25 |
| Radio propagation | Propagation range, rang = 180 m |
| Interferences | Interferences orthogonal |
| Modulation | Modulation bpsk |
| Antenna | Antenna omnidirectional |
| Battery | Energy linear |
| HELLO interval | 100 ms |
| Expire time | 300 ms |
| Initial Q-value | 0.5 |
| minspeed | 0 m/s |
| maxspeed | 15 m/s |
| Data packet | 127 Bytes |
| SINR weight | 0.7 |
| Look back for Q-Noise+ (l) | 10 |
| $w$ | $0 < w < 1$ |
| $\alpha$ | 0.6 |
| $\epsilon$ | 0.1 |

mechanism. Another contribution of Q-FANET is the "Routing Neighbor Discovery" function and its integration with the "Routing Decision" module.

Besides these original contributions, Q-FANET presents side contributions by combining and modifying features of QMR and Q-Noise+ approaches, developing a strategy that benefits from several aspects of Machine Learning and Channel Occupation techniques. The penalty mechanism and the velocity constraint are adaptations of the QMR approach. Q-FANET also simplifies QMR's reward function. Finally, Q-FANET combines its ability to discover neighbors with Q-Noise+'s strategy to evaluate the channel conditions. In this sense, the function from Q-Noise+ works as another constraint to the Q-FANET algorithm to decide which is the most suitable link to perform the transmission.

## 4. Experiments and results

Experiments use simulations to compare the performance of Q-FANET with other existing approaches, namely, Q-Geo, Q-Noise+, and QMR, using the event-driven wireless networks simulator WSNet [47]. The WSNET simulator generally applies to the simulation of large-scale sensor networks' behavior but was adapted to the FANET evaluated in this article. The simulation scenario consists of 25 nodes (representing the UAVs), randomly distributed in an area of 500 m × 500 m. The simulation tool randomly selects the source, which transmits a time-varying data flow. Table 4 summarizes the simulation parameters. These parameters were selected following the values used in work that reported QMR [36], for comparison purposes.

In the simulations performed, all of the mobile nodes move according to the Random Waypoint Mobility Model [48], and follow the study of [49]. In this manner, a mobile node makes a movement from its current location to a new random location by selecting a direction and speed (in *[minspeed, maxspeed]*). In this mobility model, after the mobile node moves to the new destination, it will pause for a certain period of time, and then resume moving to another new location. For the experiments, this period is set to 0. Also, this approach considers a lookback value of ten episodes to parametrize Q-Noise+. The simulation tool generates random weights for each episode at the beginning of the simulation. Different from the original Q-Noise+ approach, the most recent episode does not receive the higher weight. Therefore, in this manner, Q-FANET can assign random weights to each episode, not prioritizing a specific one. It is also important to highlight that, following the approach used on Q-Noise+, both the values of the learning rate ($\alpha$) and discount factor ($\gamma$) remain unchanged (0.6 and 1, respectively) throughout the whole experiment.

Two sets of simulations were performed for each algorithm. In the first one, all 25 nodes work correctly, while the second one simulates the existence of ten faulty nodes. This last set evaluates the protocols' capability to overcome the unfavorable conditions of a network with faulty nodes. The data transmission intervals vary between 10 ms an 50 ms, with an increasing pace of 10 ms [36]. This approach allows comparing the results of QMR and Q-FANET in the same conditions. For each transmission interval, initially 100 simulations runs were performed, with the final values been represented as the results' average. Then, the confidence interval was calculated using the t-student distribution and performed additional runs, if necessary, to reach a confidence interval of 95%. The evaluation considers the following metrics.

- **Maximum end-to-end delay:** the maximum delay of a a data packet transmission made from the source node to the destination node.
- **Jitter:** The degree of change in the delay of data packets transmitted from the source node to the destination node.
- **Packet delivery ratio:** The ratio of the number of received data packets by the destination node in relation to the number of data packets transmitted by the source node.

As most of the reinforcement learning techniques, it is recommended that Q-learning uses a training time or number of cases to compose a training set, which would be executed until a convergence of results was obtained. Nevertheless, since 100 simulations are executed for each time interval – with the final average representing the result – the training aspect of this algorithm is not used in the simulation.

