QLGR: A Q-learning-based Geographic FANET Routing Algorithm Based on Multi-agent Reinforcement Learning

Xiulin Qiu, Yongsheng Xie, Yinyin Wang, Lei Ye , and Yuwang Yang*
[1] School of Computer Science and Engineering,
Nanjing University of Science and Technology,
Nanjing, China.
[e-mail: yuwangyang@njust.edu.cn]
*Corresponding author: Yuwang Yang

Received March 24, 2021; revised May 24, 2021; revised September 14, 2021; revised October 10, 2021; accepted October 10, 2021; published November 30, 2021

Abstract

The utilization of UAVs in various fields has led to the development of flying ad hoc network (FANET) technology. In a network environment with highly dynamic topology and frequent link changes, the traditional routing technology of FANET cannot satisfy the new communication demands. Traditional routing algorithm, based on geographic location, can "fall" into a routing hole. In view of this problem, we propose a geolocation routing protocol based on multi-agent reinforcement learning, which decreases the packet loss rate and routing cost of the routing protocol. The protocol views each node as an intelligent agent and evaluates the value of its neighbor nodes through the local information. In the value function, nodes consider information such as link quality, residual energy and queue length, which reduces the possibility of a routing hole. The protocol uses global rewards to enable individual nodes to collaborate in transmitting data. The performance of the protocol is experimentally analyzed for UAVs under extreme conditions such as topology changes and energy constraints. Simulation results show that our proposed QLGR-S protocol has advantages in performance parameters such as throughput, end-to-end delay, and energy consumption compared with the traditional GPSR protocol. QLGR-S provides more reliable connectivity for UAV networking technology, safeguards the communication requirements between UAVs, and further promotes the development of UAV technology.

Keywords: FANET, GPSR, dynamic environment, multi-agent reinforcement learning, local information.

This work was supported in part by the National Defense Technology Foundation Research Project under Grant JCKY201760**003 and Grant JCKY201860**001.in part by the Key Technology and General Program of Jiangsu Province under Grant BE2018393, and in part by the Key Industrial Technology Innovation Project of Suzhou City under Grant SYG201826.

http://doi.org/10.3837/tiis.2021.11.020 **ISSN : 1976-7277**

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been used widely in military, civil and other fields, such as search and rescue in disaster areas, battlefield situational awareness, smart cities, scientific expeditions, etc. as shown in **Fig. 1**, a robust communication network is an important foundation for UAVs to complete complex, collaborative tasks [1-5]. The introduction of mobile ad hoc networks expanded the research scope of UAV communication networks and, thus, flying ad hoc networks (FANET) were developed [6-8]. MANET routing protocols have matured after several years of development. The routing protocols in FANET should be able to adapt to the dynamic changes of the network. The stability of routing, traffic load, and transmission scheduling is the basis for achieving a mobile tolerable network. FANET routing protocols should take into account the application, deployment, service nature, and mobility model of UAV. The routing protocols in FANET originate from the following two One is the proposed routing suitable for the FANET scenario, and the other is the reasonable improvement of routing protocol based on MANET, the latter is more common. In recent years, many improved routing protocols have been gradually developed around node mobility awareness and MAC layer and network layer traffic load awareness. They improve the existing MANET routing protocols by appropriately defining mobility characteristics and closely linking mobility-aware results, QoS(Quality of Service) requirements, and network effectiveness assessment. To solve the problem of link instability caused by the rapid change of FANET topology, the improvement of routing protocols is gradually developed in the direction of node mobility awareness and cross-layer optimization, and self-adaptation.

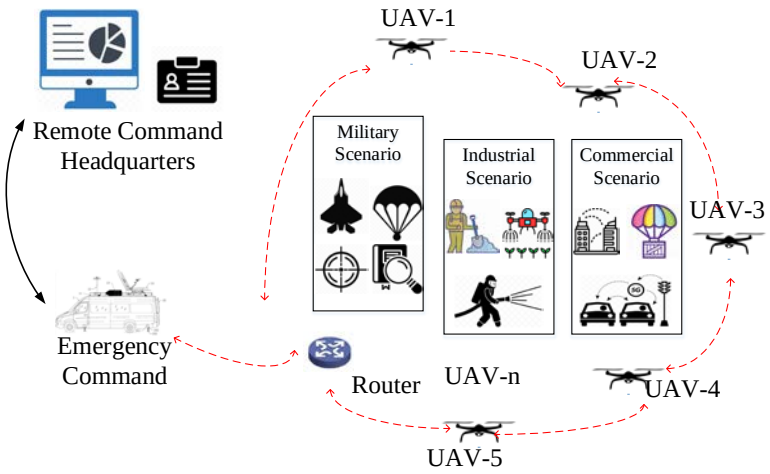


Fig. 1. A structure of FANET's application

With the development of geolocation technology, the miniaturization and cost reduction of positioning equipment, coupled with the importance of precise geolocation for military and civilian applications, has led to the widespread equipping of GPS in UAVs. The acquisition of geographic location information is a basic function of a UAV, because almost all UAV systems need geolocation information to realize path planning, especially in a UAV swarm [9]. Geolocation-based routing protocols can be divided into two categories according to the presence or absence of a route discovery process: location-assisted routing protocols and location-based routing protocols. Location-assisted routing protocols are similar to routing protocols that do not use location information and look for available routes before sending

data. The role of location information is that it is used in the route discovery process to limit the propagation range of route requests, replacing the normal flooding sending method with limited flooding and significantly reducing the network routing overhead. Location-based routing protocols do not require route discovery and choose the forwarding path based on the location information of network nodes when sending packets, and only need neighbor information and information about itself and destination nodes to complete the forwarding of packets, and the forwarding strategy can adopt greedy forwarding, limited flooding, hierarchical forwarding, etc.

The typical representatives of geolocation-based routing protocols are LAR, GeoCast, and DREAM protocols. One scheme of LAR protocol limits the sending range of route requests to a rectangular area or a sector, and the other one selects the next hop based on the principle that it is closer and closer to the destination node. The above two approaches effectively reduce the number of route control packets and the number of forwarding and lower the routing overhead. Greedy perimeter stateless routing (GPSR) [10] is a widely used protocol in the category of geolocation-based routing protocols that requires neither knowledge of the entire network topology nor routing discovery when used, making it ideal for use in a FANET. GPSR works mainly through two forwarding models, that is, under normal circumstances, the greedy mode is used to deliver data as close as possible to the destination node, but when an empty region is encountered, the mode switches to a peripheral one to forward the data.

Traditional geographic location routing considers only the distances between nodes when forwarding and, thus, not other attributes of neighbor nodes (for example, the quality of the link between one node and a neighbor node, and the traffic load of the neighbor). The node closest to the destination node is selected (in a “greedy” manner) when deciding on the next hop; long-term transmission reliability and feasibility are not considered. It is easy to fall into a local optimum and thus cause transmission to fail. When GPSR enters the peripheral forwarding mode, data packets must traverse the entire network to attain the destination node, which greatly increases the network delay and the routing packet loss rate, posing major challenges when seeking to apply GPSR to FANET.

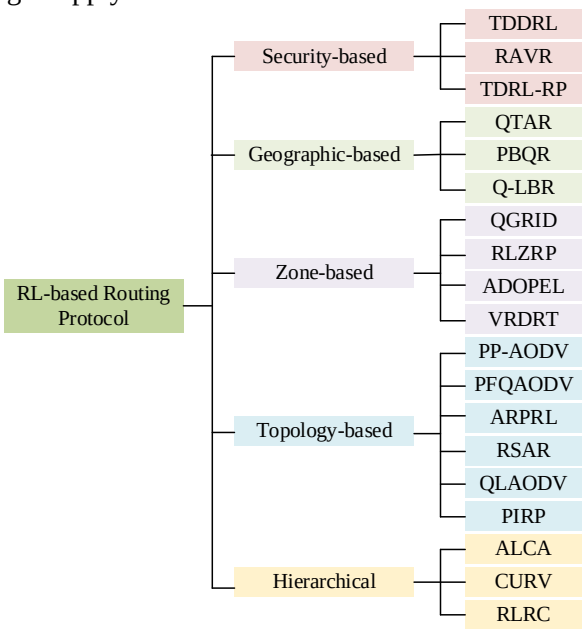


Fig. 2. Reinforcement learning-based routing protocol for FANET

Thus, Gunduz [11,12] and others have used machine-learning to design routing algorithms. The basic idea of reinforcement learning-based routing algorithm is to detect each path based on the information feedback effect of reinforcement learning and finally find an optimal path. For scenarios where the environment is constantly changing, reinforcement learning can continuously adapt to the surrounding environment. **Fig. 2** shows the common reinforcement learning-based FANET routing protocols. The most common reinforcement learning-based routing algorithm is the Q-learning-based routing algorithm. The basic components of Q-learning are intelligence, behavior, reward and punishment policy, and state. The best behavior policy in each state is selected by rewarding and punishing the state transfer after the intelligence takes action.

There are many routing techniques based on reinforcement learning. Li et al. [13]developed the QGrid protocol; Q-learning employs an on-vehicle self-organizing network (VANET) for geographic routing. This divides an area into several grids, each with a unique Q value. The agent selects the grid of largest Q value among adjacent grids, and then the node closest to the target in the selected grid. However, the Q table of QGrid must be trained offline; this requires a large amount of trajectory data. If these are lacking for a certain area, it is difficult to train the table. Jung et al. [14]developed the QGeo geographic routing protocol; this is also based on Q learning. QGeo uses a flexible discount factor to select reliable links, calculates the Q value for each received node, and makes routing decisions based on that Q value. However, it is difficult to apply QGeo to a general self-organizing network because the size of the Q value table increases linearly with the number of target nodes.

The intelligent routing algorithm of the Q-learning algorithm mostly models the packet forwarding process in the network with MDP, and later transforms the routing optimization problem into a model-based Q-learning problem, and constructs the intelligent routing algorithm based on it. Due to the characteristics of MDP modeling and model-based Q learning itself, the optimization objectives are mainly performance evaluation metrics that can be accumulated hop-by-hop, such as delay, throughput, and energy consumption. The intelligent routing algorithm designed by using the model-based Q learning method itself can self-adapt to the dynamically changing network environment, and because its MDP model is known, its decision process has better interpretability compared with other deep learning-based methods, so it has wider application in the application scenarios where the network state is highly volatile, such as FANET networks. However, it is very difficult to explicitly build MDP models for routing optimization problems with higher input-output dimensions and more complex optimization objectives, and in addition, the packet-level routing control approach commonly used by existing Q-learning-based routing optimization methods can hardly meet the high-performance requirements of backbone networks, so the application scenarios of existing Q-learning-based intelligent routing algorithms still have great limitations.

We use multi-agent reinforcement learning to process information on higher-dimensional, network state characteristics; the agent is sensitive to network changes and thus makes appropriate decisions. We term our multi-agent, geolocation-based, Q-learning routing algorithm Q-learning-based geographic routing (QLGR). We combine the trial-and-error approach of reinforcement learning with dynamic programming. Compared to the traditional routing strategy (based on fixed model-solving), QLGR is data-driven and spontaneously explores suitable model parameters. While considering the quality and load capacity of the next hop node, the protocol selects the best neighboring node based on geolocation information, and considers the data backhaul during packet transmission, which is based on punishment.
The main work in this paper is as follows:

(1)We propose a geolocation routing algorithm, QLGR, based on multi-agent reinforcement learning, with each node alone maintaining an intelligent routing table.

(2)We propose a distributed evaluation method for neighboring nodes, which includes only information about the local nodes, namely link quality, node energy and packet queue length.

(3)By contrast, we also introduce a global reward to reflect network performance, that is, the distance between the packet and the destination node, which makes all nodes cooperate to complete data transmission.

(4)Finally, to reduce the overhead of the routing algorithm, we propose an optimization method that adaptively adjusts the period required to broadcast a "HELLO" packet.

The remainder of this paper is as follows. In Section 2, we introduce research related to geolocation routing based on reinforcement learning. In Section 3, we model the various parts of the multi-intelligent routing system. In Section 4, we present the implementation of the routing algorithm and the optimization method of the proposed protocol. In Section 5, we discuss validation of the model by testing. In Section 6, we present a discussion and our conclusions.

