Inter-Satellite Routing for LEO Satellite Networks: A GNN and DRL Integrated Approach

Peng Xu[1], Mingjie Feng[1], Jiaxi Zhou[1], Lixia Xiao[1], Pei Xiao[2], and Tao Jiang[1]
[1]Research Center of 6G Mobile Communications, School of Cyber Science and Engineering
and Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology
[2]Institute for Communication Systems (ICS), University of Surrey
Email: {pengxu, mingjiefeng, zhoujiaxi, lixiaxiao}@hust.edu.cn, p.xiao@surrey.ac.uk, tao.jiang@ieee.org

2024 IEEE/CIC International Conference on Communications in China (ICCC) | 979-8-3503-7841-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCC62479.2024.10681995

***Abstract*—With the vision of supporting global internet access, satellite communication has emerged as a key component for sixth-generation (6G) communication networks. A pivotal trend for satellite communication is using Low Earth Orbit (LEO) satellites to construct a space-based backbone network, which necessitates establishing multi-hop communication paths between satellites with certain inter-satellite routing strategies. However, the vast number of satellites and the fast-varying network topology, combined with the diversified traffic demands, pose significant challenges for inter-satellite routing. In this paper, we propose a novel inter-satellite routing scheme that employs Graph Neural Networks (GNN) and Deep Reinforcement Learning (DRL) to address the above-mentioned challenges. The overall objective is minimizing network-wide average end-to-end delay, which can be achieved by selecting paths with low aggregated propagation delay and balancing the load distribution among various links. Based on this fact, we formulate the inter-satellite routing problem, in which the satellite network is modeled as a graph. Then, we design a GNN and DRL integrated solution framework, where GNN is employed to learn the complex relationship between satellite nodes in the graph and DRL is employed to make routing decisions that are adaptive to the features of satellite networks. Simulation results demonstrate that our solution can adapt to the time-varying network dynamics and diverse traffic demands, outperforming benchmark schemes in terms of average end-to-end delay, reliability, and link utilization.**

***Index Terms*—Low Earth Orbit Satellites; Inter-Satellite Routing; Graph Neural Networks; Deep Reinforcement Learning;**

I. Introduction

With the advancement of space-ground communication technology and the reduction of satellite launching cost, Low Earth Orbit (LEO) satellite communication is envisioned as the key enabler for achieving global connectivity in sixth-generation (6G) communication networks [1]. By establishing satellite-user links, users can still access the Internet even if they are outside the coverage of base stations. Due to the low altitude of LEO satellites, low-latency communication can be achieved. Meanwhile, most existing satellite communication networks rely on the ground core network to support end-to-end communication between mobile users, which is implemented by connecting satellites with ground stations and using them as gateways to access the core network. However, due to many practical constraints, ground stations cannot be deployed in many areas, limiting the operation of satellite networks. Besides, as the number of satellites keeps growing, the links between satellites and ground stations will become more congested, resulting in degraded communication performance.

To overcome the above limitations, constructing a space-based core network, which is composed of numerous LEO satellites that directly communicate with each other, has become a pivotal trend in satellite communication. Given the limited coverage of each LEO satellite, performing end-to-end communication with a space-based core network requires creating multi-hop communication routes. Thus, inter-satellite routing is a crucial design factor that impacts the performance of satellite communication [2]. Due to such importance, inter-satellite routing has been investigated in recent works. In [3], a dynamic routing-based routing scheme was proposed, which improves the inter-satellite link utility by adapting to the time-varying network topology. In [4], a network grid model, which divides the whole space into multiple cubes, was established to portray the network topology, based on which shortest path-based routing algorithms were proposed. From the perspective of quality of service (QoS) guarantee, an on-demand routing strategy was designed to fully utilize the scarce network resources and improve the QoS of users [5]. Considering load balancing among various paths, a traffic-aware routing algorithm was developed in [6], which effectively improved the network-wide performance.