This simulation parameters were chosen according to the ones stated in [36] in order to compare the results obtained with Q-Noise+ and Q-FANET in a equivalent test environment. Also, the SINR values vary during the experiments, and it determines the value of the $\eta$ parameter according to Table 2.

### 4.1. Scenario without faulty nodes

In this simulation, the source node sends one thousand data packets considering different transmission intervals. Figs. 3 to 5, show the results of Q-FANET for the different data intervals, in comparison with QGeo, Q-Noise+, and QMR, considering the performance metrics mentioned above.

In Figs. 3 and 4 it is possible to observe that Q-FANET presents a lower max end-to-end delay, as well as a lower jitter than QGeo, Q-Noise+ and QMR. There are two reasons for this better performance: the first is the use of the velocity constraint adapted from QMR, and the second is the channel selection from Q-Noise+. The velocity constraint always select the routing path with the lowest delay from source to destination. Furthermore, the use of Q-Noise+ features gives a higher weight to the channels with a good SINR value. Exploring advantageous features of both algorithms, the new proposal surpass them two. Besides, the standard deviation error bar shows that the results of Q-Noise+ and Q-FANET are inside an acceptable error margin.

Fig. 5 shows that Q-FANET increases the packet delivery ratio compared to the other algorithms. This improvement mainly occurs because the weighted last ten episodes change the learning rate and discount factor in the Q-Learning sub-module of Q-FANET. The SINR-based selection of the best channel, in the QMR sub-module, also collaborates for this result.