2. Related Work

Location-based routing protocols typically base their routing selection on the location information between local and destination nodes. In both location-aided routing [15] and the distance routing-effect algorithm for mobility [16], the network overhead is reduced by delimitation of the expected zone and the request zone around the target node. Furthermore, GPSR [17] not only reduces the connection establishment delay but also reduces the control overhead by combining greedy and peripheral modes.

Hunag [18] proposed an energy-aware dual-path geographic routing protocol to recover routing from routing holes more effectively. This protocol adaptively utilizes location information, residual energy and energy consumption characteristics to make routing decisions. Moreover, it extends such routing to three-dimensional sensor networks to provide energy-aware routing for routing hole detouring. This protocol is applicable to resource-constrained wireless sensor networks with routing holes.

Kasana [19] proposed a new geographic routing protocol based on cat swarm optimization for the unique features of vehicle-mounted self-organizing networks (such as high mobility, low bandwidth and restricted mobility), with the purpose of finding the optimal effective strategy to select the next forwarding vehicle in a highly dynamic vehicle environment. A fitness function to optimize the impact of various parameters on the selection of the next forwarding vehicle was suggested.

In FANET application scenarios, the high-speed movement of nodes will inevitably lead to frequent changes in network topology, making it difficult for traditional routing algorithms to adapt, while the application of reinforcement learning to routing algorithms can solve such problems [20].

As early as 1993, Boyan et al. [21] first applied Q-learning to routing protocols. They describe the routing and forwarding process as a Markov decision process (MDP), in which each node, as an intermediate state in the MDP, selects the next hop as the action, and the delay cost of each hop as the reward and punishment value for feedback.

There are roughly three types of applications of reinforcement learning in communication network: Q-routing, multi-agent and partially observable Markov decision routing. Hasan [22] introduced the application of traditional routing and reinforcement learning models to wireless

network routing, identified the routing challenges associated with different types of distributed wireless networks, and described the advantages of applying reinforcement learning to routing. The proposal of the ns3-gym toolkit [23] further boosted the application of reinforcement learning to routing algorithms. The toolkit is implemented by connecting the OpenAI Gym toolkit to the ns-3 network simulator, greatly simplifying the complexity of using reinforcement learning to solve problems in the network domain; also, the framework is open source, so industry can easily extend it.

Many literatures use multi-agent systems to solve network routing problems, such as[24-27].In multi-agent learning, each router or network node is considered as an agent that can only observe the local environment information and act according to its own routing policy. Liang et al.[28] proposed a MARL-based approach called Distributed Value Function-Distributed Reinforcement Learning (DVF-DRL) for routing in WSNs, which selects a next-hop neighbor node that provides a lower end-to-end delay, taking into account the Q-value of its neighbor nodes, and therefore the communication performance of the whole network is considered. Elwhishi [29]et al. proposed a MARL-based delay-tolerant network routing scheme that increases the packet delivery rate as well as reduces the transmission delay.

The robust link availability routing protocol [30] is an adaptive routing algorithm based on the gradient ascent algorithm to implement reinforcement learning. Treating each node as an independent agent, this algorithm adjusts policy parameters according to the global performance of the network, makes routing decisions, and determines the corresponding control behavior via the observed local state. Jung [14] proposed an adaptive routing model based on Q-learning to detect the movement degree of each node in the network, and proposed a new routing metric, QMetric.

The literature [31] applies the DQN-routing algorithm in Deep Reinforcement Learning DRL to solve the routing problem, which combines the advantages of Q-routing and DQN. Each router is considered as an agent whose parameters are shared and updated simultaneously during the training process (centralized training), but it provides independent packet transmission instructions (decentralized execution). The literature [32] proposes a gating mechanism in which each communicating node adaptively prunes useless information in a broadband-constrained network. The literature [24] embeds deep neural networks into multi-agent Q-routing based on Q-routing. Each router has an independent neural network that is trained without communicating with its neighbors and makes decisions locally. A multi-intelligence framework is proposed to improve the performance of existing routing methods, and this framework enables each sensor node to build a cooperative set of neighbors based on past routing experience.

Li [33] proposed an effective routing protocol for underwater sensor networks based on multi-agent reinforcement learning. It models the network as a distributed multi-agent system, then the residual energy and link quality are considered in the routing protocol design, to improve its adaptability to a dynamic environment and prolong the network life. In addition, Li proposed two optimization strategies to accelerate the convergence of reinforcement learning algorithms and, on this basis, provided a reward mechanism for distributed systems.

Zeng [34] proposed a multi-agent reinforcement learning framework for adaptive routing in communication networks, which is based on real-time Q-learning and participant criticism. It works by providing a global feedback signal; the router (agent) operates independently but is able to understand the cooperative behavior necessary to reduce packet delivery time. The algorithm is robust to some dynamic changes in the network, and each agent learns adaptive strategies to route packets.

3. System Model

In this section, we first describe the motivation for the research and then express the routing problem in a reinforcement learning model, and define the state space, action space, reward function and other elements in the Markov decision model.

3.1 Problem description

In the existing routing algorithms based on geolocation, such as the GPSR routing protocol, only the distance relationship between nodes is considered when routing and forwarding, without fully considering other attributes of neighboring nodes (link quality between nodes and their neighbors and information loads of those neighbors). Also, when the next hop is selected, only the nearest node to the destination node is greedily selected without long-term consideration of the reliability and feasibility of transmission, which can often lead to a locally optimal solution that results in transmission failure. As shown in **Fig. 3**, a source node, *S*, wants to send data to a destination node, *D*, compared with node *n2*, the distance between node *n1* and source node *S* is closer, and according to the greedy rule, source node *S* will choose the route with the red arrow. But there is a large empty area between node *n1* and destination node *D*, and no suitable next hop can be found. Therefore, the perimeter forwarding mode is triggered, and there is no other node that can be forwarded according to the right-hand rule or the left-hand rule, and the transmitted data will be sent back to the source node *S*. The source node *S* then selects the route with the green arrow, so the packet flow direction is: *S->n1->S->n2->...->D*. Because there is a hole effect in the geolocation-based routing algorithm, the routing algorithm makes the packet flow to node *n1* and back to the source node, which increases the transmission delay of communication and also increases the work pressure on the source node *S*.

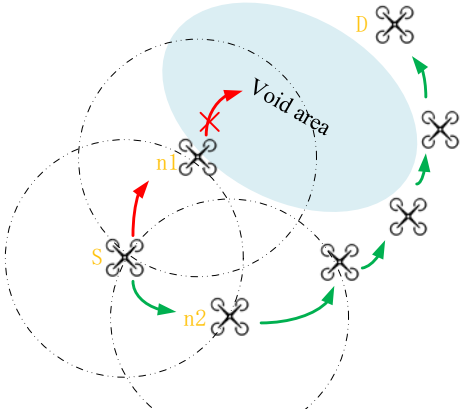


Fig. 3. A case of a data packet being trapped in locally optimal transmission based on geolocation routing

The traditional method used to discover routing voids in the network is to mark the void boundary nodes, and the study divides the void boundary into event boundary and network boundary, where the event boundary is given by the algorithm that detects the corresponding event and the network boundary is determined based on the routing topology of the network. The event boundary is obtained without complex algorithms, and the main problem faced is sensor data processing. The network boundary acquisition requires certain algorithm support and is mainly divided into geometry-based hole detection methods, statistics-based hole detection methods, and topology-based hole detection methods. These methods have high time

and space complexity as well as energy overhead. In this paper, based on the multi-intelligent reinforcement learning method, we use the multi-intelligent Q-learning algorithm to dynamically solve the routing path of the network Nash equilibrium by increasing the communication properties of nodes and intelligently and effectively avoid the routing into the hole region. Based on the method in this paper, node S can sense the state of neighboring nodes and directly select the n2 node as the next hop to directly bypass the hole region.

3.2 System Framework

In the method described in this section, the entire ad hoc network is constructed as a multi-agent system to support the information exchange between nodes, and the value function algorithm is used to obtain the reward and punishment value for interacting with the environment, so as to learn the effective transmission mode. Due to environmental factors, it is typically difficult to get an accurate data model for FANET with highly dynamic nodes. Q-learning is a model-free reinforcement learning model that is widely used, based on a value function algorithm. We use Q-learning to iterate the values of neighbor nodes employing a traditional, geographical location routing algorithm. We maintain a neighbor-value Q table. In QLGR, each data packet writes the location of the target node into a header field during construction. Our routing algorithm thus employs both the distance to the destination node and the neighbor value weight when selecting the next hop node. As shown in **Fig. 4**, the framework consists mainly of node value evaluation and the routing decision. Next, we define the reinforcement learning-related elements.

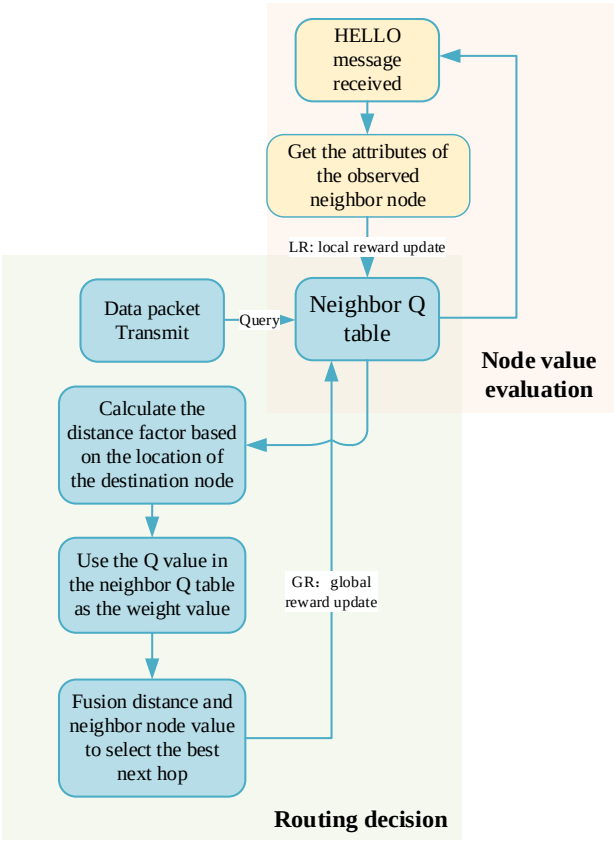


Fig. 4. The proposed Q-learning-based geographic routing algorithm framework

3.3 Modeling

3.3.1 Definitions of states and actions

For a single-agent system, the state of the two communication nodes involved in the return function and the actions they take change only their own environment, without affecting other nodes [34]. For this reason, we construct a multi-agent system in this paper, to represent the cooperation between nodes, to send packets from source nodes to destination nodes.

Before using reinforcement learning to optimize the routing algorithm, the routing decision problem should first be described as an MDP process. Let $N = \{n_1, \cdots, n_i, \cdots n_n\}$ represent a set of nodes, in which they realize multi-hop communication through direct communication by themselves or relay through other nodes. The whole network is treated as an MDP interactive environment, and all nodes are regarded as independent agents. A single agent can only perceive part of the environment, so the partially observable MDP approach should be considered first [35]. The state of the node, s_i^t, is regarded as the state set, S, at the moment, t, for the data packet, p, in node, n_i. On this node, p is sent to the action set, A_i, on n_i which is the constituent node of the action for the next jump, and the action space selected by n_i can be defined as the set of neighbors of n_i:

$$N_{\text{nbr}} = \{n_j \mid n_j \in N \text{ and } \text{Dis}(n_i, n_j) < \text{D\_max and } i \neq j\}. \quad (1)$$

where $\text{Dis}(n_i, n_j)$ is the distance between two nodes, as the bytes of a HELLO maintenance data packet contain geographic information, the distances between nodes are obtained by subtracting the node coordinates. And D_max is the maximum communication distance between nodes. After the action is completed, the agent receives environmental returns, including local (LR) and global (GR) rewards.