Despite the success of these routing schemes, the challenges caused by the unique features of LEO satellite networks were not sufficiently considered in existing works. First, the rapid movement of LEO satellites at speeds ranging from 6.9 km/s to 7.7 km/s leads to frequent changes in the network topology and the channel gains of inter-satellite links. Second, in order to achieve global coverage, LEO satellites are extensively deployed, leading to a surging number of satellites (e.g., Starlink has deployed nearly 6,000 satellites by March 2024) and complex network topology. Third, as the traffic demands are increasingly diversified, traditional solutions designed for homogeneous traffic become less efficient, resulting in unsatisfactory performance. The large scale and high dynamism of LEO satellite networks, combined with the varying traffic demand, pose significant challenges for the design of inter-satellite routing.

To develop routing strategies that are capable of adapting to the features of LEO satellite network, we propose a routing

979-8-3503-7841-2/24/$31.00 ©2024 IEEE

framework based on the integration of Graph Neural Networks (GNN) and Deep Reinforcement Learning (DRL). DRL is a machine learning paradigm renowned for solving complex decision-making problems in dynamic environments, which can optimize routing decisions in large-scale satellite networks by learning the patterns of network dynamics. Nonetheless, conventional DRL algorithms lack generalization ability, hindering their performance in satellite networks with fast-varying topologies. GNN is a class of deep learning models tailored to process graph-structured data, which can handle unseen data or topology by learning the relationship between nodes in a graph. Such generalization capability is well-suited for inter-satellite routing design in the presence of rapidly changing and complex network topology, as well as the diversified traffic demand. Due to the above-mentioned benefits, we design a GNN and DRL integrated model for generating routing decisions, which not only can handle large-scale networks, enabling it to capture the inherent complex relationships and dynamics in satellite networks, but also seamlessly adapts to different network environments, providing powerful generalization capabilities. Through iterative exploration and development, the model can approach the optimal routing strategies considering factors such as satellite orbits, link capacities, and traffic patterns. Our contributions can be summarized as follows:

- Based on a virtual topology for the satellite constellation, we model the satellite network as a graph. We formulate the routing problem with the overall objective of minimizing network-wide average end-to-end delay, which can be achieved by selecting the paths with low aggregated propagation delay and balancing the load distribution among various links. Based on this fact, we define the objective function as the sum of the weighted sums of the inverse of propagation delay and the residual capacity of all traffic demands along all paths.
- Considering the network dynamics, we further formulate the routing problem as a Markov Decision Process (MDP) and propose a GNN and DRL integrated approach to optimize routing decisions. To reduce the implementation complexity, we employ the k-shortest path algorithm to generate the action space, resulting in reduced dimension for MDP and faster convergence during training.
- Simulation results show that our approach outperforms benchmark solutions in terms of end-to-end latency and link reliability.

In the remainder of this paper, we first present the problem formulation in Section II. The solution algorithms are introduced in Section III. We then show the simulation results and conclude the paper in Section IV and V, respectively.

II. Problem Formulation

We consider a LEO satellite network composed of a vast number of LEO satellites, where communication between satellites relies on Inter-Satellite Links (ISL). The four-link algorithm is used for ISL establishment between neighboring satellites [7]. We divide time into time slots. In each time slot, the network topology is regarded as fixed [8], which is modeled as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the sets of satellites and ISLs, respectively. Let N and M denote the cardinalities of $\mathcal{V}$ and $\mathcal{E}$, respectively. The traffic demand is characterized by an $N \times N$ traffic matrix $\mathbf{F}$, where each element $f_{o,d}$ represents the required data transmission rate between each origin-destination pair o and d. Each ISL is parameterized by two parameters: (i) capacity $R_{i,j}^{\text{total}}$, which specifies the aggregated data rate of all traffic demands that can be supported by ISL (i, j); (ii) distance $W_{i,j}$, which is the straight line distance between satellites i and j. We assume that each ISL operates in a full-duplex pattern with symmetric capacity.

