A Distributed and Adaptive Routing Protocol for UAV-aided Emergency Networks

Jie Tang*, Zihao Zhou*, Wanmei Feng†, Kai Kit Wong‡,
*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China
†College of Electronic Engineering(college of Artificial Intelligence), South China Agricultural University, Guangzhou, China
‡Department of Electronic and Electrical Engineering, University College London, WC1E 6BT London, UK
eejtang@scut.edu.cn, eezihaozhou@gmail.com, wmfeng@scau.edu.cn, kaikit.wong@ucl.ac.uk

***Abstract*—Due to its strong flexibility, easy deployment, high maneuverability and extensive connectivity, unmanned aerial vehicle (UAV) swarm has been widely used in the construction of emergency communication network in recent years. Among them, packet routing in a resilient and adaptive manner is one of the fundamental problems for cooperation between multiple UAVs to complete search and rescue tasks. Recently, reinforcement learning (RL) technique has provided a new opportunity for network-related applications, including routing. However, most existing RL-based routing protocols suffer from issues such as local optimum, blind exploration and slow convergence speed. Additionally, the routing protocols based on deep reinforcement learning (DRL) has high computational complexity, making them unsuitable for energy-limited emergency relief scenarios. In this paper, we proposed a Q-learning aided resilient routing protocol with hindsight pre-calculation ($\mathrm{QR^2HPC}$) in UAV swarm for the construction of the emergency networks. Firstly, a dynamic exploration and exploitation coefficient is proposed based on the number and speed of neighbors. Secondly, a warm-start mechanism is proposed in the exploration phase that modifies the traditional random next hop selection to a routing approach guided by various indicators. Finally, we introduce a hindsight pre-calculation (HPC) mechanism to improve the robustness of Q-table to traffic flow changes. The experimental results manifest that our protocols can make effective routing decisions in dynamic wireless multi-hop networks, thereby enhancing the system performances in terms of packet delivery ratio, end-to-end delay, throughput and network lifetime.**

***Index Terms*—UAV swarm, emergency wireless communication networks, routing protocol, Q-Learning**

I. Introduction

The establishment of the emergency network is crucial in the event of the natural or man-made disasters as it enables time communication connectivity [1]- [4]. However, traditional solutions that rely on ground emergency vehicles lack flexibility and are limited by environmental and spatial constraints [5]. Recently, due to the strong flexibility, easy deployment, high maneuverability and extensive connectivity, UAV swarm is emerging as a promising emergency situation option for deploying an intelligent mobile and flexible network, which is called flying ad-hoc network (FANET) [6]. Within a UAV swarm, packet routing plays a vital role in facilitating cooperation among multiple UAVs to accomplish complex missions. However, in an emergency situation, the uncertainty of the environment, the rapid changes in the network topology, and the frequent communication needs of the rescue team and the disaster area have brought great challenges to the design of routing protocols.

Over the past few decades, researchers have proposed many classical routing protocols such as OLSR [7], AODV [8] and GPSR [9]. Proactive routing protocols like OLSR require nodes to periodically store and maintain routing tables, resulting in high routing overhead. On the other hand, reactive routing protocol like AODV establish routing paths only when the packets need to be sent, which will lead to high end-to-end (E2E) delay. Position-based routing protocol, which is represented by GPSR, selects the next hop solely based on the location information. Nevertheless, the frequent occurrence of routing holes due to the high mobility of UAV swarms significantly increases latency. Moreover, due to the lack of intelligent awareness about the environments and the limited adaptability and flexibility, the aforementioned traditional routing protocols face challenges when they are applied in the construction of the emergency communication networks.