To achieve cooperation with neighbor nodes, when n_i makes routing decisions, the influence of the neighbor nodes' local returns and global returns on itself should be considered. To this end, n_i needs to interact with the information of its neighbors to ensure that routing decisions can respond to the dynamic network promptly. In the QLGR routing protocol, periodically broadcast beacon data (HELLO message) can be used to inform the surrounding neighbor nodes of their location and LR and GR information. Furthermore, to control the broadcast HELLO message cycle, the adaptive HELLO time slot algorithm proposed in Section4 of this paper can be used to reduce the control message overhead in the network.

3.3.2 Definitions of reward function

In a single-agent system, each node only perceives its immediate environment and is unaffected by the actions of other agents. If all agents execute actions with their own optimal strategy, the network load may be unbalanced because, if multiple routes are relayed through a node, the network will be congested, thus shortening the network life. To evaluate the rationality of the strategy, the load capacity and link quality of the link task are considered in the LR. The ultimate purpose of routing and the transfer of packets to the destination node, or the next-hop node closer to it, is incorporated into the GR. To establish the LR, a node broadcasts a HELLO message to a neighboring node (the design of the HELLO message format is considered in a later section). The GR updates the Q-value with location information based on the message's successful transfer. LR and GR are defined as follows:

LR:

$$LR(i, j) = \alpha L_Q(i, j) + (1-\alpha)L(i, j). \tag{2}$$

where $j \in N_i$ means that j is a neighbor node of i,

$$L_Q(i, j) = \frac{P_{\text{rec}}}{P_{\text{totle}}}. \tag{3}$$

$$L(i, j) = 1 - \frac{D_{\text{len}}}{C_{\text{len}}}. \tag{4}$$

$L_Q(i, j)$ represents the link quality, as a ratio, between the sending and receiving nodes, and P_{rec} and P_{totle} are the numbers of data packets received by the next hop node and of all data packets sent, respectively. $L(i, j)$ represents the remaining load capacity of the normalized node, C_{len} represents the length of the cache queue, D_{len} represents the length of the existing data queue in the cache queue and α represents the weighting value of balancing load quality and the remaining load capacity.

GR: The ultimate purpose of a routing decision is to choose the appropriate path to transmit a packet to a destination node, while the LR is used to measure the value of the neighboring node, which only gives feedback for the optimal next hop. However, whether this optimal next hop can send data packets to the destination node is unknown. Therefore, a global return value is needed to ensure the best possible effort to deliver the data to the next hop, closer to the destination node when forwarding it. GR was proposed by Li [36] to optimize the performance of the whole multi-agent system, to evaluate the multi-agent coordination in a continuous space state. In that study, it was first integrated into the agent routing algorithm as a quality of service metric to optimize the static network topology path decisions of multi-agent systems. Inspired by this, a GR function was designed to evaluate whether the next hop selected is closer to the destination node:

$$GR(i, j) = \begin{cases} 1, & \text{if } j \in N_i \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

4 Geographic routing algorithm based on multi-agent reinforcement learning

In this section, we analyze the two most important parts of the proposed routing algorithm, namely the node value assessment and the routing decision; then, we propose an optimization method for the routing overhead of the protocol; finally, we describe the implementation and flow of the routing protocol in detail.

4.1 Neighbor node value assessment based on multi-agent Q-learning

When defining an agent, an action is defined as sending a packet to a neighbor node. According to the definitions of conventional reinforcement learning, a reward value is obtained only after an action is performed and the Q-value is updated. However, in the dynamic network environment of a FANET, when no packets are sent, such a Q-value would be constant, which is obviously not appropriate. Therefore, receiving the HELLO message is

also regarded as an action, and the value of the neighboring node relative to another is evaluated by the attributes of the neighbor node in the HELLO message as the basis for updating the Q-value.

Each node maintains a one-hop neighbor Q-table within its communication range, as listed in **Table 1**. The Q-value in the table entry is used as a routing decision weight, and its range of values is [0,1]. Also, to save storage space, only the surrounding active neighbors are saved, and a life span is set for the information about each neighbor, so that if no more HELLO messages are received from the node then, after a certain period of time, it is considered to have left the communication range of the node, with the information being deleted after the total of the three longest HELLO message slots. When a HELLO message is received from a new neighbor, this process begins for it and a Q-value is initialized.

Table 1. Q-table Structure

Neighbor nodes	Destination	Coordinate position	Destination node d_1	Destination node d_2	…
n_1	1.1.1.1	$< x_1, y_1, z_1 >$	$Q(n_1, d_1)$	$Q(n_1, d_2)$	…
n_2	1.1.1.2	$< x_2, y_2, z_2 >$	$Q(n_2, d_1)$	$Q(n_1, d_2)$	…
…	…	…	…	…	

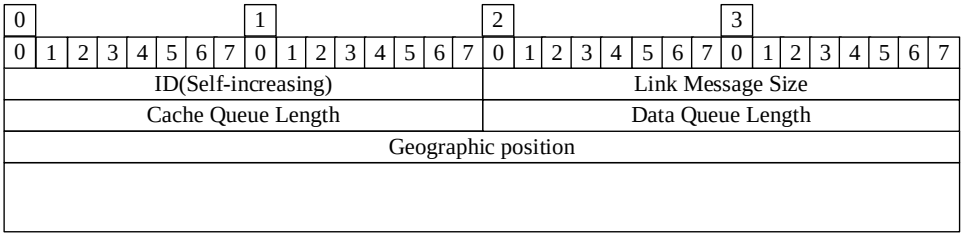


Fig. 5. Hello message format

The HELLO message plays an important role in neighbor discovery and Q-value updates. The HELLO message should include not only the node location information but also the HELLO sequence number, message length, Q-table information, cache queue length and the length of the data queue that is already in the cache queue, as shown in **Fig. 5**.

When the current node, i, receives the HELLO message sent by the neighbor node, j, it counts how many of these it has received and compares this with the message identity originally assigned by j. Through this self-added identity number, it can identify whether a HELLO message was lost and then calculate L_Q according to (3). By a similar method, according to the cache and data queue lengths of the HELLO messages, the remaining load capacity of the neighbor node is calculated using (4).

In accordance with the above information, it is easy to calculate the local reward, $LR(i, j) \in [0,1]$ using (2). Nodes with good link stability and strong residual load capacity obtain a greater LR.

Each node is viewed as an agent; interactions among nodes affect the rewards for each node. Assume that Π_i includes all strategies of the i-th node (the set of all optional neighbors) and that $V_i(s, \pi_1^*, \cdots, \pi_n^*)$ is the reward for the i-th node that balances all intelligence in the state s, where $\pi^*(a|\ s)$ is the optimal strategy. Then, the optimal route of the entire network is the

Nash equilibrium strategy $(\pi_1^*, \pi_2^*, \cdots, \pi_n^*)$ of all agents. All agents $i = 1, 2, \cdots, n$ of state $s \in S$, satisfy

$$V_i(s, \pi_1^*, \cdots, \pi_i^*, \cdots, \pi_n^*) \geq V_i(s, \pi_1^*, \cdots, \pi_i, \cdots, \pi_n^*), \tag{6}$$

Define the Q function of the i-th agent as $Q_i(s, a_i, \boldsymbol{a}_{-i})$, where $\boldsymbol{a}_{-i} = \{a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n\}$ represents the actions of all agents except the i-th agent. Then, the Nash Q function of the i-th agent is:

$$Q_i(s, a_i, \boldsymbol{a}_{-i}) = r_i(s, a_i, \boldsymbol{a}_{-i}) + \beta \sum_{s' \in S} p(s' / s, a_i, \boldsymbol{a}_{-i}) V_i(s', \pi_1^*, \cdots, \pi_n^*) \tag{7}$$

where $(\pi_1^*, \cdots, \pi_n^*)$ is the joint Nash equilibrium strategy and $r_i(s, a_i, \boldsymbol{a}_{-i})$ the instant reward of the i-th agent. The agent learns its Q value via repeated guessing from the time the game starts. At each time step t, agent i observes the current state, takes an action a_i, and receives an instant reward r_i. When action $\boldsymbol{a}_{-i}$ is taken by other agents, and they are rewarded, the environmental state changes to s'. Next, agent i calculates the Nash equilibrium strategy $(\pi_1(s'), \cdots, \pi_n(s'))$ using the action value function $(Q_1^t(s'), \cdots, Q_n^t(s'))$, the update rule of which is:

$$Q_i^{t+1}(s, a_i, \boldsymbol{a}_{-i}) = (1 - \alpha_t) Q_i^t(s, a_i, \boldsymbol{a}_{-i}) + \alpha_t [r_i^t + \gamma V_i^t(s')] \tag{8}$$

where α_t is the learning rate and γ the discount factor. According to that value, the corresponding Q-value of the neighbor node in the Q-table can be updated, and the estimated value of the link i to j can be calculated as follows:

$$\begin{aligned} Q_i^{t+1}(j, d) \leftarrow & (1 - \alpha) Q_i^t(j, d) \\ & + \alpha \left\{ LR^{t+1}(i, j) + \gamma \cdot w_1 \cdot V^t(j, d) + \gamma \cdot w_2 \cdot \sum_{i' \in N_i, i' \neq j} V^t(i', d) \right\}. \end{aligned} \tag{9}$$

Where

$$V^t(j, d) = \max_{j' \in N_j} Q_j^t(j', d). \tag{10}$$

$$V^t(i', d) = \max_{i' \in N_j} Q_j^t(i', d). \tag{11}$$

$V^t(j, d)$ and $V^t(i', d)$ represent the state value functions of j and other neighboring nodes, respectively, with respect to the destination node, d, which are used to estimate the joint value of j selected as the next hop and the transmission trend of surrounding nodes to d. w_1 and w_2 are the weights of the two functions, which are set to 0.2 and 0.05, respectively, in the literature [33] to ensure that the algorithm performs well.

It can be seen from (9) that the higher the link value between the neighbor and the current nodes, the higher the LR obtained according to the neighbor information in the HELLO message. Moreover, the Q-value also gives a better evaluation of this node after each iteration, which is in line with the design idea of selecting the next hop with stable and sufficient residual load capacity. When a node receives a HELLO message, as sent periodically by a neighbor node, it will maintain the Q-table of its evaluated neighbor nodes in real time. When data forwarding is needed, it will select the optimal neighbor node at that moment as the next hop by its evaluation of the values of its neighbor nodes and the location information of the destination node.

4.2 Routing decision based on geographical location

Routing decisions are reflected in the current packets forwarded by nodes under a certain policy. In the previous section, we described the algorithm for evaluating the value of neighbor nodes by the QLGR protocol. When a node has the task of transmitting data, it must select the next hop for forwarding. At that time, the optimal next hop in the current state should be selected according to the information relayed to the node combined with the location information of the destination node, to ensure that the information can reach that node. To facilitate the maintenance and update of node information, the geographical location and Q-value execution are stored in a hash table. Compared with polling lookup, which has O(n) time complexity, a query method with constant time complexity is more conducive to shortening the packet forwarding delay.

For the selection of the next hop, the value of the neighbor node should be considered on the basis of the proximity principle, based on the distance to the destination node. For this reason, the distance between the neighbor and the destination nodes is quantified, according to

$$Dis_i(j,d) = \begin{cases} e^{\frac{D(i,d)-D(j,d)}{r_{\max}}-1}, & D(i,j) < r_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $r_{\max}$ represents the communication radius of i and $D(\bullet)$ represents the Euclidean distance between two nodes:

$$D(i,j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2 + (i_z - j_z)^2}. \tag{13}$$

As can be seen by the use of the distances in (12), if j is further from d than i, then the probability of it being selected as the next hop is smaller. However, there is no operation that prohibits i from choosing a hop farther away from d, so the ability of the node to balance distance and transmission feasibility is retained.

The Q-value corresponding to each neighbor node in the current node Q-table is taken as the weight of the distance quantification value, and the product of the two is defined as the discounted Q-value, $\bar{Q}(j,d)$. At the current node, QLGR adopts a softmax policy to select the node for the next hop forward. The softmax strategy is calculated as follows:

$$\pi(a \mid s) = \frac{e^{\frac{\bar{Q}(s,a)}{\tau}}}{\sum_{a' \in A} e^{\frac{\bar{Q}(s,a')}{\tau}}}. \tag{14}$$

where $\tau > 0$ is the temperature. Different to the balanced exploration in the strategy ε-greedy, in this softmax strategy, each Q-value is mapped exponentially, focusing on exploration of the neighbor nodes with better $\bar{Q}$, so as to distribute the network traffic on different network nodes and avoid the network congestion caused by packets being concentrated.