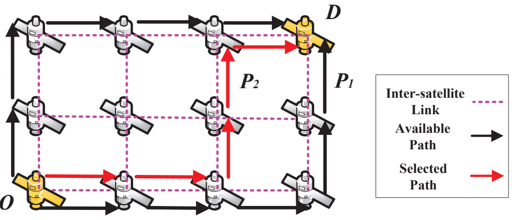


Fig. 1. System model of inter-satellite routing in this paper. The set $\mathcal{P}_{o,d} = \{p_1, p_2, \cdots, p_K\}$ represents all available paths, with the red path indicating the selection of path p for demand (o, d). In this example, $x_{o,d}^{p_2} = 1$.

As shown in Fig. 1, $\mathcal{P}_{o,d}$ denotes all available paths for demand (o, d) and $x_{o,d}^{p}$ is a binary decision variable indicating the selection of path p for demand (o, d). Specifically, $x_{o,d}^{p} = 1$ indicates that path p is selected for demand (o, d), while $x_{o,d}^{p} = 0$ indicates otherwise. We denote $f_{i,j}^{p}$ as the flow on link (i, j) for demand (o, d) along path p, which is the required data rate for demand (o, d) along path p.[1] The flow on link (i, j) for demand (o, d) can be expressed as:

$$f_{i,j}^{o,d} = \sum_{p \in \mathcal{P}_{o,d}} x_{o,d}^{p} \cdot f_{i,j}^{p}. \tag{1}$$

For each ISL (i, j), its average traffic load $\lambda_{i,j}$ can be calculated by:

$$\lambda_{i,j} = \sum_{o,d \in \mathcal{V}} f_{i,j}^{o,d}. \tag{2}$$

As mentioned, load balancing is a critical factor that impacts performance, which is characterized by the residual capacity of a link or path. We define the residual capacity of a path as the minimum residual capacity among its edges (i.e., the ISLs that compose the path). Hence, the residual capacity along path p for demand (o, d) is given by:

$$R_{p,o,d}^{\text{res}} = \min_{(i,j) \in p} \left\{ R_{i,j}^{\text{total}} - \lambda_{i,j} \right\}, \quad \forall p \in \mathcal{P}_{o,d}. \tag{3}$$

Let $L_{p,o,d}^{\text{prop}}$ be the propagation delay of demand (o, d) along path p, it is the summation of the propagation delays across all edges along path p, given by:

$$L_{p,o,d}^{\text{prop}} = \sum_{(i,j) \in p} \frac{W_{i,j}}{c}, \quad \forall p \in \mathcal{P}_{o,d}, \tag{4}$$

[1] The resource of each ISL (e.g., wavelength) is shared by various traffic demand or flow, which corresponds to the capacity of each ISL shared among various flows with certain data rate requirements.

where $W_{i,j}$ is the distance of ISL (i,j) and c is the speed of light.

The overarching objective of our inter-satellite routing design is to minimize the network-wide average end-to-end delay, which is impacted by multiple practical factors (e.g., delays for data transmission, queuing, and processing). As routing decisions must be made before data transmission, we can only rely on measurable parameters to estimate the achievable performance and make decisions accordingly. Thus, we formulate the routing problem with the objective of maximizing the sum of the weighted sums of the inverse of delay and the residual capacity of all traffic demands along all paths. Such a problem is formulated as:

$$\max \sum_{o,d\in\mathcal{V}} \sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}\cdot(\alpha\frac{1}{L^{\text{prop}}_{p,o,d}}+\beta R^{\text{res}}_{p,o,d}). \tag{5}$$

$$\text{s.t. } \lambda_{i,j}\le R^{\text{total}}_{i,j},\quad \forall (i,j)\in\mathcal{E}, \tag{6}$$

$$\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}(f^p_{i,j}-f^p_{j,i}) = \begin{cases} f_{o,d}, & \text{if } i=o,\\ -f_{o,d}, & \text{if } i=d,\\ 0, & \text{otherwise}\end{cases}$$

$$\forall o,d\in\mathcal{V},\ \forall i\in\mathcal{V}, j\in\mathcal{V}\setminus\{o,d\}, \tag{7}$$

$$\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}=1,\quad \forall o,d\in\mathcal{V}. \tag{8}$$

