Computer Communications 150 (2020) 304–316

Contents lists available at ScienceDirect

Computer Communications

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

QMR:Q-learning based Multi-objective optimization Routing protocol for Flying Ad Hoc Networks

Jianmin Liu [a,b], Qi Wang [a,*], ChenTao He [a,b], Katia Jaffrès-Runser [c], Yida Xu [a,b], Zhenyu Li [a], YongJun Xu [a]

[a] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China
[c] Université de Toulouse, IRIT/ENSEEIHT, F-31061, Toulouse, France

ARTICLE INFO

Keywords:
Multi-objective routing
FANETs
Q-learning
Adaptive parameters

ABSTRACT

A network with reliable and rapid communication is critical for Unmanned Aerial Vehicles (UAVs). Flying Ad Hoc Networks (FANETs) consisting of UAVs is a new paradigm of wireless communication. However, the highly dynamic topology of FANETs and limited energy of UAVs have brought great challenges to the routing design of FANETs. It is difficult for existing routing protocols for Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs) to adapt the high dynamics of FANETs. Moreover, few of existing routing protocols simultaneously meet the requirement of low delay and low energy consumption of FANETs. This paper proposes a novel Q-learning based Multi-objective optimization Routing protocol for FANETs to provide low-delay and low-energy service guarantees. Most of existing Q-learning based protocols use a fixed value for the Q-learning parameters. In contrast, Q-learning parameters can be adaptively adjusted in the proposed protocol to adapt to the high dynamics of FANETs. In addition, a new exploration and exploitation mechanism is also proposed to explore some undiscovered potential optimal routing path while exploiting the acquired knowledge. Instead of using past neighbor relationships, the proposed method re-estimates neighbor relationships in the routing decision process to select the more reliable next hop. Simulation results show that the proposed method can provide higher packet arrival ratio, lower delay and energy consumption than existing good performing Q-learning based routing method.

1. Introduction

Nowadays, Unmanned Aerial Vehicles (UAVs) have become one of the most important technical areas. To achieve more complex applications which are very difficult for traditional Mobile Ad Hoc Networks (MANETs) or individual UAV, Flying Ad Hoc Networks (FANETs) consisting of UAVs have been intensively studied. FANETs play an important role in the Internet-of-Everything. Significant applications of FANETs are related to the quick deployment of connectivity in situations, such as surveillance [1], emergency communications and rescue [2]. However, FANETs are a kind of wireless networks with dynamic and unsustainable topologies due to the high mobility of nodes. Thus, routing in a network with easily disconnected feature is regarded as one of the main challenges in FANETs.

Many routing protocols have been designed for wireless ad hoc networks: proactive, reactive and hybrid routing. Proactive routing creates routes before packets are forwarded. However, maintaining the routing table information makes larger control overhead. Instead, reactive routing creates routes when packets are forwarded. But it brings larger delay due to discovery delivery paths. Hybrid routing takes a trade-off between proactive and reactive routing. It combines the advantages of low delay for proactive routing and low network control overhead for reactive routing. It is mainly suitable for networks with stable network topology.

Due to the changing network topology, geographic information based routing becomes the primary option for improving routing performance. For example, GPSR [3] is a typical geographic information based routing protocol, which designs a packet delivery strategy based on the geographic information of the network node. However, when routing hole problem occurs frequently, the hop count increases because the next hop of the decision is random. The majority of algorithms so far have not paid significant attention to the movement pattern of nodes or numerous assumptions about nodes and network topology are considered. However, in FANETs, the high mobility of nodes leads to the highly dynamic and unsustainable topology. An adaptive and highly autonomous protocol is desired, which means that

[*] Corresponding author.
E-mail addresses: liujianmin18z@ict.ac.cn (J. Liu), wangqi08@ict.ac.cn (Q. Wang), hechentao@ict.ac.cn (C. He), katia.jaffres-runser@irit.fr (K. Jaffrès-Runser), xuyida@ict.ac.cn (Y. Xu), zyli@ict.ac.cn (Z. Li), xyj@ict.ac.cn (Y. Xu).

https://doi.org/10.1016/j.comcom.2019.11.011
Received 19 July 2019; Received in revised form 4 November 2019; Accepted 8 November 2019
Available online 27 November 2019

the routing protocol of FANETs should have the ability to find a reliable neighbor to complete the transmission through perceiving the change of the environment adaptively. Q-learning is an adaptive machine learning with environmental feedback as input, which contributes to adaptive routing design. In Q-learning, agents could constantly adjust their action strategies according to the reward of environmental feedback to better adapt to the dynamic and unsustainable topology.

The routing protocol based on Q-learning relies on the local data of the neighboring nodes and it did not make any assumptions about the environment. The existing Q-learning based routing protocols such as QGrid [4], QLAR [5], QGeo [6], make the best choice among the neighbors at any moment to transmit a packet to the destination. Due to the requirement of real-time data transmission and the limited energy of UAVs, it is crucial for a routing protocol to provide low delay and low energy consumption service guarantees. However, the delay and energy metrics often conflict. The optimal routing solution maximizing for instance delay may not be the one that minimized energy. A multi-objective optimization routing protocol is desired to concurrently meet the low delay and low energy consumption requirements. However, most of the existing routing protocols focus on mono-objective optimization, and seldom pay attention to multi-objective optimization. Thus, this paper proposes a multi-objective optimization routing protocol to provide the service guarantees, which cannot be ensured in the existing Q-learning based routing protocols. In addition, most of them use fixed values for Q-learning parameters and lack of a reasonable scheme to balance exploration and exploitation. Due to the high dynamic of UAV networks, if Q-learning parameters including learning rate and discount factor are fixed, the accuracy of action selection declines, and the selected link may have the low probability of connecting to a neighbor node. The faster the network topology changes, the bigger the learning rate should be to pay more attention to new information, and the smaller discount factor should be to reflect unstable future expectations. However, the fixed learning rate and discount factor are not capable of reflecting the dynamic mobile environment, which leads to poor performance. Meanwhile, the reasonable compromise between exploration and exploitation, could make it possible to explore some undiscovered potential optimal routing path while exploiting the acquired knowledge. Due to these limitations, in this paper, we propose a novel Q-learning based Multi-objective optimization routing protocol. The main contributions of this paper are as follows:

- **Joint optimization of delay and energy consumption.** Without fixed route vector table, QMR utilizes Q-learning to perform multi-objective optimization routing instead of mono-objective as in [6]. In the multi-objective optimization, end-to-end delay and energy consumption are simultaneously minimized.
- **Adaptively adjust Q-learning parameters.** Due to the mobility of nodes, the link quality is extremely unstable. In this method, each link is given a different learning rate, and each node is given a different discount factor. Furthermore, the learning rate and discount factor are adaptively adjusted according to the network condition.
- **Re-estimate neighbor relationships.** High mobility of nodes leads to unstable neighbor relationships. Therefore, it is incorrect to use the past neighbor relationships to determine the current relationships. In our QMR, neighbor relations are re-estimated in the routing decision process to get the most reliable next hop.
- **Improve the exploration and exploitation mechanism.** In FANETs, the balance process of exploration and exploitation should not be simply regulated by learning time, but regulated by the network condition. Different from traditional methods, such as ε greedy strategy, Boltzmann and Upper-Confidence-Bound(UCB), we propose a adaptive mechanism of exploration and exploitation, which balances exploration and exploitation according to the network condition.

This paper is organized as follows. The related work of routing protocol for ad hoc networks is presented in Second 2. Section 4 proposes a Q-learning based Multi-objective optimization Routing protocol for FANETs. The performance evaluation is then given in Section 5. Finally, Section 6 makes a conclusion about this paper.

2. Related work

At present, there is no a proprietary routing protocol for FANETs [7]. Most routing protocols of FANETs are modifications of routing protocols of MANETs. MANET routing protocols can be divided into static routing, proactive routing, reactive routing and hybrid routing.

2.1. Static routing protocols

In static routing protocols, static routing tables of static routing protocols must be computed and loaded before the task starts and cannot be updated during an operation. Due to this limitation, these protocols are not fault tolerant and do not apply to dynamically changing environments.

Load Carry and Delivery Routing (LCAD) [8] is the first routing protocol in FANET, where a UAV flies and carries data from a ground node through to the destination. The purpose of LCAD is to maximize the throughput and increase security. Although LCAD achieves higher throughput, data delivery delay is longer in LCAD due to use of a single UAV.

Multi Level Hierarchical Routing (MLHR) [9] solves the scalability problems of large-scale vehicle networks which performance degrades as the size increases. The size and operation area can be increased by organizing the network as hierarchical structure. Analogously, UAV networks can be grouped into multiple clusters in which only cluster head is connected outside the cluster. However, frequent change of cluster heads leads to large network overhead.

2.2. Proactive routing protocols

In Proactive routing protocols, all nodes store the routing information from this node to other nodes by a routing table. When the topology of the network changes, nodes need maintain and update their routing tables by exchanging routing information. However, in FANETs, the high mobility of nodes leads to the frequent change of the network topology. So, such routing protocols are not suitable to be used in FANETs due to bandwidth constraints.

The Optimized Link State Routing (OLSR) [10] is a well-known proactive routing protocol where two types of messages including "hello" and "topology" are used to finish routing. The "hello" message is used to find neighbor nodes in the communication rang and maintain neighbor node list. Whereas "topology" message is use for maintaining the topology information in routing tables. However, this protocol has high control overhead due to periodically exchanging messages.