After making a routing decision, it is necessary to reward the selection of this action according to the GR, that is, the associated Q-value is updated by calculation as follows:

$$Q_i^{t+1}(j,d) \leftarrow Q_i^t(j,d) + \alpha[GR_{t+1} + \gamma \max_{j' \in N_i} Q(j',d) - Q_i^t(j,d)]. \tag{15}$$

After that, the subsequent forwarding nodes will gradually transmit the packets to d in accordance with the above principles.

QLGR routing is conducted according to Algorithm1. When the protocol selects nodes, the value function is first calculated according to the local reward and the global reward, and the Q value is updated according to (15). If the neighbor node-set is not empty, the distance between the neighbor node and the destination node is quantified, and the probability of each alternative path can be calculated by multiplied corresponding Q value.

Algorithm 1 Q-learning-based geographic routing algorithm

Input: data pack p, Current node c, Set of neighbors N_{nbr}

Initialize: Q_{table}

If packet p ' destination node is c **then**
  Return
End if
If packet p returned to c **then**
  Update $Q(p.last, p.des)$ with (12) and $GR = -GR$
End if
If $N_{nbr}(c) \neq \varnothing$ **then**
  For all $n_i \in N_{nbr}(c)$ **do**
    If $p.pre = n_i$ **then**
      Continue;
    End if
    $P[n_i] = Q_{table}(n_i, p.des) * Dis(n_i, p.des)$
  End for
  $P_{next} = \text{softmax}(P)$
  Select next hop a according to probability P_{next}
  Update $Q_{table}(a, p.des)$ with (15)
Else
  Select a using GPSR' Perimeter mode
End if

Output: Next hop a

As long as we maximize the selection probability among the alternative paths, we can decide the best path for the next hop. If the neighbor node set is empty, the route enters the peripheral mode, and the route falls back to GPSR to perform route selection. When the data packet exits from the GPSR peripheral mode, the mode switches to the greedy mode again, and the QLGR algorithm is restored for routing. The complete flowchart of proposed QLGR algorithm as shown in **Fig. 6**.

In summary, each node maintains a neighbor node value Q-table for the next hop that can reach d, and the size of the Q-table is determined by the numbers of next-hop neighbor nodes and previous destination nodes. The Q-table entries are maintained automatically when neighbor nodes are added or deleted, so the algorithm is robust in a multi-hop network environment.

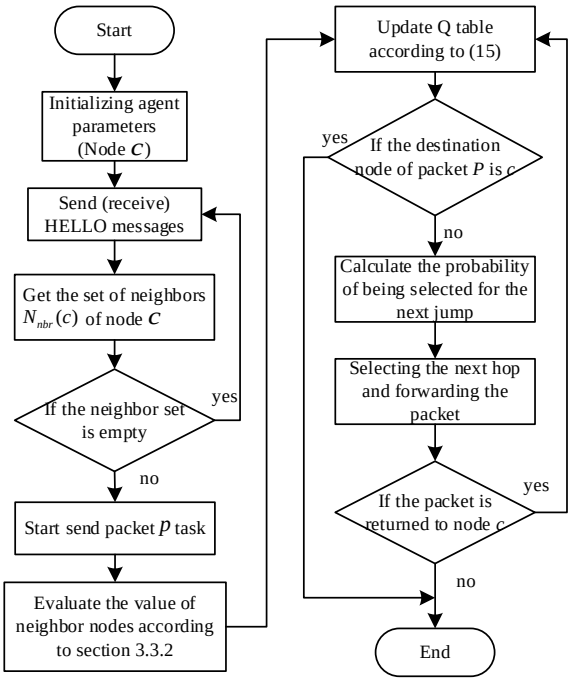


Fig. 6. Flow chart of proposed QLGR algorithm

4.3 Protocol optimization

For a highly dynamic FANET, this scheme of periodically exchanging HELLO messages is insufficient to adapt to the changing network environment, because the selection of the HELLO message slot plays a decisive role in link discovery in geographically based routing protocols. The shorter this time slot, the faster the detection of new neighbors or link outages, but this generates a higher overhead and hinders the transmission of normal packets. Conversely, the longer the time slot, the lower the overhead, but this limits the ability to discover neighbors and detect a broken link. Therefore, in this section, we propose an optimization scheme, which the system can learn online and use to adjust adaptively the broadcast cycle of HELLO messages.

The HELLO time slot adjustment problem is described as an MDP process with a state, $S \in < s_{nbr}, s_{load}, s_{solt} >$, where s_{nbr} represents the change degree of neighbor nodes in the period of time step, Δt, of the current node, as given by:

$$s_{nbr} = \frac{\left| num_{t+1} - num_t \right|}{\Delta t}. \tag{16}$$

where num_t represents the number of nodes in the neighbor table; s_{load} represents the number of packets to be sent in the interface queue of a node; and s_{solt} represents the length of the HELLO slot on the current node.

How does a node change its perceived time slot in response to changes in the network environment? Reducing the time slot will improve the speed of sensing the neighbor nodes but will increase the network overhead. In the current paper, the action is set as:

$$T_H^{t+1} = T_H^t + T_a. \tag{17}$$

where the value of T_a is $(+1000, 0, -500)$ in milliseconds, with three possible actions concerning the broadcast HELLO cycle compared to the previous decision cycle, namely adding 1000 ms, remaining unchanged or decreasing it by 500 ms.

To accomplish this, a measure of the positive and negative value of the return given by the environment needs to be defined, and thus we define the following utility function:

$$Utility = -\alpha \times log(C) + \delta \times log(L). \tag{18}$$

where, C represents the change degree of neighbor nodes in Δt, calculated as:

$$C = \frac{\left| s_{nbr}^{t+1} - s_{nbr}^{t} \right|}{\Delta t}. \tag{19}$$

L represents the information load capacity of the current node [13], as calculated in (3), where l represents the queue cache length of the node.

$$L = 1 - \frac{s_{load}}{l}. \tag{20}$$

α and δ represent the relative weights of the change degree of the neighbor nodes and the load capacity, respectively, which are both set to 50% in this paper. It can be seen intuitively that this function can represent the ability to detect neighbor nodes quickly and smoothly to the maximum extent while minimizing the loss of forwarded information.

Returns are defined by the difference between successive benefit values:

$$R_t = \begin{cases} U_{t+1} - U_t, & if \ \left| U_{t+1} - U_t \right| > \zeta \\ 0, & if \ \left| U_{t+1} - U_t \right| < \zeta \end{cases}. \tag{21}$$

where ζ represents an adjustable parameter. When the difference between the two benefit functions is greater than ζ, that difference is taken as the return function. When the return function is positive (negative), it represents a reward (punishment). The value function can be defined as follows:

$$\begin{aligned} Q_\pi(s,a) &\doteq \mathbb{E}_\pi \left[G_t \middle| \ S_t = s, A_t = a \right] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| \ S_t = s, A_t = a \right]. \end{aligned} \tag{22}$$

where $G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$. When the value function is maximized, the required periodic strategy for adjusting the broadcast HELLO message, $\pi(a \mid s)$, can be obtained:

$$Q_*(s,a) \doteq \max_\pi Q_\pi(s,a) \tag{23}$$

Compared with traditional routing technology, its advantages lie in that it improves the ability of nodes to find neighbors and increases the stability of the routing protocol in the face of a highly dynamic topological environment; it also reduces the protocol's overhead under a relatively stable topology environment.

4.4 Routing protocol implementation

QLGR extends the message format beyond that of the GPSR routing protocol to support data transmission and message sharing. There are two main formats of the message packets: HELLO and information. The design of the message format is conducive to the acquisition of node information and the realization of an optimal path strategy. On this basis, the specific process of routing protocol is as follows:

Initial work: Set up the routing table at the start node and initialize the parameters related to setting up the network.

Route discovery: Each node periodically broadcasts HELLO messages across the network, informs the surrounding nodes of its own node status, determines the node link quality and load capacity within the single-hop communication range according to the received HELLO messages and evaluates the LR of the node quality. Based on these, each node will update its Q-table after receiving a HELLO message and monitor the communication requirements in the network to prepare for information packet transmission at any time.

Message receiving: When the current node receives a message sent by a neighbor node, it will determine the message type. If it is HELLO, it will perform the routing discovery operation and update the corresponding Q-value. If the received message is information, it will judge whether the packet has already passed through this node according to the packet's source node and serial number. If so, it executes (15), setting $GR = -GR$, and updates the Q-table, then routes and forwards the packet; if not, it goes directly to the route forwarding step.

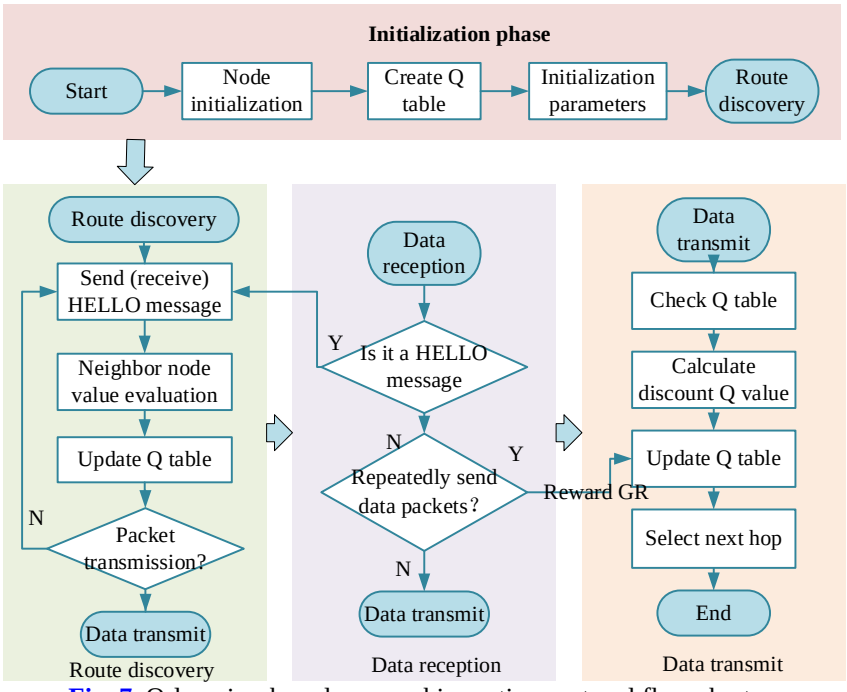


Fig. 7. Q-learning-based geographic routing protocol flow chart

Route forwarding: In the current node's message queue, when an information packet needs to be sent, the location of the packet's first destination node is obtained and combined with the Q-value information in the Q-table to calculate the discounted Q-value of each neighbor, and the next hop for forwarding is selected according to the softmax strategy. The detailed process for this is shown in **Fig. 7**.

5 Simulation Experiment

In this section, we simulated the protocol on the ns3-gym simulation platform [23]. We compared and analyzed several important performance indicators of routing protocols: average transmission delay, throughput, packet loss rate and average delay jitter. We then optimized the routing algorithm using the protocol in Section4.3, calling the resulting algorithm QLGR-S, and compared it with GPSR.

In the experiment, each node was regarded as an agent, that is, each node was required to have corresponding computing power. Also, for convenience, the state of each node was input to an agent so as to output different time slot results, so that the agent could obtain more states and learn to give better results, as shown in **Fig. 8**.

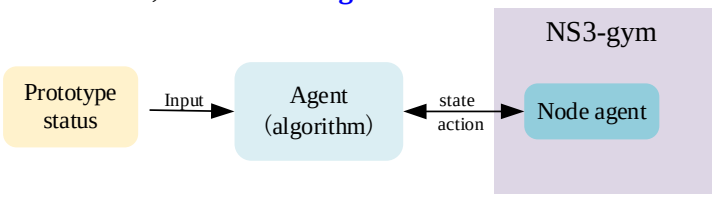


Fig. 8. Simulation architecture

Obtain the dataset: QLGR learns in the interaction with the environment, and thus a training dataset can be generated by combining the required parameters obtained using the simulation environment, NS3-gym. Specifically, the NS3 side provides interface functions pertaining to passing status: MyGetObservation(), reward MyGetReward(),end marker MyGetGameOver() and receiving action MyExecuteActions(action). Correspondingly, the OpenAI Gym side provides the interface functions obs, done, reward=step(action) for receiving status, as well as reward, end flag, and passing action information.