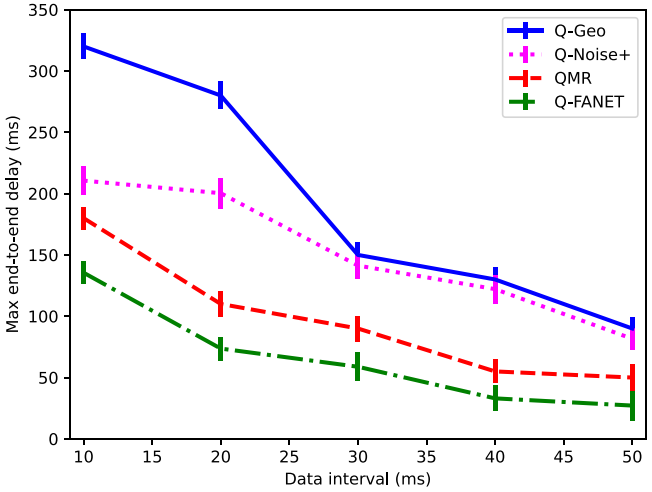



**Fig. 3.** Max end-to-end delay for the first scenario with all nodes working properly.

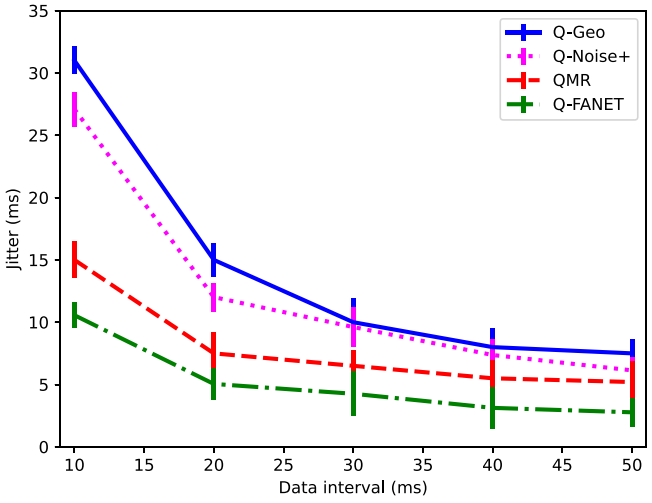



**Fig. 4.** Jitter for the first scenario with all nodes working properly.

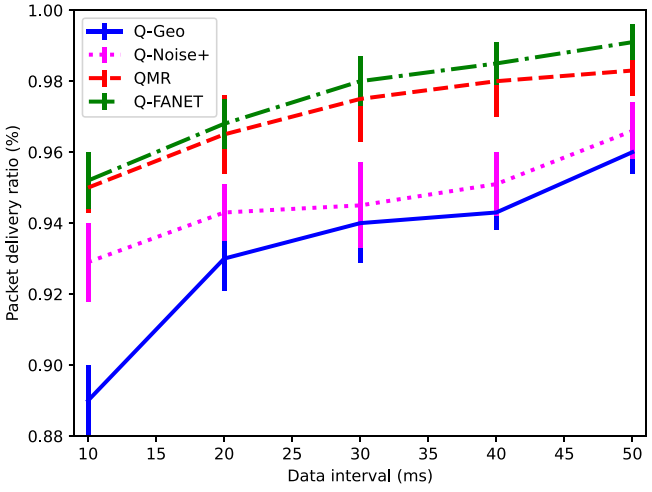



**Fig. 5.** Packet delivery ratio for the first scenario with all nodes working properly.

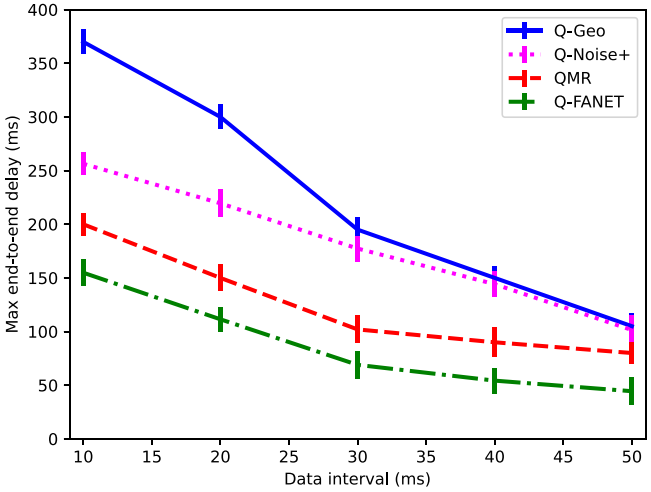



**Fig. 6.** Max end-to-end delay for the scenario with faulty relay nodes.

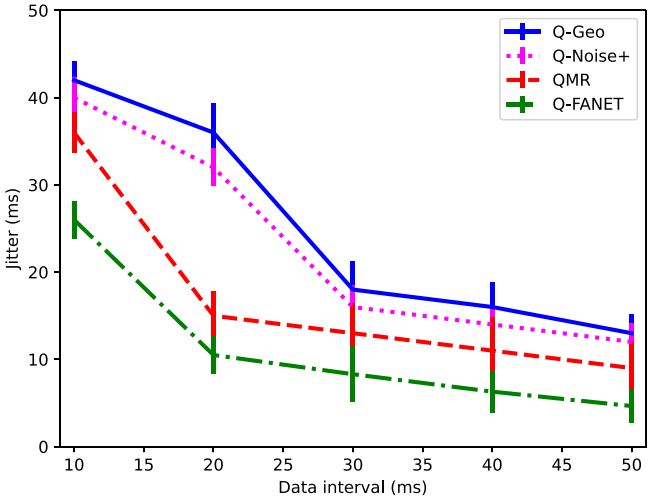



**Fig. 7.** Jitter for a scenario with faulty relay nodes.

### 4.2. Scenario with faulty nodes

In this second set of simulations, 10 out of the 25 nodes stop working by powering them off 1 s after starting the simulation. As in the first set of simulations, Q-FANET was compared with QGeo, Q-Noise+, and QMR under different data intervals.

From Figs. 6 to 8, it is still possible to observe that Q-FANET presents a better performance in all the evaluation metrics. As observed in Figs. 6 and 7, Q-Geo, and Q-Noise+ are greatly affected by the presence of the faulty nodes while both QMR and Q-FANET present a better adaptive behavior, and can overcome the problem by selecting better routes to transmit. The packet delivery is also more affected in Q-Geo and Q-Noise+ compared to Q-FANET and QMR. The difference between the latter ones is smaller in this situation with faulty nodes, but still significant, particularly considering applications such as video streaming, which are very sensitive to the Quality of Service (QoS) degradation, as discussed by [50].