Link-quality and traffic-load aware optimized link state routing protocol (LTA-OLSR) has improved OLSR to apply to FANETs [11]. LTA-OLSR integrates the link quality and traffic load schemes with OLSR. The link quality scheme is designed to distinguish the link quality between the node and its neighboring nodes based on the statistical information of received packets. The traffic load scheme can ensure a light-load path by considering MAC layer channel contention information and the number of packets stored in the buffer. Compared to OLSR, LTA-OLSR can provide reliable and efficient communication in FANETs, because the link quality and the traffic load are considered.

2.3. Reactive routing protocols

In reactive routing protocols, a route is created only when a packet need to be transmitted from the source to the destination, and nodes do not need to maintain the routing information in real time. So,

Table 1
Comparison between routing protocols.

Routing protocol	Consider end-to-end delay	Consider energy consumption	Q-learning parameters	Exploration and utilization policy	Type of routing	Type of network
LCAD [8]	No	No	–	–	Static routing	FANETs
MLHR [9]	No	No	–	–	Static routing	FANETs
LTA-OLSR [11]	Yes	No	–	–	Proactive routing	FANETs
LEPR [12]	Yes	No	–	–	Reactive routing	FANETs
NEMA [13]	No	Yes	–	–	Hybrid routing	FANETs
OPT-EQ-Routing [14]	No	Yes	Both α and γ are fixed	ε greedy	Q-learning based routing	WSNs
QLAR [5]	Yes	No	Both α and γ are fixed	ε greedy	Q-learning based routing	MANETs
QGrid [4]	No	No	α is fixed, γ is variable	Greedy and Markov	Q-learning based routing	VANETs
QGeo [6]	Yes	No	α is fixed, γ is selected form two values	ε greedy	Q-learning based routing	FANETs

Type of network: (WSNs: Wireless Sensor Networks, MANETs: Mobile Ad Hoc Networks, VANETs: Vehicular Ad Hoc Networks, FANETs:Flying Ad Hoc Networks).

this type of routing protocols are also called on-demand routing protocols. Although reactive routing protocols reduce overhead problem of proactive routing protocols, they have large delay due to route construction.

Ad-hoc On-demand Distance Vector (AODV) [15] is a well-known reactive protocol in mobile ad hoc network. AODV consists of three phases: routing discovery, transmission of packet and route maintenance. In AODV, nodes (source and relay nodes) hold only one entry for each destination and store next hop information corresponding to each data communication.

Link Stability Estimation-based Preemptive Routing (LEPR) protocol for FANETs is proposed on the basis of AODV [12]. LEPR constructs multiple reliable link-disjoint paths with a new link stability metric which takes into account the past, current and future statuses of link stability. In addition, a preemptive route maintenance mechanism is proposed to repair links that may be broken soon.

2.4. Hybrid routing protocols

Hybrid routing takes a trade-off between proactive and reactive routing. It combines the advantages of low delay for proactive routing and low network control overhead for reactive routing. In Hybrid routing protocols, the network is divided into different zones where proactive protocol and reactive protocol are used for routing in intra zone and inter zone, respectively.

Zone Routing Protocol (ZRP) [16] is a hybrid routing protocol which works on the concept of zones. Each node has an alternate zone separated by a predefined range called R. Proactive routing is used to route inside the zone. When information needs to be sent outside the zone, reactive strategy is used.

Node Energy Monitoring Algorithm for Zone Head Selection (NEMA) improves ZRP by adding energy constraints to hence the lifetime of MANET [13]. NEMA consists of two parts. In the first part, a zone head selection algorithm is designed to select a zone head with maximum residual power. In the second part, node energy monitoring algorithm is designed to monitor the change of the residual energy of each node and set different phase for each node according to residual energy level.

2.5. Q-learning based routing protocol

Complex flight environments and diverse flight tasks have caused FANETs to be in an unpredictable random fluctuation state. Therefore, the above routing protocols are difficult to adapt to the change of the network in real time, which may deteriorate network communication performance for a long time. Hence, an adaptive and highly autonomous protocol which is capable of discovering reliable communication links adaptively and autonomously is desired. Reinforcement learning (RL) is an adaptive learning method that belongs to the category of machine learning. It is a good idea to use reinforcement learning to solve routing problems of FANETs.

For the first time, Boyan and Littman used RL to solve the routing problem in static networks. Based on Q-learning, an adaptive algorithm Q-routing is proposed [17]. Getting the shortest path from the source to the destination may be a good way to achieve fast routing. However, a path with the minimum number of hops cannot be the best route because it may be heavily congested. Thus, Q-routing learns a routing policy that balances minimizing the number of "hops" a packet will take with the possibility of congestion along popular routes. As a result, Q-routing is better than nonadaptive algorithm based on precomputed shortest path. The [14] extends the Q-Routing and proposes an energy balancing routing algorithm, designed for Wireless Sensor Networks. The goal of the energy balancing routing algorithm is to optimize the network lifetime by balancing the routing effort among the sensors in consideration of their current remaining batteries. Today, Q-learning based routing protocols for Ad Hoc Networks are gradually emerging. In [4], a Q-learning Based Routing Protocol (QGrid) for Vehicular Ad Hoc Networks is proposed. QGrid divides the area into different grids to make routing decision from macroscopic and microscopic aspects. The optimal next-hop grid is determined in macroscopic aspect, and then a specific vehicle is selected in the optimal next-hop grid as next-hop vehicle in microscopic aspect. Although the simulation confirmed that QGrid has better performance than the existing position based routing protocols, such as higher packet delivery ratio, end-to-end delay, energy consumption of nodes are not considered in QGrid. The [5] proposes a Q-Learning based adaptive routing (QLAR) for MANETs. QLAR develops a new model to detect the mobility level of each node in the network and a new metric to account for the static and dynamic routing. Although end-to-end delay of QLAR is lower than OLSR, QLAR still does not consider energy consumption of nodes.

Recently, Jung et al. proposed a Q-Learning-Based geographic ad hoc routing protocol for Unmanned Robotic Networks (QGeo). QGeo uses a distributed routing decision mechanism based on the geographical location information of nodes. In QGeo, the reward value of the action is related to the packet travel speed. Meanwhile, link condition and location estimation error are considered when calculating the travel time. Experimental results show that QGeo has higher packet delivery ratio and lower end-to-end delay compared with QGrid [4] in mobile scenarios. However, there are still some shortcomings in QGeo. QGeo does not consider energy consumption, which means it could not simultaneously provide low delay and low energy consumption service guarantees. Limited battery lifetime is considered a major drawback of UAVs [18]. Although carrying an extra battery pack can power the UAV continuously for a period of time, the battery pack is still limited. Thus, it is very necessary to design a routing protocol with low energy consumption. In addition, QGeo utilizes a fixed learning rate in Q-learning approach. Learning rate is used to control the speed of updating Q-value. A fixed learning rate implies that the speed of updating Q-value is constant. However, links in FANETs are extremely unstable. Therefore, the speed of updating Q-value associated with the link should be adaptively adjusted as network environment changes.

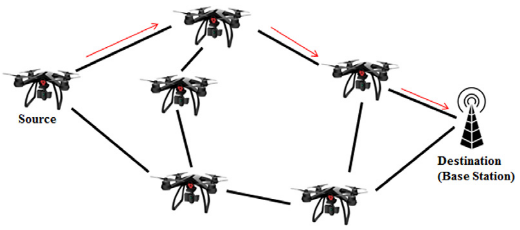


Fig. 1. A multi-hop FANET with multi-UAVs.

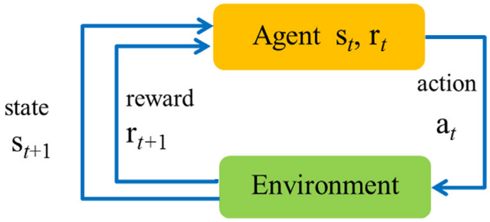


Fig. 2. The interaction between agent and environment in Q-learning.

Based on the above analysis, Table 1 compares the characteristics of exiting routing protocols for FANETs and Q-learning based routing protocols. Static, proactive, reactive and hybrid routing protocols do not belong to adaptive routing protocols, which are not able to discover reliable communication links adaptively and autonomously. Although there are some adaptive routing protocols based on Q-learning, they still have some shortcomings. On one hand, existing Q-learning based routing protocols pay little attention on multi-objective optimization to consider end-to-end delay and energy consumption simultaneously. On other hand, Q-learning parameters, learning rate and discount factor, cannot be adaptively adjusted according to the network condition.

For the above motivation, we proposed a multi-objective routing protocol where end-to-end delay and energy consumption are optimized concurrently. Furthermore, we also propose a method of adaptively adjusting Q-learning parameters and a new exploration and exploitation mechanism in Q-learning, in order to adapt to the high dynamics of FANETs.

3. System model

In this section, we present a multi-hop FANET model and Q-learning model of the FANET.

3.1. Multi-hop FANET model

In this paper, we consider a UAV network comprising multiple UAVs and a ground station, as shown in Fig. 1. The ground station is regarded as the destination node to receive signals from the UAVs, one UAV is regarded as the source node to send signals, and the rest UAVs is regarded as the relay nodes to forward signals. The key issue is how to find an optimal path so that signals transmitted along the path can successfully reach the destination with low delay and low energy consumption.