To compare the performance between routing protocols more comprehensively, we refer to the experimental approach of the literature [37]. The simulations of the experiments defined four scenarios, Grid size = 5km×5km and Number of nodes = 20, Grid size = 3km×3km and Number of nodes = 20, Grid size = 5km×5km and Number of nodes = 40, Grid size = 3km×3km and Number of nodes = 40. To form a usable network topology, considering the size of the grid and the node density, we set the maximum communication distance of the nodes to 600 m and the movement speed between 100-300 m/s. Other network parameters and algorithm training parameters are shown in **Table 2**.

Table 2. Experimental Parameter Settings

		GPSR	QLGR	QLGR-S
Environment parameters	Simulation area	5 km × 5 km, 3 km × 3 km	5 km × 5 km, 3 km × 3 km	5 km × 5 km, 3 km × 3 km
	Number of nodes	20,40	20,40	20,40
	Mobility model	Gauss–Markov mobility model	Gauss–Markov mobility model	Gauss–Markov mobility model
	Simulation time	200 s	200 s	200 s
	Node movement speed	100−300 m/s	100−300 m/s	100−300 m/s
	altitude of UAVs	200m	200m	200m
	transmission ranges for UAVs	600m	600m	600m
	HELLO message interval	0.5s	0.5s	Adaptive
	Send buffer size	32kB	32kB	32kB
	Receive buffer size	32kB	32kB	32kB
	Data transmission mode	CBR	CBR	CBR
	Packet size	512 bytes	512 bytes	512 bytes
	Data transmission efficiency	1−11 Hz	1−11 Hz	1−11 Hz
	MAC protocol	IEEE 802.11b	IEEE 802.11b	IEEE 802.11b

Training parameters	Learning rate α_t	-	1×10^{-3}	1×10^{-3}
	Discount factor γ	-	0.9	0.9
	Value weight w_1 of node j	-	0.2	0.2
	Value weight w_2 of other node	-	0.05	0.05
	Weight neighbor node change degree α	-	-	0.5
	Weight of network load capacity δ	-	-	0.5
	HELLO message adjustment range	-	0ms	-1000ms, 0ms, +1000ms

A model should exhibit good network performance and capture UAV movement accurately. The existing mobility models include random, temporally dependent, and spatially dependent models with geographical constraints, and hybrids. The Gauss Markov mobility model (GMM) considers time correlations when simulating aerodynamic constraints on nodes and, thus, models real-world situations effectively[2], as follows: (1) Set the initial speed and direction of the node; (2) the node moves for a preset time; (3) node direction and speed are updated using (24); and, (4) step (2) is repeated and a cycle develops.

$$\begin{aligned} s_n &= \alpha_{s_{n-1}} + (1-\alpha)\bar{s} + \sqrt{\left(1-\alpha^2\right)}s_{x_n-1} \\ d_n &= \alpha d_{n-1} + (1-d)\bar{d} + \sqrt{\left(1-\alpha^2\right)}d_{x_n-1} \end{aligned} \tag{24}$$

where s_n and d_n are the new speed and direction of the mobile node over the time interval n, and $\bar{s}$ and $\bar{d}$ are constants when $n\to\infty$. The GMM reflects the high-level dynamics of the network topology and accommodates UAV flight well under real-world conditions.

The performance parameters compared in the experiment were defined as follows.

Throughput: the speed at which nodes actually send data through the network; throughput is the average rate at which messages are successfully transmitted through the communication channel, calculated as:

$$Thoughput = \frac{C_{\text{totle}}}{\tau}. \tag{25}$$

where C_{totle} represents the total number of data packets successfully transmitted, and τ represents the time taken to transmit them.

Routing overhead: the number of messages, other than information packets, in the whole network required to maintain routing transmission,

$$RO = \sum_{i=1}^{n} C_i. \tag{26}$$

where C_i represents the number of messages in the network other than information packets.

Average end-to-end delay: the average time taken to transmit messages from the source node to the destination node,

$$\overline{DT} = \frac{1}{N}\times\sum_{i=1}^{N} DT(i) \tag{27}$$

where N represents the number of transmitted data packets, and $DT(i)$ represents the transmission delay of the ith data packet.

Packet loss rate: an important indicator of network quality; the number of packets lost in the network during data transmission, typically evaluated as the number of such packets as a proportion of the total number sent during a characteristic period,

$$Loss = \frac{NS - NR}{NS} \tag{28}$$

where *NS* and *NR* represent the total number of packets sent and received, respectively, by a node.

Residual energy: To reflect the energy consumption of the routing algorithm, it is necessary to statistically analyze the residual energy of the UAV to complete several tasks, and the energy consumed by the UAV to transmit data is

$$E_{TX} = \begin{cases} nE_{elec} + n\varepsilon_{fs} d^2 & d < d_0 \\ nE_{elec} + n\varepsilon_{mp} d^4 & d \geq d_0 \end{cases} \tag{29}$$

Where n is the number of bits, E_{elec} is the energy consumed per bit of data received and transmitted, ε is the energy per bit transmitted per unit square meter by the transmitting amplifier, if the transmission distance is less than the threshold d_0, the power amplification loss is used in the free space model; conversely, the multi-path fading model is used. The energy consumed for receiving data is

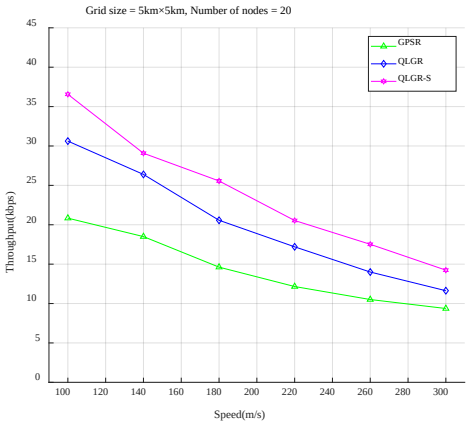
$$E_{RX}(n) = nE_{elec} \tag{30}$$

So the residual energy of the UAV node is

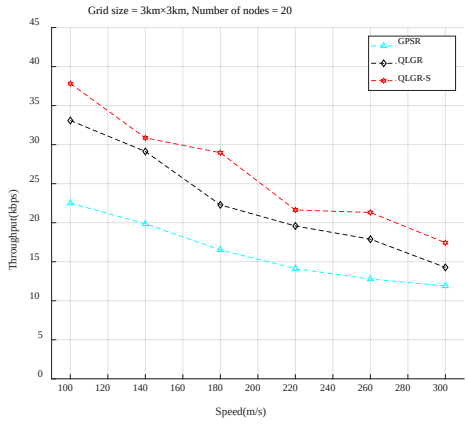
$$E_{residual} = 1 - \frac{(E_{TX} + E_{RX}) \times R}{E_{ALL}} \times 100\% \tag{31}$$

where R is the number of missions performed by the UAV.

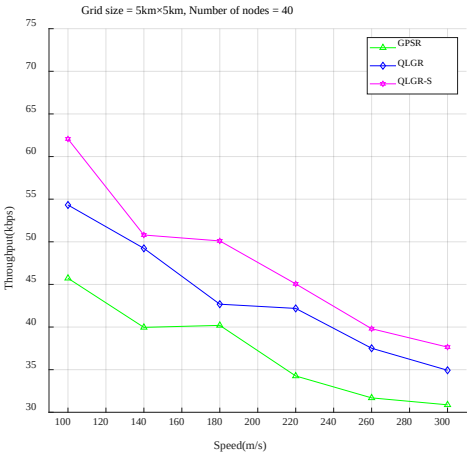
We performed simulations for four scenarios, and then the results of each performance parameter were counted.



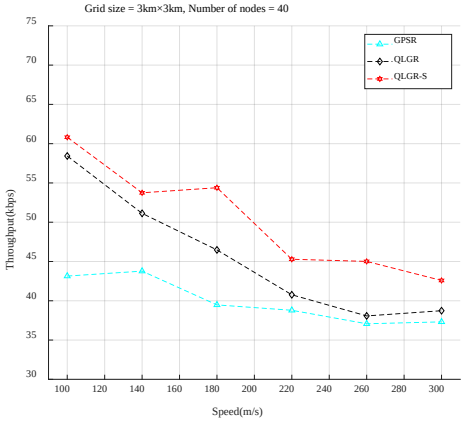
(a)



(b)

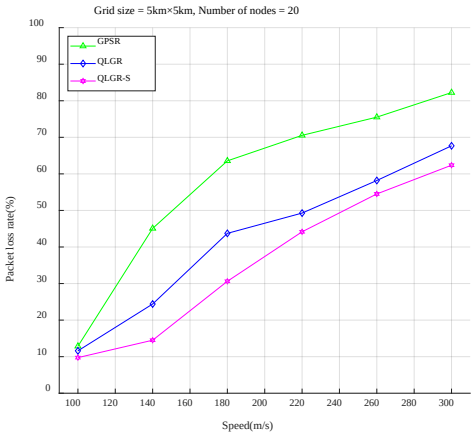


(c)

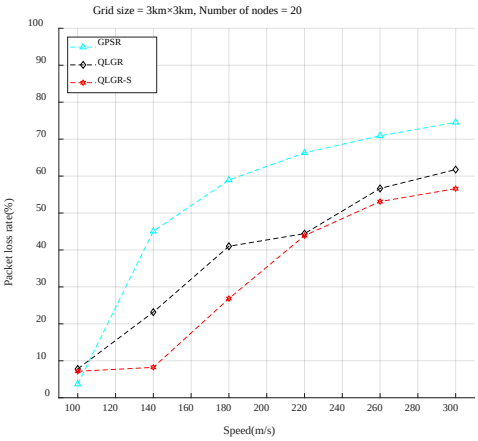


(d)

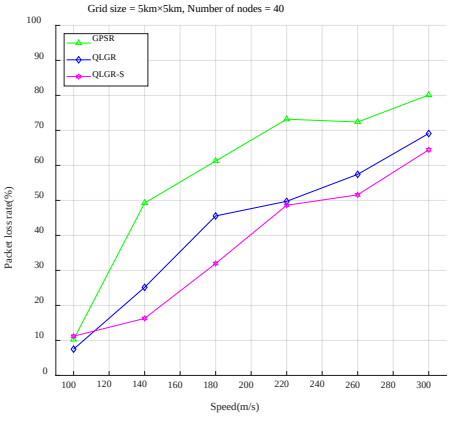
Fig. 9. Relationship between node speed and throughput



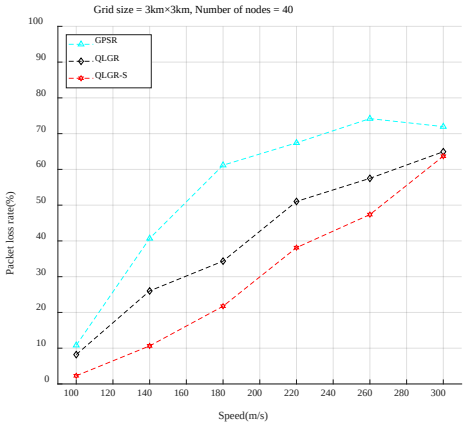
(a)



(b)



(c)



(d)

Fig. 10. Relationship between node speed and packet loss rate

Fig. 9 reflects the relationship between the speed of the UAV nodes and the network throughput in the simulation state. Where the left (a) and (c) are the results for a network area size of 5km*5km, represented by solid lines, and the right (b) and (d) are the results for 3km*3km, represented by dashed lines, with 20 nodes in (a) and (b) and 40 nodes in (c) and (d), and are consistent with this below. **Fig. 10** reflects the relationship between node speed and network packet loss rate. It can be seen from **Fig. 9** and **Fig. 10** that the throughput of all three routing protocols decreases and the packet loss rate increases as the speed of the nodes increases. From overall view, the increase in speed causes a significant change in the network topology, where links capable of transmission are quickly disconnected and new links suitable for transmission are generated in a very short time, too late for the routing protocol to take advantage of the link. This leads to an increase in network transmission uncertainty and many packets are dropped due to lifecycle arrivals before reaching the destination node, hence the network throughput keeps decreasing, and the packet loss rate increases. Concretely, our proposed QLGR protocol outperforms the GPSR routing protocol in terms of throughput and packet loss rate because QLGR improves the reliability of routing by taking into account the link quality and the distance factor from the destination node when routing and forwarding. In addition, QLGR-S routing protocol using an optimized link probing algorithm outperforms QLGR in terms of throughput because the load and mobility of nodes are taken into account when selecting the sending time slot during the route discovery phase.