In recent years, reinforcement learning (RL) has demonstrated its strength in decision-making and is widely adopted in routing problems in ad-hoc networks. In [10], a Q-learning based geographic routing protocol is proposed for UAV swarm, where link stability, link capacity and interference information were considered to select the next hop. Liu et al. [11] proposes a multi-objective optimization routing protocol using Q-learning to optimize the E2E delay and energy consumption of the network. In [12], an enhanced Q-Learning routing algorithm based on OLSR is proposed, where Kalman filter is used to predict the trajectory of the node in advance for calculating the Q-value. However, Kalman filter is computationally intensive and is not suitable for UAVs with limited computing power and resources. In [13], by exploiting the information of the two-hop neighbors, a Q-learning based topology-aware routing protocol is studied for FANET. The selection of the next hop is based on delay constraints, speed constraints, and energy constraints. Serhani et al [14] proposed a Q-learning based adaptive routing (QLAR) for MANETs, where a new model was developed to detect the mobility level of each node. However, in the actual disaster relief scenario, there are often numerous two-way communication needs between the disaster-affected area and the rescue troops, which requires the routing protocol to quickly adapt to the changes of traffic flow direction. In addition, the fixed ratio of agent exploration and exploitation in the above routing protocols and the random selection of the next hop during exploration are not well-suited for the rapidly changing network topology of FANETs. Poor selection of the next hop may result in unnecessary flight delays, increased energy consumption, or ineffective path selection, thereby reducing routing efficiency.

In this paper, we propose a Q-learning aided resilient routing protocol with hindsight pre-calculation ($\mathrm{QR^2HPC}$) in UAV swarm for the construction of the emergency networks. Firstly, a dynamic exploration and exploitation coefficient is proposed based on the number and speed of neighbors. Secondly, a warm-start mechanism is proposed in the exploration phase that modifies the traditional random next hop selection to a routing approach guided by various indicators. Finally, we introduce a hindsight pre-calculation (HPC) mechanism to improve the robustness of Q-table against traffic flow changes.

The remainder of this paper is organized as follows. Section II describes the system model. The routing algorithm proposed in this paper is described in Section III. Then, Section IV shows the simulation results and discussions to demonstrate the significant performance of the proposed scheme. Finally, Section V concludes this paper.

II. SYSTEM MODEL

In this paper, we consider a UAV swarm with a set of UAV nodes $\mathcal{M}$, which can be denoted as $\mathcal{M} = [U_1, U_2, \cdots, U_M]$. Each UAV is equipped with an omni-directional antenna whose maximum communication range is D_{max}. If the Euclidean distance between UAV U_i and U_j ($U_i, U_j \in \mathcal{M}$) is $d_{U_iUj} < D_{max}$, it means that a potential transmission link can be established between two UAVs. The mutual perception between two UAVs requires regular exchange of hello packets. Hence the network is modeled as a directed graph $\mathcal{G} = (\mathcal{M}, \xi)$, ξ is defined as a finite set of the transmission links between UAVs. $e(U_i, U_j) \in \xi$ indicates the establishment of UAV U_i's perception of U_j. The Gauss-Markov Mobility Model [15] is adopted to formulate the mobility of UAV nodes. Each UAV $U_m \in \mathcal{M}$ can obtain its location, speed and direction by equipping with Global Navigation Satellite Systems (GNSS). In addition, we assume that the network operates in a time-slotted fashion with normalized time slot. Therefore, operations such as packet sending and receiving occur at specific time slots.

Four ground stations are regarded as the destination nodes to receive data packets from the UAV nodes. We assume a sequential data provision scheme in the UAV swarm. Specifically, the UAV swarm serves one destination node for a certain period of time and then moves on to serve another destination node in the subsequent period. Therefore, in each time slot, one UAV node is considered as the source node to send data packet while the remaining UAV nodes act as relay nodes to forward the packets.

III. PROPOSED ALGORITHM

A. Routing Decision

At each time slot t, nodes with data packet forwarding tasks need to determine the next-hop according to some routing strategies. One of the commonly used strategies in RL-based routing protocols is ϵ-greedy [16]. However, this strategy also presents the following issues.

On one hand, the balance between exploration and exploitation plays a crucial role to achieve a more efficient transmission of the date packets in ϵ-greedy strategy. The exploration coefficient ϵ is used to control the exploration and exploitation of the agent. However, in most existing Q-Learning based routing protocols in FANET, the exploration coefficient remains fixed. When the topology of the FANET undergoes frequent changes, indicating a highly dynamic environment, the use of a traditional fixed exploration coefficient may hinder the ability of the algorithm to adapt to the new environment changes in a timely manner. On the one hand, during exploration, choosing an inappropriate next-hop can lead to unnecessary transmission delays, increased energy consumption, or ineffective path selection, ultimately diminishing the overall routing efficiency. Besides, random exploration choices may introduce noise and uncertainty, making the agent more susceptible to inefficient paths. In this paper, we propose an adaptive exploration and exploitation strategy. Furthermore, during exploration, we introduce a warm-start mechanism that modifies the traditional random next hop selection to a routing approach guided by various indicators.