The constraints in (6) ensure that the total flow on each edge does not exceed its capacity. Eq. (7) represents flow conservation constraints, which indicates the balance of inflow and outflow for each demand at each intermediate node [9]. Eq. (8) reflects the path selection constraints, which ensures that only one path can be selected for each demand (o,d).

III. Solution Algorithms

In this section, we present our GNN and DRL integrated solution for inter-satellite routing.

A. GNN in the Context of Inter-Satellite Routing

The core of GNN is encapsulated within the Message Passing Neural Network (MPNN) framework [10]. In satellite routing problems, MPNN enables the local propagation of information across the network by iteratively aggregating and updating node features based on the features of neighboring nodes.

The process of MPNN begins with the Input Representation step, where graph-structured data undergoes transformation into tensor representations that are suitable for neural network processing. Second, the Node Features step endows each node with a feature vector encompassing its essential information, such as the key parameters of satellite nodes and link statuses. The third step is Message Passing and Aggregation, where local information is exchanged across the graph, refining the features of each node by considering the attributes of neighboring nodes. This is expressed mathematically as message aggregation, represented by:

$$m_v^{(t+1)} = \mathbb{A}(\{\mathbb{M}(h_v^{(t)}, h_u^{(t)})|u\in B(v)\}), \tag{9}$$

where $m_v^{(t+1)}$ represents the updated message for node v at iteration $t+1$, $B(v)$ is the set of neighboring nodes of node v, $h_v^{(t)}$ is the feature vector (hidden state) of node v at iteration t, $\mathbb{M}$ represents the message function and $\mathbb{A}$ denotes the aggregation function (with summation used for approximation in this work). Following message aggregation, the Update Rules, governed by the function $\mathbb{U}$, determine how each node's feature state is iteratively updated:

$$h_v^{(t+1)} = \mathbb{U}(x_v^{(t)}, m_v^{(t+1)}), \tag{10}$$

where $h_v^{(t+1)}$ represents the updated feature vector for node v at iteration $t+1$. Lastly, the Readout Function $\mathbb{R}$ transforms the final state of nodes $h_v^{(T)}$ into the network's output:

$$\hat{y} = \mathbb{R}(\mathbb{A}(\{h_v^{(T)} \mid v\in V\})), \tag{11}$$

B. DRL in the Context of Inter-Satellite Routing

The estimation for Q-values is the key for DRL. Q-values are the expected cumulative rewards associated with various actions for a given state, which are calculated by:

$$Q(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \mid s_t=s, a_t=a, \pi\right]. \tag{12}$$

$Q(s,a)$ is the sum of rewards, discounted by γ, attainable at each time step t following observation of state s and execution of action a, guided by policy π.

DRL is implemented by training the deep Q-network (DQN), which is a class of deep neural networks to approximate the Q-values. A DQN takes the current state as input and generates Q-values for all possible actions. However, utilizing neural networks to represent Q-functions often leads to instability or even divergence during training [11]. To address this challenge, DQN incorporates two crucial techniques into its model. The first technique is Experience Replay, which bolsters training stability and sampling efficiency by storing experiences in a replay buffer and randomly sampling them for training. This approach disrupts the temporal correlations between consecutive events. The second technique is Fixed Q-Targets. Specifically, two neural networks are employed during training: the Q-network (referred to as the Behavior Network in this paper) and the Target Network. The Behavior Network continually learns and updates, while the Target Network periodically copies the parameters of the Behavior Network to compute the target Q-values. The asynchronous parameter updates of the two networks decouple the target Q-values from the parameters of Q-values of the Behavior Network, thus enhancing training stability. The target Q-values can be expressed as:

$$Q_{\text{target}} = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-). \tag{13}$$

