Deep Reinforcement Learning based Routing in an Air-to-Air Ad-hoc Network

Zhe Wang
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, KY, United States
zhe.wang@louisville.edu

Ruixuan Han
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, KY, United States
ruixuan.han@louisville.edu

Hongxiang Li
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, KY, United States
h.li@louisville.edu

Eric J. Knoblock
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, OH, United States
eric.j.knoblock@nasa.gov

Rafael D.Apaza
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, OH, United States
rafael.d.apaza@nasa.gov

Michael R. Gasper
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, OH, USA
michael.r.gasper@nasa.gov

***Abstract*—This paper studies the Multiple Sources and Multiple Destinations (MSMD) routing problem in a dynamic Air-to-Air Ad-hoc Network (AAAN). We consider a spectrum limited scenario where multiple links have to share the same frequency channel so that co-channel interference becomes inevitable. As a result, routing decisions and spectrum access are coupled and must be jointly considered. This paper proposes a deep Q-learning based algorithm to find an optimal routing and channel selection strategy that minimizes the end-to-end communication delay. Specifically, under the assumption that only local information is available to every node, the Deep Q-Network (DQN) is trained offline to learn the optimal routing and channel selection strategy. After the trained DQN is implemented in every node, multiple relay nodes can simultaneously determine their next-hop relay and channel selections in real-time. Simulation results demonstrate the efficacy of our proposed algorithm.**

***Index Terms*—AAAN, Routing, Deep Reinforcement Learning, Channel Selection**

I. Introduction

The ever-increasing number of aerial vehicles flying in the national airspace leads to a new type of wireless network that establishes ad-hoc connections among aircraft in highly dynamic environment [1]–[6]. In recent years, considerable progress has been made in aerial network structures and network topology [7], [8]. The command and control of aircraft flying at low-to-medium altitude is generally managed by ground control stations [9], requiring reliable air-ground communications for critical information exchange. However, various limitations exist in air-ground communication networks, such as high infrastructure cost, shortage of licensed spectrum, and limited coverage. On the other hand, an AAAN can offload data traffic and extend the coverage area of air-ground communications in a number of applications such as search and rescue, patrolling, and delivery of goods [10]. Currently, the National Aeronautics and Space Administration (NASA) is investigating Artificial Intelligence (AI) assisted aeronautical communications including AAAN to modernize the spectrum management in the National Airspace System (NAS). In particular, some learning-based algorithms were developed to improve the spectrum utilization efficiency in a single-hop AAAN [11], [12].

This paper extends our effort to consider routing and spectrum management in a multi-hop AAAN. Multi-hop air-to-air (A2A) communications are necessary when the source and destination nodes are out of reach for direct communications. In this case, a good routing strategy must select several relay nodes to guarantee the End-to-End (E2E) quality of service (QoS). A hard deadline constrained packet transmission needs an intelligent and efficient routing solution to coordinate a group of aircraft sharing limited frequency channels.

Among existing wireless routing studies, geographical locations are considered one of the most important information to make routing decisions [13]–[15]. Some popular routing algorithms are widely adopted in different applications, such as Dijkstra's Shortest Path Routing (DSPR) algorithm [16]–[18], Bellman-Ford algorithm [19], and Floyd Warshall algorithm [18]. However, global geographical information is needed in these algorithms, which is impractical in AAAN. On the other hand, local information is more accessible in a dynamic network. In [20], the authors proposed a Greedy Perimeter Stateless Routing (GPSR) protocol based on local geographical locations, where greedy and perimeter forwarding are used to make decisions. The GPSR was adopted in [21] for dynamic routing in AAAN. Furthermore, [22], [23] redesigned the GPSR to enhance the routing performance in AAAN.