### 4.3. Discussing the improvements of Q-FANET

Figs. 9 and 10 show that Q-FANET can enhance routing performance and presents a significant improvement over QMR, which is the best among the other three protocols tested in the performed experiments.

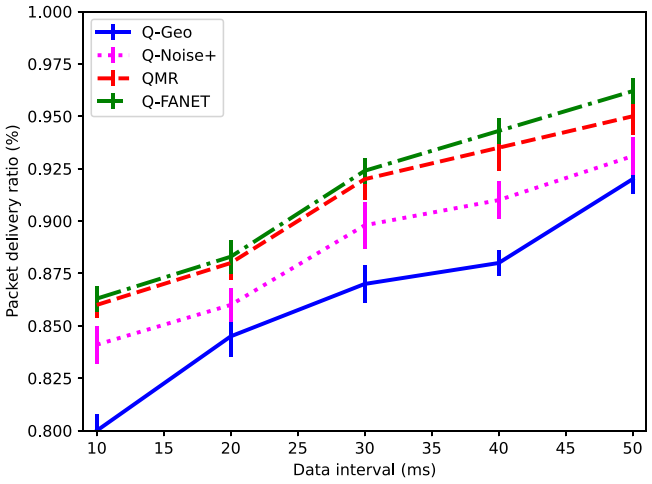



**Fig. 8.** Packet delivery ratio for the scenario with faulty relay nodes.

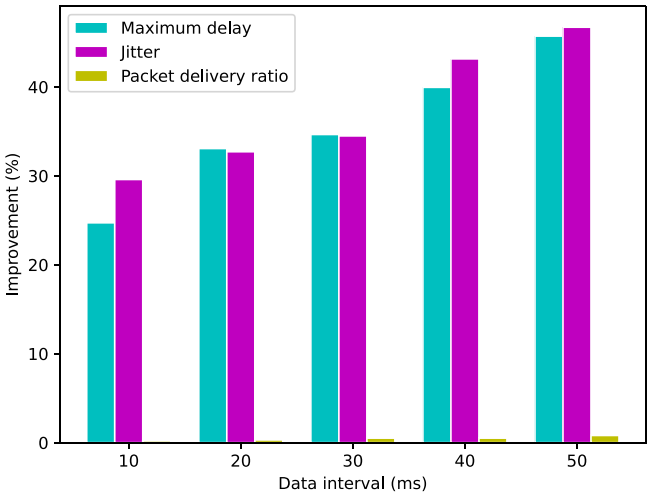



**Fig. 9.** Improvement percentage of Q-FANET over QMR.

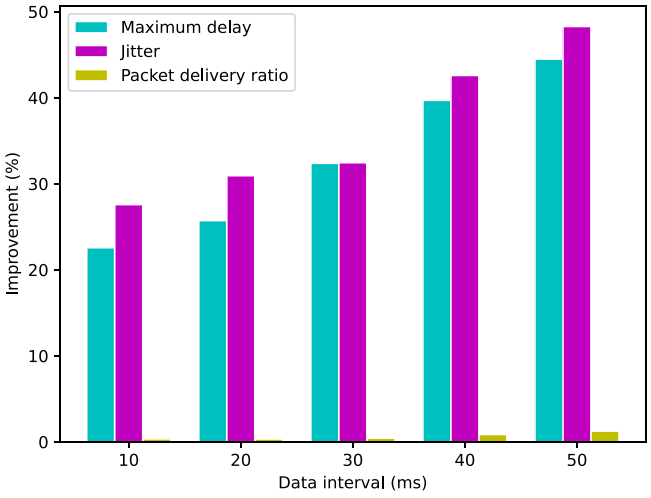



**Fig. 10.** Improvement percentage of Q-FANET over QMR in the faulty nodes simulation scenario.

Q-FANET presents an increasing improvement in terms of maximum delay and jitter over QMR as the time intervals between the data transmission increase, achieving between 45.71% and 46.75%, respectively, of better performance than QMR for the 50 ms data sending time interval. Even in the scenario with the faulty nodes, Q-FANET shows improvements of 44.49% and 48.28% for the maximum delay and jitter. It is crucial to observe that the results under tight intervals are not as good as those obtained for larger intervals, which happens because the Q-FANET algorithm improves its knowledge of the links' status and channel conditions as the experiment runs. Variable intervals result in network topology changes that lead to adjustments in the algorithm's behavior, consequently improving the results. However, even in these challenging conditions, the results in terms of delay and jitter are significant and represent an important contribution to the type of video-stream based applications UAV-networks usually are employed to, which are very sensitive to QoS degradation.