3.2. Q-learning model of FANETs

3.2.1. The basic of Q-learning technology

Reinforcement learning is an area of machine learning where agents alter their actions in a specific environment with the goal of maximizing results. In order to achieve the goal, agents constantly adjust their action strategies through the reward of environmental feedback. Reinforcement learning usually considers the reward by estimating value functions, state value functions or action value functions [19]. Q-learning is an optimization of the off-policy temporal difference(TD) [20]. The foundational idea of the Q-learning is learning from interaction with environment. It uses action value functions to get the feedback from environment. In the technique, the agent chooses an action in a particular state according to the reinforcement (Q-value). Reinforcement consists of direct reward and the future Q-value expectation. Through reinforcement, agents can assess how good an action in the current state and makes a better action at the next step. The goal of the agent is to maximize expectation for cumulative rewards over the long run. The basic iterative formula for the Q-value is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \underset{a}{max} Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{1}$$

where, α is the learning rate, γ is the discount factor, and they are set between 0 and 1. r_{t+1} is the direct reward value that the agent, at time t, takes action a_t in state s_t. After the agent taking action a_t, the state of the agent changes from the s_t state to the s_{t+1} state. $max_a Q(s_{t+1}, a)$, future Q-value expectation, is maximum Q-value when agent selects possible action a in the next state s_{t+1}. Fig. 2 shows the interaction between agent and environment in Q-learning.

3.2.2. Q-learning model

In our QMR routing, considering a packet coming from a source and directed to a destination by multi-hop communication, the whole network is considered as an environment, and each packet in the network represents an agent. Each state of the agent indicates a node who holds the packet. For example, when a packet is at node i, the current state associated with this packet is s_i. An action $a_{i,j}$ represents the decision of the packet (agent) to be forwarded from node i to neighbor node j. By this action, the state of the agent moves from s_i to s_j and the agent receives a reward about the action.

4. Q-learning based multi-objective optimization routing

In this section, QMR, a Q-learning based Multi-objective optimization Routing protocol for FANETs, is introduced. In QMR, nodes utilize a reinforcement learning algorithm without knowledge of entire networks to make optimal routing decisions considering low-delay and low-energy service. To solve the routing problem caused by the high mobility within FANETs, adaptive Q-learning parameters and a new exploration and exploitation mechanism for Q-learning are also proposed to enhance the routing performance. The QMR module consists of Routing neighbor discovery, Q-learning, Routing decision and Penalty mechanism. The flowchart of the QMR framework is shown in Fig. 3.

In QMR, nodes acquire their geographic location information by GPS. Further, the routing neighbor discovery is implemented by sending HELLO packets. When a data packet generated from a source and directed to a destination, Q-learning is the key component of routing decision. In the routing decision, if a routing hole problem [21] is encountered (i.e., all the neighbors of a node are distant than the distance from this node to the destination), the penalty mechanism is triggered. Algorithm 1 shows QMR method and the details of the work at each phase of QMR.

4.1. Routing neighbor discovery

Each node periodically sends HELLO packets that include the node's geographic location, energy, mobility model, queuing delay, and discount factor. When a node receives HELLO packets, it uses the information in HELLO packets to establish and maintain its neighbor table. The Neighbor table contains the important information of each neighbor including the geographic location, energy, mobility model (i.e., moving speed and direction of the neighbor), arrival time, discount factor,

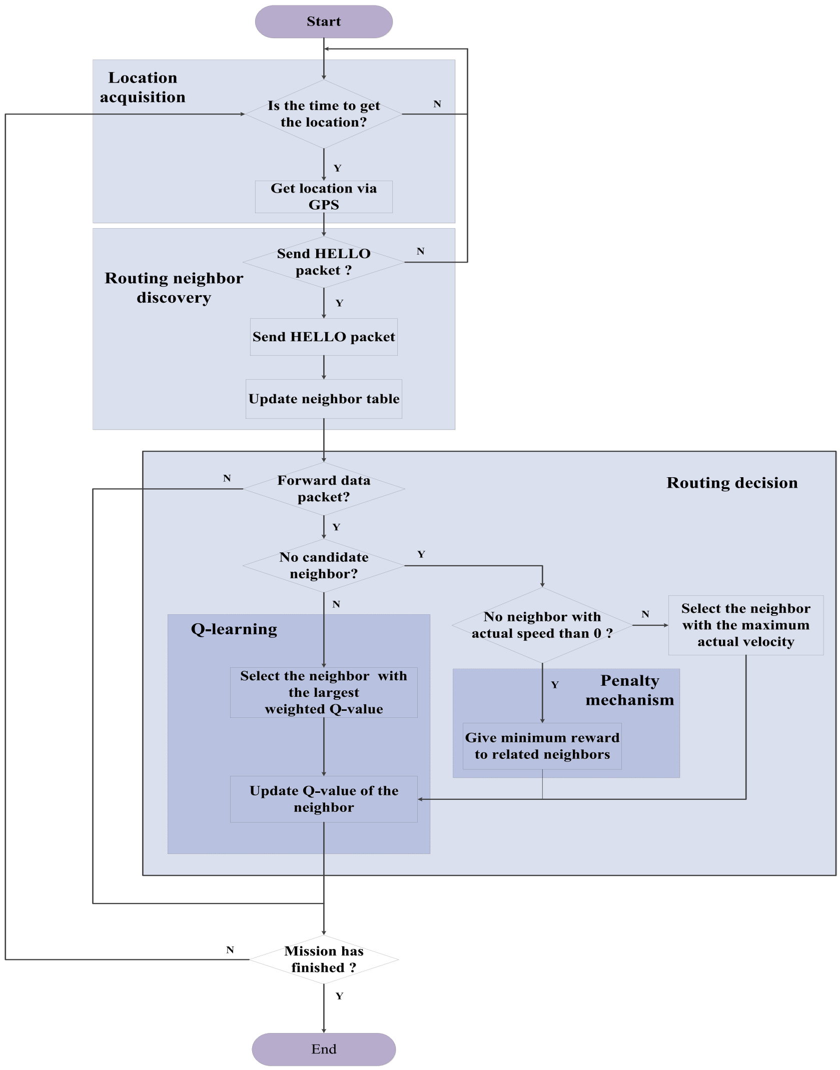

Start
Location acquisition
Is the time to get the location?
N
Y
Get location via GPS
Routing neighbor discovery
Send HELLO packet ?
N
Y
Send HELLO packet
Update neighbor table
Routing decision
Forward data packet?
N
Y
No candidate neighbor?
Y
N
Q-learning
Select the neighbor with the largest weighted Q-value
Update Q-value of the neighbor
No neighbor with actual speed than 0 ?
N
Select the neighbor with the maximum actual velocity
Y
Penalty mechanism
Give minimum reward to related neighbors
Mission has finished ?
N
Y
End

Fig. 3. Flowchart of QMR framework.

queuing delay and Q-value. Moreover, neighbor table also stores some information about links from current node to its neighbors, such as learning rate and MAC delay. Each node utilizes the information of its neighbor table to perceive the network condition. If the information of a neighbor is not refreshed after a ExpireTime [22], it will be removed from the neighbor table. The time interval of the HELLO packet and the ExpireTime can be adjusted according to the moving speed of the node. The higher the moving speed of the node is, the smaller the time interval of the HELLO packet and the ExpireTime are. On the contrary, the time interval of the HELLO packet and the ExpireTime become larger.

4.2. Q-learning in QMR

In our QMR, Q-learning is used to make multi-objective optimization routing decisions, where end-to-end delay and energy consumption are considered in the reward function. To adapt to the dynamic mobile environment, Q-learning parameters including learning rate and discount factor are adaptively adjusted according to the delay and neighbor mobility. Furthermore, an adaptive mechanism of the exploration and exploitation in the Q-learning is also proposed to adapt to the dynamic network.

4.2.1. Energy metric

In order to balance the energy consumption of nodes, we consider the energy of the node when selecting the next hop. In QMR, we use the ratio of the residual energy of the node to the initial energy (i.e., node residual energy level [23]) as a energy metric to measure the energy consumption of the node. The energy metric E_i of node i can be expressed as follows:

$$E_i = \frac{E\_res_i}{E\_init_i} \tag{2}$$

where E_res_i is the residual energy of the node i, and E_init_i is the initial energy of the node i. The larger the E_i is, the lower the energy consumption of the node i is; otherwise, the higher the energy consumption is.

4.2.2. Reward function

In QMR, we jointly consider the delay and energy consumption in the reward function. The expression of the reward function $f_R(s_t, a_t)$ is as follows:

$$f_R(s_t, a_t) =$$

$$\begin{cases} r_{max} & , when\ s_{t+1}\ is\ destination \\ r_{min} & , when\ s_t\ is\ local\ minimum \\ \omega * e^{-delay_{i,j}} + (1-\omega) * E_j & , otherwise \end{cases} \tag{3}$$

where $\omega(0 < \omega < 1)$ is the weight for one-hop delay. Suppose that at time t, the state of the data packet (agent) is s_i, meaning that the data packet stays at node i. The action a_t is to select node j for forwarding

Algorithm 1 QMR

```
Phase 1: Location acquisition
if the time of updating node location arrives then
    each node obtains its location via GPS;
end if
Phase 2: Routing neighbor discovery
if the time of sending HELLO packets arrives then
    Each node sends a HELLO packet;
    These nodes receiving HELLO packets reconfirm their neighbors
    and update neighbor tables according to HELLO packets.
end if
Phase 3: Routing decision
while Data packets need to be sent do
    if the set of candidate neighbors is not empty then
        Make routing decision based on Q-learning;
    else
        if there are neighbors whose actual velocity is greater than 0
        then
            Select the neighbor associated with the maximum actual
            velocity;
        else
            Penalty mechanism;
        end if
    end if
end while
Phase 4: Q-learning
if No information about neighbors then
    Initialize the Q-value of each neighbor;
end if
Select the neighbor associated with the largest $\kappa$-weighted Q-value;

if a neighbor is selected to forward data packets then
    Update Q-value of the neighbor by reward;
end if
Phase 5: Penalty mechanism
if the routing hole problem is encountered, or forwarding nodes do
not receive ACK packets then
    Give minimum reward to related neighbors.
    Update Q-value of related neighbors.
end if
```

data packet. The state of the data packet will be converted from state s_i to state s_j, i.e., at the next time t+1, the state of the packet is s_j. $delay_{i,j}$ is the one-hop delay from node i to node j, E_j is the residual energy level of the next hop j. When the next hop j is the destination node, it means that the destination node is the neighbor of node i, so the link from node i to node j gets the maximum reward value r_{max}. When node i is a local minimum, it means that all neighbors of node i are farther away from the destination than node i. If the packet is forwarded to any neighbor of node i, the packet will take longer time to reach the destination node. Therefore the minimum reward value r_{min} is derived.