Recently, learning-based algorithms were studied to provide dynamic solutions for packet forwarding through AAAN [24]–

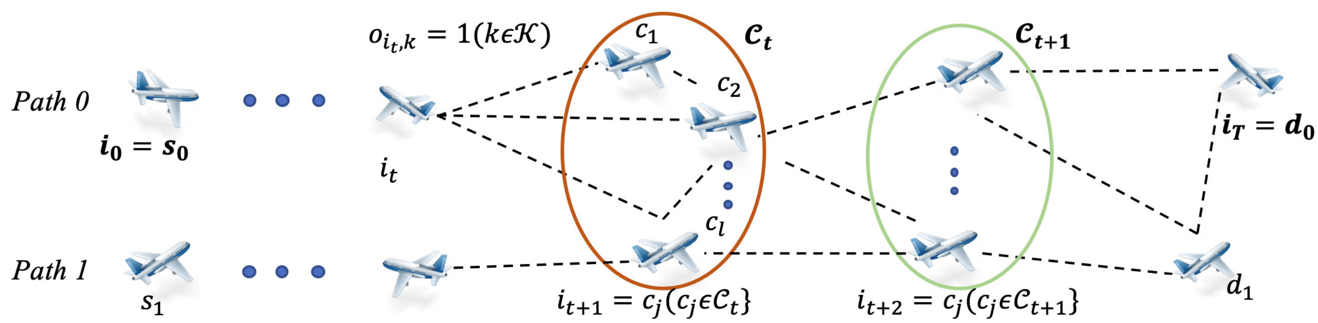

Fig. 1. Multiple Sources and Multiple Destinations (MSMD) Routing in an Air-Air Ad-hoc Network.

[27]. For example, Liu et al. proposed a deep learning-aided packet routing in AAAN based on real flight data to achieve minimized E2E delay [24]. However, the proposed algorithms focus on a Single Source and Single Destination (SSSD) scenario, without considering Multiple Sources and Multiple Destinations (MSMD) routing.

This paper considers MSMD routing, where packets go from different sources to different destinations through the same network. Due to spectrum scarcity, a frequency channel may be used by multiple links in MSMD routing so that co-channel interference is inevitable in AAAN. This paper aims to design a learning-based routing and channel selection strategy that minimizes the E2E communication delay. The challenges are multifold: First, each node may have direct connections with many nodes, making the search space intractable in a high density AAAN. Second, each node is constantly moving with high speed so the A2A channel conditions are highly dynamic. Third, multiple routing paths may share the same relay node, making the queuing delay intractable. Finally, physical channels are frequency selective so the channel selection is coupled with relay selection.

To tackle these challenges, we propose a Deep Reinforcement Learning (DRL) based MSMD A2A Routing (DMAR) Algorithm in support of multi-hop A2A communications. Our contributions are summarized as follows:

- DMAR considers joint channel selection and MSMD routing, which is a multi-dimensional combinatorial problem that suffers from the curse of dimensionality.
- After centralized training, DMAR is implemented distributively, where each relay can independently make routing and channel selection decisions by analyzing local geographical and channel information.
- The computation complexity of DMAR is tractable even in high complex AAAN, making it suitable for real-time decisions.
- Extensive simulation results corroborate the effectiveness of the proposed solution.

The rest of the paper is organized as follows. Sec. II describes the system model and problem formulation. The DMAR algorithm is discussed with details in Sec. III. Experimental results are presented in Sec. IV to evaluate the performance of DMAR algorithm in different scenarios. Finally, a conclusion is drawn in Sec. V.

II. System Model and Problem Formulation

We consider a multi-hop AAAN with N aircraft randomly located in a given airspace. There are K frequency channels $\mathcal{K} = \{1, 2, ..., K\}$ available for communications. The objective is to minimize the average E2E communication delay, which consists of propagation delay, transmission delay, and queuing delay. Fig. 1 shows two source-destination pairs coexist to find their routing paths.

We use t and T to denote the hop index and the total number of hops needed to deliver a data packet from its source to destination. The entire process of transmitting a packet from its source to destination is called an episode, and the maximum number of hops allowed for one episode is set as T_{max}. An optimal routing means to find the next relay node in every hop to minimize the E2E delay. In Fig. 1, let's use path-0 as an example, where the packet is forwarded to node i_t after the t^{th} hop. To minimize the search space for the next relay, node i_t's neighboring nodes (within direct communication range R_{def}) are ranked based on their distance to the destination in ascending order, where a subset of the top ranked nodes is selected as the candidate set $\mathcal{C}_t = \{c_1, c_2, ..., c_l\}$ for node i_{t+1}, as shown in Fig. 2.