Consequently, the Target Network is fixed while the Behavior Network is updated using the gradient descent method during training. The Loss Function L, typically based on

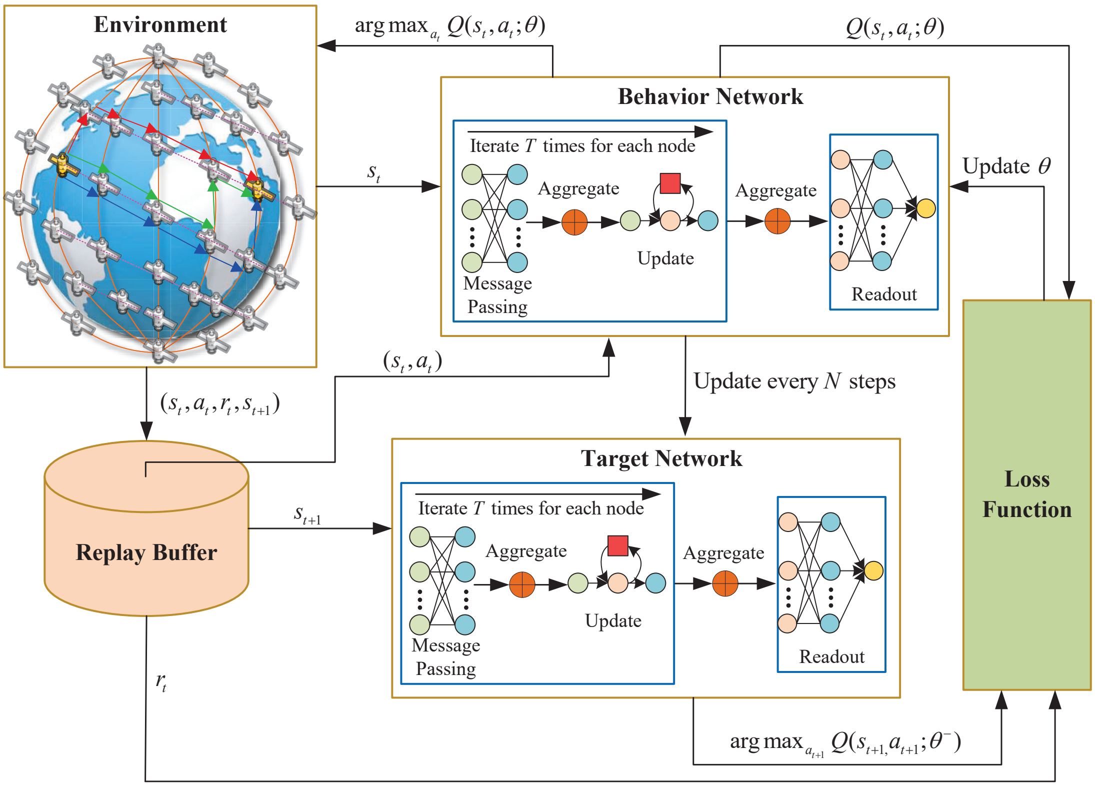


Fig. 2. Framework of our proposed GNN-DRL integrated solution.

mean squared error (MSE), quantifies the discrepancy between predicted and target Q-values, given by:

$$L(\theta) = \mathbb{E}\left[(Q_{\text{target}} - Q(s_t, a_t; \theta))^2 \right]. \tag{14}$$

Finally, DRL balances exploration and exploitation through an ϵ-greedy strategy. Specifically, the DRL agent selects a random action with a certain probability (ϵ) to explore the environment, while the selecting action with the highest estimated Q-value to maximize the reward.