4.2.3. Adaptive Q-learning parameters

In Q-learning, the learning rate determines to what extent the newly acquired information overrides the old information. If the learning rate is higher, the Q-value is updated faster. Most of the existing Q-learning based routing protocols have a fixed learning rate. However, in FANETs, if the link between nodes is more unstable, the one-hop delay will be larger. Then the speed of updating Q-values should be faster. Therefore, we introduce a method of adjusting the learning rate to adapt to the speed of updating Q-value by assigning a corresponding learning rate for each link.

We use the one-hop delay to measure the stability of the link, where a relatively stable link has a smaller delay. The normalized one-hop delay $\varepsilon_{i,j}$ can be expressed as follows:

$$\varepsilon_{i,j}=\frac{|delay_{i,j}-\mu_{i,j}|}{\sigma_{i,j}} \tag{4}$$

where $delay_{i,j}$ is the one-hop delay from node i to node j, $\mu_{i,j}$ and $\sigma_{i,j}$ ($\sigma_{i,j}\neq 0$) are the mean value and variance of one-hop delay respectively. Since the learning rate ranges from 0 to 1, we introduce an exponential function for adaptive learning rate. The adaptive learning rate $\alpha_{i,j}$ associated with the link from node i to node j can be expressed as follows:

$$\alpha_{i,j}=\begin{cases}1-e^{-\varepsilon_{i,j}} & \sigma_{i,j}\neq 0\\ 0.3 & \sigma_{i,j}=0\end{cases} \tag{5}$$

In the Q-learning algorithm, discount factor γ represents the stability of future Q-value expectation. A high value of discount factor indicates that future Q-value expectation is stable, while the low value indicates the vulnerable Q-value expectation. Since the routing decision is to find the reliable neighbor to forward the packet, it is important to adjust the discount factor according to the mobility of neighbors in the adjacent time intervals. The faster neighbors of the node move in adjacent time intervals, the more unstable the future Q-value expectation associated with the node is. Then the discount factor of the node is lower. On the contrary, the discount factor is higher. For the node i, the discount factor γ_i is defined as follow:

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

where $N_i(t-1)$ and $N_i(t)$ are the neighbor sets of the node i at time $t-1$ and time t respectively.

4.2.4. Exploration and exploitation

In order to select the most reliable neighbor to forward the packet in FANETs, we also investigate how to balance the relationship between exploration and exploitation in the Q-learning. The exploration is to search for unknown actions (get new knowledge), but too much exploration makes it difficult to retain some better actions. While the exploitation is to take advantage of explored actions which may generate high returns, but too much exploitation makes it difficult to select some undiscovered potential optimal actions.

In order to balance the relationship between exploration and exploitation, the ε greedy strategy, Boltzmann mechanism [24] and Upper-Confidence-Bound (UCB) [25] are usually adopted in the field of reinforcement learning. The ε greedy strategy is that the quantitative allocation method is used to explore with a small probability ε, which cannot be adaptively adjusted. Although UCB and Boltzmann mechanism can change the degree of exploration, the degree of exploration is regulated according to time. This method of regulating exploration through time may be more adaptive to static environments. However, for dynamic environments such as FANETs, the balance between exploration and exploitation should be regulated by the network condition rather than simply by time.

In QMR, we propose a new mechanism of exploration and exploitation in Q-learning routing. The mechanism balances exploration and exploitation according to network condition, including actual velocity of a data packet traveling over the link, link quality and intimacy of neighbor relationship. The balance between exploration and exploitation in this method is mainly reflected in the following two aspects:

- When selecting the next hop, instead of directly selecting the link with the largest Q-value, the actual velocities of a data packet traveling over links from the current node to its neighbors are evaluated. These neighbors associated with links that meet velocity requirements are selected (more details are discussed in 4.3.2). By filtering the neighbors with actual velocities, the data packet may arrive at the destination node with smaller delay. Meanwhile, neighbors that have not been selected in the past but meet the velocity requirement may also be used as the next hop, which means the agent wants to explore new actions.

- Among the neighbors that meet the velocity requirement, the κ-weighted Q-value of each link is calculated. The neighbor with the maximum κ-weighted Q-value is selected. The weight κ that depends on link quality and intimacy of neighbor relationship (see 4.3.3 for details), is used to evaluate the current condition of the link. Furthermore, Q-value is used to measure the past condition of the link. The Q-value can be regarded as the empirical value of learning link, which reflects that the agent is exploiting the learned knowledge. On the basis of Q-value, the weight is introduced, which actually reflects that the agent is balancing the relationship between exploration and exploitation.

4.3. Routing decision

Considering the real-time transmission, each data packet has a corresponding deadline to constrain the velocity of the data packet during the transmission. Neighbors whose actual velocities are not less than the constraint velocity will be selected as the candidate neighbors for routing decision. Furthermore, Q-learning is utilized to perform the optimal routing decision upon these candidate neighbors.

4.3.1. Delay constraint

For the real-time application, data packets generated by source have to be transmitted to the destination before a certain deadline. In QMR, when a relay node decides to forward a data packet, the latest deadline for this data packet will be updated by subtracting its passed time.

Assuming that node i selects node j as the next hop to forward data packets, node i calculates the one-hop delay $delay_{i,j}$ from node i and node j using the medium access delay (MAC delay) and the queuing delay recorded in the neighbor table. Since data packets travel at the speed of light in wireless media, propagation delay is approximately nanoseconds in the communication range of the order of hundred meters. Therefore, propagation delay is negligible compared to the MAC delay and queuing delay. The one-hop delay $delay_{i,j}$ is expressed as follows:

$$delay_{i,j} = D\_mac_{i,j} + D\_que_{i,j} \tag{7}$$

where $D\_mac_{i,j}$ is the MAC delay, which is the time needed by the medium access protocol to either successfully deliver the packet or drop it in case of repeated failures. $D\_que_{i,j}$ is the queuing delay, which is the time for the packet to reach the head of the transmission queue.

The MAC delay t_m is estimated by ACK packets and is calculated by

$$t_m = t_{ACK} - t_{send} \tag{8}$$

where t_{ACK} is the moment when a node receives the ACK packet from its neighbor, and t_{send} is the moment when the node sends the data packet to its neighbor.

The method of window mean with exponentially weighted moving average (WMEWMA) is used to update MAC delay. For a node i with m neighbors, it always maintains m sliding windows with length n. Each window records the MAC delay of the last n data packets sent by node i to node j. The formula for the lth updated MAC delay is as follows:

$$D\_mac_{i,j}(l) = (1-\beta)\frac{\sum_{k=l-n}^{l\text{-}1} D\_mac_{i,j}(k)}{n} + \beta t_m \tag{9}$$

where $\beta (0 < \beta < 1)$ is the tunable weighting coefficient. The MAC delay t_m is measured from the time of node i sends the packet to the time node i gets an acknowledgment (ACK) from the node j, which is expressed as follows:

Similarly, the queuing delay is updated by

$$D\_que_{i,j}(l) = (1-\beta)\frac{\sum_{k=l-n}^{l\text{-}1} D\_que_{i,j}(k)}{n} + \beta t_q \tag{10}$$

where t_q is the waiting time for the data packet to reach the head of the transmission queue.

Assume that node i sends a data packet to node j, and the deadline of the data packet at nodes i and j are respectively deadline_i and deadline_j. The deadline is updated by:

$$\text{deadline}_j = \text{deadline}_i - delay_{i,j} \tag{11}$$

4.3.2. Velocity constraint

To meet delay constraint, we define the requested velocity and actual velocity. The requested velocity to transmit the data packet from node i to destination D has to meet the constraint of the deadline for end-to-end packet delivery. When node i will forward a data packet, it calculates the requested velocity according to the current deadline of the data packet. The requested velocity to transmit the data packet at node i can be expressed as follows:

$$V_i = \frac{d_{i,D}}{deadline_i} \tag{12}$$

where $d_{i,D}$ is the distance from node i to destination D, and $deadline_i$ is the deadline of the data packet at node i.