The communication delay of the direct link from node i_t to node i_{t+1} using channel k can be expressed as:

$$D(i_t \to i_{t+1}, k) = \frac{Dist(i_t, i_{t+1})}{c} + \frac{S}{R_{i_t \to i_{t+1}}}, \qquad (1)$$

The first term is propagation delay, where $Dist(i_t, i_{t+1})$ is the distance between nodes i_t and i_{t+1} and c is the speed of light. The second term is transmission delay, where S is the packet size and $R_{i_t \to i_{t+1}}$ denotes the achievable data rate of the link $i_t \to i_{t+1}$ on channel k. Compared to the transmission delay, the propagation delay is trivial and can be omitted.

We adopt the decode-and-forward (DF) relaying strategy, where the decoding delay D_p can be considered as a constant for every receiver. Additionally, queuing delay is inevitable in multi-hop transmission and the average queuing delay at a given node can be calculated based on its queuing status:

$$D_q = \frac{1}{\mu - \lambda}(\mu > \lambda) \qquad (2)$$

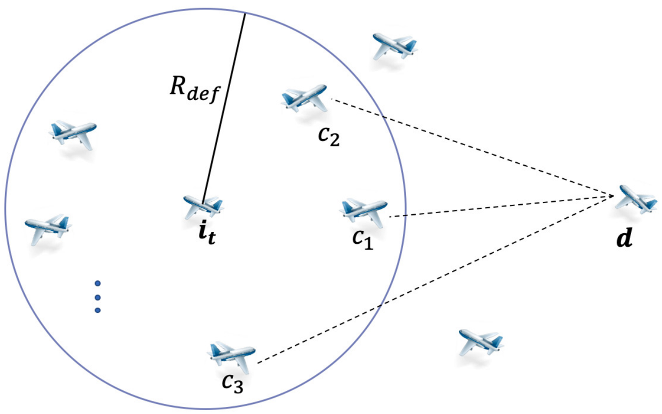

Fig. 2. Candidate Set Selection for Current Relay

where μ and λ are the packet service rate and arrival rate respectively.

Accordingly, the one-hop delay of a packet from node i_t to i_{t+1} is given by:

$$D_{i_t \to i_{t+1}} = D_q(i_t) + D_p(i_t) + D(i_t \to i_{t+1}, k). \quad (3)$$

Since D_p is a constant, only queuing delay and link transmission delay need to be considered in the optimization. Given the packet size, the transmission delay is solely determined by the achievable rate of the underlying physical channel, whereas the queuing delay is more affected by the incoming traffic load.

As shown in Fig. 1, from a source to its destination, there are many routing and channel selections to choose from. The question is how to find the optimal route $\mathcal{P} = \{i_t \to i_{t+1}, t = 0, ..., T\}$ and spectrum allocation $\mathcal{O} = \{o_{i_t,k}, k = 1, .., K\}$ that minimize the E2E delay. The joint optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\mathcal{O},\mathcal{P}} \sum_{t=0}^{T-1} D_q(i_t) + D(i_t \to i_{t+1}, k) \\ &s.t. \quad \sum_{k=1}^{K} o_{i_t,k} \leq 1, t = 0, ..., T-1 \end{aligned} \quad (4)$$

where $o_{i_t,k} \in \{0, 1\}$ is a binary channel selection indicator ($o_{i_t,k} = 1$ means node i_t chooses channel k for transmission, otherwise $o_{i_t,k} = 0$). We consider a spectrum-limited scenario where the number of available frequency channels is always less than the number of A2A links. As a result, multiple communication links may have to share the same channel, causing co-channel interference. For fairness, we also impose that no link can use more than one channel at any given time.