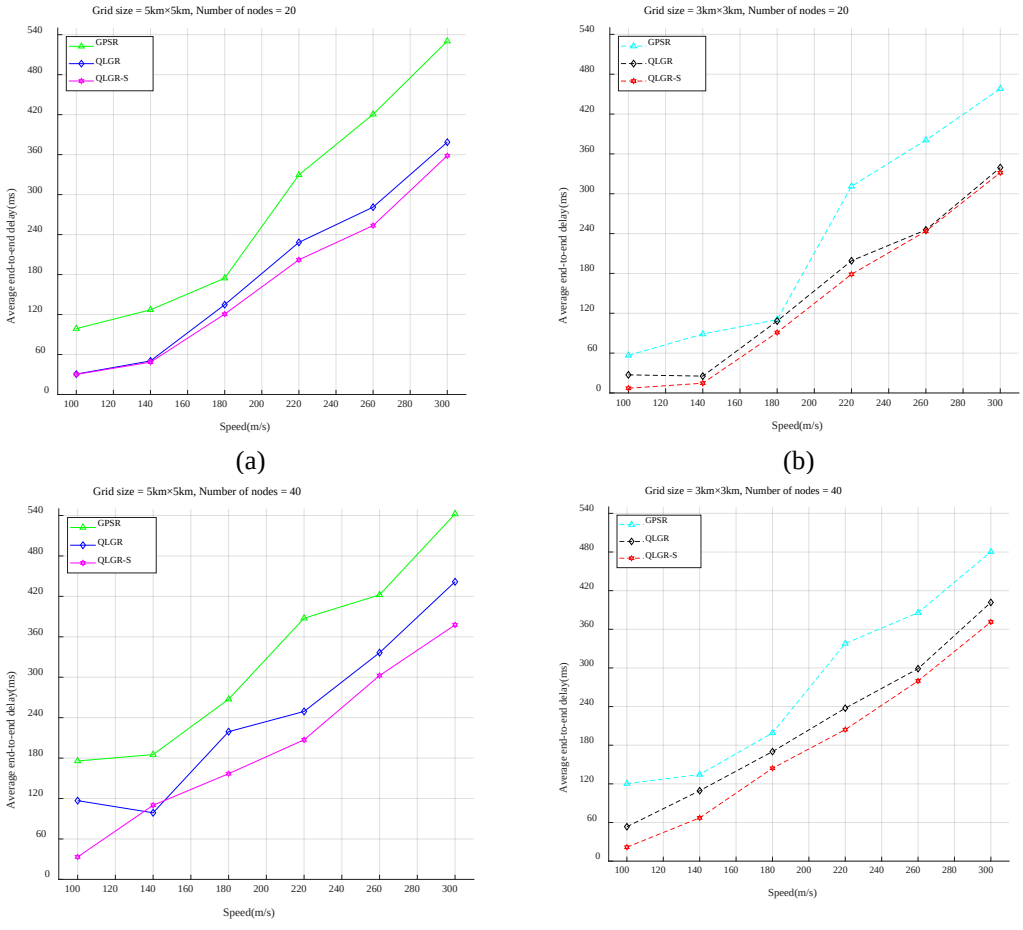


Fig. 11. Relationship between node speed and average end-to-end delay

Fig. 11 shows the simulation results of the average end-to-end delay with node movement speed for the three routing protocols in the four grids. As a whole, the average end-to-end delay of all routing protocols increases with increasing node speed. Because stable link quality relies on stable network topology, an increase in node speed leads to a decrease in the number of communicable links. Specifically, the average end-to-end delay of our proposed QLGR protocol is smaller than that of GPSR protocol because the QLGR series routing takes into account the location relationship between the next-hop neighbor node and the destination node and reduces the routes into the perimeter forwarding mode using the reward and penalty mechanism, which is good at reducing the data transmission delay. So as seen from the curves in the figure, although the latency of all three protocols increases with speed, QLGR series routing is better than GPSR and QLGR-S routing is, in turn, better than QLGR due to the maintenance of Q at the update node indicates that the QLD algorithm is more adaptive.

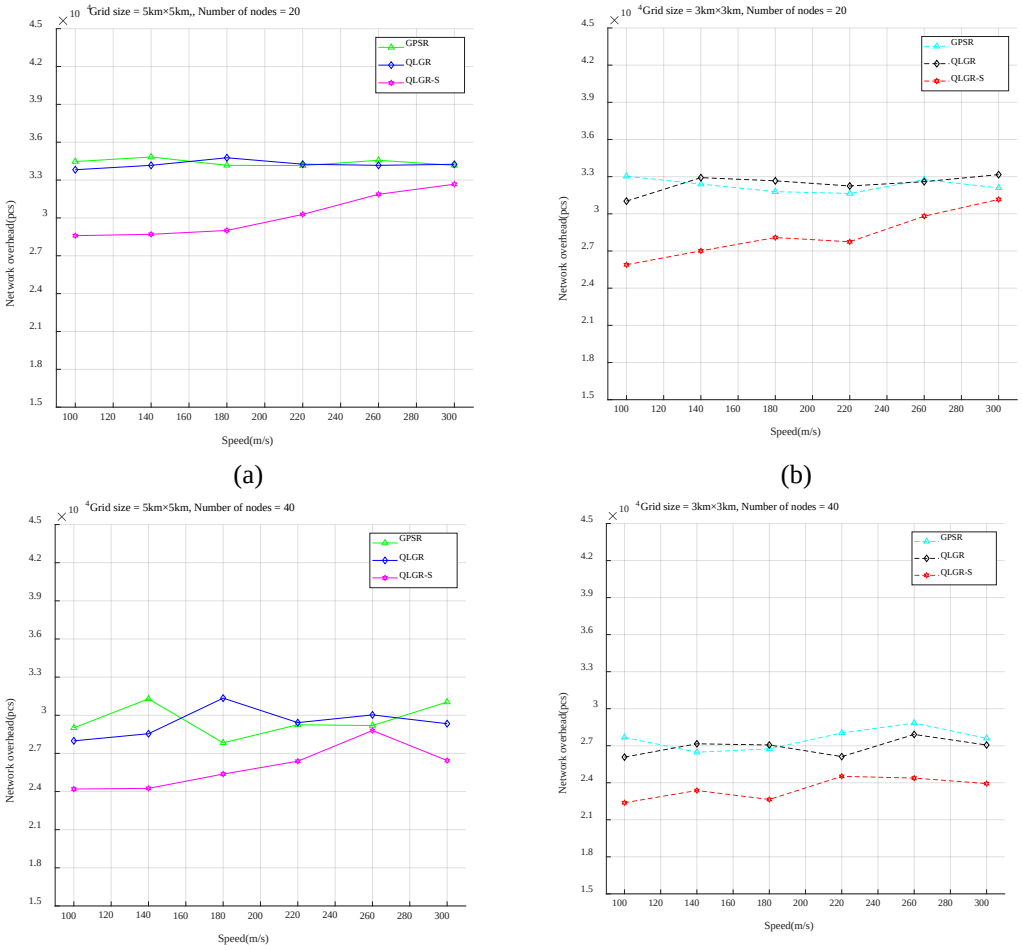


Fig. 12. Relationship between node speed and network overhead

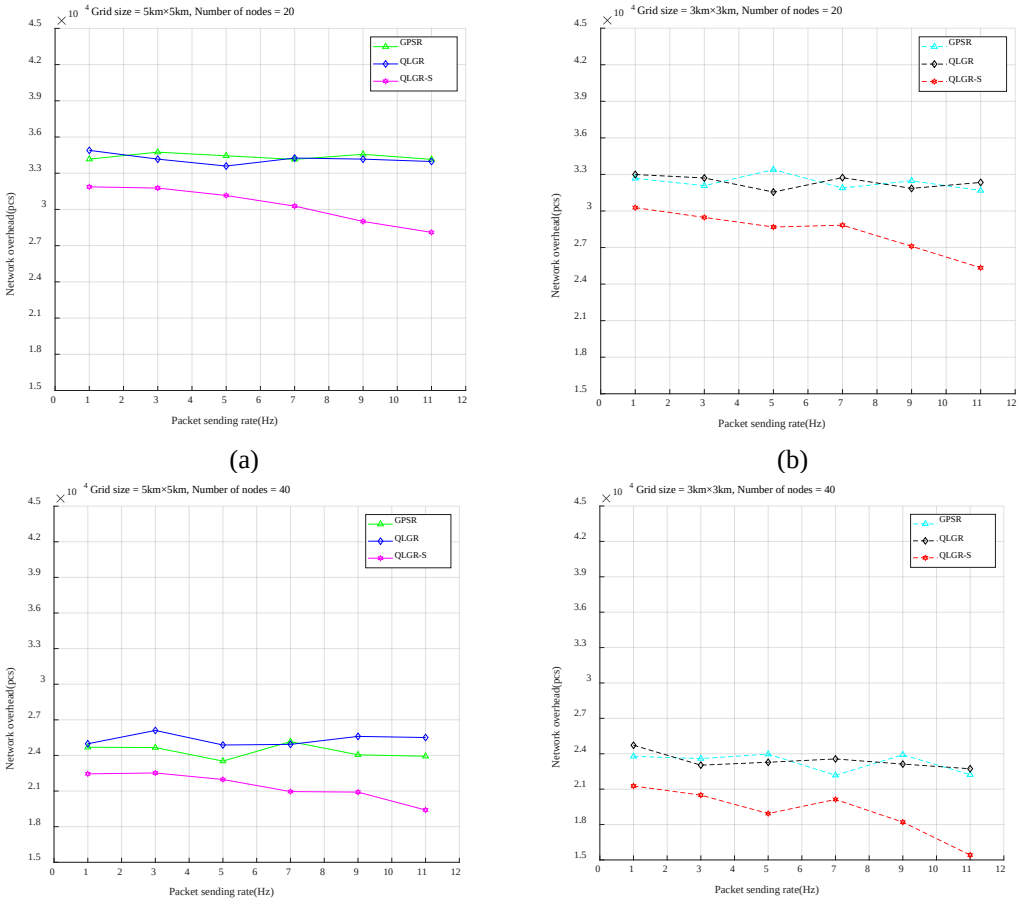


Fig. 13. Relationship between network load and network overhead

Fig. 12 illustrate the relationship between the network overhead and the node movement speed for the three routing protocols in the four grids. According to the design principle of GPSR and QLGR, their routing overheads are mainly link detection packets, i.e., HELLO packets, and are sent at a fixed time interval, so the network overheads are more fixed. However, the QLGR with the addition of the optimized link detection algorithm (QLGR-S) increases the network overhead when the node speed increases to adapt to the node topology changes. **Fig. 13** shows the simulation results of network overhead versus packet sending rate, where GPSR and QLGR remain essentially the same, while QLGR-S sends data for free bandwidth when the packet sending rate increases, while the network overhead decreases.