It is worth mentioning that Q-FANET has a minor improvement over QMR in terms of packet delivery ratio, showing a better performance of 0.81% and 1.26%, for the scenario with all the nodes working and the scenario with the faulty nodes, respectively. Under tight intervals, the improvement is still smaller and they can be explained by the same reasons mentioned above, i.e. the learning process of the algorithm. However, considering the possible video streaming applications, this small improvement can represent a significant result for the final user, considering Quality of Experience (QoE) evaluations, as discussed in [51]. Analyzing these results in light of the discussions provided by [51] and [50], it is possible to assess that the improvements provided by Q-FANET can benefit end-users of video streaming applications through lowering the number and length of stalls in the videos, particularly considering that these QoE metrics are directly affected by the QoS metrics delay and jitter.

The realism of the adopted nodes' mobility model could be argued as a threat to the validity of the obtained results. However, the adopted Random Waypoint Mobility Model is realistic enough for the scenarios considered in this present study, as analyzed in [49]. Thus, no bias in the result is expected due to the used mobility model. Anyway, other mobility models can be considered in the continuation of this study, such as the Gaussian-Markov explored in [52].

Still about the mobility, as stated in 3.1, the considered UAVs are those flying with low to medium speed. However, studying adaptations in the proposed solution to consider faster UAV platforms, flying at high speeds, is also possible with the developed simulation framework setup. Despite of the importance of this subject, it is out of the scope of this current study.

Finally, Fig. 11 shows that Q-FANET respects the timer limit for convergence in both simulation scenarios. Following Q-FANET knowledge and performance improvement over time, the convergence time for the algorithm slowly decreases as it learns the most optimal paths to transmit data, even with higher transmission intervals and the adjustments of the network topology.

## 5. Conclusion and future work

This paper proposed Q-FANET, an improved Q-learning based routing protocol for FANETs. The proposed approach has brought together the leading techniques and elements used in two different routing protocols that make use of Reinforcement Learning: QMR and Q-Noise+ in a new protocol. By combining and adapting elements of these base protocols into the new conceived protocol architecture, the goal was to propose a protocol that better suites the dynamic behavior of FANETs, improving the network reliability and performance.

The proposal was evaluated, having its performance compared with QMR, Q-Noise+ and Q-GEO protocols in two scenarios. In the first one, all the nodes were up and running, while the second one considered the presence of faulty nodes. Q-FANET obtained significantly lower maximum end-to-end delay and jitter than the competitors in both scenarios. There was also a minor increase in the packet delivery ratio.

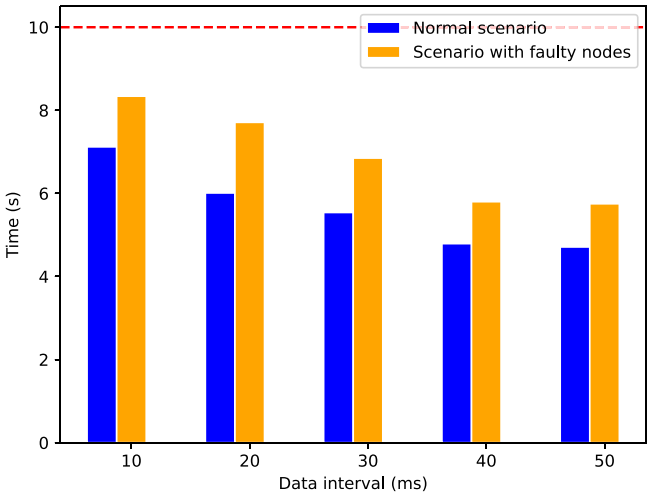



**Fig. 11.** Time for the convergence of Q-FANET in both simulation scenarios.

Directions for future investigations include dealing with other issues, like energy consumption, an essential concern regarding small UAVs with constrained energy resources. Online inner parameter adaptation is a possible direction to further enhance the proposed solution. Particular movement patterns are also of interest for future exploratory experiments as well as further adaptations of the proposed solution to support networks composed of high-speed UAVs.