III. Deep Reinforcement Learning-based Routing Algorithm

In (4), the next-hop relay and channel selections depends only on the current state so that the AAAN dynamics can be modeled as a Markov Decision Process (MDP), which can be cast into our DRL framework [11], [12]. Specifically, our proposed solution takes the geolocation and Channel State Information (CSI) as input to make the optimal routing and channel selection decisions, where DQN is applied to learn the optimal strategy through trial and error. In this case, each aircraft node is an agent. We use centralized learning in the training process and the learned parameters will be shared among all agents. That is, there is only one agent in the learning network which will be trained by the experiences from all agents.

A. DRL Framework

The MDP is characterized by several essential elements: action, state, reward, and policy. Specifically, at hop t, the agent observes the environment and collect data to have state s_t. Based on s_t, the agent takes action a_t on the next relay and channel selection. Then, the agent transits into a new state s_{t+1} and receives a reward r_t from the environment. The agent aims to learn a mapping function (i.e., policy $\pi(s_t, a_t)$) from s_t to a_t that can maximize the accumulated reward. The Bellman Expectation Equation for the state-action value function (Q-function) is $Q^\pi(s_t, a_t) = \mathbb{E}^\pi\{\sum_{j=0}^{T} \gamma^t r_{t+j+1} | s_t, a_t, \pi\}$. The proposed routing algorithm aims to find an optimal policy that achieves the minimum E2E delay:

$$Q^*(s_t, a_t) = \max_{\pi} Q^\pi(s_t, a_t) \quad (5)$$

a) Action: The action vector $< i_{t+1}, o_{i_t,k} >$ indicates the selection of the next relay i_{t+1} and channel $o_{i_t,k}$, which are selected from the relay candidate set $\mathcal{C}_t$ and frequency channel set $\mathcal{K}$ respectively.

b) State: In AAAN, we assume each agent only knows the local information and the location of a packet's final destination. Thus, the input to the learning agent is the state vector consisting of two parts: (i) the locations of the current node, its relay candidates, and the destination node; (ii) the local information including the CSI between the current node and its neighbors, the CSI between its neighbors, the transmission power and channel selections of its neighbors. This information will be used to calculate the channel quality, measured in SINR, of the next-hop link.

c) Reward: According to the objective function in (4), the reward must be directly related to the E2E delay. We design a negative reward (i.e., penalty) to solve proposed minimization problem (4). Our reward function is defined as:

$$r_t = -(D_q(i_t) + D(i_t \to i_{t_1}, k) + w * RD) + r_p \quad (6)$$

where RD denotes the relative distance between the selected node and the destination node, weight w controls the penalty of the remaining distance, r_p is an added penalty for not reaching the destination (i.e., the selected node is not the destination node).

d) Policy: The policy maps the environment state into an action that maximizes the E2E delay. Specifically, the agent adopts the fully connected Neural Networks (NNs) as the policy $\pi(s, a)$ to formulate the DQN structure, which has Input, Hidden, and Output layers. The Input layer takes the current state as input so it has the same dimension as the state vector. The detail of the hidden layer is stated in Sec. IV. The Output layer has the same dimension as the action vector.

B. Offline Training

In offline training, the centralized virtual agent equipped with the policy is shared by all nodes, where the policy is trained by experiences from all nodes to find out an optimal Q-function defined in (5). While there could be more than one optimal policies in the MDP, they all achieve the same optimal Q-function. Let θ^{train} and θ^{target} denote the DQN parameters in the virtual agent and real agents, θ^{target} will be updated by θ^{train} in every T_c training events. An optimal policy can be found by maximizing $Q^*(s_t, a_t; \theta^{target})$. Note that θ^{target} is trained offline in a simulated dynamic environment, and more details are provided in Sec. IV.

