A Q-learning-based routing scheme for smart air quality monitoring system using flying ad hoc networks

Jan Lansky¹, Amir Masoud Rahmani²*, Seid Miad Zandavi³, Vera Chung⁴, Efat Yousefpoor⁵, Mohammad Sadegh Yousefpoor⁵, Faheem Khan⁶* & Mehdi Hosseinzadeh⁷,⁸,⁹

Air pollution has changed ecosystem and atmosphere. It is dangerous for environment, human health, and other living creatures. This contamination is due to various industrial and chemical pollutants, which reduce air, water, and soil quality. Therefore, air quality monitoring is essential. Flying ad hoc networks (FANETs) are an effective solution for intelligent air quality monitoring and evaluation. A FANET-based air quality monitoring system uses unmanned aerial vehicles (UAVs) to measure air pollutants. Therefore, these systems have particular features, such as the movement of UAVs in three-dimensional area, high dynamism, quick topological changes, constrained resources, and low density of UAVs in the network. Therefore, the routing issue is a fundamental challenge in these systems. In this paper, we introduce a Q-learning-based routing method called QFAN for intelligent air quality monitoring systems. The proposed method consists of two parts: route discovery and route maintenance. In the part one, a Q-learning-based route discovery mechanism is designed. Also, we propose a filtering parameter to filter some UAVs in the network and restrict the search space. In the route maintenance phase, QFAN seeks to detect and correct the paths near to breakdown. Moreover, QFAN can quickly identify and replace the failed paths. Finally, QFAN is simulated using NS2 to assess its performance. The simulation results show that QFAN surpasses other routing approaches with regard to end-to-end delay, packet delivery ratio, energy consumption, and network lifetime. However, communication overhead has been increased slightly in QFAN.

Industrial growth has led to technological advances in the world, but these advances have forced industrial and chemical contaminations that negatively affect human health and environment, including agricultural products, insects, and animals. Air quality standards are usually evaluated by measuring various air pollutants, including nitrogen dioxide (NO₂), ground-level ozone (O₃), particulate matter (PM), carbon dioxide (CO₂), and carbon monoxide (CO). These pollutants cause serious health injuries. In the long-term, they damage the human respiratory system and lead to his early death. For this reason, air quality monitoring is very important and vital. This issue has been considered by environmental organizations and governmental institutions. Traditional air quality monitoring systems are dependent on fixed monitoring stations. Although, these solutions are faced many challenges and restrictions because these stations can cover a limited area, and their deployment is difficult and costly. Today, air quality in urban, rural, and industrial areas is controlled using several advanced networking technologies such as flying ad hoc networks (FANETs), Internet of things (IoT), vehicular ad hoc networks (VANETs), and wireless sensor networks (WSNs).

Among these technologies, FANETs have special importance because these networks can cover a large operational environment and have been deployed easily and with acceptable costs. FANET includes a group of

¹Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague, Czech Republic. ²Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan. ³School of Biotechnology and Biomolecular Science, The University of New South Wales, Sydney, Australia. ⁴School of Computer Science, The University of Sydney, Sydney, Australia. ⁵Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran. ⁶Department of Computer Engineering, Gachon University, Seongnam, Republic of Korea. ⁷Institute of Research and Development, Duy Tan University, Da Nang, Vietnam. ⁸School of Medicine and Pharmacy, Duy Tan University, Da Nang, Vietnam. ⁹Department of Computer Science, University of Human Development, Sulaymaniyah, Iraq. *email: rahmania@yuntech.edu.tw; faheem@gachon.ac.kr
unmanned aerial vehicles (UAVs) organized in an ad hoc form. UAVs, which are also called drones, are known as a promising technology for intelligent air quality monitoring. UAVs move in a three-dimensional space, and their speed is usually in the range of 30–46 km/h. UAVs have powerful computing power and energy resources because they require a lot of energy to fly, which is much more compared to the energy needed for data processing. Although, energy is a serious issue for small drones. Figure 1 shows the FANET application for smart air quality monitoring. Today, some researchers study the application of UAVs for intelligent air pollution monitoring systems. For example, the authors have presented a component for the UAV structure to control air pollutants using a real-time manner. In the authors have expressed the importance of UAVs to monitor hazardous pollutants such as leakage of chemical materials. Furthermore, UAVs and IoT have been integrated with each other to monitor air pollution parameters. In FANET-based air quality monitoring system, each UAV is equipped with cameras and specific sensors to measure air pollutants. With this equipment, UAVs can collect their environmental data and send them to ground station (GS) using wireless communication technologies.

Flying ad hoc networks have special features such as low density, rapid topological changes, rapid changes in UAV conditions such as removing and adding UAVs. These features cause link failure when sending data related to the measurement of air pollutants to the ground station (GS). In an intelligent air quality monitoring system based on FANET, the time required to deliver data packets from source to destination should be short, and collision between packets should not occur. Due to the specific features of FANETs, providing an efficient routing method for this smart system is challenging. Routing means sending and receiving packets between two UAVs called source UAV and destination UAV. An appropriate routing process helps the intelligent air quality monitoring system to stably and continuously control air quality in an environment. In different studies, researchers have stated various routing methods for flying ad hoc networks, for example position-based, topology-based, hybrid, and nature-inspired routing methods. Existing classical routing protocols cannot provide a good solution to meet the routing problem in FANET. Recently, many researchers have tried to solve the routing challenge in FANETs using artificial intelligence (AI) and machine learning (ML) techniques. Therefore, UAVs can independently learn the routing process in FANET. In ML, reinforcement learning (RL) is an efficient technique, which can efficiently and independently design routing algorithms in FANETs. RL is popular due to the use of trial and error technique. This algorithm includes an agent and an environment. In a reinforcement learning-based routing method, the agent discovers the network environment by performing appropriate actions to achieve an optimal routing policy. To this end, the agent must discover the best paths between different source-destination pairs by considering the optimization criteria. Usually, this process uses the local information of nodes to lower energy consumption and improve network connections. Although to achieve an optimal routing technique, it is necessary to identify the total system.

In this paper, we propose a Q-learning-based routing method called QFAN for the intelligent air quality monitoring system using FANETs. QFAN includes two parts: route discovery and route maintenance.

- **Route discovery**: QFAN uses a Q-Learning-based route discovery mechanism to discover a new path between source-destination pairs. In this mechanism, the route request packet (RREQ) plays the role of an agent. Furthermore, the network is the learning environment. The state of the agent indicates the node, which has received the RREQ message. In this process, the action represents a set of allowed neighboring nodes, which can receive the RREQ message in the current state. In the route discovery process, we present the filtering parameter according to several factors such as the motion direction, remaining energy, link quality, and distance to restrict the search space. As a result, the agent can achieve the optimal response (i.e. the best route) at a lower time. Additionally, QFAN utilizes a reward function, which considers three parameters, including delay, the number of hops, and route fitness.
• **Route maintenance:** It includes two parts. In the first part, QFAN detects and corrects routes close to failure. This helps QFAN to keep the data transmission process without interruption. In the second part, QFAN quickly identifies the failed paths to replace them with alternative paths to lower latency in the data transfer process.

In the following, the paper is organized as follows: in "Related works", some routing approaches are reviewed in FANETs. "Basic concepts" introduces reinforcement learning, which is used for designing QFAN. Also, the system model used in our proposed approach is presented in this section. "Proposed method" describes QFAN in detail. "Simulation and evaluation of results" provides the simulation results of our scheme. Finally, the paper is concluded in "Conclusion".