## Declaration of competing interest

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

## Acknowledgment

This study was financed in part by the Coordenação de Aperfeicoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001 and in part by Conselho Nacional de Desenvolvimento Cientifico e Tecnologico - Brasil (CNPq) Projects 309505/2020-8 and 420109/2018-8, and Fundacao de Amparo a Pesquisa do Estado do Rio Grande do Sul (FAPERGS).

## References

[1] A. Merwaday, I. Güvenç, Uav assisted heterogeneous networks for public safety communications, in: 2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2015, pp. 329–334.

[2] R.S. de Moraes, E.P. de Freitas, Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services, J. Intell. Robot. Syst. 92 (3) (2018) 645–656.

[3] R.S. de Moraes, E.P. de Freitas, Multi-uav based crowd monitoring system, IEEE Trans. Aerosp. Electron. Syst. 56 (2) (2020) 1332–1345.

[4] L.F.F.G. de Assis, L.P. Behnck, D. Doering, E.P. de Freitas, C.E. Pereira, F.E.A. Horita, J. Ueyama, J. Porto de Albuquerque, Extending sensor web for near real-time mobile sensor integration in dynamic scenarios, in: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), 2016, pp. 303–310.

[5] D. Orfanus, E.P. de Freitas, F. Eliassen, Self-organization as a supporting paradigm for military uav relay networks, IEEE Commun. Lett. 20 (4) (2016) 804–807.

[6] C. Caillouet, F. Giroire, T. Razafindralambo, Efficient data collection and tracking with flying drones, Ad Hoc Netw. 89 (2019) 35–46, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870518305985.

[7] O.K. Sahingoz, Mobile networking with uavs: Opportunities and challenges, in: 2013 International Conference on Unmanned Aircraft Systems (ICUAS), 2013, pp. 933–941.

[8] E. Pignaton de Freitas, L.A.L.F. da Costa, C. Felipe Emygdio de Melo, M. Basso, M. Rodrigues Vizzotto, M. Schein Cavalheiro Corrêa, T. Dapper e Silva, Design, implementation and validation of a multipurpose localization service for cooperative multi-uav systems, in: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 2020, pp. 295–302.

[9] İlker Bekmezci, O.K. Sahingoz, Şuamil Temel, Flying ad-hoc networks (fanets): A survey, Ad Hoc Netw. 11 (3) (2013) 1254–1270, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870512002193.

[10] M.B. Yassein, N.A. Damer, Flying ad-hoc networks: Routing protocols, mobility models, issues, Int. J. Adv. Comput. Sci. Appl. 7 (6) (2016) [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2016.070621.

[11] C.J. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (3–4) (1992) 279–292.

[12] J. Jiang, G. Han, Routing protocols for unmanned aerial vehicles, IEEE Commun. Mag. 56 (2018) 58–63.

[13] Y. Fu, Ding Prof. M., C. Zhou Prof., H. Hu, Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization, IEEE Trans. Syst. Man Cybern.: Syst. 43 (6) (2013) 1451–1465.

[14] M.S.-H. Jean-Daniel Medjo Me Biomo, Thomas. Kunz, Routing in Unmanned Aerial Ad hoc Networks: A Recovery Strategy for Greedy Geographic Forwarding Failure, 3, (2014) 2236–2241. [Online]. Available: https://goo.gl/aBgvgQ.

[15] A. Boukerche, B. Turgut, N. Aydın, M.Z. Ahmad, L. Bölöni, D. Turgut, Routing protocols in ad hoc networks: A survey, Comput. Netw. 55 (2011) 3032–3080.