For a SSSD routing, the offline training process is summarized in Algorithm 1. After t hops, a relay node has the packet that needs to be forwarded to the next relay. The relay applies ϵ-greedy method to take action on the next relay and channel selections. Specifically, with probability ϵ, the relay randomly selects an action for exploration; otherwise, it takes action $a_t = \max_{a'} Q(s_t, a'; \theta^{target})$ by exploiting the experiences. Then, the experience vector $e_t = [s_t, a_t, r_t, s_{t+1}]$ is recorded in a replay memory $\mathcal{E}$, which stores the experience vectors from all paths.

Algorithm 1 DMAR

Initialize Q-function parameter θ^{train} with random values, and $\theta^{target} \leftarrow \theta^{train}$.
Initialize $\mathcal{E}$, s, j, n_{ep}
for $ep = 0$, $ep = n_{ep}$ **do**
  for $t = 0$, $t = T_{max}$ **do**
    if $j - 1 < |\mathcal{B}|$ or $rand(t) < \epsilon$ **then**
      Randomly select an action a_t
    else
      Action $a_t = \max_{a'} Q(s_t, a'; \theta^{target})$
    end if
    Perform action a_t on environment
    Get updated state s_{t+1}, and reward r_t
    Store $e_t = [a_t, s_t, r_t, s_{t+1}]$ in $\mathcal{E}$
    if $j - 1 > |\mathcal{B}|$ **then**
      Sample random mini-batch $\mathcal{B}$ from $\mathcal{E}$
      Calculate equation (7) and update θ^{train}
    end if
    $j := j + 1$
    Update parameters every T_c: $\theta^{target} \leftarrow \theta^{train}$
  end for
end for

The stochastic gradient descent algorithm is applied to train DQN using mini-batch $\mathcal{B}$, which is randomly sampled from $\mathcal{E}$. Then, the parameter of DQN θ^{train} is updated as:

$$\theta^{train}_{t+1} = \arg\min_{\theta^{train}} \frac{1}{b} \sum_{e \in \mathcal{B}} (y^{target} - Q(s, a; \theta^{train}))^2, \quad (7)$$

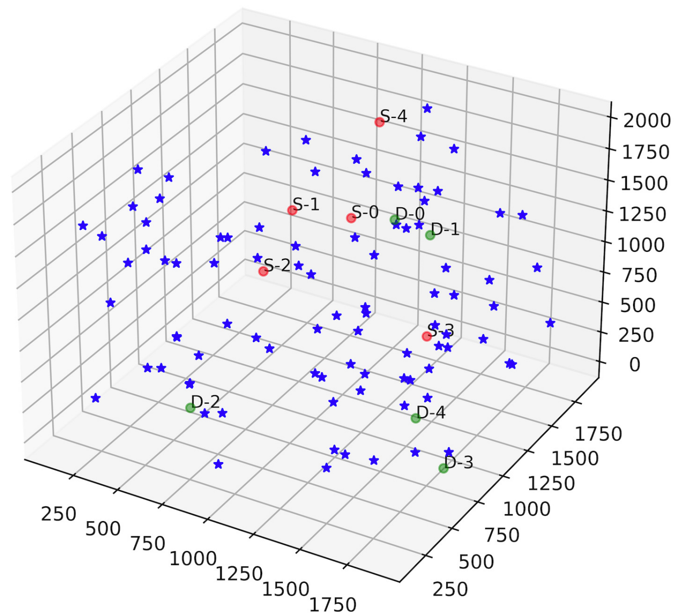

Fig. 3. Initial Node Deployment for MSMD Routing in a 3D Airspace.

where the target value y^{target} to train DQN is calculated by:

$$y^{target} = r + \gamma \max_{a'} Q(s, a'; \theta^{target}) \quad (8)$$

After the training converges, an optimal policy is obtained with maximum $Q(s, a)$. Then, DQN parameters θ^{train} and θ^{target} are synchronized to implement the optimal routing strategies. In the case of multiple source-destination pairs, multiple routing paths work in parallel.