**Related works**
Arafat and Moh in25 have presented the Q-learning-based topology-aware routing (QTAR) scheme for FANETs. To create reliable paths, QTAR not only utilizes the information of single-hop neighbors but also uses the information of two-hop neighbors. The authors believe that this method improves the path discovery process, reduces the time required for calculating routes and improves the selection of the next-hop node. Although this technique increases routing overhead and system complexity, QTAR considers scales such as spatial information, delay, speed, and energy when selecting next-hop node. Moreover, QTAR provides a technique for calculating the link lifetime to estimate the Hello interval and the link holding time. In this protocol, Q-Learning factors such as learning rate and reward factor are adaptively determined with regard to network conditions. QTAR is executed on a 3D environment that is acceptable for FANET.

Jung et al. in26 have suggested the Q-Learning-based Geographic routing scheme (QGeo). It is a distributed scheme, which uses local information for routing decisions. This method seeks to reduce routing overhead and delay in the routing process. In this process, QGeo has used spatial information and node speed in the network and ignored parameters such as energy and delay. This issue may create low-energy, unstable, and high-delay routes. In this method, the discount factor in Q-learning is specified based on distance and the mobility pattern of neighbors. However, this approach is executed on a two-dimensional space that is not compatible with FANET.

Karp and Kung in27 have provided the greedy perimeter stateless routing (GPSR) scheme for ad hoc networks. GPSR applies the location information of the single-hop neighboring UAVs in routing decisions. GPSR diminishes routing overhead and accelerates the path discovery operation. This method uses a greedy routing technique to acquire the next-hop node. However, GPSR does not pay attention to parameters such as energy, delay, and speed of nodes in dynamic environments. As a result, it has limitations in the route selection process in FANETs.

Lee et al. in28 have offered the energy-aware fuzzy logic-based routing method for FANETs. This method, which is an improved model of AODV, builds stable routes in the network, distributes the energy consumption evenly between network nodes and increases the network lifetime. This routing method consists of two parts: finding and maintaining routing paths. In the first part, each node obtains a score to decide on the RREQ rebroadcasting. Only high-score nodes are tasked to disseminate RREQs in the network. Furthermore, this work manages the flood of RREQs in the network and prevents the broadcast storm problem. When choosing the path, a fuzzy system is designed to pick out paths with low delay, high fitness and less hops. The second part involves the prevention of the route failure via discovering and modifying the paths on the failure threshold and the reconstruction of failed paths via identifying and replacing rapidly these paths. This method is executed on the 3D environment that is acceptable for FANET. Although this approach has a lot of communication overhead.

Liu et al. in29 have suggested the Q-learning-based multi-objective optimization routing (QMR) in FANETs. QMR has improved the exploration and exploitation system to earn the most effective routes, shorten latency, and lessen energy consumption in the network. In QMR, nodes regularly disseminate hello messages to obtain information about neighboring UAVs to reliably select the next-hop node. In QMR, Q-learning factors (i.e. the learning rate and the discount factor) are adjusted according to network conditions. This issue helps QMR to better be adapted to the dynamic network. When these parameters are adjustable, the routing process is more accurate. This decreases disconnections in the network. QMR proposes a mechanism for preventing path failure. Although, it is executed on a two-dimensional area that is not machetable for FANET. Additionally, QMR does not design an appropriate control mechanism for controlling network congestion.

Oubbati et al. in30 have presented a routing technique called ECaD for FANETs. It is rooted in AODV. In ECaD, high-energy UAVs can participate in route discovery. This helps ECaD to create stable paths. However, this method is challenging when a node cannot find high-energy neighboring nodes as the next-hop node. Thus, this node cannot create any route to other nodes in the network. Therefore, this UAV is isolated in the network. In addition, ECaD considers only energy parameter when finding the next-hop node. This scheme evenly distributes energy consumption and builds stale paths in the network. Furthermore, it prevents link failure using a predictive mechanism. ECaD is simulated in a three-dimensional space.

Costa et al. in31 have proposed a Q-learning-based routing protocol called Q-FANET for flying ad hoc networks. Q-FANET can lower delay in the network. It has been inspired by two methods, including QMR and Q-Noise+. It includes two main parts: discovering neighbors and routing. In the neighbor discovery process, UAVs exchange their information periodically. This helps UAVs to be aware of their neighbors at any moment and use the information in the routing phase. In the second part, Q-FANET utilizes an improved Q-learning technique called Q-Learning+ to improve QMR and Q-Noise+. Q-FANET considers channel quality conditions when determining Q-values so that high-quality links can obtain a higher Q-value. Although this protocol does not pay attention to energy when finding routing paths. This approach is executed on a two-dimensional area that is not accepted for FANETs.
Perkins et al. in 32 have offered the ad hoc on-demand distance vector (AODV) approach in ad hoc networks. This method is on-demand. This means that routing paths are constructed only when there is a route request. In recent years, many routing techniques are inspired by AODV. However, this method has very challenges in FANETs because of their properties such as high-speed flying nodes and fast link failure. Additionally, AODV has a long delay when discovering a new route. AODV introduces a maintenance mechanism to discover and correct the failed paths. AODV requires high bandwidth and high delay for creating paths and correcting the failed routes, especially in large-scale networks.

Rahmani et al. in 33 have presented an enhanced model of the optimized link state routing protocol (OLSR) called OLSR+ for FANETs. They have suggested a novel solution for assessing the link lifetime by considering various scales, namely connection quality, distance, relative speed, and motion direction of flying nodes. In addition, OLSR+ has designed a fuzzy logic-based algorithm to choose multipoint relays (MPRs) in the network. This fuzzy model involves three fuzzy inputs: the remaining energy of UAV, the connection lifetime of UAVs, and their neighborhood degree. It allocates a score to each UAV. This score is used to select MPRs. In this method, the routing table is calculated by considering three parameters: the number of hops, path energy, and path lifetime. OLSR+ creates stable and energy-efficient paths and reduces packet loss. Moreover, this method is executed on a 3D space that is acceptable for FANET. However, OLSR+ suffers from a lot of routing overhead. Due to dynamic topology of FANETs, it is very difficult to update neighborhood table and routing table because it requires a lot of delay and communication overhead.

Qiu et al. in 34 have designed a Q-learning-based geographic routing (QLGR) method for FANETs. This protocol utilizes a multi-agent learning algorithm for designing the routing model. In this model, each drone is regarded as an agent, and the whole network is assumed to be the learning system. The state set involves the status of nodes, and the action set indicates neighbors of the current UAV. In QLGR, each UAV uses a distributed technique to estimate the fitness value of its neighbors based on local information, namely link quality, energy, and queue length to choose the most suitable relay UAV. This information is obtained by broadcasting hello messages between the UAVs periodically. After each action, the agent receives two rewards from the environment: the local reward (LR) that represents the local returns of neighboring nodes and is calculated based on connection quality and load capacity, and the global reward (GR) that indicates global returns of UAVs after taking this action. GR is evaluated based on distance from the current location of the packet to the destination. This method presents a rebroadcast control mechanism to determine the broadcast interval of hello messages and manage the communication overhead in the network. This protocol is executed on a two-dimensional space that is not appropriate for FANET.

Table 1 expresses the most important strengths and weaknesses of the related works and compares these schemes with our scheme.

Basic concepts
In this section, we demonstrate Q-learning and system model briefly because we consider these concepts for designing the route discovery mechanism in QFAN.

Reinforcement learning. Reinforcement learning (RL) is a robust and useful implement in artificial intelligence (AI). RL includes two main modules: agent and environment. In this approach, the agent takes various actions to explore the environment35. It decides on an action according to the Markov decision process (MDP) that is a useful structure for simulating various issues using dynamic programming and RL techniques. It helps the agent to control this process randomly. Usually, MDP is known based on four parameters that are displayed as \( (S, A, p, r) \). So that \( S \) and \( A \) indicate the finite state and action sets, respectively. Moreover, \( p \) defines a probability value for replacing the current state with the next state after running the action \( a \). Furthermore, \( r \) is a reward given by the environment after completing the action \( a \). According to Fig. 2, at the time \( t \), the agent takes the action \( a_t \) with regard to its latest state \( s_t \) to get the reward \( r_t \) and its new state \( s_{t+1} \) from the environment. The agent seeks to learn the policy \( \pi \) that enlarges the reward obtained from the environment. In the long term, the expected discounted total reward must be increased through \( \max \left\{ \sum_{t=0}^{\infty} \delta^t r_t(s_t, \pi(s_t)) \right\} \). \( \delta \in [0, 1] \) is known as the discount factor, which indicates the importance of the reward and the effort of the agent for discovering the environment. In the proposed method, \( \delta = 0.7 \). According to the discounted reward, when the transition probabilities (i.e. \( p \)) are known, the Bellman equation called Q-function (Eq. 1) is created to perform the next action \( a_{t+1} \) using MDP.