[16] C.M. Cheng, P.H. Hsiao, H.T. Kung, D. Vlah, Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm, in: IEEE Wireless Communications and Networking Conference, WCNC, 2007, pp. 4420–4427.

[17] J. Ko, A. Mahajan, R. Sengupta, A network-centric UAV organization for search and pursuit operations, IEEE Aerosp. Conf. Proc. 6 (2002) 2697–2713.

[18] P. long Yang, C. Tian, Y.H. Yu, Analysis on optimizing model for proactive ad hoc routing protocol, in: MILCOM 2005-2005 IEEE Military Communications Conference, Vol. 5, 2005, pp. 2960–2966.

[19] M. Bani, N. Alhuda", Flying ad-hoc networks: Routing protocols, mobility models, issues, Int. J. Adv. Comput. Sci. Appl. 7 (6) (2016) 162–168, [Online]. Available: https://goo.gl/Tu95TD.

[20] D.S. Vasiliev, A. Abilov, V.V. Khvorenkov, Peer selection algorithm in flying ad hoc networks, in: 2016 International Siberian Conference on Control and Communications (SIBCON), 2016, pp. 1–4.

[21] S. Habib, S. Saleem, K.M. Saqib, Review on manet routing protocols and challenges, in: 2013 IEEE Student Conference on Research and Developement, 2013, pp. 529–533.

[22] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, 1996.

[23] V.A. Maistrenko, L.V. Alexey, V.A. Danil, Experimental estimate of using the ant colony optimization algorithm to solve the routing problem in fanet, in: 2016 International Siberian Conference on Control and Communications (SIBCON), 2016, pp. 1–10.

[24] J. Jailton, T. Carvalho, J. Araújo, C. Frances, Relay positioning strategy for traffic data collection of multiple unmanned aerial vehicles using hybrid optimization systems: A fanet-based case study, Wirel. Commun. Mob. Comput. 2017 (2017) 1–11.

[25] V. Park, S. Corson, Temporally-ordered routing algorithm (tora) version 1. [Online]. Available: https://tools.ietf.org/html/draft-ietf-manet-tora-spec-00, 0000.

[26] D.K.L. Ram Shringar Raw, S. Das, An analytical approach to position-based routing protocol for vehicular ad hoc network, 2012, p. 9.

[27] B. Karp, H.T. Kung, Gpsr: greedy perimeter stateless routing for wireless networks, in: MobiCom, 2000.

[28] C. Zang, S. Zang, Mobility prediction clustering algorithm for uav networking, in: 2011 IEEE GLOBECOM Workshops (GC Wkshps), 2011, pp. 1158–1161.

[29] L. Kesheng, Z. Jun, Z. Tao, The clustering algorithm of uav networking in near-space, in: 2008 8th International Symposium on Antennas, Propagation and EM Theory, 2008, pp. 1550–1553.

[30] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[31] P.S. Bithas, E.T. Michailidis, N. Nomikos, D. Vouyioukas, A.G. Kanatas, A survey on machine-learning techniques for uav-based communications, Sensors (Basel, Switzerland) 19 (2019).

[32] R. Li, F. Li, X. Li, Y. Wang, Qgrid: Q-learning based routing protocol for vehicular ad hoc networks, in: 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), 2014, pp. 1–8.

[33] A. Serhani, N. Naja, A. Jamali, Qlar: A q-learning based adaptive routing for manets, in: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016, pp. 1–7.

[34] W.-S. Jung, J. Yim, Y.-B. Ko, Qgeo: Q-learning-based geographic ad hoc routing protocol for unmanned robotic networks, IEEE Commun. Lett. 21 (2017) 2258–2261.