IV. Simulation Results

A. Simulation Environment

We conduct simulation experiments to evaluate the effectiveness of our DMAR algorithm for MSMD routing in a dynamic AAAN. In Fig. 3, a total of 100 nodes are uniformly distributed in a 3-D airspace of $2000 \times 2000 \times 2000$ cubic, where the red and green circular points represent the source and destination nodes respectively. There are five concurrent source-destination pairs. All nodes are flying at constant speed towards random directions, with a minimum separation distance enforced for flight safety. When a node hits the airspace boundary, it will be bounced back at the same speed. The achievable data rate of each link is determined by its bandwidth and the received SINR, which can be calculated as in [11]. We assume $5MHz$ bandwidth for each channel and the packet size is set as $S = 1KB$. The size of each relay candidate set is 8. The queuing delay is calculated by (2), where the service rate is set as $\mu = 200$ packets/s, and the arrival rate is estimated based on recent historical traffic. We use Python program for real-time simulation with a resolution of 1 ms, i.e., the system updates the node locations and channel information every 1 ms.

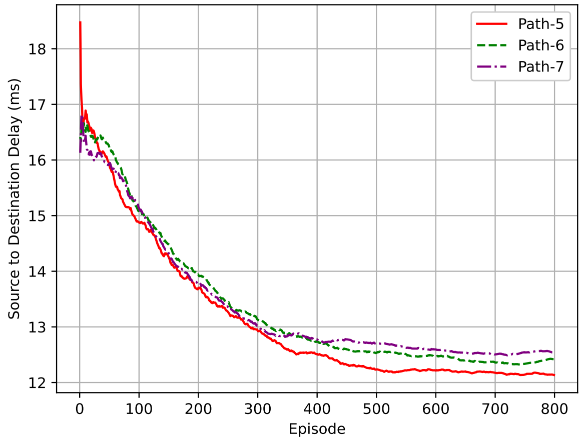

Fig. 4. DMAR Learning with Path = 5, 6, 7.

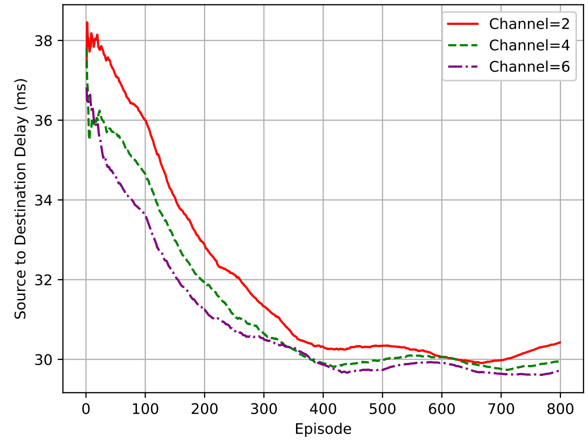

Fig. 5. DMAR Learning with Channel = 2, 3, 4.

B. DQN Parameters

The DQN has four hidden layers with 200, 200, 100, 50 neurons. The Rectified Linear Unit (ReLU) is selected as the activation function for each hidden layer. In the training process, the initial exploration probability is set as $\epsilon = 0.5$, then it decreases to 0.001 with a decaying rate of 0.5% per episode. The reply memory size is $B = 200$, and the mini-batch size is $b = 64$. To achieve "quasi-static" of the neural network, the target network is updated every 50 training events. The reward discount factor is set as $\gamma = 0.4$. Furthermore, Root Mean Square Propagation (RMSPro) is selected as the optimization method, which adopts an adaptive learning rate in the mini-batch stochastic gradient descent.

C. Evaluations for MSMD Transmission

To test the reliability and scalability of our DMAR, we conduct several experiments with different configurations. In the first experiment, there are 100 nodes and 2 channels. Fig. 4 shows the E2E delay as a function of the training episode for different number of paths (5, 6, and 7), where the delay is averaged over all paths. To smooth the curves, the moving average of 50 episodes is applied. We can see that our DMAR algorithm is able to converge after about 500 episodes. We also observe that the converged E2E delay increases slightly with the number of paths, which is expected.