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

where \( \alpha \) represents the learning rate so that \( 0 < \alpha \leq 1 \). \( \alpha \) (also called step size) is one of the Q-learning parameters. It is between 0 and 1. If \( \alpha = 0 \), then the Q-values are never updated. As a result, the agent does not learn any new knowledge and only exploits prior knowledge, else if \( \alpha = 1 \), this means that the learning process is very fast and the agent focuses only on the most recent information and ignores its prior knowledge when exploring possibilities. In practice, researchers usually consider a constant learning rate, such as \( \alpha = 0.1 \)36. In the proposed method, \( \alpha = 0.1 \).

The Q-Learning algorithm presents a model-free and off-policy RL technique, which uses Q-function presented in Equation 1. It helps an agent to learn the best actions36,37. Q-learning applies a table (i.e. Q-Table) for storing all optimal state-action pairs. In this table, there are two inputs (i.e. the state-action pairs) and one output called Q-value. Q-Learning seeks to achieve maximum Q-value. To this end, agents learn the best action strategy through the environmental feedback. According to the learning operation, the agent analyzes whether an action is appropriate in the current state or not. Based on this decision, the agent chooses a better action in
| Method     | Strengths                                                                 | Weaknesses                                                                                           |
|------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| QTAR       | Considering parameters such as energy, spatial information, delay, and movement of UAVs for discovering routes, adjusting the broadcast hello interval based on the path lifetime, adjusting learning rate and discount factor dynamically, implementing this routing method in a 3D space | High overhead because of the need to information of single-hop and two-hop neighbors                  |
| QGeo       | Adjusting the discount parameter dynamically by calculating the distance and movement pattern of UAVs | Not considering the residual energy of flying nodes and delay in the routing process, the possibility of making unstable and low-energy paths, implementing QGeo in a two-dimensional environment, enlarging Q-table by growing the number of nodes in the network |
| GPSR       | Reducing routing overhead and delay when creating paths, high scalability | Constructing paths only based on distance and ignoring parameters such as energy, delay, link quality in this process, the possibility of making unstable and low-energy routes, not suitable for 3D environments such as FANETs |
| QFANET     | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |
| Q-FANET    | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QGeo       | Adjusting the discount parameter dynamically by calculating the distance and movement pattern of UAVs | Not considering the residual energy of flying nodes and delay in the routing process, the possibility of making unstable and low-energy paths, implementing QGeo in a two-dimensional environment, enlarging Q-table by growing the number of nodes in the network |
| GPSR       | Reducing routing overhead and delay when creating paths, high scalability | Constructing paths only based on distance and ignoring parameters such as energy, delay, link quality in this process, the possibility of making unstable and low-energy routes, not suitable for 3D environments such as FANETs |
| QFAN       | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |
| Q-FANET    | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |
| Q-FANET    | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |
| Q-FANET    | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |
| Q-FANET    | Improving delay in the routing process, Predicting path failure            | Implementing Q-FANET in a 2D space, not considering the energy parameter in the routing process, the possibility of forming unstable routes in the network, fixed Q-Learning parameters |
| AODV       | Designing an on-demand algorithm, designing a route maintenance model     | High delay and routing overhead when creating a route, the possibility of the broadcast storm, not considering the energy parameter when finding paths, not considering the features of FANETs |
| OLSR+      | Implementing this routing method in a 3D space, calculating link lifetime based on connection quality, distance, relative speed, and motion direction of UAVs, presenting a fuzzy logic-based algorithm for selecting MPRs, considering the route energy and the path delay when choosing a route, reducing packet loss, improving stability of paths | high delay and routing overhead when updating routing table                                             |
| QMR        | Adjusting Q-Learning parameters adaptively, designing an adaptive mechanism to build a trade-off between exploration and exploitation, presenting a penalty mechanism for addressing the routing hole problem | Not designing a broadcast control mechanism, not considering energy parameter in the routing process, the possibility of forming unstable routes, simulating QMR in a two-dimensional environment |
| ECD        | Considering the energy of UAVs when selecting paths, not participating low-energy nodes in the route construction operation, implementing the routing method in a 3D space | Flooding routing messages                                                                            |

Table 1. Strengths and weaknesses of related works.

The next step. Q-values are refreshed in each step according to Eq. (1). Then, the agent exploits the environment by performing actions with most suitable Q-value. This policy is known as ϵ-greedy. According to this policy, the agent explores or exploits the environment based on the probability value ϵ.

**System model.** This section illustrates the network system applied in QFAN. This network model is homogeneous and consists of a number of flying nodes. They are randomly scattered in the network and fly in a three-dimensional space. These drones include similar energy source, computing power, and storage capacity. In each UAV, the wireless interface in the Mac layer includes IEEE 802.11n that has suitable features such as high throughput, high data rate, support from long-range communication, high resistance to interference, and uniform coverage within an area. The flying nodes move very fast, and distance between them varies at any moment. Each UAV consists of a special ID. They are equipped with a localization mechanism like global posi-
tioning system (GPS). Therefore, UAVs know their position and speed at any moment. In QFAN, the network model utilizes two connection types, namely UAV-to-UAV and UAV-to-GS. The system model is illustrated in Fig. 3.

Proposed method

In this section, we present a Q-Learning routing approach called QFAN in FANETs. QFAN improves AODV protocol using Q-learning algorithm. Note that AODV is a well-known reactive algorithms defined in ad hoc networks because of its main property (i.e. it is on-demand). Today, many algorithms such as are inspired by AODV because it is valuable and useful. However, AODV cannot be applied in FANETs because it does not consider the unique properties of these networks, for example high-mobile UAVs, and frequent disconnections. Therefore, we try to improve this routing protocol to be suitable for FANETs. In QFAN, we consider different parameters such as link quality, motion direction, route delay, distance between nodes, and energy to create stable paths with low delay and rise the packet delivery rate (PDR). QFAN seeks to extend the network lifetime by distributing the consumed energy between network nodes uniformly. QFAN involves two parts: 1) Route discovery 2) Route maintenance. In the following, we illustrate each part item by item.

Route discovery. In QFAN, when a source node \( U_i \) wants to build a connection link between itself and the destination node \( U_d \) for exchanging data packets, \( U_i \) examines its routing table for achieving a reliable path. If \( U_i \) cannot obtain a valid path between itself and \( U_d \), it uses a Q-Learning-based routing operation to discover the best path \( \text{BestRoute} \) for exchanging data packets. In this process, each flying node \( U_i \) exchanges Hello messages with its neighboring UAVs at regular time intervals. This message contains spatial information \( (x'_i, y'_i, z'_i) \), velocity vector \( (v'_{x,i}, v'_{y,i}, v'_{z,i}) \), delay information, Q-value, and energy \( (E_i') \) at the time \( t \). \( U_i \) stores information about its neighbors in a neighboring table. Based on information stored in this table, \( U_i \) calculates the filtering parameter \( S_{\text{filtering}} \) for each neighboring node such as \( U_j \). This parameter helps \( U_i \) to filter a number of its neighbors that are not in good conditions for rebroadcasting RREQ, and put other neighbors in a set of licensed neighbors \( \text{Set}_{\text{licensed}} \). For calculating \( S_{\text{filtering}} \), \( U_i \) uses four parameters, including motion direction, remaining energy, link quality, and distance.