In QMR, node i considers the mobility of neighbor j when calculating the actual velocity of a data packet from node i to its neighbor j. Here, we assume that the node's moving speed is fixed for a certain period of time. The position of the neighbor node is predicted by the moving speed and moving direction of the neighbor node recorded in the neighbor table. Let node i add node j to the neighbor table at time t_1. The location of node j at time t_1 is $(x_j(t_1), y_j(t_1))$. The moving speed is Vm_j and the moving direction is $angle\_xy_j$. t_2 is the current moment, that is, node i makes a routing decision at time t_2. Assuming that node i selects node j as the next hop and the data packet reaches node j at time t_3, the position of node j at time t_3 can be estimated as:

$$\hat{x}_j(t_3) = x_j(t_1) + Vm_j * \cos(angle_j) * (t_3 - t_1) \tag{13}$$

$$\hat{y}_j(t_3) = y_j(t_1) + Vm_j * \sin(angle_j) * (t_3 - t_1) \tag{14}$$

$$t_3 = t_2 + delay_{i,j} \tag{15}$$

According to the current position of node i and the predicted position of neighbor node j, the actual velocity $v_{i,j}$ from node i to node j can be obtained.

$$v_{i,j} = \frac{d_{i,D} - \hat{d}_{j,D}}{delay_{i,j}} \tag{16}$$

where $d_{i,D}$ is the distance between the real position of the node i at the time t_2 and the destination D. $\hat{d}_{j,D}$ is the distance between the predicted position of the node j at the time t_3 and the destination D.

To meet the deadline, the actual velocity to forward the packet should be not less than the requested velocity. It means that the candidate node must satisfy $v_{i,j}$ no less than V_i.

4.3.3. The selection of the optimal forwarding node

Due to the high mobility of nodes, the neighbor relationship and link quality between two nodes are extremely unstable. To solve this problem, we introduce a weighted Q-value as an indicator for selecting the next hop. The weight κ of the Q-value depends on the neighbor relationship coefficient $M_{i,j}$ and the link quality $LQ_{i,j}$. The expression of κ is as follows:

$$\kappa = M_{i,j} * LQ_{i,j} \tag{17}$$

The link quality $LQ_{i,j}$ is calculated using the forward delivery ratios $df_{i,j}$ and reverse delivery ratios $dr_{i,j}$ of the link. $df_{i,j}$ represents the probability that a data packet successfully arrives at the recipient (i.e., node j); $dr_{i,j}$ represents the probability of sender (i.e., node i) successfully receiving ACK packets. In [5], $df_{i,j}$ and $dr_{i,j}$ are measured using hello message. In this paper, we also use hello message to measure $df_{i,j}$ and $dr_{i,j}$. The expression of $LQ_{i,j}$ is as follows:

$$LQ_{i,j} = df_{i,j} * dr_{i,j} \tag{18}$$

The neighbor relationship coefficient $M_{i,j}$ indicates the intimacy between node i and node j. Due to the high dynamics of nodes, the neighbor relationships may vary constantly. Therefore, it is not a good way to determine the current neighbor relationships by past neighbors. We need to re-estimate the neighbor relationships between nodes. Let node i add node j to its neighbor table at time t_1, and node i makes routing decision at time t_2. Here assuming that node i selects node j as the next hop, the data packet arrives at node j at time t_3, then we can use (13) and (14) to estimate the position of node j at time t_3. According to the estimated position of node j at time t_3 and the actual position of node i at time t_2, the intimacy of node i and node j within $(t_3 - t_2)$ time can be estimated. The coefficient of neighbor relationship $M_{i,j}$ as:

$$M_{i,j} = \begin{cases} 1 - \frac{d_{i,j}}{R}, & d_{i,j} \leqslant R \\ 0, & d_{i,j} > R \end{cases} \tag{19}$$

$$d_{i,j} = \sqrt{(\hat{x}_j(t_3) - x_i(t_2))^2 + (\hat{y}_j(t_3) - y_i(t_2))^2} \tag{20}$$

where $d_{i,j}$ is the distance between the position of node i at time t_2 and the estimated position of node j at time t_3. R is the propagation range of the node. If $d_{i,j}$ is greater than R, the data packet cannot reach node j from node i, indicating that node i has a weak neighbor relationship with node j. So, $M_{i,j}$ is equal to 0. On the contrary, if $d_{i,j}$ is smaller than R, $M_{i,j}$ becomes larger as the decrease of $d_{i,j}$, which indicates that node i has strong intimacy with neighbor node j.

When the node i makes a routing decision, a candidate neighbor associated with the largest κ-weighted Q-value is selected as the next hop. It can be expressed as following:

$$max \quad \kappa * Q(s_i, a_{i,j}) \quad s.t. \; v_{i,j} \geq V_j \tag{21}$$

where s_i is the state of the data packet (agent) at current time. One possible action $a_{i,j}$ is to select a neighbor node j as the next hop.

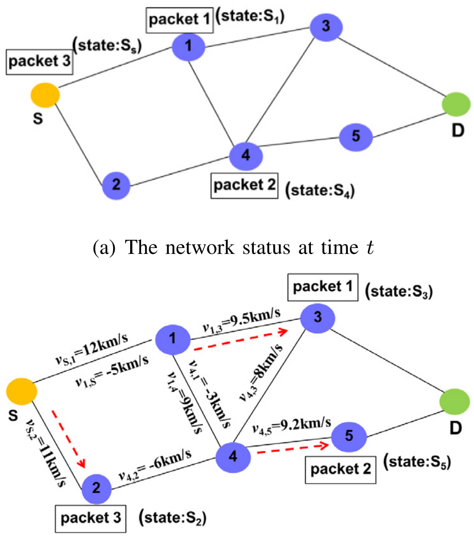
If the set of candidate neighbors is empty, while there are neighbors whose actual velocities are greater than 0, the neighbor associated with the maximum actual velocity will be selected as the next hop. Moreover, if all neighbors whose actual velocities are no greater than 0 (i.e., the routing hole problem is encountered), the penalty mechanism is triggered.

4.4. Penalty mechanism

The emergence of routing holes will increase the delay of the data packet. In order to reduce the routing hole problem, penalty mechanism is proposed in QMR. Penalty mechanism is triggered in the following two cases. One case is that when node j makes routing decision, if the routing hole problem is encountered, it gives previous hop node i a feedback. After receiving the feedback, the previous node i will give a minimum reward r_{min} to the link from node i to node j and use Q-learning to update the Q-value of the link again. Another case is that forwarding node i does not receive the ACK packet from the next hop node j, node i will give the minimum reward r_{min} of the link from node i to node j, and update the corresponding Q-value.

4.5. Illustration for QMR routing decision

Fig. 4 shows a simple network topology diagram. In the network, there are source node, destination node and 5 relay nodes. Suppose that at the current moment t, there are three data packets in the network, i.e., three agents. Packet 1, packet 2 and packet 3 (i.e., agent 1, agent 2, agent 3) stay on node 1, node 4 and node S respectively. Therefore, at time t, the state of packet 1, packet 2, and packet 3 are s_1, s_4, and s_s, respectively, as shown in Fig. 4(a). Suppose that at time t, the distances between node S, node 1, node 4 and destination node D are 500 m, 360 m and 245 m respectively. The deadline of packet 1 at node 1 is 45 ms; the deadline of packet 2 at node 4 is 35 ms; the deadline of packet 3 at node S is 50 ms. Using (12), according to the distance between the node and the destination and the deadline of the packet on the corresponding node, the requested velocities of the node S, node 1 and node 4 are 10 km/s, 8 km/s and 7 km/s, respectively. As can be seen from Fig. 4(a), at the current moment t, the sets of neighbors of node S, node 1 and node 4 are {node 1, node 2}, {node S, node 3, node 4}, {node 1, node 2, node 3, node 5} respectively. The node S, node 1 and node 4 need to select one of their neighbors as the next hop to forward the packet. It is assumed that at time t, the actual velocities for one-hop forwarding are shown in Fig. 4(b).



(a) The network status at time t

(b) The network status at time $t + 1$

Fig. 4. The network status at time t and $t + 1$.

Table 2
Link information.

Link	(s,1)	(s,2)	(1,3)	(1,4)	(4,3)	(4,5)
Weight κ	0.6	0.78	0.6	0.65	0.63	0.85
Q-value	0.62	0.5	0.8	0.72	0.8	0.68
Weight Q-value	0.372	0.39	0.48	0.468	0.504	0.578

According to the requested velocities of source node S, node 1 and node 4 and the actual velocities of their neighbors, we can obtain the set of candidate neighbors of each node. The sets of candidate neighbors of node S, node 1 and node 4 are {node 1, node 2}, {node 3, node 4}, {node 3, node 5} respectively, according to the velocity requirement that the actual velocity is not less than the requested velocity. Suppose that at time t, the Q-value, weight κ and weighted Q-value of the link corresponding to the candidate neighbors of node S, node 1 and node 4 are shown in Table 2.

In Table 2, (i, j) represents the link from node i to node j. At the beginning of the learning phase, the initial Q-value of each link is 0.5. As the weighted Q-value of the link (1,3) is greater than the weighted Q-value of the link (1,4), packet 1 (agent 1) will select the link (1,3) to forward. In fact, the Q-value of the link (1,3) is also greater than the Q-value of the link (1,4), which implies link (1,3) is better than link (1,4) in the past. Packet 1 (agent 1) selects the best link in the past, which means it mainly exploits the knowledge (experience) that has been learned in the past.

For packet 2 (agent 2), although the Q-value of the link (4,5) is smaller than the Q-value of the link (4,3), the weighted Q-value of the link (4,5) is greater than the weighted Q-value of the link (4,3). Therefore, packet 2 (agent 2) will select the link (4,5). Since the Q-value of the link (4,5) is not equal to the initial value 0.5, the link (4,5) has been explored before time t. However, packet 2 (agent 2) still selects the link (4,5) to forward, which indicates that links that have been explored in the past but in poor condition will have the opportunity to be re-explored.