In the second experiment, there are 200 nodes and 80 paths. Fig. 5 shows similar E2E delay convergence curves under different number of channels (2, 4, and 6). We can see that, for a denser AAAN, the E2E delay still converges in all cases. As expected, the E2E delay decreases with the number of available channels. This is because more frequency channels will reduce the co-channel interference, and thus improve A2A link's achievable data rate.

D. Non-learning based Routing Algorithms

To better evaluate our learning based solution, we also use two non-learning based baselines for performance comparison.

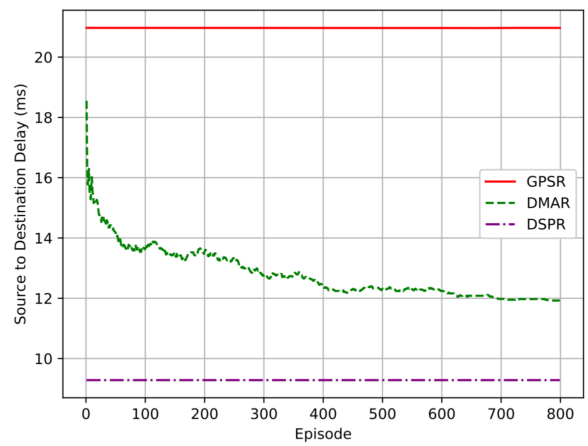

Fig. 6. Average Delay Comparison Among DQN Learning, DSPR, and GPSR.

a) DSPR: Dijkstra Shortest Path Routing (DSPR) algorithm is a widely adopted routing algorithm in many studies [16]–[18]. However, the original DSPR is not applicable to the problem in (4) because the co-channel interference and channel selection are not considered. In this paper, we revise the DSPR by assuming a fixed data rate for all channels, so it provides a performance upper bound for our solution.

b) GPSR: Greedy Perimeter Stateless Routing (GPSR) is an efficient routing protocol for wireless networks [20]. GPSR exploits the geographic locations and connectivity in a wireless network to find the shortest-paths routes. Specifically, a relay gathers the location information from its neighbors, and applies greedy forwarding or perimeter forwarding to make forwarding decisions. The original GPSR algorithm doesn't consider channel selection, so the channel is randomly selected in our experiment.

We conduct experiments to compare our proposed DMAR with the modified DSPR and GPSR algorithms. The simulation setup includes 100 nodes, 2 channels and 5 paths. Fig. 6 illustrates the average E2E delay vs. the number of episode for different algorithms, where GPSR and DSPR provide performance lower and upper bounds for DMAR. In particular, the converged delay values for DMAR, DSPR, and GPSR are 11.9 ms, 9.3 ms, and 20.8 ms, respectively.

V. Conclusion

This paper studies the joint MSMD routing and channel selection problem to minimize the E2E delay in a highly dynamic AAAN environment. We propose a DRL-basd solution to predict the next relay and channel selection that can minimize the E2E delay. In particular, our DMAR algorithm supports concurrent MSMD routing in a distributed manner where each agent only has local information from its neighbors. The simulation results demonstrate the effectiveness of our learning based approach in complex and dynamic aerial networks.

References

[1] E. Sakhaee, A. Jamalipour, and N. Kato, "Aeronautical ad hoc networks," in *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, vol. 1, pp. 246–251, IEEE, 2006.

[2] V. Sharma and R. Kumar, "Cooperative frameworks and network models for flying ad hoc networks: a survey," *Concurrency and computation: Practice and experience*, vol. 29, no. 4, p. e3931, 2017.

[3] J. Zhang, T. Chen, S. Zhong, J. Wang, W. Zhang, X. Zuo, R. G. Maunder, and L. Hanzo, "Aeronautical $a\breve{d}hoc$ networking for the internet-above-the-clouds," *Proceedings of the IEEE*, vol. 107, no. 5, pp. 868–911, 2019.

[4] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 136–142, 2018.

[5] J. Wu, L. Zou, L. Zhao, A. Al-Dubai, L. Mackenzie, and G. Min, "A multi-uav clustering strategy for reducing insecure communication range," *Computer Networks*, vol. 158, pp. 132–142, 2019.