Motion direction \( \theta_{i,j} \). \( U_i \) calculates the angle between itself and \( U_j \) at the time \( t \). If the movement direction of two nodes \( U_i \) and \( U_j \) is similar (i.e. the angle between \( U_i \) and \( U_j \) is zero), \( U_i \) increases \( S_{\text{filtering}} \) corresponding to \( U_j \) to achieve more chance to be placed in \( \text{Set}_{\text{licensed}} \) because the communication link between \( U_i \) and \( U_j \) is valid for a longer time. Thus, more stable paths are created. Equation (2) calculates the angle between the motion directions of \( U_i \) and \( U_j \).

\[
\theta_{i,j} = \cos^{-1} \left( \frac{v'_{x,i}v'_{x,j} + v'_{y,i}v'_{y,j} + v'_{z,i}v'_{z,j}}{|V'_i||V'_j|} \right), \quad 0 \leq \theta \leq \pi, \tag{2}
\]

where \( (v'_{x,i}, v'_{y,i}, v'_{z,i}) \) and \( (v'_{x,j}, v'_{y,j}, v'_{z,j}) \) are velocity vectors of \( U_i \) and \( U_j \), respectively. In addition, \( |V'_i| \) and \( |V'_j| \) indicate the length of these vectors. They are obtained through Eqs. (3) and (4), respectively.

\[
|V'_i| = \sqrt{(v'_{x,i})^2 + (v'_{y,i})^2 + (v'_{z,i})^2}. \tag{3}
\]

Also,
\[
|V_j^i| = \sqrt{(v_{x,j}^i)^2 + (v_{y,j}^i)^2 + (v_{z,j}^i)^2}.
\]  

(4)

Remaining energy (\(E^i_j\)). \(U_i\) increases \(S_{\text{filtering}}\) corresponding to high-energy neighboring nodes to achieve more chance for being placed in \(\text{Set}_{\text{licensed}}\). The purpose of choosing \(E^i_j\) is to distribute energy consumption uniformly in the network. According to this parameter, high-energy nodes are responsible for discovering new paths and forming stable paths.

Link quality (\(Q^i_j\)). \(U_i\) increases \(S_{\text{filtering}}\) corresponding neighbors that include a better connection quality compared to others. This increases their chances for being placed in \(\text{Set}_{\text{licensed}}\). The purpose of choosing \(Q^i_j\) is to construct high-quality paths. When there is a low-quality link between \(U_i\) and \(U_j\), the path is unstable. This causes path failure. In QFAN, we assume that the link quality is evaluated using the received signal strength indication (RSSI). It can exactly assess link quality because research literature show that the higher RSSI leads to a better packet delivery rate (PDR) in the receptor and the transmitter\(^{39,40}\). Also, RSSI is stable for a short time period (about 2 s) and its standard deviation is close to 1 dBm\(^41\). In our scheme, we assume that RSSI registers are embedded in radio transceivers to calculate the signal strength when receiving packets.

Distance (\(D^i_{ij}\)). If the distance between \(U_i\) and its neighbor such as \(U_j\) is suitable, \(U_i\) increases \(S_{\text{filtering}}\) corresponding to \(U_j\). Thus, \(U_j\) achieves more chance for being in \(\text{Set}_{\text{licensed}}\). The suitable distance means that \(U_i\) and \(U_j\) are not very close to each other because if \(U_i\) is selected as relay node, it rises the hop count in the path, and this issue does not help \(U_j\) to find the path. On the other hand, the suitable distance means that \(U_i\) and \(U_j\) are not very far away from each other because these two nodes quickly get out from each other communication range. Therefore, the path created by these nodes will be invalid quickly. The suitable distance between the two nodes is called \(D^i_{ij}\) and is between \(D_{\text{Min}}\) and \(D_{\text{Max}}\), where \(0 \leq D_{\text{Min}} < D_{\text{Max}}\) and \(0 < D_{\text{Max}} \leq R\). \(R\) indicates the transmission radius of UAV\'s in the network. In this case, \(U_i\) and \(U_j\) can form stable paths. \(D^i_{ij}\) is the Euclidian distance between \(U_i\) and \(U_j\) that is computed through Eq. (5):

\[
D^i_{ij} =  \begin{cases} 
1 - \dfrac{\sqrt{(x^i_i - x^j_j)^2 + (y^i_i - y^j_j)^2 + (z^i_i - z^j_j)^2}}{D_{\text{Min}}} , & 0 \leq D^i_{ij} < D_{\text{Min}} \\
1 , & D_{\text{Min}} \leq D^i_{ij} \leq D_{\text{Max}} \\
1 - \dfrac{\sqrt{(x^i_i - x^j_j)^2 + (y^i_i - y^j_j)^2 + (z^i_i - z^j_j)^2}}{D_{\text{Max}}} , & D_{\text{Max}} < D^i_{ij} \leq R 
\end{cases}
\]

where \((x^i_i, y^i_i, z^i_i)\) and \((x^j_j, y^j_j, z^j_j)\) indicate the spatial coordinates of \(U_i\) and \(U_j\), respectively.

After calculating these parameters, \(U_i\) calculates \(S_{\text{filtering}}\) corresponding to \(U_j\) through Eq. (6):

\[
S_{\text{filtering}} = \omega_1 \left(1 - \dfrac{\theta^i_{ij}}{\pi}\right) + \omega_2 \left(\dfrac{E^i_j - E_{\text{Min}}}{E_{\text{Max}} - E_{\text{Min}}}\right) + \omega_3 \left(\dfrac{Q^i_{ij} - Q_{\text{Min}}}{Q_{\text{Max}} - Q_{\text{Min}}}\right) + \omega_4 \left(D^i_{ij}\right),
\]

(6)

where \(E_{\text{Min}} \geq 0\) indicates the minimum energy (20% of initial energy) and \(E_{\text{Max}} > 0\) is the primary energy of UAV\'s. In addition, \(Q_{\text{Min}} \geq 0\) and \(Q_{\text{Max}} > 0\) are the least link quality and the highest link quality between two nodes, respectively. According to\(^{39}\), PDR is zero when RSSI = 0. Also, PDR is about 99% and when RSSI = \(-87\) dBm. We consider these values as \(Q_{\text{Min}}\) and \(Q_{\text{Max}}\), respectively. \(\omega_1, \omega_2, \omega_3,\) and \(\omega_4\) are weight coefficients, and \(\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1\). These coefficients indicate the effect of each factor on \(S_{\text{filtering}}\) in Eq. (6). The important point is that the sum of all weight coefficients must be one. This paper considers similar values for these coefficients, i.e. \(\omega_1 = \omega_2 = \omega_3 = \omega_4 = \frac{1}{4}\). Analysis of the effect of these coefficients is outside the scope of this paper.

Now, we have standardized \(S_{\text{filtering}}\) using Eq. (7). It evaluates the number of standard deviations by which \(S_{\text{filtering}}\) is above or below the mean value. \(S_{\text{filtering}}\) above the mean has a positive value, while \(S_{\text{filtering}}\) below the mean has a negative value. A standardized value is computed by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. For more details, refer to\(^{42}\).

\[
S_{\text{Standard}} = \dfrac{S_{\text{filtering}} - \mu}{\sigma},
\]

(7)

\(\mu\) and \(\sigma\) indicate the mean value and standard deviation of \(S_{\text{filtering}}\) of neighbors, respectively:

\[
\mu = \dfrac{1}{n_{\text{neighbor}}} \sum_{i=1}^{n_{\text{neighbor}}} S_{\text{filtering}_i},
\]

(8)
where $n_{\text{neighbor}}$ indicates the number of neighbors of $U_i$.