For packet 3 (agent 3), although the Q-value of the link (S,2) is smaller than the Q-value of the link (S,1), the weighted Q-value of the link (S,2) is greater than the weighted Q-value of the link (S,1). Therefore, packet 3 (agent 3) will select the link (S,2). The link (S,2) has not been explored before time t, because the Q-value of the link is equal to the initial value of 0.5. However, packet 3 (agent 3) still selects link (S,2) to forward, which implies that packet 3 (agent 3) decides to explore a new link (i.e., explore new action) due to the link (S,2) with bigger weight (i.e., the link in better condition at present). This means that previously undiscovered links might be explored.

In a word, packet 1 selects the link that was regarded as the best in the past (exploitation), while packet 2 selects the link that was regarded as the poor in the past but may be better at present (exploration). Packet 3 selects the link not previously explored (exploration). Then at the next time $t+1$, packet 1, packet 2 and packet 3 respectively stay on node 3, node 5 and node 2. Therefore, at the time $t+1$, the states of packet 1, packet 2 and packet 3 (i.e., agent 1, agent 2 and agent 3) are s_3, s_5 and s_2 respectively, as shown in Fig. 4(b). Meanwhile, these actions (i.e., link(1,3), link(4,5) and link (S,2)) will receive corresponding rewards.

From this example, we can see that both explored links and unexplored links are likely to be explored when forwarding packets. Due to the frequent change of topology in FANETs, the quality of link is extremely unstable, so the exploration and exploitation mechanism mentioned in this paper can better adapt to the situation of link instability to find a better forwarding path.

5. Performance evaluation

In this section, our QMR algorithm is implemented and compared with the existing good performing QGeo [6], using an event-driven wireless networks simulator WSNet.[1]

In QGeo, the packet travel speed is considered as the reward to select optimal next hop. The discount factor selected from two constant values and a fixed learning rate are used to update Q-value in QGeo. The packet travel speed is only considered in reward function, However, the energy consumption is ignored, which makes QGeo difficult to provide low energy consumption service. In contrast, discount factor can be adjusted adaptively according to the mobility of neighbors in the adjacent time intervals, and learning rate can be adjusted adaptively according to one-hop delay in our QMR. In addition, QMR simultaneously considers delay and energy consumption to provide low delay and low energy consumption service.

For the considered scenario, 25 nodes are evenly distributed in an area of 500 m × 500 m, and the coordinates of destination node are (500, 500). We randomly select one node as the source node to transmit data to the destination node, and the rest of nodes except the destination node are relay nodes. The source emits a periodic flow of data packets whose data interval is set differently for comparison. Initially, Q-value of each link is 0.5. The parameters for the scenario are shown in Table 3.

The performance metrics including average end-to-end delay, maximum end-to-end delay, packet arrival ratio and energy consumption are considered:

- Average end-to-end delay: The average delay of the data packet from the source node to the destination node.
- Maximum end-to-end delay: The maximum delay of the data packet from the source node to the destination node.
- Packet arrival ratio: The ratio of the number of data packets received by the destination node (excluding redundant data packets) to the number of data packets transmitted by the source node.
- Energy consumption: We consider as a first approximation that the main energy consumption factor is due to the emission and reception of a data packet. Thus, the energy consumption is defined as the average number of emissions and reception operations performed by all nodes (sources and relays). This value is normalized by the number of data packets sent by the sources. The complete definition is given in [26]

Table 3
Parameter configuration.

Parameters	Settings
Area size	500 m × 500 m
Number of nodes	25
MAC	802.11 DCF
Radio propagation	Propagation_range, rang = 180 m
Interferences	Interferences_orthogonal
Modulation	Modulation_bpsk
Antenna	Antenna_omnidirectionnal
Battery	Energy_linear
HELLO interval	100 ms
ExpireTime	300 ms
Update interval of γ	100 ms
Size of data packet	127 Bytes
β	0.5
ω	0.6

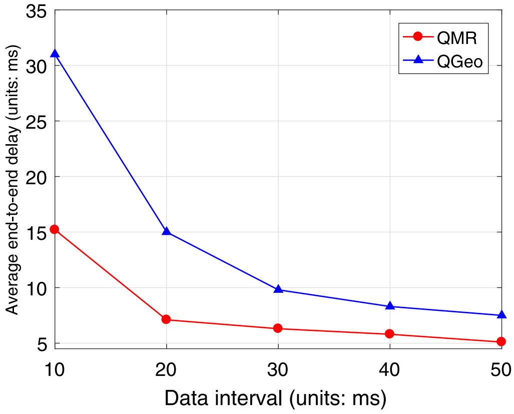


Fig. 5. Average end-to-end delay for different data interval.

In the following simulations, all mobile nodes (including source and relay nodes) utilize the Random Waypoint Mobility Model. In this mobility model, a mobile node moves from its current location to a new location by choosing a direction and speed [27]. The new location is randomly chosen in the range of simulation area, while the new speed is evenly chosen from [minspeed, maxspeed]. When the mobile node moves to the newly selected destination at the selected speed, the mobile node pauses for a specified period and then starts to move to another new location. In all simulations, the pause time of mobile nodes set to 0.

5.1. Evaluation metrics for different data interval

In this simulation, the minspeed and the maxspeed of mobile nodes are 0 m/s and 15 m/s respectively. Source node sends one thousand data packets at different interval. For each interval, we repeated the simulation 100 times. From Figs. 5 to 8, the performance of our QMR for different data interval of source is compared with QGeo, considering average end-to-end delay, maximum end-to-end delay, packet arrival ratio and energy consumption.

From Figs. 5 and 6, we can see that the average end-to-end delay and max end-to-end delay of our algorithm QMR are lower than QGeo. Compared to the QGeo, the average end-to-end delay and max end-to-end delay are averagely reduced by 42% and 47%, respectively. The

[1] http://wsnet.gforge.inria.fr/.

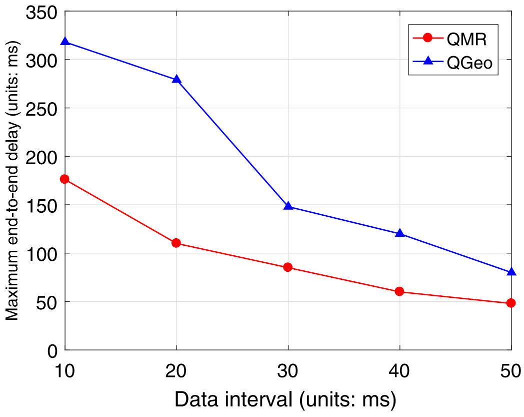


Fig. 6. Max end-to-end delay for different data interval.

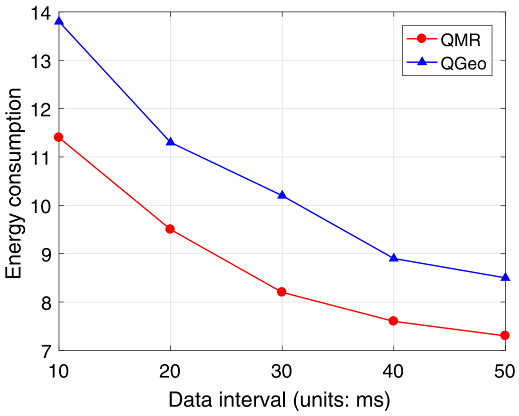


Fig. 7. Energy consumption for different data interval.

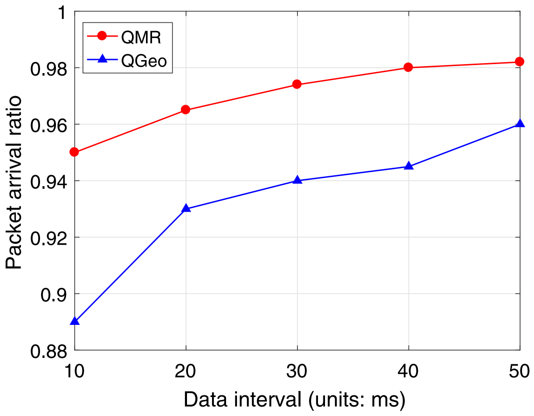


Fig. 8. Packet arrival ratio for different data interval.

main reason is that our algorithm constrains the velocity of data packets in the transmission process. QMR requires that the actual velocity of the data packets in the transmission process is not less than the requested velocity within the delay deadline. Thus, the routing path with low delay from source to destination is selected for data transmission. In addition, compared with QGeo, our algorithm not only considers the MAC delay, but also considers the average queuing delay in one-hop delay, which is more realistic end-to-end delay.

Fig. 7 shows that energy consumption of our QMR is lower. Compared with QGeo, energy consumption is reduced by 18%. One of the major reasons is that our algorithm considers the energy consumption of the node in the Q-learning reward function. By considering the initial energy and the residual energy of the nodes in a comprehensive manner, the nodes with smaller energy consumption are selected as the next hop, thereby balancing the energy consumption of all nodes in the network. Besides, QMR has good capability to find a low delay path, which results in fewer retransmissions and higher energy utilization efficiency.

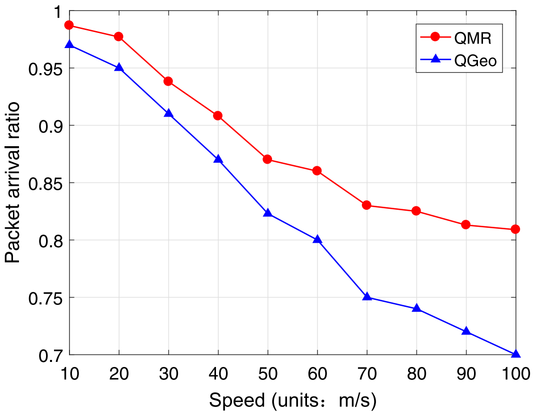


Fig. 9. Packet arrival ratio for different moving speed.