Firstly, the exploration coefficient of any node U_i is defined as follows:

$$\epsilon_{U_i} = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \times e^{-\lambda \times (\bar{v}_{U_i} + \bar{n}_{U_i})}, \tag{1}$$

where ϵ_{min} and ϵ_{max} represent the minimum and maximum exploration coefficient, respectively. λ is a control parameter. $\bar{v}_{U_i}$ is the normalized average speed of the neighbors of node U_i. The method of window mean with exponentially weighted moving average (WMEWMA) is adopted to update the neighbor average speed $\bar{v}_{U_i}$. Node U_i maintains a sliding window with length l which records the neighbor velocity of the last l hello packets sent by the neighbors of U_i. The k-th updated neighbor average speed is given by:

$$\bar{v}_{U_i}(k) = (1 - \beta) \times \frac{\sum_{q=k-l}^{k-1} \frac{v_{U_m, U_m \in N_{U_i}}(q)}{V_{max}}}{l} + \beta \times v_{U_j}^{new}. \tag{2}$$

where $\beta (0 < \beta < 1)$ in eq.(2) is the tunable weighting coefficient. $v_{U_m, U_m \in N_{U_i}}$ indicates the node velocity recorded in the neighbor table and $v_{U_j}^{new}$ is the speed of node U_j recorded in the packet received from U_j.

$\bar{n}_{U_i}$ in eq.(1) is the normalized average number of neighbors of node U_i, which can be calculated as:

$$\bar{n}_{U_i} = \frac{n - N_{min}}{N_{max} - N_{min}}, \tag{3}$$

where N_{max} is the maximum number of nodes in the network. N_{min} is the minimum number of neighbors of a node in the network. n is the number of valid entries in the neighbor table of node U_i at the current moment. The node U_i updates $\bar{n}_{U_i}$ once that it receives a hello packet.

In the above formula, when the velocity of the node is low or there are numerous neighboring nodes, the agent increases the exploration probability. On the contrary, when the UAV has a higher speed or fewer neighbor nodes, the exploration rate decreases, and the node is more likely to choose the known optimal action for utilization. The underlying reasons are as follows:

Firstly, when the node speed is low, the network topology is relatively stable, allowing the drone to frequently try different routing options in order to discover potential better solutions. When the node speed is high, selecting the known optimal action enables faster achievement of the target position or completion of the task, thereby reducing time overhead. This is particularly crucial in emergency applications.

Secondly, when the number of neighboring nodes is high, it indicates a greater number of alternative paths and neighboring nodes for the drone. By increasing the exploration rate, drones can actively explore a wider range of routing options, aiming to uncover potentially superior solutions. Additionally, a large number of neighboring nodes signifies a more intricate network topology, which may introduce more changes and uncertainties. By increasing the exploration rate, drones can enhance their adaptability to fluctuations in neighboring nodes and strengthen their resilience when faced with changes in the network topology. Conversely, in scenarios where the number of neighboring nodes is low, excessive exploration can lead to increased communication overhead and computational costs, while the potential benefits from exploration may be relatively limited.

We have redirected our focus towards on routing decisions. When a node is required to choose the next hop for forwarding, it employs a probability of 1-ϵ to select the maximum weighted Q for forwarding, as described in [11]. During the exploration phase, We have modified the original method of randomly selecting neighboring nodes as the next hop instead of selecting the next hop based on several specific methods. The specific methods are described as follows:

- *Greedy*: Under the greedy strategy, a node forwards the data packet to the neighbor node that minimize the distance to the destination node, which can be expressed as:

$$Nexthop(U_i) = \underset{U_j \in N_{U_i}}{argmax}\, d_{U_j d}, \tag{4}$$

  where $Nexthop(U_i)$ indicates the next hop from node U_i to the destination d. If the void area is encountered, the node will hold this data packet.
- *Compass*: Compass is a routing policy that utilizes direction information to guide nodes towards their destination. Its objective is to ensure that each step brings the node closer to the desired direction by selecting the neighboring node with the smallest angle deviation as the next hop. The strategy can be expressed as:

$$Nexthop(U_i) = \underset{U_j \in N_{U_i}}{argmax}\, \angle U_j U_i d. \tag{5}$$