Now, if $S_{\text{standard}} \geq 0$, $U_i$ is placed in $\text{Set}_{\text{Licensed}}$. In QFAN, each RREQ message plays the role of an agent. The RREQ format in QFAN is similar to that in the AODV protocol. However, it includes two additional fields: $F_{\text{Route}}$ and $T_{\text{Route}}$. $F_{\text{Route}}$ represents the route fitness, which is in $[0, 1]$. It will be equal to the least value of the filtering parameter of nodes that create a path. This parameter is calculated in accordance with Eq. (10):

$$F_{\text{Route}} = \min_{U_j \in \text{Route}_{\text{neg}}} \left( S_{\text{filtering}} \right), \quad j = 1, \ldots, M_f,$$

where $M_f$ is the number of $U_j$ in $\text{Route}_{\text{neg}}$.

$T_{\text{Route}}$ indicates the route delay, and its value is obtained from the total time needed to forward RREQ from $U_S$ to $U_D$. This parameter is calculated using Eq. (11):

$$T_{\text{Route}} = \sum_{i = 1}^{M_i} t_{i,j+1},$$

where $M_i$ is the number of flying nodes in $\text{Route}_{i}$. Also, $t_{i,j+1}$ indicates the single-hop delay between two nodes, including $U_i$ and $U_{i+1}$. This is the time needed to forward RREQ from $U_i$ to $U_{i+1}$, we use the equation presented in29. According to29, $t_{i,j+1}$ is calculated by two parameters, including the medium access delay ($D_{\text{mac},i+1}$) and the queuing delay ($D_{\text{queue},i+1}$). $D_{\text{mac},i+1}$ is the time needed for the medium access protocol to deliver this message or delete the duplicated message. $D_{\text{queue},i+1}$ illustrates the time interval required for the message to reach the head of the transfer queue. Note that29 does not consider the propagation delay because this delay is very short and depends on the light speed. As a result, $t_{i,j+1}$ is expressed based on Eq. (12):

$$t_{i,j+1} = D_{\text{queue},i+1} + D_{\text{mac},i+1}.$$

Note that our learning system regards the network as the environment. The state of agent represents the flying node (i.e. $U_i$), which has received the RREQ message. In this process, the action represents a set of licensed neighboring nodes (i.e. $U_{\text{neg}}, \in \text{Set}_{\text{Licensed}}$), which can receive the RREQ message in the current state (i.e. $U_i$). This set is displayed as $A = \{ U_i \rightarrow U_{\text{neg}}, U_i \rightarrow U_{\text{neg}}, \ldots, U_i \rightarrow U_{\text{neg}} \}$. When the agent (i.e. the RREQ message) is in the state $U_i$ and takes the action $U_i \rightarrow U_{\text{neg}}$, the state is converted into $U_{\text{neg}}$. When starting the route discovery operation, we initialize different Q-Learning factors, like the learning rate ($\alpha$), discount factor ($\gamma$) and Q-value.

In the following, we demonstrate how to obtain the reward function. It is a major component of Q-Learning. The reward function indicates environmental feedback relative to the action $a_t$ taken in the state $U_t$. The proposed learning model must maximize the reward value when transferring RREQ from $U_S$ to $U_D$. In QFAN, we consider three scales, including route time ($T_{\text{Route}}$), the number of hops ($\text{hopcount}$), and the route fitness ($F_{\text{Route}}$), to design the reward function for discovering the best path ($\text{BestRoute}$) to transfer data packets from $U_S$ to $U_D$. Thus, $\text{BestRoute}$ is a route that has the highest fitness, the lowest hops, and the lowest delay.

- $F_{\text{Route}}$: When $F_{\text{Route}}$ is close to one, the discovered route consists of high-energy nodes and high-quality links. Moreover, the distance between these nodes is suitable and their movement direction is similar. This guarantees that the created path is stable for longer time.
- $\text{hopcount}$: When the number of hops is low in one path, that path has less delay when transferring data packets. It enhances the performance of QFAN.
- $T_{\text{Route}}$: It is an essential factor when finding various paths in FANETs because the high velocity of flying nodes causes link failure and unstable paths. When delay is high in a path, the paths may be disconnected before transferring data to $U_D$. This increases packet loss in the network.

Therefore, the reward function is calculated according to Eq. (13):

$$\text{Reward}(U_t, a_t) = \begin{cases} R_{\text{max}}, & U_{t+1} \text{ is destination.} \\ R_{\text{min}}, & U_{t+1} \text{ is local minimum.} \\ 1 - \frac{T_{\text{Route}}}{\text{max}(T_{\text{Route}})} + \left( 1 - \frac{\text{hopcount}}{N+1} \right) + F_{\text{Route}}, & \text{Otherwise} \end{cases},$$

where $T_{\text{Route}}$ indicates delay in the path from $U_S$ to $U_{t+1}$. $\text{max}(T_{\text{Route}})$ is the maximum delay in the discovered paths to $U_{t+1}$. Moreover, $\text{hopcount}$ indicates the number of hops in the path. This parameter is inserted in RREQ. $N$ is the total number of UAVs. With regard to the reward function, if the next-hop node is $U_D$, the link between $U_t$ and $U_{t+1}$ acquires the most reward value. On the other hand, when a local minimum occurs, i.e. among the states that can be selected, those that are farther away from $U_D$ compared to the current node acquire the minimum reward. In other modes, the reward function is evaluated using the route delay, the number of hops and the route fitness.

In QFAN, the convergence condition is the time required to find the best path ($\text{BestRoute}$) by the learning algorithm. In our scheme, when the learning algorithm are repeated five steps and the response remains unchanged,
this means that the algorithm has converged and achieved the optimal response ($\text{BestRoute}$). Then, $U_S$ records the novel path in its routing table. Algorithm 1 expresses the pseudo-code of this operation.

### Algorithm 1 Q-learning-based route discovery mechanism

**Input:** RREP message  
RREQ message  
$U_S$  
$U_D$  
$U_i, i = 1, 2, ..., N$  
$N$: Total number of flying nodes.  

**Output:** $\text{BestRoute}$

**Begin**

1: for $i = 1$ to $N$ do
2: $U_i$: Disseminate a hello message for the neighboring UAVs;
3: $U_i$: Store the information of its neighboring nodes in the neighborhood table;
4: $U_i$: Calculate $S_{\text{filtering}}$ for each single-hop neighbors according to Equation 6;
5: $U_i$: Standardize $S_{\text{filtering}}$ using Equation 7;
6: for $j = 1$ to $n_{\text{neighbor}}$ do
7: if $S_{\text{Standard}}_j \geq 0$ then
8: $U_i$: Insert $U_j$ in $\text{Set}_{\text{Licensed}}$;
9: end if
10: end for
11: end for
12: if $U_S$ wants to transmit the packets to $U_D$ then
13: $U_S$: Examine the routing table to earn an appropriate path to $U_D$;
14: if there is an appropriate path to $U_D$ then
15: $U_S$: Transmit the data packets to $U_D$ via this path;
16: else
17: while the convergence condition is not met do
18: if there is no information about neighbors then
19: Initialize all Q-learning factors, for example Q-table, learning rate, and discount factor;
20: end if
21: Select a neighbor node with maximum Q-value from $\text{Set}_{\text{Licensed}}$;
22: Update the Q-value corresponding to the neighbor in the Q-table according to the reward value;
23: end while
24: $U_S$: Send its data packet to $U_D$ trough $\text{BestRoute}$;
25: end if
26: end if
End

**Route maintenance.** The existing paths may be disconnected because of the specific properties of FANETs, like the high-speed UAVs and the very dynamic topology. In the route maintenance process, the goal is to immediately detect a failed path and replace a new path for transferring data packets. When one of the following modes is met, the created path should be corrected to avoid its failure:

- **Mode 1:** When $U_i$ is dying in a path, meaning that its energy level is lower than the energy threshold $E_{\text{Threshold}}$ (i.e. $E_{U_i} < E_{\text{Threshold}}$). Thus, this path should be replaced because $U_i$ cannot continue the data transfer process.
- **Mode 2:** When the traffic level of $U_i$ in one path is more than the traffic threshold $T_{\text{Threshold}}$ (i.e. $T_{U_i} > T_{\text{Threshold}}$), its buffer capacity is at the overflow state. Therefore, the path has very high delay when transferring data because the path is blocked. Therefore, this path should be replaced because it cannot forward data packets.
• **Mode 3:** When the communication link between $U_i$ and $U_j$ is disconnecting in a path, meaning that its connection quality is lower than the quality threshold $Q_{Threshold}$ (i.e., $Q_{Link_{i-j}} < Q_{Threshold}$). Therefore, this path should be corrected.