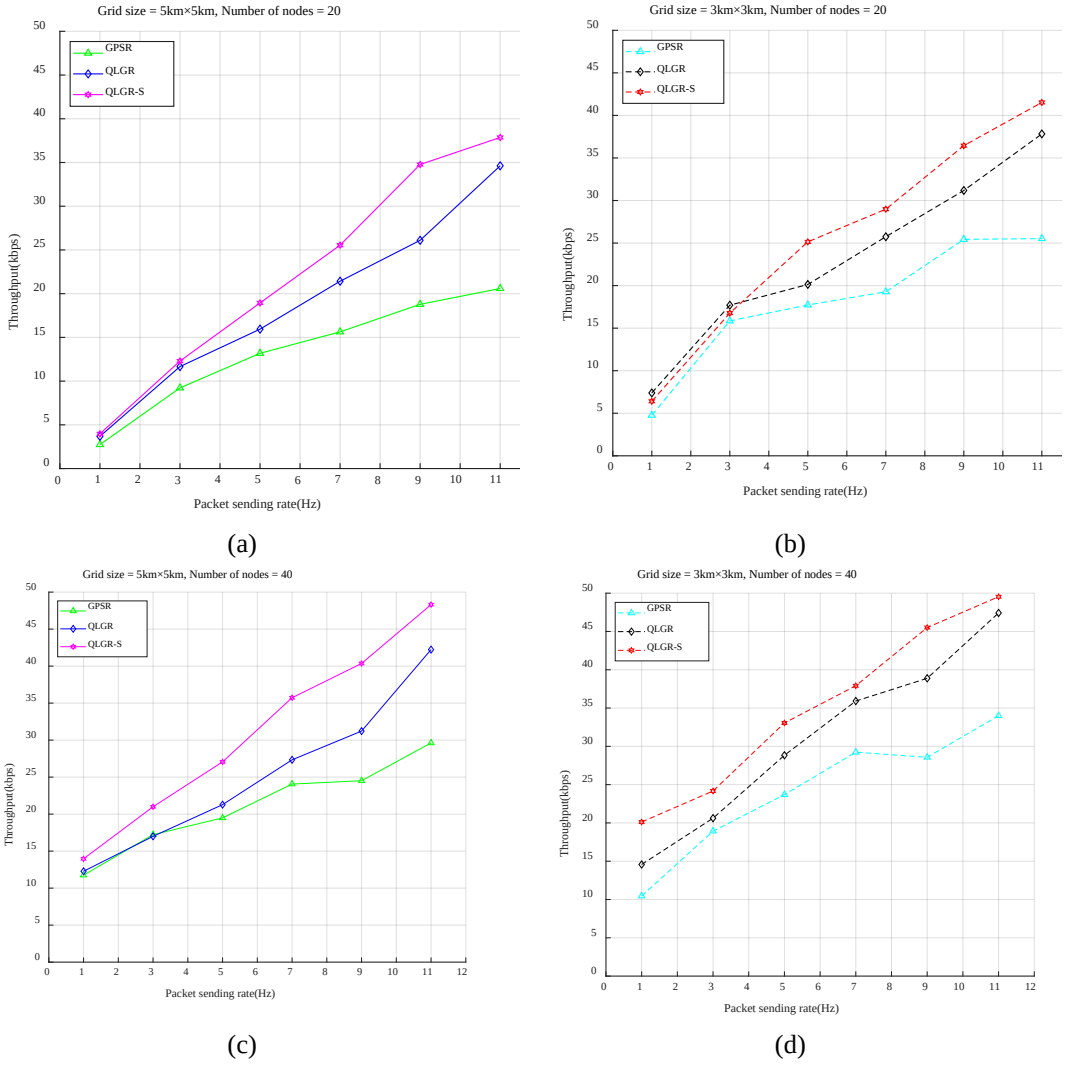


Fig. 14. Relationship between network load and throughput

Fig. 14 shows the network load versus throughput for the three routing protocols in the four grids. From an overall perspective, the network throughput of GPSR, QLGR, and QLGR-S all increase to different degrees as more data are available in the network, and the QLGR series routing increases significantly. Specifically, the network load increases with the limited capacity of nodes in the network, and QLGR takes a load of neighboring nodes into account in the routing decision when selecting the next hop, so it can distribute the network on different paths to spread the network load and increase the throughput of the network.

Energy is an important factor in UAV scenarios. To examine the energy consumption of the routing protocol, we counted the residual energy as a performance parameter, which is calculated as shown in (31). R is the number of rounds of UAV executing tasks, we set the amount of data to be distributed to execute one round of tasks to $1000bit/round$, the communication distance threshold $d_0 = 300m$, and the initial energy of all UAV nodes to 100%, and the simulation results are shown in **Fig. 15**.

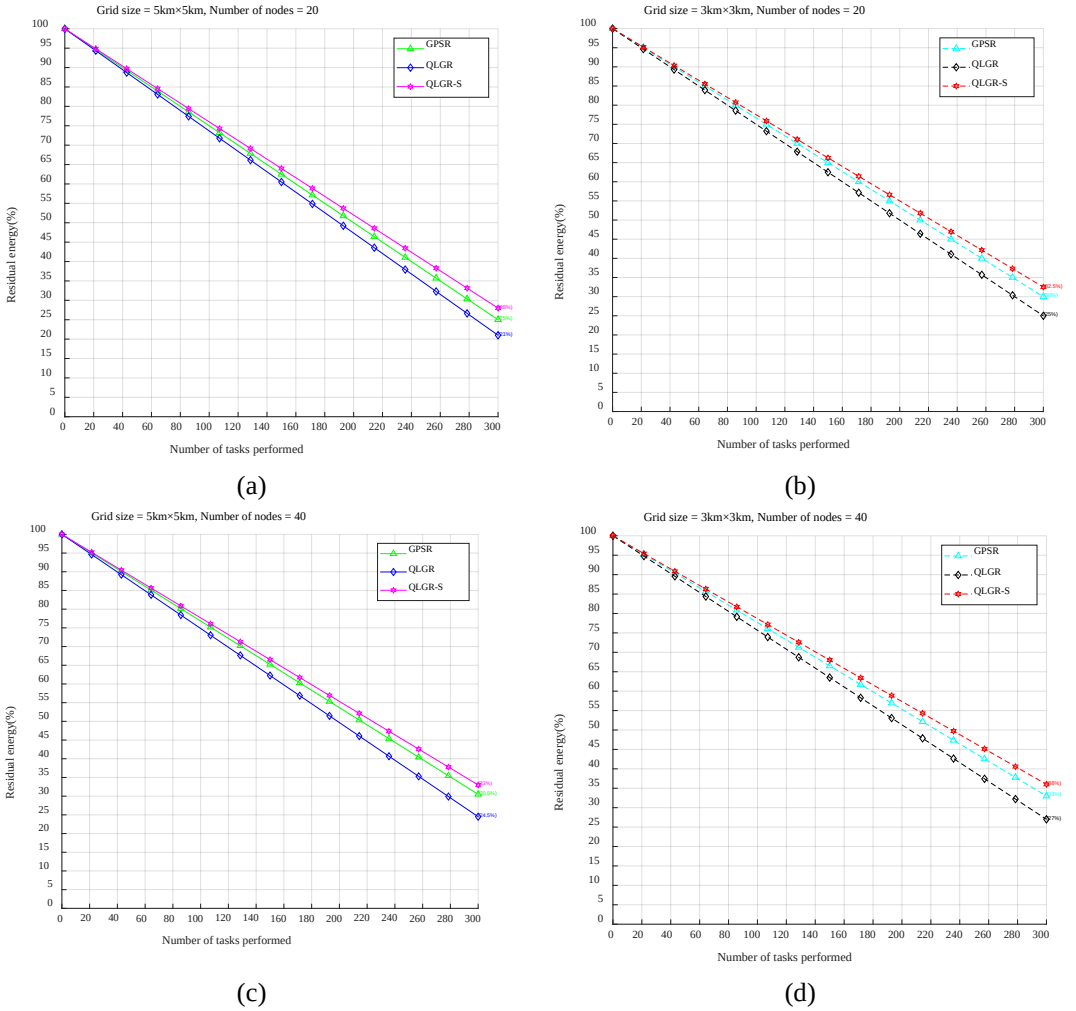


Fig. 15. Relationship number of tasks performed and residual energy

We recorded the power consumption of the UAV for 300 missions, and the remaining energy of the UAV was calculated every 15 missions. As can be seen from **Fig. 15**, the remaining energy of the UAV decreases linearly as the number of missions performed increases. among the four simulation scenarios, the scenario with the smallest node density (Grid size = 5km×5km and Number of nodes = 20) has the fastest energy consumption for the UAV and the least remaining energy; the scenario with the largest node density (Grid size = 3km×3km and Number of nodes = 40) has a lower energy consumption for the UAV and the most remaining energy. In each scenario, our proposed QLGR-S algorithm has the most residual energy and performs in addition to the best performance. In the scenario with Grid size = 5km × 5km and Number of nodes = 20, after 300 missions, the residual energy of the UAV with GPSR algorithm is 25%, with QLGR algorithm is 21%, and with QLGR-S algorithm is 28%. It is worth noting that GPSR outperforms QLGR for energy consumption, probably because the QLGR algorithm is not optimized, while the traditional GPSR algorithm has relatively low complexity and it shows better performance.

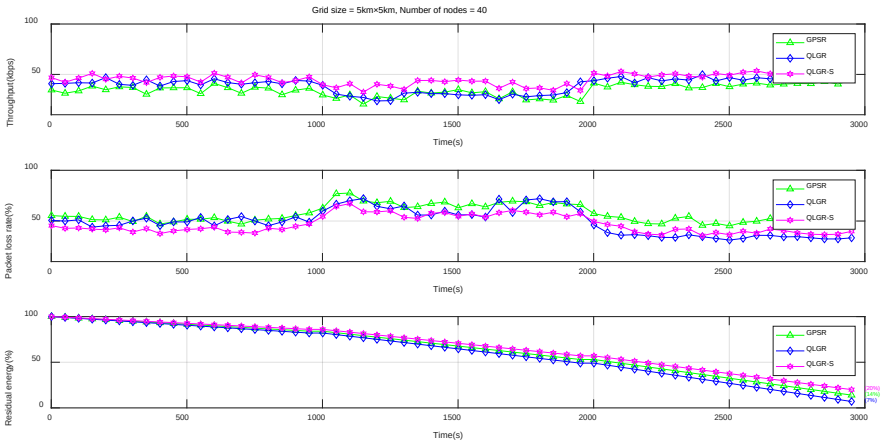


Fig. 16. UAV speed and topology changes on the performance of routing protocols in terms of throughput, packet loss and energy consumption

Topology change is also a factor affecting the performance of the protocol. To verify the networking performance of QLGR-S when the UAV is performing a mission, we simulated the changes of UAV flight speed, altitude, and topology, and conducted a statistical analysis of the protocol throughput, packet loss rate, and UAV residual energy. The experiments set the UAVs to complete three consecutive missions in the order from cruise to target search and then to target tracking. The literature [38] corresponds these three application scenarios of highly dynamic FANET to different mobility models, namely Reference Point Group Mobility (RPGM), Random Way Point (RWP), and Pursue. during the execution of the three tasks, the UAV dynamically adjusts the network topology, and some of the UAVs undergo speed and sudden changes in altitude, so it meets the requirements for routing protocol examination. The experiment simulates each model for 1000 seconds, a total of 3000 seconds, and obtains the following performance statistics.

As shown in the **Fig. 16**, the simulation results show that the throughput of the three protocols decreases at 1000 seconds and increases at 2000 seconds. This indicates that the topology of the UAV changes faster when the UAV performs the target search mission, causing the networking performance to drop, but the QLGR-S algorithm still maintains the highest throughput, followed by QLGR and the lowest by GPSR. For the packet loss rate, the impact of a topology change and UAV movement change on the performance of the proposed algorithm can also be seen from the simulation results. The switch of UAV performing the task means the change of network topology and the packet loss rate of the routing protocol. From the residual energy of the UAV, it is clear that the UAV consumes the fastest energy when performing the target search task, the second-fastest when performing the task of target tracking, and the slowest when performing the cruise task. Finally, the residual energy of the UAV with the QLGR-S algorithm reaches 20%, while the residual energy of the UAV with GPSR and QLGR is 14% and 7%.

6 Conclusions and future directions

In this paper, we have proposed a geolocation routing algorithm based on multi-agent reinforcement learning to address the shortcomings of traditional geolocation-based routing protocols. In this algorithm, each node is regarded as an agent that separately executes the

routing strategy in accordance with local information; through the definition of global reward, all nodes cooperate to complete data transmission. The node value function considers information about the link quality, node energy and queue length, reducing the possibility that geographic routing is trapped in the hole effect. To reduce the routing overhead of the proposed protocol, we have also proposed an optimization method in this paper, which adaptively adjusts the cycle for broadcasting the HELLO packets, so that the link quality is maintained while the overhead of that maintenance is minimized. Our simulation showed that our new protocol improved the performances of all network parameters compared to those of a traditional, geographic location-based routing algorithm. Our protocol enhances the connectivity of UAV networking, guarantees low overhead and high-throughput communication among UAVs, and will aid the further development of UAV technology.