[35] Q. Yang, S. Jang, S. Yoo, Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks, Wirel. Pers. Commun. (2020) 1–24.
[36] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, Y. Xu, Qmr: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks, Comput. Commun. 150 (2020) 304–316.
[37] C. He, Q. Wang, Y. Xu, J. Liu, Y. Xu, A q-learning based cross-layer transmission protocol for manets, in: 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), 2019, pp. 580–585.
[38] N. Lyu, G. Song, B. Yang, Y. Cheng, Qngpsr: A q-network enhanced geographic ad-hoc routing protocol based on gpsr, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018, pp. 1–6.
[39] L.R. Faganello, R. Kunst, C.B. Both, L.Z. Granville, J. Rochol, Improving reinforcement learning algorithms for dynamic spectrum allocation in cognitive sensor networks, in: 2013 IEEE Wireless Communications and Networking Conference (WCNC), 2013, pp. 35–40.
[40] K. Dalamagkidis, Classification of UAVs, Springer Netherlands, Dordrecht, 2015, pp. 83–91, [Online]. Available: https://doi.org/10.1007/978-90-481-9707-1_94.
[41] D. of Defense, Lighter-than-air vehicles, 2012, [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a568211.pdf.
[42] 3DRobotics, iris plus quadcopter, 2014, [Online]. Available: http://3dr.com/support/articles/207358106/iris/.
[43] R.E. Mohemed, A.I. Saleh, M. Abdelrazzak, A.S. Samra, Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks, Comput. Netw. 114 (2017) 51–66.
[44] T.S. Rappaport, Wireless communications - principles and practice, 1996.
[45] M. Wunder, M.L. Littman, M. Babes, Classes of multiagent q-learning dynamics with epsilon-greedy exploration, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), Citeseer, 2010, pp. 1167–1174.
[46] L. Matignon, G. Laurent, N.L. Fort-Piat, Reward function and initial values: Better choices for accelerated goal-directed reinforcement learning, in: ICANN, 2006.
[47] Wsnet simulator, 2019, http://wsnet.gforge.inria.fr/, (accessed: 2019-09-30).
[48] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wirel. Commun. Mob. Comput. 2 (2002) 483–502.
[49] D. Orfanus, E.P. de Freitas, Comparison of uav-based reconnaissance systems performance using realistic mobility models, in: 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014, pp. 248–253.
[50] I. Zacarias, J. Schwarzrock, L. Gaspary, A. Kohl, R. Fernandes, J. Stocchero, E.P. de Freitas, Enhancing mobile military surveillance based on video streaming by employing software defined networks, Wirel. Commun. Mob. Comput. 2018 (2018) 1–12.
[51] Z. Zhao, P. Cumino, A. Souza, D. Rosário, T. Braun, E. Cerqueira, M. Gerla, Software-defined unmanned aerial vehicles networking for video dissemination services, Ad Hoc Netw. 83 (2019) 68–77, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870518306231.
[52] X. Li, T. Zhang, Stgm: A spatiotemporally correlated group mobility model for flying ad hoc networks, in: ChinaCom, 2016.

**Luis Antonio L.F. da Costa** is currently an Assistant Researcher at the Applied Computing Group of the University of Vale do Rio dos Sinos (Unisinos). He also holds a M.Sc. degree in Computer Science from the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS). His research interests include computer networks, optimization, artificial intelligence and wireless networks.

**Rafael Kunst** is a professor and Researcher at the Applied Computing Graduate Program of the University of Vale do Rio dos Sinos (Unisinos), Brazil, where he is also a member of the Software Innovation Laboratory — SOFTWARELAB. He is also an ad-hoc consultant for the Brazilian Ministry of Education. He holds a Ph. D. and an M.Sc. degree in Computer Science; both received from the Federal University of Rio Grande do Sul (UFRGS). His current research interests involve next-generation mobile communications, such as 5G and 6G, military communications, Industry 4.0, smart cities, Internet of Things, and the application of Big Data Analytics and Machine Learning to optimize telecommunications. He has extensive experience as a consultant, coordinating and participating in projects with companies from Brazil and abroad.

**Edison Pignaton de Freitas** Computer Engineer by the Military Institute of Engineering in Brazil (2003), he received the M.Sc. in Computer Science by Federal University of Rio Grande do Sul (UFRGS) in Brazil (2007), and the Ph.D. degree in Computer Science and Engineering from Halmstad University in Sweden in 2011. He is currently associate professor at UFRGS, developing research in the Graduate Program on Computer Science and in the Graduate Program in Electrical Engineering. His research interests are mainly in the following areas: Computer Networks, Internet of Things, Real-Time Systems, Multiagents Systems and Unmanned Aerial Vehicles.