If one of the three modes occurs, the reward corresponding to $U_i$ will be equal to $R_{min}$. Then, $U_i$ searches its routing table and finds the paths that include itself. $U_i$ sends a feedback containing the reward value to its previous-hop nodes. Then, the previous-hop node executes the Q-Learning algorithm to pick out a new UAV as the next-hop node and modify the path. Note that the old route is valid and used for transferring data packet until the new path is discovered and replaced. After creating the new path, the old path is removed and data packets are sent through the new path.

To discover and replace failed paths, UAVs must regularly control the paths recorded in their routing table. To do this, $U_i$ regularly transmits the path control message to $U_D$ via the created path. If $U_D$ receives this message successfully, this path is still valid. Then, $U_D$ sends the ACK message to $U_S$. Otherwise, if $U_D$ does not receive the message on time, $U_S$ concludes that the path is cut off. At this step, the reward value related to the failed UAV will be equal to $R_{min}$. This node sends this feedback to its previous-hop UAV to correct the failed path. Therefore, the previous-hop node runs the Q-Learning algorithm to specify a new UAV as the next-hop node and make a new path. Algorithm 2 presents the pseudo-code of the route maintenance operation.

**Algorithm 2** The route maintenance process

| Input: | $U_i,\ i=1,2,\ldots,N$ |
| N: Total number of flying nodes. |
| Output: | Altered path |
| Begin |
| 1: for $i = 1$ to $N$ do |
| 2: if $E_{U_i} < E_{Threshold}$ or $Tr_{U_i} > Tr_{Threshold}$ or $Q_{Link_{i-j}} < Q_{Threshold}$ then |
| 3: $U_i$: Set its reward value as $R_{min}$; |
| 4: $U_i$: Update its Q-value; |
| 5: $U_{prev-hop}$: Select $U_{alternative}$ as its next hop node in the routing table; |
| 6: end if |
| 7: end for |
| 8: $U_S$: Transfer a path control message to $U_D$; |
| 9: if this path is credible then |
| 10: $U_D$: Transmit an ACK message to $U_S$; |
| 11: Go to line 8; |
| 12: else |
| 13: $U_S$: Set the reward value corresponding to failed UAV as $R_{min}$; |
| 14: $U_S$: Update the Q-table; |
| 15: $U_S$: Select $U_{alternative}$ as its next hop node; |
| 16: end if |
| End |

**Simulation and result evaluation**

In this section, QFAN is implemented using the network simulator version 2 (NS2) to analyze its results. For this purpose, QFAN is compared with three routing schemes, namely QTAR, GPSR and QGeo in terms of five parameters, including packet delivery rate (PDR), average end-to-end delay (Average E2E), communication overhead, consumed energy, and network lifetime. In the simulation process, the three-dimensional Gauss Markov mobility model (3D GM) simulates drone movement in the network environment. The network dimensions are $2000 \times 2000 \times 300$ m$^3$. Furthermore, the number of UAVs changes between 20 and 100, and their velocities are different from 10 m/s to 40m/s. The initial energy of UAVs is 1000J, and their communication radius is 250m. In the simulation process, the IEEE 802.11n standard is regarded as a wireless interface in the MAC layer because this standard has suitable features such as high throughput, high data rate, support from long-range communication, high resistance to interference, and uniform coverage within an area. Table 2 presents the most important simulation parameters in summary.

**Average end-to-end delay.** The average end-to-end delay (Average E2E) is the average required time for successfully transmitting data packets between source and destination nodes. Figure 4 compares the average E2E in various approaches when changing the density of UAVs (between 20 and 100 nodes). Figure 4, shows if
Table 2. Simulation parameters.

| Parameter                  | Value                                           |
|----------------------------|-------------------------------------------------|
| Simulator                  | NS-2.35                                         |
| Environment size           | $2000 \times 2000 \times 300 \text{ m}^3$      |
| The number of UAVs         | 20–100                                          |
| UAV speed (m/s)            | 10–40                                           |
| Primary energy of UAVs (J) | 1000                                           |
| Mobility model             | 3D GM                                           |
| Carrier frequency (GHz)    | 2.4                                            |
| Transport protocol         | UDP                                            |
| Communication radius (m)   | 250                                            |
| Mac layer protocol         | IEEE 802.11n                                    |
| Antenna                    | Omni-directional                                |
| Propagation model          | Free-space                                      |
| Traffic model              | Constant bit rate (CBR) traffic                |
| CBR rate (Mbps)            | 2                                              |

Figure 4. Comparison of end-to-end delay in various approaches in relation to the density of UAVs.

Figure 5. Comparison of end-to-end delay in various approaches in relation to the speed of UAVs.
there are a small number of UAVs in the network, delay is also low in all routing methods. However, delay will be high when growing the number of UAVs due to the time needed to create paths in the network. Note that our scheme has the minimum delay compared to QTAR, QGeo, and GPSR. QFAN reduces delay by 13.33%, 43.48% and 60.6% compared to QTAR, QGeo, and GPSR, respectively. Moreover, Fig. 5 evaluates delay in different approaches according to speed changes (10–40 m/s). This figure shows the successful performance of our method compared to other schemes. As shown in Fig. 5, QFAN lowers delay by 25%, 52.63% and 68.97% compared to QTAR, QGeo, and GPSR, respectively. According to this figure, we find that delay is increased in all routing approaches when increasing the speed of UAVs in the network because this issue reduces the stability of communication links, which leads to the rapid disconnection of existing routes between nodes and increases the need for finding new paths. This issue increases delay in the network. In general, based on Figs. 4 and 5, it can be found that QFAN has a desirable delay because it considers delay for discovering paths and chooses the paths with less delay for transferring data. In contrast, GPSR focuses only on distance between nodes to choose the next-hop node, this is why it has the worst delay. QTAR considers delay in the routing process and has better performance than QGeo and GPSR. Although, this method has longer delay than QFAN because it requires information of its two-hop neighboring UAVs. On the other hand, QGeo uses spatial information and speed of nodes to choose the next-hop node in the routing operation, but it does not regard delay in this operation. This has increased delay in this method compared with QFAN.

**Packet delivery rate.** The packet delivery rate (PDR) is defined as the ratio of packets delivered at the destination to the produced packets. Figure 6 illustrates PDR in various approaches according to the density of UAVs. In accordance with this figure, when growing the number of UAVs, the connections between nodes will be improved. This increases PDR in all routing methods. QFAN has the highest PDR compared to other schemes. It increases PDR by 4.45%, 12.01%, and 20.20% in comparison with QTAR, QGeo, and GPSR, respectively. Fig-
Figure 7 evaluates various methods in terms of PDR in relation to the speed of nodes. Based on this figure, QFAN has improved this parameter by 4.64%, 14.73%, and 29.03% compared to QTAR, QGeo, and GPSR, respectively. When rising the speed of UAVs, the packet delivery rate decreases in all routing schemes because in this case, the paths made between nodes will be unstable, and the probability of path failure is high. As shown in Figs. 6 and 7, QFAN has a good performance in terms of PDR because it rebuilds the failed paths to lower packet loss due to the path breakage. QFAN considers energy, movement direction, and link quality to form stable paths and reduce path breakages in the network. This will increase PDR in QFAN. In contrast, GPSR and QGEO do not pay attention to energy and link quality. This can increase the probability of path failure, which has lowered PDR in these methods. QTAR has a good performance in terms of PDR. However, its performance is weaker than our method because QFAN uses global information such as the path delay, the path energy, and hop count to calculate the best path, but QTAR depends on local information, such as latency, velocity, and energy of nodes to select the paths.