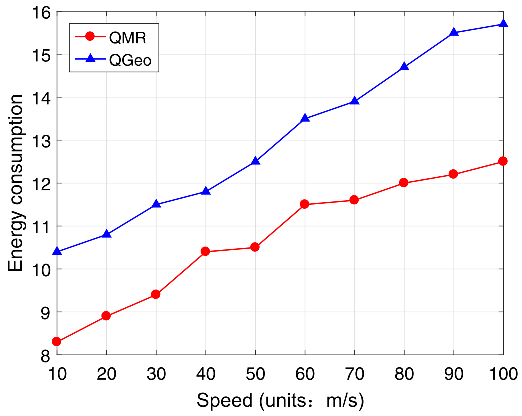


Fig. 10. Energy consumption for different moving speed.

Fig. 8 represents packet arrival ratio of our QMR. Compared with QGeo, the packet arrival rate of our QMR has increased by 6%. The main reason is that considering the high mobility of nodes, our algorithm re-estimates neighbor relationships between nodes when requiring the next hop. In addition, learning rate and discount factor in Q-learning are adaptively adjusted according to the one-hop delay and the mobility of neighbors to predict the status of the link. According to the prediction results, the most stable link is selected for data transmission, which in turn increases the packet arrival ratio.

To be concluded, our QMR performs better than the existing good performing QGeo for different data interval.

5.2. Evaluation metrics for different moving speed

In this simulation, source node sends one thousand data packets at the interval of 30 ms. The minspeed of mobile nodes is 0 m/s, and the maxspeed of mobile nodes varies from 10 m/s to 100 m/s. For each maximum speed of mobile nodes, we repeated the simulation 100 times.

From Figs. 9 to 12, the performance of our QMR for different moving speed is compared with QGeo. Fig. 9 shows that packet arrival ratio is higher than Qgeo, which is increased by 10% on average. We can see in Fig. 10 that energy consumption of the node is lower. Compared

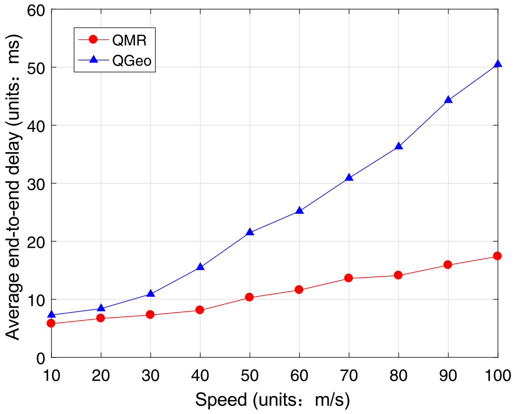


Fig. 11. Average end-to-end delay for different moving speed.

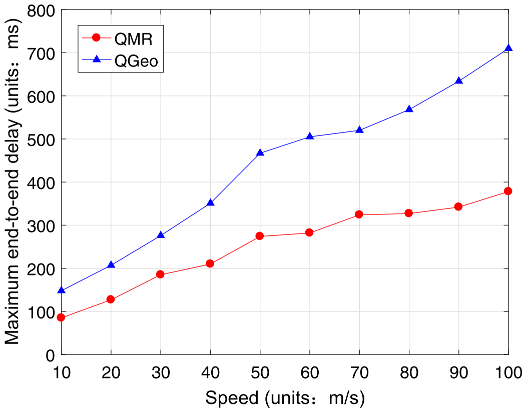


Fig. 12. Max end-to-end delay for different moving speed.

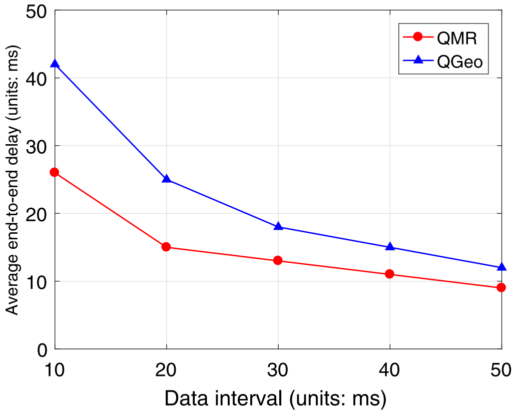


Fig. 13. Average end-to-end delay for a scenario with faulty relay nodes.

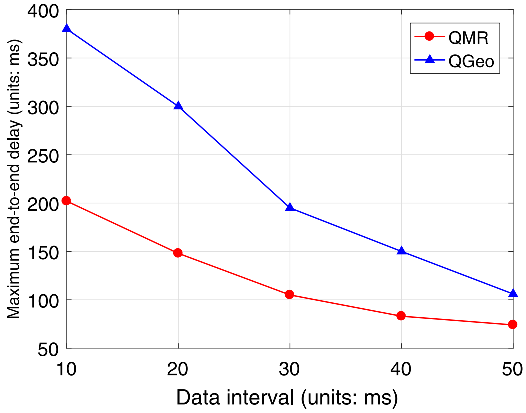


Fig. 14. Max end-to-end delay for a scenario with faulty relay nodes.

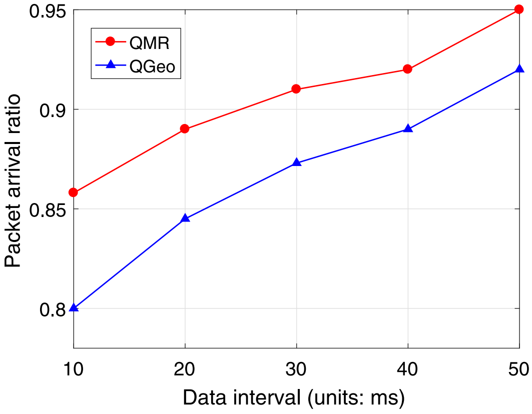


Fig. 15. Packet arrival ratio for a scenario with faulty relay nodes.

with QGeo, energy consumption is reduced by 19% on average. From Figs. 9 and 12, we can see that in our algorithm, the average end-to-end delay and max end-to-end delay are lower, which is averagely reduced by 48% and 44%, respectively. We can include that our QMR performs better than the existing good performing QGeo for different moving speed.

5.3. Evaluation metrics for a scenario with faulty relay nodes

In this simulation, we randomly select 10 nodes among 25 nodes to stop working by power off at 1 s after the start of the simulation, to achieve the scenario where numbers of relay nodes are faulty. The performance of the proposed method QMR and the existing QGeo under different data interval of source are compared. The minimum speed of mobile nodes is 0 m/s, and the maximum speed of mobile nodes is 15 m/s. We repeated the simulation 100 times.

From Figs. 13 to 16, we can see that our method still have better performance than the existing method QGeo, even if numbers of intermediate nodes are faulty. From Figs. 13 and 14, it can be seen that our method QMR still has lower average and max end-to-end delay. Compared to QGeo, the average end-to-end delay and max end-to-end delay are averagely reduced by 40% and 46%, respectively. Meanwhile, energy consumption is reduced by 17% on average, and packet arrival ratio is increased by 6% in comparison to QGeo.

5.4. Comparisons of exploration and exploitation mechanism

In this simulation, we use QMR to compare the exploration and exploitation mechanism proposed in this paper with traditional methods (ε greedy strategy, UCB, Boltzmann mechanism) by average end-to-end delay and packet arrival ratio. The minspeed and the maxspeed of mobile nodes are 0 m/s and 15 m/s respectively. The data interval of the source to transmit one packet is 10 ms and 40 ms. The experimental results are shown in Fig. 17.

From Fig. 17, we can see that our mechanism can enhance the routing performance, where the average end-to-end delay is smallest and the packet arrival ratio is the highest. This is because the regulation method is adaptive to the variation of network condition. Therefore, it can better balance the relationship between exploration and exploitation, which is more suitable for FANETs and other dynamic environments.

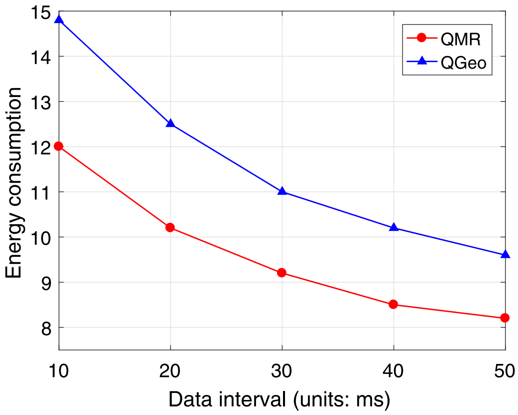
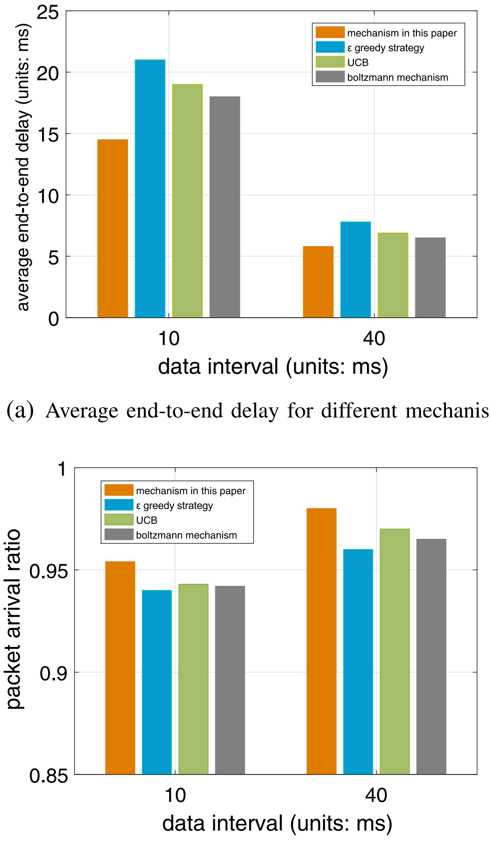


Fig. 16. Energy consumption for a scenario with faulty relay nodes.