C. *The Proposed GNN and DRL Approach*

Our proposed architecture, depicted in Fig. 2, combines GNN and DRL to improve satellite routing decisions. Initially, we extract the initial environment state from the LEO satellite network and select action a using the ϵ-greedy strategy. Then, we execute action a, observe the reward r and new state s', and store the transition (s, a, r, s') in the replay buffer. Afterward, we sample an episode (s, a, r, s') from the replay buffer and input it into the Behavior Network and the Target Network, both constructed with the MPNN architecture. In the MPNN, the state s undergoes preprocessing before passing through the fully connected layer responsible for message passing. Next, through T iterations, historical messages are updated using RNN and aggregated with a summation function. Finally, the readout function outputs $Q(s, a; \theta)$. Similarly, the new state s' is fed into the target network to obtain $Q(s', a'; \theta^-)$, i.e., the target Q value. The weights of the Behavior Network are updated via gradient descent until convergence.

The proposed GNN and DRL integrated architecture empowers the model to dynamically adapt to the ever-changing network topologies, resulting in versatile routing strategies that achieve enhanced performance. Next, we first introduce the details of the MDP formulation.

1) State Space Design: The state vector of the MDP is obtained by concatenating the state vectors of all ISLs. The state vector of an ISL encompasses the traffic demand received by the ISL and the residual capacity of the ISL. To accommodate the operation of GNN, the concatenated vector is further augmented with multiple zero elements at specific positions to store the aggregated information from adjacent ISLs. As discussed in Section III-A, MPNN requires obtaining feature vectors for all nodes and their neighboring nodes in the graph. Since the features of ISLs are the dominant factors that impact the routing design, we set ISLs as nodes in the GNN model.

2) Reward Design: The reward comprises two components: one is related to the path propagation delay and the other is determined by the residual capacity. The first component assesses the performance of path selection by comparing the length of the action path to the reference path. The second part evaluates load balancing by comparing the residual capacity of the action path to that of the reference path. This reward setting promotes efficient path selections that consider both path delay and load balancing. The reward is calculated by:

$$\text{Reward} = \alpha \frac{L_{\text{standard}}}{L_{\text{action}}} + \beta \frac{R_{\text{action}}}{R_{\text{standard}}}, \tag{15}$$

where α and β are the respective weights for the two components. L_{action} and L_{standard} are the delays of the selected and reference paths, respectively. R_{action} and R_{standard} are the residual capacities of the selected and reference paths, respectively.

3) Action Space Design: In contrast to traditional DQN-based routing algorithms, which typically define the selection of the next hop as the action, we consider the available paths between origin and destination nodes as part of the action space, which enhances the model's global awareness. To address the challenges of large-scale satellite networks, we employ Yen's k-shortest path algorithm [13] to generate the candidate paths $\mathcal{P}_{o,d}$, which constitute the set of actions.

Algorithm 1 GNN with DRL Integrated Inter-Satellite Routing Algorithm

```
Input: IRETraffic matrix $\mathbf{F}$, number of shortest paths $k$
Output: Routing table composed of optimal paths for various traffic demands
Initialize state $s$
repeat
    Based on the matrix $\mathbf{F}$, calculate $k$ available paths for each origin-destination pair using Yen's k-shortest path algorithm
    For all nodes, calculate hidden state $h_v^{(t)}$ and their neighboring nodes' state $h_u^{(t)}$ using state $s$
    for t = 1 to $T$ do
        for v = 1 to $V$ do
            $m_v^{(t+1)} = \sum_{u\in B(v)} \mathbb{M}(h_v^{(t)}, h_u^{(t)})$
            $h_v^{(t+1)} = \mathbb{U}(x_v^{(t)}, m_v^{(t+1)})$
        end for
    end for
    $Q(s,a;\theta) = \mathbb{R}(\sum_{v\in V} h_v^{(T)})$
    Select $a = \text{argmax}_a\, Q(s,a;\theta)$
until (the episode is done)
```

Algorithm 1 illustrates the process of generating an inter-satellite routing strategy with the GNN and DRL integrated network, which is trained by the process shown in Fig. 2.