Routing overhead. Routing overhead means all control packets sent by the routing method. Figure 8 analyzes this parameter in different approaches with regard to the density of UAVs. In this figure, when there are the small number of UAVs, routing overhead is decreased in all routing protocols. When there are the high number of UAVs, they exchange the high number of control messages in the routing operation. According to Fig. 8, QFAN has a higher routing overhead than QTAR (about 11.49%), QGeo (67.79%), and GPSR (30.72%). Figure 9 evaluates different approaches in terms of routing overhead with regard to the speed of UAVs. In accordance with this figure, when the speed of UAVs is increasing in all approaches, routing overhead is high because in this case, the unstable routes are created in the network and the probability of the path failure is very high. As shown in Fig. 9, QFAN has grown routing cost by 9.83%, 32.91%, and 14.66% compared to QTAR, QGeo, and GPSR, respectively. Regarding Figs. 8 and 9, we find that our method has a weaker performance than other methods in
terms of routing overhead. This is rooted in the exchange of RREQ and RREP messages when discovering paths in the network. While QTAR and QGeo do not use these control messages in their routing process.

**Energy consumption.** This parameter has a great influence on network lifetime and indicates the amount of energy consumed by each UAV when communicating with other nodes. Figure 10 compares energy consumption in different approaches with regard to the density of UAVs. According to Fig. 10, UAVs consume more energy when increasing the density of UAVs because they exchange more control messages for building paths. According to this figure, QFAN has the best performance in terms of energy consumption in comparison with other methods and improves this parameter by 14.75%, 31.76%, and 45.05% compared with QTAR, QGeo, and GPSR, respectively. In addition, Fig. 11 evaluates various methods in terms of energy consumption with regard to the velocity of UAVs. This evaluation shows that QFAN has improved the average energy consumption by 13.14%, 30.73%, and 47.8% compared to QTAR, QGeo, and GPSR, respectively. All methods consume more energy when increasing the speed of UAVs because this issue increases instability of the created paths, which leads to the higher need for rebuilding of failed paths. GPSR and QGeo do not pay attention to the energy of nodes when discovering paths. As a result, they may create low-energy paths, which reduces network lifetime. In contrast, QFAN and QTAR consider energy of UAVs in the routing operation. This balances energy consumption in the network and enhances network lifetime. However, QTAR consumes more energy than QFAN because it needs information of two-hop neighbors. Additionally, QFAN considers different factors such as distance, motion direction, remaining energy, connection quality, number of hops, and delay in the routing operation to build stable paths. Thus, QFAN can reduce path breakages, which improves energy consumption in the network.

**Network lifetime.** The network lifetime indicates the entire routing time until the first node dies (FND) in the network. Figure 12 shows network lifetime in different approaches in relation to the density of UAVs. Accord-
ing to this figure, QFAN has the best network lifetime in comparison with other approaches. It has increased this parameter by 12.41%, 36.45%, and 95% compared to QTAR, QGeo, and GPSR. Figure 13 evaluates various methods in terms of network lifetime with regard to the speed of UAVs. This assessment shows that QFAN has improved this parameter by 7.92%, 24.81% and 36.82% compared to QTAR, QGeo, and GPSR because the proposed method balances energy consumption in the network. As a result, QFAN enhances the network performance in terms of network lifetime. We have described the reasons for this issue in "Energy consumption".

Conclusion
In this paper, an energy-aware Q-learning-based routine method (QFAN) was presented for smart air quality monitoring system using FANETs. QFAN involves two parts: route discovery and route maintenance. In the first part, QFAN uses a Q-Learning-based path discovery mechanism to select the best path between source and destination nodes. In this mechanism, each node calculates a filtering parameter for each neighboring UAV. The node chooses a set of licensed neighbors based on this parameter and restricts the search space to quickly achieve the final response (the best path). In the route maintenance phase, QFAN firstly tries to discover and correct paths close to failure. In the second stage, QFAN quickly identifies and replaces the failed paths. Finally, QFAN was simulated by NS2 and compared with QTAR, QGeo, and GPSR in terms of end-to-end delay, packet delivery rate, energy consumption, network lifetime, and routing overhead. We have analyzed the simulation results with regard to density of UAVs and speed changes. The simulation results based on the density of nodes show that our scheme decreases delay compared to QTAR (13.33%), QGeo (43.48%), and GPSR (60.6%). Additionally, QFAN has higher PDR than QTAR (4.45%), QGeo (12.01%), and GPSR (20.20%). Furthermore, our scheme has the minimum energy consumption in comparison with QTAR (14.75%), QGeo (31.76%), and GPSR (45.05%). Moreover, it has the best network lifetime compared to QTAR (12.41%), QGeo (36.45%), and GPSR (95%). However, the routing overhead has been increased slightly in QFAN compared with QTAR (about

![Figure 12. Comparison of network lifetime in various approaches with regard to the density of UAVs.](image)

![Figure 13. Comparison of network lifetime in various approaches with regard to the speed of UAVs.](image)
References

1. Simo, A., Dzitac, S., Dzitac, I., Frigura-Iliasa, M. & Frigura-Iliasa, F. M. Air quality assessment system based on self-driven drone and LoRaWAN network. *Comput. Commun.* 175, 13–24. https://doi.org/10.1016/j.comcom.2021.04.032 (2021).

2. Lambe, V. & Prasad, A. D. A review on air quality measurement using an unmanned aerial vehicle. *Water Air Soil Pollut.* 232(3), 1–32. https://doi.org/10.1007/s11270-020-04973-5 (2021).

3. Singh, D., Dahiyana, M., Kumar, R. & Nanda, C. Sensors and systems for air quality assessment monitoring and management: A review. *J. Environ. Manage.* 289, 112510. https://doi.org/10.1016/j.jenvman.2021.112510 (2021).

4. Javaheri, D., Hosseinzadeh, M. & Rahmani, A. M. Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access* 6, 78321–78332. https://doi.org/10.1109/ACCESS.2018.2884964 (2018).

5. Heikalabad, S. R., Navt, A. H., Hosseinzadeh, M. & Oladghaferi, T. Erratic memory structure for data-oriented models implementation. *J. Circ. Syst. Comput.* 25(02), 1692001. https://doi.org/10.1142/S021812661920018(2016).

6. Mesbah, M. R., Rahmani, A. M. & Hosseinzadeh, M. Highly reliable architecture using the 80/20 rule in cloud computing datacenters. *Future Gener. Comput. Syst.* 77, 77–86. https://doi.org/10.1016/j.future.2017.06.011 (2017).

7. Yousefpoor, E., Barati, H. & Barati, A. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer-to-Peer Netw. Appl.* 14(1), 1917–1942. https://doi.org/10.1080/21535561.2018.2011136-3 (2021).

8. Yousefpoor, M. S. & Barati, H. DSKMS: A dynamic smart key management system based on fuzzy logic in wireless sensor networks. *Wirel. Netw.* 26(4), 2515–2535. https://doi.org/10.1007/s11217-019-01980-1 (2020).

9. Shahzadi, R., Ali, M., Khan, H. Z. & Naeem, M. UAV assisted 5G and beyond wireless networks: A survey. *Scientific Reports* 9. https://doi.org/10.1038/s41598-022-20353-x (2022).