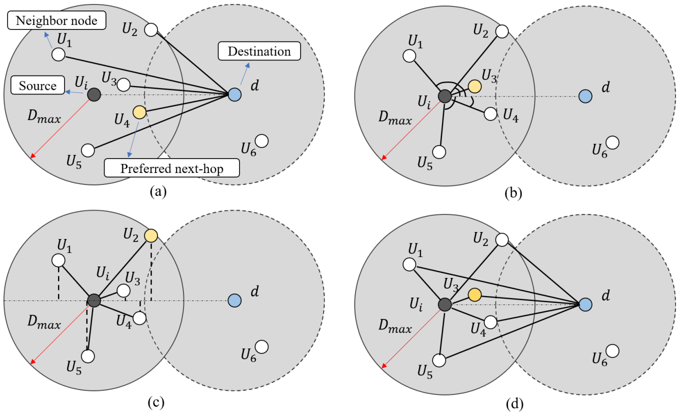

Fig. 1. Average E2E delay under different UAV speeds.

- *Most forward*: In this case, node U_i will forward the data packet to the neighbor U_j whose projection on the line $(U_i d)$ is closer to d. The strategy can be expressed as:

$$Nexthop(U_i) = \underset{U_j \in N_{U_i}}{argmax}\, \frac{\overrightarrow{U_j U_i} \cdot \overrightarrow{dU_i}}{\sqrt{(x_d - x_{U_i})^2 + (y_d - y_{U_i})^2}}. \tag{6}$$

- *Ellipsoid*: In Ellipsoid mode, node U_i will forward the packet to the neighbor U_j that minimizes the sum of the distance from U_i to U_j and the distance from U_j to destination d. The strategy can be expressed as:

$$Nexthop(U_i) = \underset{U_j \in N_{U_i}}{argmax}(d_{U_i U_j} + d_{U_j d}). \tag{7}$$

Fig.1 illustrates the selection of the next hop for the four modes. When the agent enters the exploration mode, it randomly selects one of the above four strategies to select the next hop. It is worth noting that in more complex network environments, the strategies available during the exploration phase can also take into account factors such as energy, buffer, channel quality, etc.

B. Reward Function

As the sole source of feedback for the agent, rewards play a crucial role in guiding the Q-Learning algorithm. The objective of the Q-Learning based routing algorithm is to enable nodes to maximize cumulative rewards while transmitting data packets. In this paper, we aim to achieve efficient transmission by balancing energy consumption, controlling congestion, and considering the distance factor, angle factor, energy factor, and buffer remaining capacity of the node in the reward function. Firstly, we define the joint metric as follows:

$$r_{U_i \to U_j}(d) = \omega_1 \times f_1 + \omega \times f_2 + \omega \times f_3 + \omega \times f_4, \tag{8}$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ are the weights which hold $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$, f_1 is the distance factor, which is expressed as:

$$f_1 = \pm \frac{\sqrt{(x_{U_j} - x_d)^2 + (y_{U_j} - y_d)^2}}{\sqrt{(x_{U_i} - x_d)^2 + (y_{U_i} - y_d)^2}}, \tag{9}$$

f_1 is positive when the next hop U_j is closer to the destination than node U_i and vice versa.

f_2 is the angle factor. In order to reduce the number of hops of the route, the data packets should be transmitted along a straight line to the destination node. Therefore, f_2 can be obtained as follows:

$$f_2 = \frac{(x_{U_j} - x_{U_i}) \times (x_d - x_{U_i}) + (y_{U_j} - y_{U_i}) \times (y_d - y_{U_i})}{d_1 + d_2}, \tag{10}$$

where $d_1 = \sqrt{(x_{U_j} - x_{U_i})^2 + (y_{U_j} - y_{U_i})^2}$ and $d_2 = \sqrt{(x_d - x_{U_i})^2 + (y_d - y_{U_i})^2}$, which represent the distance between U_j and U_i and the distance between destination and U_i. Larger f_2 indicates closer to straight line transmission.