[6] G. Gui, Z. Zhou, J. Wang, F. Liu, and J. Sun, "Machine learning aided air traffic flow analysis based on aviation big data," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4817–4826, 2020.

[7] D. Medina and F. Hoffmann, "The airborne internet," in *Future Aeronautical Communications*, IntechOpen, 2011.

[8] Q. Vey, A. Pirovano, J. Radzik, and F. Garcia, "Aeronautical ad hoc network for civil aviation," in *International Workshop on Communication Technologies for Vehicles*, pp. 81–93, Springer, 2014.

[9] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019.

[10] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for uav airborne networks toward on-demand coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2015–2031, 2018.

[11] Z. Wang, H. Li, E. J. Knoblock, and R. D. Apaza, "Joint spectrum access and power control in air-air communications - a deep reinforcement learning based approach," in *2021 40th Digital Avionics Systems Conference (DASC)*, 2021.

[12] Z. Wang, H. Li, E. J. Knoblock, and R. D. Apaza, "Learning based edge computing in air-to-air communication network," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 333–338, IEEE, 2021.

[13] M. Iordanakis, D. Yannis, K. Karras, G. Bogdos, G. Dilintas, M. Amirfeiz, G. Colangelo, and S. Baiotti, "Ad-hoc routing protocol for aeronautical mobile ad-hoc networks," in *Fifth international symposium on communication systems, networks and digital signal processing (CSNDSP)*, pp. 1–5, Citeseer, 2006.

[14] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A geographic routing strategy for north atlantic in-flight internet access via airborne mesh networking," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1231–1244, 2011.

[15] S. Wang, C. Fan, C. Deng, W. Gu, Q. Sun, and F. Yang, "A-gr: A novel geographical routing protocol for aanets," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 931–937, 2013.

[16] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[17] J. Hu, L.-L. Yang, and L. Hanzo, "Energy-efficient cross-layer design of wireless mesh networks for content sharing in online social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8495–8509, 2017.

[18] B. N. Karp, *Geographic routing for wireless networks*. Harvard University, 2000.

[19] Y. Xu, J. Liu, Y. Shen, X. Jiang, Y. Ji, and N. Shiratori, "Qos-aware secure routing design for wireless networks with selfish jammers," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 4902–4916, 2021.

[20] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, 2000.

[21] B.-S. Kim, K.-I. Kim, B. Roh, and H. Choi, "A new routing protocol for uav relayed tactical mobile ad hoc networks," in *2018 wireless telecommunications symposium (WTS)*, pp. 1–4, IEEE, 2018.

[22] A. Silva, N. Reza, and A. Oliveira, "Improvement and performance evaluation of gpsr-based routing techniques for vehicular ad hoc networks," *IEEE Access*, vol. 7, pp. 21722–21733, 2019.

[23] Y.-n. Chen, N.-q. Lyu, G.-h. Song, B.-w. Yang, and X.-h. Jiang, "A traffic-aware q-network enhanced routing protocol based on gpsr for unmanned aerial vehicle ad-hoc networks," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 9, pp. 1308–1320, 2020.

[24] D. Liu, J. Cui, J. Zhang, C. Yang, and L. Hanzo, "Deep reinforcement learning aided packet-routing for aeronautical ad-hoc networks formed by passenger planes," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5166–5171, 2021.

[25] J. Cui, D. Liu, J. Zhang, H. Yetgin, S. X. Ng, R. G. Maunder, and L. Hanzo, "Minimum-delay routing for integrated aeronautical ad hoc networks relying on real flight data in the north-atlantic region," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 310–320, 2021.

[26] B. Du, X. Di, D. Liu, and H. Zhang, "Dynamic graph optimization and performance evaluation for delay-tolerant aeronautical ad hoc network," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6018–6036, 2021.

[27] J. Yang, K. Sun, H. He, X. Jiang, and S. Chen, "Dynamic virtual topology aided networking and routing for aeronautical ad-hoc networks," *IEEE Transactions on Communications*, 2022.