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

11. Nawaz, H., Ali, H. M. & Laghari, A. A. UAV communication networks issues: A review. *Arch. Comput. Methods Eng.* 28(3), 1349–1369. https://doi.org/10.1007/s11831-020-09418-0 (2021).

12. Alsalmi, S. H. et al. Green internet of things using UAVs in 5G networks: A review of applications and strategies. *Ad Hoc Netw.* 117, 102505. https://doi.org/10.1016/j.adhoc.2021.102505 (2021).

13. Koo, V. et al. A new unmanned aerial vehicle synthetic aperture radar for environmental monitoring. *Prog. Electromagn. Res.* 122, 245–268. https://doi.org/10.2528/PIER11092604 (2012).

14. Smidl, V. & Hofman, R. Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmos. Environ.* 67, 425–436. https://doi.org/10.1016/j.atmosenv.2012.10.054 (2013).

15. Zang, W., Lin, J., Wang, Y. & Tao, H. Investigating small-scale water pollution with UAV remote sensing technology. In *World Automation Congress 2012* 1–4. (IEEE, 2012).

16. Burgués, J. & Marco, S. Environmental chemical sensing using small drones: A review. *Sci. Total Environ.* 748, 141172. https://doi.org/10.1016/j.scitotenv.2020.141172 (2020).

17. Al Tahtawi, A. R., Andika, E., Yusuf, M. & Harjanto, W. N. Design of quadrotor UAV and internet-of-things based air pollution monitoring systems. *Int. J. Inf. Technol. Electr. Eng.* 3(4), 120–127. https://doi.org/10.22141/ijitee.2020.51203 (2019).

18. Vijitha Ananthi, J. & Subha Hendy Jose, P. A review on various routing protocol designing features for flying ad hoc networks. In *Mobile Comput. Sustain. Inform.* 315–325 (Springer, 2022). https://doi.org/10.1007/978-3-662-58502-0_23.

19. Alalbty, H., Atiquzzaman, M., Lorenz, P., Tareque, M. H. & Hossain, M. S. Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives. *IEEE Access* 7, 81057–81105. https://doi.org/10.1109/ACCESS.2019.2923840 (2019).

20. Arvedo, M. I. B., Coutinho, C., Toda, E. M., Carvalho, T. C. & Jailton, J. Wireless communications challenges to flying ad hoc networks (FANET). *Mobile Comput.* 3, 25 (2020).

21. Rezwan, S. & Choi, W. A survey on applications of reinforcement learning in flying ad-hoc networks. *Electronics* 10(4), 449. https://doi.org/10.3390/electronics10040449 (2021).

22. Yousefpoor, M. S. et al. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *J. Netw. Comput. Appl.* 190, 103118. https://doi.org/10.1016/j.jnca.2021.103118 (2021).

23. Rahmani, A. M. et al. An area coverage scheme based on fuzzy logic and shuffled frog-leaping algorithm (sfla) in heterogeneous wireless sensor networks. *Mathematics* 9(18), 2251. https://doi.org/10.3390/math9182251 (2021).

24. Yousefpoor, M. S. & Barati, H. Dynamic key management algorithms in wireless sensor networks: A survey. *Comput. Commun.* 134, 52–69. https://doi.org/10.1016/j.comcom.2018.11.005 (2019).

25. Ashraf, M. Y. & Moh, S. A Q-learning-based topology-aware routing protocol for flying ad hoc networks. *IEEE Internet Things* Journal. https://doi.org/10.1109/JIOT.2021.3089759 (2021).

26. Jung, W. S., Yim, J. & Ko, Y. R. QGeo: Q-learning-based geographic ad hoc routing protocol for unmanned robotic networks. *IEEE Commun. Lett.* 21(10), 2258–2261. https://doi.org/10.1109/LCOMM.2017.2658679 (2017).

27. Karp, B. & Kung, H. T. August. GSPR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* 243–254 (2000). https://doi.org/10.1145/345910.345953.

28. Lee, S. W. et al. An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). *IEEE Access* 9, 129977–130005. https://doi.org/10.1109/ACCESS.2021.3114444 (2021).

29. Liu, J. et al. QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks. *Comput. Commun.* 150, 304–316. https://doi.org/10.1016/j.comcom.2019.11.011 (2020).

30. Oubbati, O. S. et al. ECalD: Energy-efficient routing in flying ad hoc networks. *Int. J. Commun Syst* 32(18), e4156. https://doi.org/10.1002/ics.4156 (2019).

31. da Costa, L. A. L., Kunst, R. & de Freitas, E. P. Q-FANET: Improved Q-learning based routing protocol for FANETS. *Comput. Netw.* 198, 108379. https://doi.org/10.1016/j.comnet.2021.108379 (2021).
32. Perkins, C. E. & Royer, E. M. Ad-hoc on-demand distance vector routing. In Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications 90–100 (IEEE, 1999). https://doi.org/10.1109/MCSA.1999.749281.
33. Rahmani, A. M. et al. OLSR+: A new routing method based on fuzzy logic in flying ad-hoc networks (FANETs). Veh. Commun. 20, 100489 (2022).
34. Qiu, X., Xie, Y., Wang, Y., Ye, L. & Yang, Y. QLGR: A Q-learning-based geographic FANET routing algorithm based on multi-agent reinforcement learning. KSII Trans. Internet Inf. Syst. 15(11), 4244–4274 (2021).
35. Jang, B., Kim, M., Harettina, G. & Kim, J. W. Q-learning algorithms: A comprehensive classification and applications. IEEE Access 7, 133653–133667. https://doi.org/10.1109/ACCESS.2019.2941229 (2019).
36. Sutton, Richard & Barto, Andrew. Reinforcement Learning: An Introduction (MIT Press, 1998).
37. Rahmani, A. M. et al. Machine learning (ML) in medicine: Review, applications, and challenges. Mathematics 9(22), 2970. https://doi.org/10.3390/math9222970 (2021).
38. Stallings, W. IEEE 802.11: Wireless LANs from a to n. IT Profess. 6(5), 32–37. https://doi.org/10.1109/MITP.2004.62 (2004).
39. Baccour, N. et al. Radio link quality estimation in wireless sensor networks: A survey. ACM Trans. Sens. Netw. 8(4), 1–33. https://doi.org/10.1145/2240116.2240123 (2012).
40. Lowrance, C. I. & Lauf, A. P. Link quality estimation in ad hoc and mesh networks: A survey and future directions. Wirel. Pers. Commun. 96(1), 475–508. https://doi.org/10.1007/s11277-017-4180-9 (2017).
41. Vlavianos, A., Law, L. K., Broustis, I., Krishnamurthy, S. V. & Faloutsos, M. Assessing link quality in IEEE 802.11 wireless networks: Which is the right metric? In 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications 1–6 (IEEE, 2008). https://doi.org/10.1109/PIMRC.2008.4699837.
42. Holický, M. Introduction to Probability and Statistics for Engineers (Springer, 2013). https://doi.org/10.1007/978-3-642-38300-7.
43. Broyles, D., Jabbar, A. & Sterbenz, J. P. (2010). Design and analysis of a 3-D Gauss–Markov mobility model for highly-dynamic airborne networks. In Proceedings of the International Telemetering Conference (ITC), San Diego, CA.

Acknowledgements
The result was created through solving the student project “Security analysis and developing lightweight ciphers and protocols” using objective-oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic. Authors thank Michal Merta, and Zdeněk Truhlář for their help with the research connected with the topic of the article.

Author contributions
A.M.R., M.S.Y., and E.Y.: initial conceptualization. S.M.Z.: experimental setup. J.L., M.H.: field testing J.L., M.H., S.M.Z., V.C., F.K., A.M.R., M.S.Y., and E.Y.: wrote the main manuscript text. M.S.Y., and E.Y.: prepared figures. M.S.Y., M.H., V.C., F.K.: wrote the analysis section. M.H. and S.M.Z.: reviewed the final manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to A.M.R. or F.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022