f_3 is the energy factor, which is defined as the ratio of the residual energy of the node to the initial energy. The energy factor of node U_j can be expressed as follows:

$$f_3 = \frac{E_{U_j}^{res}}{E_{U_j}^{init}}, \tag{11}$$

where $E_{U_j}^{res}$ is the residual energy of the node U_j, and $E_{U_j}^{init}$ is the initial energy of the node U_j. The larger the f_3 is, the lower energy consumption of the node U_j is.

f_4 represents the buffer remaining capacity. Similar to the definition of f_3, f_4 is defined as the ratio of the residual buffer capacity to the buffer initial capacity. The buffer remaining capacity of node U_j can be expressed as follows:

$$f_4 = \frac{B_{U_j}^{res}}{B_{U_j}^{init}}, \tag{12}$$

where $B_{U_j}^{res}$ is the buffer residual capacity of the node U_j, and $B_{U_j}^{init}$ is the initial capacity of the buffer of node U_j. A larger f_4 indicates a lower likelihood that the link is experiencing congestion.

Therefore, we can define the joint reward function as follows:

$$r_t(U_i, U_j, d) = \begin{cases} r_{max} & when\ U_j\ is\ destination \\ r_{min} & when\ U_j\ is\ local\ minimum \\ r_{U_i \to U_j}(d) & otherwise. \end{cases} \tag{13}$$

If the next hop is the destination, the agent can obtain the maximum reward r_{max}. When the next hop selected by the agent has no neighbors closer to the destination, which is called local minimum, the agent will receive the minimum reward r_{min}. The last item can be adopted to jointly optimize transmission efficiency, energy consumption, and congestion issues.

C. HPC-based Q-table Update Mechanism

Let us examine the iterative performance of the conventional Q-learning based routing algorithm:

$$\begin{aligned} Q_d(U_i, U_j) \leftarrow (1-\alpha) Q_d(U_i, U_j) + \alpha(r_t(U_i, U_j, d) \\ + \gamma_{U_i U_j}(1 - f_t^p) Q_d(U_j, U_m)), \end{aligned} \tag{14}$$

where f_t^p is the task completion indicator, whose definition is shown in eq.(19). U_m can be expressed as follows:

$$U_m = \underset{n \in N_{U_j}}{argmax}\, Q_d(U_j, n). \tag{15}$$

However, when the destination node changes, the Q-value for the new destinations needs to be trained from scratch. When the network consists of a large number of nodes, it results in significant training time overhead and diminishes the efficiency of data packet transmission. In this paper, in order to alleviate this problem, we propose a hindsight pre-calculation (HPC) mechanism to improve the robustness of Q-table to task changes. The idea behind HPC is that the agent updates the Q-table during each transition not only with the original goal for that transmission but also with other goals. The pseudo-code of the HPC algorithm is shown in Algorithm 1.

Algorithm 1 Hindsight pre-calculation (HPC) mechanism

```
Input : Discounted factor $\gamma$, learning rate $\alpha$, routing policy $\pi$, source node $U_i$, Q-table of node $U_i$ $Q_d(s, a)$, destination set $\mathcal{D}$
Output : Updated Q-table $Q'_d(s, a)$

while $U_i$ has a packet to transmit to $d, d \in \mathcal{D}$ do
    Choose a next hop $U_j$ in $N_{U_i}$ according to the given routing policy $\pi$
    Transmit the data packet to node $U_j$ and observe the reward $r_{U_i \to U_j}$ and $f_t^p$
    Update Q-value: $Q'_d(U_i, U_j) = (1-\alpha) Q_d(U_i, U_j) + \alpha(r_t(U_i, U_j, d) + \gamma_{U_i U_j}(1 - f_t^p) \underset{x, x \in N_{U_j}}{max} Q_d(U_j, x))$
    for $d' \in \mathcal{D}$ do
        $r' := r_t(U_i, U_j, d')$
        Update Q-value: $Q'_{d'}(U_i, U_j) = (1-\alpha) Q_{d'}(U_i, U_j) + \alpha(r' + \gamma_{U_i U_j}(1 - f_t^p) \underset{x, x \in N_{U_j}}{max} Q_{d'}(U_j, x))$
    end for
end while
```

In summary, in our algorithm, each UAV node is an agent that updates the status information of its neighbors (such as location, speed, remaining energy, etc.) by periodically exchanging hello packets. During the routing decision stage, the adaptive exploration and utilization coefficient determines whether to choose the neighbor with the largest Q-value as the next hop or adopt the hot start mechanism. After receiving the ACK packet sent by the next hop node, the node updates the Q-table through HPC mechanism.