(a) Average end-to-end delay for different mechanism

(b) Packet arrival ratio for different mechanism

Fig. 17. Comparison of exploration and exploitation mechanism.

6. Conclusion

The drastically changing topology of FANETs has brought great challenges to routing protocols for FANETs. Existing routing protocols for MANETs and VANETs cannot be directly applied in FANETs. In this paper, we propose a novel Q-learning based routing protocol for FANETs. Considering the requirement of low delay and low energy consumption of FANETs, a Q-learning based Multi-objective optimization Routing protocol (QMR) is proposed aiming directly at simultaneously providing low-delay and low-energy service guarantees. In addition, a method of adaptively adjusting Q-learning parameters is proposed to adapt to the high dynamics of FANETs. In this method, learning rate is adaptively adjusted according to one-hop delay, and discount factor is adjusted according to the mobility of neighbors in the adjacent time intervals. The results have demonstrated outstanding performance of our QMR in comparison with the QGeo. The transmission of multi-flows with different requirements of QoS will be investigated in the future. Moreover, the implementation of QMR in a physical UAVs would be a challenge and could provide many useful insights.

Declaration of competing interest

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

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61602447.

References

[1] Luis C. Batista Da Silva, Ricardo Maroquio Bernardo, Hugo A. De Oliveira, Paulo F.F. Rosa, Multi-uav agent-based coordination for persistent surveillance with dynamic priorities, in: International Conference on Military Technologies, 2017, pp. 765–771.

[2] Milan Erdelj, Micha Krl, Enrico Natalizio, Wireless sensor networks and multi-uav systems for natural disaster management, Comput. Netw. 124 (2017) 72–86.

[3] Brad Karp, H.T. Kung, Gpsr:greedy perimeter stateless routing for wireless networks, in: International Conference on Mobile Computing and NETWORKING, 2000, pp. 243–254.

[4] Ruiling Li, Fan Li, Xin Li, Yu Wang, Qgrid: Q-learning based routing protocol for vehicular ad hoc networks, in: PERFORMANCE Computing and Communications Conference, 2014, pp. 1–8.

[5] Abdellatif Serhani, Najib Naja, Abdellah Jamali, Qlar: A q-learning based adaptive routing for manets, in: Computer Systems and Applications, 2017, pp. 1–7.

[6] Woo Sung Jung, Jinhyuk Yim, Young Bae Ko, Qgeo: Q-learning based geographic ad-hoc routing protocol for unmanned robotic networks, IEEE Commun. Lett. PP (99) (2017) 1.

[7] Hua Yang, Zhiyong Liu, An optimization routing protocol for fanets, EURASIP J. Wireless Commun. Networking 2019 (2019) 120.

[8] Chen-Mou Cheng, Pai-Hsiang Hsiao, H.T. Kung, Dario Vlah, Maximizing throughput of uav-relaying networks with the load-carry-and-deliver paradigm, in: 2007 IEEE Wireless Communications and Networking Conference, IEEE, 2007, pp. 4417–4424.

[9] Ozgur Koray Sahingoz, Networking models in flying ad-hoc networks (fanets): Concepts and challenges, J. Intell. Robot. Syst. 74 (1–2) (2014) 513–527.

[10] Thomas Clausen, Philippe Jacquet, Optimized link state routing protocol (olsr), Rfc 527 (2) (2003) 1–4.

[11] C. Pu, Link-quality and traffic-load aware routing for uav ad hoc networks, in: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), 2018, pp. 71–79.

[12] Xianfeng Li, Jiaojiao Yan, Lepr: Link stability estimation-based preemptive routing protocol for flying ad hoc networks, in: 2017 IEEE Symposium on Computers and Communications (ISCC), 2017, pp. 1079–1084.

[13] S.D. Ghode, K.K. Bhoyar, Nema: Node energy monitoring algorithm for zone head selection in mobile ad-hoc network using residual battery power of node, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 1999–2004.

[14] Guido Oddi, Antonio Pietrabissa, Francesco Liberati, Energy balancing in multi-hop wireless sensor networks: An approach based on reinforcement learning, in: Adaptive Hardware and Systems, 2014, pp. 262–269.

[15] Samir R. Das, Elizabeth M. Belding-Royer, Charles E. Perkins, Ad hoc on-demand distance vector (aodv) routing, 2003, IETF RFC 3561.

[16] Zygmunt Haas, Marc Pearlman, Prince Samar, The zone routing protocol (zrp) for ad hoc networks, in: IETF Internet Draft draft-ietf-manet-zone-zrp-04, 2002.

[17] Justin A. Boyan, Michael L. Littman, Packet routing in dynamically changing networks: a reinforcement learning approach, in: International Conference on Neural Information Processing Systems, 1993, pp. 671–678.

[18] H.V. Abeywickrama, B.A. Jayawickrama, Y. He, E. Dutkiewicz, Empirical power consumption model for uavs, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018, pp. 1–5.

[19] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, Mach. Learn. 8 (3–4) (1992) 225–227.

[20] Richard S. Sutton, Learning to predict by the methods of temporal differences, Mach. Learn. 3 (1) (1988) 9–44.

[21] Y. Yu, D. Estrin, R. Govindan, Geographical and energy-aware routing: A recursive data dissemination protocol for wireless sensor networks, UCLA Computer Science Department Technical Report, 2001, pages UCLACSD TR–01–0023.

[22] Tian He, John A. Stankovic, Chenyang Lu, Tarek Abdelzaher, Speed: A stateless protocol for real-time communication in sensor networks, in: International Conference on Distributed Computing Systems, 2003. Proceedings, 2003, pp. 46–55.

[23] Yanjun Li, Chung Shue Chen, Yeqiong Song, Wang Zhi, Youxian Sun, Enhancing real-time delivery in wireless sensor networks with two-hop information, IEEE Trans. Ind. Inf. 5 (2) (2009) 113–122.

[24] Daan Bloembergen, Karl Tuyls, Daniel Hennes, Michael Kaisers, Evolutionary dynamics of multi-agent learning: A survey, J. Artificial Intelligence Res. 53 (1) (2015) 659–697.

[25] Jean Yves Audibert, Sbastien Bubeck, Regret Bounds and Minimax Policies under Partial Monitoring, volume 11, 2010.

[26] Wang Qi, Katia Jaffres-Runser, Yongjun Xu, Jean Luc Scharbarg, Zhulin An, Christian Fraboul, Tdma versus csma/ca for wireless multi-hop communications: a stochastic worst-case delay analysis, IEEE Trans. Ind. Inf. PP (99) (2017) 1.

[27] Tracy Camp, Jeff Boleng, Vanessa Davies, A survey of mobility models for ad hoc network research, Wirel. Commun. Mobile Comput. 2 (5) (2010) 483–502.

Jianmin Liu received her B.Eng. degree in computer science and technology from Shanxi University (China), in 2018, and has been a Ph.D. candidate of Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, since 2018. Her current research focuses on collaboration in multi-agent system, and real-time protocols of mobile ad hoc networks.

Qi Wang is an associate professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. She received the Ph.D. degree in computer science from University of Chinese Academy of Sciences, Beijing, China in 2015. In 2010, she received a 1-year fellowship from INRIA under the joint program with the Chinese Academy of Sciences to pursue her research within the SWING team of INRIA, at CITI laboratory, INSA Lyon, France. She visited IRIT laboratory at University of Toulouse because she was a recipient of the 2012 EIFFEL doctoral fellowship from the French Ministry of Foreign Affairs. Her research focuses on performance evaluation of wireless networks for delay sensitive applications.

ChenTao He received his B.Eng. degree in computer science and technology from University of Science and Technology Beijing (China), in 2018, and has been a Master candidate of Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, since 2019. His current research focuses on collaboration in multi-agent system, and routing protocols of mobile ad hoc networks.

Katia Jaffrès-Runser (M'05) received the Dipl.Ing. and M.Sc. degree in 2002, and the Ph.D. degree in computer science from INSA Lyon, Villeurbanne, France, in 2005. From 2002 to 2005, she was with INRIA. In 2006, she joined the Stevens Institute of Technology as a Postdoctoral Researcher. She has been an Associate Professor with the University of Toulouse, Toulouse, France, since 2011. She received a three-year Marie-Curie OIF fellowship from the European Union (2007–2010). Her research interest includes the performance evaluation of wireless networks in general, with a special focus on real-time guarantees provision.

Yida Xu received his B.Eng. degree in computer science and technology from University of Science and Technology of China, Hefei, China, in 2008, received his M.Eng. degree in computer science and technology from China University of Petroleum, Beijing, China, 2017, and has been a Ph.D. candidate of Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, since 2017. His current research focuses on real-time protocols for wireless sensor networks.

Zhenyu Li received his Ph.D. degree from the in 2009. He is a full professor at ICT/CAS and an adjunct professor at the University of CAS. His research interests include Internet architecture and Internet measurement.

Yongjun Xu is a professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. He received his B.Eng. and Ph.D. degree in computer communication from Xi'an Institute of Posts & Telecoms (China) in 2001 and Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2006, respectively. His current research interests include wireless sensor network, cyber–physical Systems and multi- sensor Data Fusion.