Our research plan and applications of the proposed method will derive from some limitations of the current work. Multi-agent reinforcement learning is not the first time to be used in network routing, and many works have made some attempts, but this paper is the first attempt to use multi-agent reinforcement learning for optimizing the location-based routing protocol in FANET. With the utilization of multi-agent systems in continuous state space and complex scenarios, the scalability of algorithms is the latest challenge, and later research can try other algorithms for multi-intelligent body systems, such as MADDPG (Multi-agent Deep Deterministic Policy Gradient), DRQN (Deep recurrent Q- network), etc.

In terms of the simulation, we have completed the simulation of flight environment based on NS3-Gym, but among FANET applications, there are relatively few network simulation tools that incorporate reinforcement learning, and it appears to be somewhat difficult to fully and completely simulate the flight environment of UAVs and perform network performance statistics. In the follow-up work, we will consider proposing more general network settings and providing more comprehensive tools to facilitate the development of reinforcement learning in network routing.

Reinforcement learning has been applied to various routing schemes in distributed wireless networks, including wireless LANs, wireless sensor networks, cognitive radio networks, and delay-tolerant networks, and it has been shown to improve network performance, such as higher throughput and lower end-to-end delay. RL enables wireless nodes to observe their local operating environment and subsequently learn to make global routing decisions efficiently. It is foreseeable that the advantages that RL brings to routing will attract a great deal of research interest shortly.

Acknowledgement

This work was supported in part by the National Defense Technology Foundation Research Project under Grant JCKY201760**003 and Grant JCKY201860**001.in part by the Key Technology and General Program of Jiangsu Province under Grant BE2018393, and in part by the Key Industrial Technology Innovation Project of Suzhou City under Grant SYG201826.

The authors thank the Associate Editor and the anonymous reviewers for their constructive comments, which helped us improve the presentation of the work considerably.

References

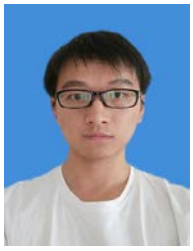
[1] Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E., "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Special Issue on Drone Delivery Systems*, 72(4), 411-458, 2018.. Article (CrossRef Link)

[2] Hayat, S., Yanmaz, E., Muzaffar, R., "Survey on unmanned aerial vehicle networks for civil applications, A communications viewpoint," *IEEE Communications Surveys & Tutorials*, 18(4), 2624-2661, 2016. Article (CrossRef Link)

[3] Mowla, N.I., Tran, N.H., Doh, I., Chae, K., "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET," *Journal of Communications and Networks*, 22(3), 244-258, 2020. Article (CrossRef Link)
[4] Chmaj, G., Selvaraj, H., "Distributed Processing Applications for UAV/drones: A Survey," *Progress in Systems Engineering*, pp 449-454, 2015. Article (CrossRef Link)
[5] Lakew, D.S., Sa'ad, U., Dao, N., Na, W., Cho, S., "Routing in Flying Ad Hoc Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, 22(2), 1071-1120, 2020. Article (CrossRef Link)
[6] Sharma, M., Singh, M., Walia, K., Kaur, K., "A Comprehensive Study of Performance Parameters for MANET, VANET and FANET," in *Proc. of 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0643-0646, 2019. Article (CrossRef Link)
[7] Gankhuyag, G., Shrestha, A.P., Yoo, S.J., "Robust and Reliable Predictive Routing Strategy for Flying Ad-Hoc Networks," *IEEE Access*, vol. 5, pp. 643-654, 2017. Article (CrossRef Link)
[8] Cruz, E.P.F.d., "A Comprehensive Survey in Towards to Future FANETs," *IEEE Latin America Transactions*, 16(3), 876-884, 2018. Article (CrossRef Link)
[9] Arafat, M.Y., Moh, S., "Localization and Clustering Based on Swarm Intelligence in UAV Networks for Emergency Communications," *IEEE Internet of Things Journal*, 6(5), 8958-8976, 2019. Article (CrossRef Link)
[10] Wang, S., Huang, C., Wang, D., "Delay-aware relay selection with heterogeneous communication range in VANETs," *Wireless Networks*, 26(2), 995-1004, 2020. Article (CrossRef Link)
[11] Gunduz, D., De Kerret, P., Sidiropoulos, N.D., Gesbert, D., Murthy, C., Mihaela, V.D.S., "Machine Learning in the Air," *IEEE Journal on Selected Areas in Communications*, 37(10), 2184-2199, 2019. Article (CrossRef Link)
[12] Valadarsky, A., Schapira, M., Shahaf, D., Tamar, A., "Learning to Route," in *Proc. of the 16th ACM Workshop on Hot Topics in Networks*, pp. 185-191, 2017. Article (CrossRef Link)
[13] Li, R., Li, F., Li, X., Wang, Y., "QGrid: Q-learning based routing protocol for vehicular ad hoc networks," in *Proc. of 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, pp. 1-8, 2014. Article (CrossRef Link)
[14] Jung, W., Yim, J., Ko, Y., "QGeo: Q-Learning-Based Geographic Ad Hoc Routing Protocol for Unmanned Robotic Networks," *IEEE Communications Letters*, 21(10), 2258-2261, 2017. Article (CrossRef Link)
[15] Defrawy, K.E., Tsudik, G., "ALARM: Anonymous Location-Aided Routing in Suspicious MANETs," *IEEE Transactions on Mobile Computing*, 10(9), 1345-1358, 2011. Article (CrossRef Link)
[16] Narayanan, P.S., Joice, C.S., "Vehicle-to-Vehicle (V2V) Communication using Routing Protocols: A Review," in *Proc. of 2019 International Conference on Smart Structures and Systems (ICSSS)*, pp. 1-10, 2019. Article (CrossRef Link)
[17] Karp, B., Kung, H.T., "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of the 6th annual international conference on Mobile computing and networking*, Boston, Massachusetts, USA, pp. 243-254, 2000. Article (CrossRef Link)
[18] Huang, H., Yin, H., Min, G., Zhang, J., Wu, Y., Zhang, X., "Energy-Aware Dual-Path Geographic Routing to Bypass Routing Holes in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, 17(6), 1339-1352, 2018. Article (CrossRef Link)
[19] Kasana, R., Kumar, S., "A geographic routing algorithm based on Cat Swarm Optimization for vehicular ad-hoc networks," in *Proc. of 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 86-90, 2017. Article (CrossRef Link)
[20] Mammeri, Z., "Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches," *IEEE Access*, 7, 55916-55950, 2019. Article (CrossRef Link)
[21] Boyan, J.A., Littman, M.L., "Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach," *J.A.i.n.i.p.s.*, 6, 671-678, 1993. Article (CrossRef Link)

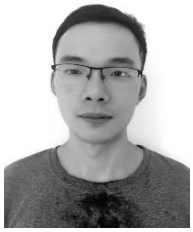
[22] Al-Rawi, H.A.A., Ng, M.A., Yau, K.-L.A., "Application of reinforcement learning to routing in distributed wireless networks: a review," *Artificial Intelligence Review*, 43(3), 381-416, 2015. Article (CrossRef Link)
[23] Gawłowicz, P., Zubow, A., "ns3-gym: Extending openai gym for networking research," *J.a.p.a.*, 2018. Article (CrossRef Link)
[24] You, X., Li, X., Xu, Y., Feng, H., Zhao, J., Yan, H., "Toward Packet Routing with Fully-distributed Multi-agent Deep Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-14, 2020. Article (CrossRef Link)
[25] Singh, N., Elamvazuthi, I., Nallagownden, P., Ramasamy, G., Jangra, A.J.S., "Routing Based Multi-Agent System for Network Reliability in the Smart Microgrid," *Sensors*, 20(10), 2020. Article (CrossRef Link)
[26] Kaviani, S., Bo, R., Ahmed, E., Larson, K.A., Kim, J.H., "Robust and Scalable Routing with Multi-Agent Deep Reinforcement Learning for MANETs," a*rXiv preprint arXiv:2101.03273*, 2021. Article (CrossRef Link)
[27] Pourpeighambar, B., Dehghan, M., Sabaei, M., "Multi-agent learning based routing for delay minimization in cognitive radio networks," *Journal of Network and Computer Applications*, 84, 82-92, 2017. Article (CrossRef Link)
[28] Liang, X., Balasingham, I., Byun, S.-S., "A multi-agent reinforcement learning based routing protocol for wireless sensor networks," in *Proc. of 2008 IEEE International Symposium on Wireless Communication Systems*, pp. 552-557, 2008. Article (CrossRef Link)
[29] Elwhishi, A., Ho, P.H., Naik, K., Shihada, B., "ARBR: Adaptive reinforcement-based routing for DTN," in *Proc. of 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2010. Article (CrossRef Link)
[30] Zheng, L.-M., Li, X.-D., Li, X.-Y., "RLAR: Adaptive routing algorithm based onreinforcement learning," *J.C.E.*, (4), 13, 2011. Article (CrossRef Link)
[31] Mukhutdinov, D., Filchenkov, A., Shalyto, A., Vyatkin, V., "Multi-agent deep learning for simultaneous optimization for time and energy in distributed routing system," *Future Generation Computer Systems*, 94, 587-600, 2019. Article (CrossRef Link)
[32] Mao, H., Gong, Z., Zhang, Z., Xiao, Z., Ni, Y., "Learning multi-agent communication under limited-bandwidth restriction for internet packet routing," *J.a.p.a.*, 2019. Article (CrossRef Link)
[33] Li, X., Hu, X., Zhang, R., Yang, L., "Routing Protocol Design for Underwater Optical Wireless Sensor Networks: A Multi-Agent Reinforcement Learning Approach," *IEEE Internet of Things Journal*, 7(10), 9805-9818, 2020. Article (CrossRef Link)
[34] Zeng, S., Xu, X., Chen, Y., "Multi-Agent Reinforcement Learning for Adaptive Routing: A Hybrid Method using Eligibility Traces," in *Proc. of IEEE 16th International Conference on Control & Automation (ICCA)*, pp. 1332-1339, 2020. Article (CrossRef Link)
[35] Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y., Kim, D.I., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, 21(4), 3133-3174, 2019. Article (CrossRef Link)
[36] Li, X., Hu, X., Zhang, R., Yang, L., "Routing Protocol Design for Underwater Optical Wireless Sensor Networks: A Multiagent Reinforcement Learning Approach," *IEEE Internet of Things Journal*, 7(10), 9805-9818, 2020. Article (CrossRef Link)
[37] Afzal, K., Tariq, R., Aadil, F., Iqbal, Z., Ali, N., Sajid, M., "An Optimized and Efficient Routing Protocol Application for IoV," *Mathematical Problems in Engineering*, 9977252, 2021. Article (CrossRef Link)
[38] Hong, J., Zhang, D., Niu, X., " Impact Analysis of Node Motion on the performance of FANET routing protocols," in *Proc. of 14th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2018)*, pp. 147-162, 2018. Article (CrossRef Link)

Xiulin Qiu was born in Ganzhou, Jiangxi Province, master's degree. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include deep reinforcement learning, resource allocation for 5G and artificial intelligence based future mobile network



Yongsheng Xie was born in Jiangxi Yichun, master's degree, School of computer science and engineering, Nanjing University of technology, computer application technology major, research direction, reinforcement learning, flight ad hoc network.

Yinyin Wang was born in Yancheng, Jiangsu, China. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include machine learning, high performance computing, and medical big data.



Lei Ye was born in Taizhou, Jiangsu, China. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include network coding, big data analysis, and machine learning.

Yuwang Yang received the B.S. degree from North Western Polytechnical University in 1988, the M.S. from the University of Science and Technology of China in 1991, and the Ph.D. degree from the Nanjing University of Science and Technology in 1996. He is currently a Professor with the School of Computer Science and Engineering, NUST. His research interests are high performance computing, machine learning, and intelligent system.
(e-mail: yuwangyang@njust.edu.cn)