IV. SIMULATION AND PERFORMANCE EVALUATION

A. Simulation Parameters

In this section, our proposed $\mathrm{QR^2HPC}$ is compared with GPSR [9] and QMR [11] in a FANET simulation platform based on Python. The velocity of UAV nodes is 10m/s to 30m/s to reflect the mobility impact on routing performance. For the considered scenario, nodes are randomly distributed in an area of 1500m × 1500m. The coordinates of the four destinations are located at (400, 750), (750, 400), (750, 1100) and (1100, 750). The total simulation time is set to 250 seconds and divided into 5000 time slots, each of which is 0.05 seconds in length. At the beginning of each time slot, a node is randomly selected as the source node to transmit data packets to the destination node. The detailed parameters used in our simulation are summarizes in Table I.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation area	1500m × 1500m
Total simulation time	250s
Time slot interval	0.05s
Node number	40
Node velocity	10-30 m/s
UAV transmission radius	250m
UAV transmit power	1.0 W
UAV received power	1.0 W
Antenna	Omni-directional
Mobility model	Gauss Markov mobility model
Initial energy of UAVs	900J
Energy threshold	20J
Hello interval	0.5s
Max TTL	15
Initial value of Q-table	0.5
Initial learning rate	0.3
ϵ	0.9
λ	0.5
β	0.5

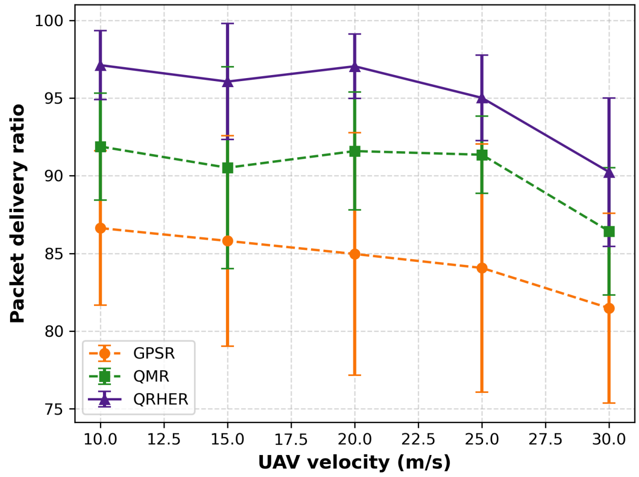

Fig. 2. PDR VS. Node velocity

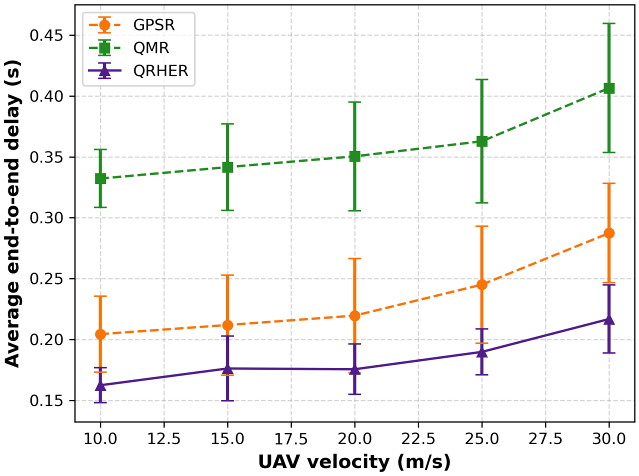

Fig. 3. Average E2E delay VS. Node velocity

B. Simulation Results and Evaluation

Fig.2 illustrates the PDR of different routing protocols under different moving speed of UAV nodes. In the case of GPSR, the selection of the next hop is solely based on geographic location information. This approach results in a substantial number of nodes entering the routing hole area as the movement speed accelerates. Moreover, the frequent utilization of the perimeter forwarding mode leads to an increase in packet forwarding hops, potentially exceeding the maximum Time-to-Live (TTL) value. Consequently, this leads to a significant decline in the PDR. For QMR, the incorporation of Q-Learning and dynamic hyper-parameters have contributed to an improvement in the PDR performance. However, when the destination node changes, QMR requires the retraining of the Q-table, which leads to a slow learning speed and makes it difficult to adapt to high-speed node motion. For the algorithm proposed in this paper, the incorporation of the hindsight experience replay (HER) concept during Q-table updates allows for improved learning efficiency in Q-Learning when the target node changes. Consequently, this enhancement in Q-Learning efficiency leads to an overall improvement in the PDR.

Fig.3 shows the average E2E delay experienced by UAV nodes based on their velocities. For GPSR, as the node velocity increases, the occurrence of void regions becomes more frequent. The adoption of perimeter forwarding mode to bypass these void regions results in a significant increase in hop count, leading to increased E2E delay. For QMR, it takes into account the constraints of packet deadlines and the actual velocity, resulting in a partial reduction in the number of hops required for packet forwarding. However, QMR solely selects the next hop with the largest weighted Q-value, potentially trap the algorithm in local optima. On the contrary, $\mathrm{QR^2HPC}$ introduces a warm start mechanism into ϵ-greedy strategy, greatly enhancing the search efficiency of Q-Learning. This improvement effectively reduces the number of hops needed for packet transmission in $\mathrm{QR^2HPC}$.

The throughput performance of the proposed routing proto-

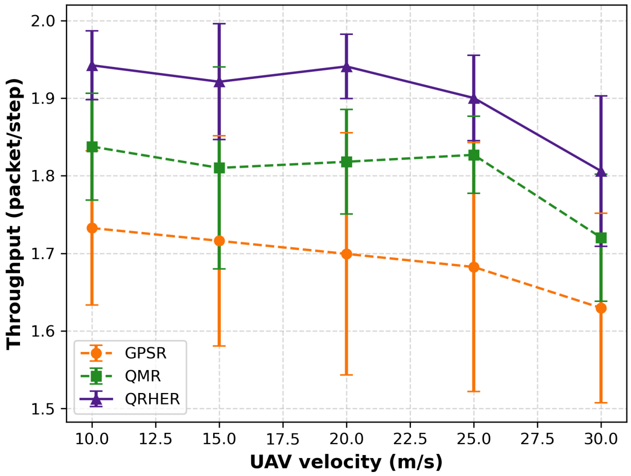

Fig. 4. Throughput VS. Node velocity

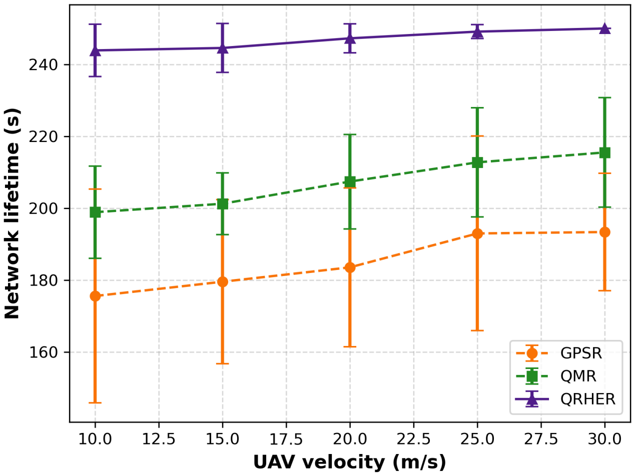

Fig. 5. Network lifetime VS. Node velocity

col with different velocity of UAV nodes are evaluated in Fig.4. It can be seen that both throughput and PDR exhibit similar trends. Our proposed algorithm outperforms other schemes, as demonstrated by the results. Specifically, when the UAV node moves at a speed of 20 m/s, the throughput under GPSR is merely 1.7 packets/slot. In contrast, our proposed method achieves a throughput of nearly 1.95 packets/slot, representing a significant improvement of 14.7%.

Fig.5 shows the network lifetime under different velocity of UAV nodes. In our simulation, we consider as an approximation that the main energy consumption is due to the emission and reception of a packet. Therefore, as the speed of the UAV node increases, there is a greater probability of encountering scenarios where the node becomes isolated without any neighboring nodes. In such cases, the UAV is unable to forward the packet and can only carry it on its own. Consequently, this leads to an increase in end-to-end delay and an extended network lifetime.

CONCLUSION

In this paper, we proposed a Q-learning aided resilient routing protocol with hindsight pre-calculation (QR^{2}HPC) in UAV swarm for the construction of the emergency networks. Our protocol utilizes a dynamic exploration and exploitation scheme to adapt to changes in network topology. To mitigate the impact of blind exploration and poor experience on agent training, we propose a warm-start mechanism during the exploration phase. This mechanism replaces the traditional random next hop selection with a routing approach guided by various indicators. Finally, with purpose of increasing the convergence speed of Q-learning, a hindsight pre-calculation (HPC) mechanism is proposed. Extensive simulations illustrate that our proposed routing algorithm can effectively reduce the E2E delay and improve PDR, throughput and network lifetime in UAV swarm network.

REFERENCES

[1] T. Do-Duy, L. D. Nguyen, T. Q. Duong, S. R. Khosravirad and H. Claussen, "Joint Optimisation of Real-Time Deployment and Resource Allocation for UAV-Aided Disaster Emergency Communications," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3411-3424, 2021.

[2] W. Feng et al., "UAV-Enabled SWIPT in IoT Networks for Emergency Communications," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 140-147, 2020.

[3] W. Feng et al., "NOMA-based UAV-aided networks for emergency communications," *China Communications*, vol. 17, no. 11, pp. 54-66, 2020.

[4] N. Lin, Y. Liu, L. Zhao, D. O. Wu and Y. Wang, "An Adaptive UAV Deployment Scheme for Emergency Networking," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2383-2398, 2022.

[5] N. Zhao et al., "UAV-Assisted Emergency Networks in Disasters," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 45-51, 2019.

[6] Z. Yao, W. Cheng, W. Zhang and H. Zhang, "Resource Allocation for 5G-UAV-Based Emergency Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3395-3410, 2021.

[7] T. Clausen and P. Jacquet, "Rfc:3626: Optimized Link State Routing Protocol (OLSR)," 2003.

[8] C. Perkins, E. Beldingroyer, S. Das, "RFC3561: Ad hoc On-demand Distance Vector (AODV) Routing," 2003.

[9] B. Karp, "GPSR : Greedy Perimeter Stateless Routing for Wireless Networks," in *Proceedings of the 6th International Conference on Mobile Computing and Networking*, 2000, pp. 243-254.

[10] W. -S. Jung, J. Yim and Y. -B. Ko, "QGeo: Q-Learning-Based Geographic Ad Hoc Routing Protocol for Unmanned Robotic Networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2258-2261, 2017.

[11] J. Liu, Q. Wang, C. He et al., "QMR: Q-learning based Multi-objective Optimization Routing Protocol for Flying Ad hoc Networks," *Computer Communications*, vol. 150, no. 15, pp. 304-316, 2020.

[12] H. Ye and J. Liu, "An Enhanced Q-Learning Routing Algorithm Based on Trajectory Prediction for UAV Networks," in *13th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2021, pp. 1-5.

[13] M. Y. Arafat and S. Moh, "A Q-Learning-Based Topology-Aware Routing Protocol for Flying Ad Hoc Networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1985-2000, 2022.

[14] A. Serhani, N. Naja and A. Jamali, "QLAR: A Q-learning based adaptive routing for MANETs," in *IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, 2016, pp. 1-7.

[15] N. Meghanathan, "Impact of the Gauss-Markov Mobility Model on Network Connectivity, Lifetime and Hop Count of Routes for Mobile Ad hoc Networks," *Journal of networks*, vol. 5, no. 5, pp. 509, 2010.

[16] J. Lansky et al,. "Reinforcement learning-based routing protocols in flying ad hoc networks (FANET): A review," *Mathematics*, vol. 10, no. 16, pp. 3017, 2022.