IV. Performance Evaluation

A. Experimental Setup

We utilize the keras-based framework in Python to construct our MPNN and DQN models. We evaluate the performance of our proposed approach with a satellite network consisting of 9 inclined orbits. Each inclined orbit in the LEO constellation accommodates 12 satellites with an inclination of 50 degrees, and all satellites are positioned at an altitude of 550 km. We establish 4 inter-satellite links for each satellite, with two links within the same orbit and two links between adjacent orbits. We set the capacity of all ISLs to 10 Mbps, while the traffic demands $f_{o,d}$ fluctuate between 1.8 and 2.7 Mbps, with a packet size η of 10 Kbs.

We compare our proposed GNN and DQN integrated approach (termed GQN) with three baseline schemes: the Shortest Path (SP), the Load Balancing (LB), and the DQN (DQN). The Shortest Path scheme selects the shortest path among all possible paths, the Load Balancing scheme randomly chooses a path with non-zero "residual capacity," and the DQN scheme employs a conventional DNN to represent the Q-function instead of using an MPNN.

B. Performance Metrics

We evaluate both the average end-to-end delay and packet loss rate under various sum required data rates (i.e., the sum of all flows).

1) Average End-to-End Delay: The average end-to-end delay L_{E2E} can be computed as follows:

$$L_{\text{E2E}} = L_{\text{prop}} + L_{\text{que}} + L_{\text{tran}}, \tag{16}$$

where L_{prop}, L_{que}, and L_{tran} represent the average propagation delay, the average queuing delay of the whole network [14], and the average transmission delay, respectively. These are calculated as:

$$L_{\text{prop}} = \frac{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d} L^{\text{prop}}_{p,o,d}}{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}}, \tag{17}$$

$$L_{\text{que}} = \sum_{(i,j)\in\mathcal{E}} \frac{\eta}{\lambda}\left(\frac{\lambda_{i,j}}{R^{\text{total}}_{i,j} - \lambda_{i,j}}\right), \tag{18}$$

$$L_{\text{tran}} = \frac{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d} H_p(\eta/f^p_{i,j})}{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}}, \tag{19}$$

where $\lambda = \sum_{o,d\in\mathcal{V}} f_{o,d}$ is the sum of all traffic demands, and H_p represents the number of hops for each path.

Fig. 3(a) demonstrates that when the sum required data rate is low, propagation delay has a significant impact on end-to-end delay, and GQN does not show significant advantages over SP. As the sum required data rate increases, the average end-to-end delays of all schemes increase. GQN achieves the lowest delay among all schemes, and the advantage is more significant under high traffic demands (i.e., high sum required data rate).

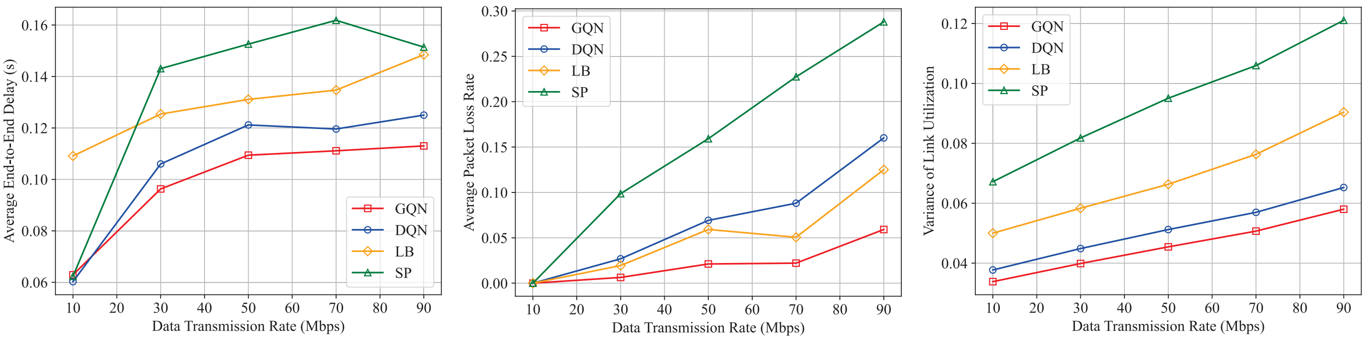
2) Packet Loss Rate: Based on Constraint (6), the packet loss rate for demand (o,d) along path p is calculated by:

$$\phi^p_{o,d} = \max_{(i,j)\in p} \frac{f^p_{i,j} - R^{\text{total}}_{i,j}}{f^p_{i,j}}, \quad \forall p \in \mathcal{P}_{o,d}. \tag{20}$$

Then, the average packet loss rate for all traffic demands is:

$$\Phi = \frac{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}\cdot\phi^p_{o,d}}{\sum_{o,d\in\mathcal{V}}\sum_{p\in\mathcal{P}_{o,d}} x^p_{o,d}}. \tag{21}$$

The performance of packet loss rate is shown in Fig. 3(b). SP exhibits the highest packet loss rate, which grows rapidly as the sum required data rate increases. In contrast, DQN and LB maintain relatively lower packet loss rates, but they are significantly affected by the system load, represented in higher packet loss rates under high required data rates. GQN consistently outperforms all other strategies, maintaining lower packet loss rates across different data transmission rates, highlighting the reliability of our approach.



(a) End-to-end delay (b) Packet loss rate (c) Link utilization

Fig. 3. The average end-to-end delay, the average packet loss rate, and the variance of link utilization under various data transmission rates.

3) Link Utilization: We define the utilization of link (i, j) as $\rho_{i,j} = \lambda_{i,j}/R_{i,j}^{\text{total}}$. To show how well load balancing is achieved under different routing strategies, we evaluated the variance of link utilization for each scheme. The variation of link utilization of all schemes is illustrated in Fig. 3(c). SP, which selects fixed paths, leads to a situation where a few shortest paths bear most of the traffic demands, resulting in uneven distribution of network loads. However, owing to the intelligent decision-making capability of DQN, it exhibits superior load-balancing performance compared to the LB scheme. Benefiting from the traffic awareness of GQN, our proposed algorithm consistently maintains the lowest variance of link utilization under different data transmission rates, demonstrating superior network load balancing performance among all schemes.

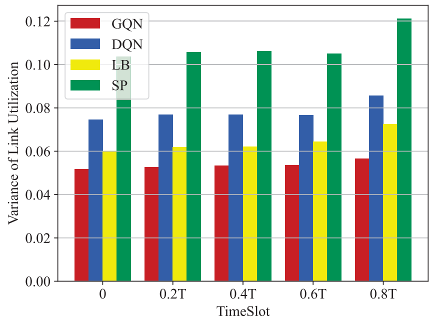


Fig. 4. Variance of link utilization under different time slots.

To assess the generalization ability of our solution, we evaluate the average performance over the satellite orbit period T and show the performance at time 0, $0.2T$, $0.4T$, $0.6T$, and $0.8T$. As depicted in Fig. 4, GQN maintains the lowest variance of link utilization across different time slots, showcasing outstanding performance in generalization capability.

V. Conclusions

In this paper, we proposed a GNN and DRL integrated approach for inter-satellite routing, which is adaptive to the evolving satellite topologies and varying traffic demands. Simulation results show that our proposed approach exhibits strong performance in reducing end-to-end delay and packet loss, as well as achieving better load balancing, demonstrating its capability to adapt to the dynamic network topology and diverse traffic patterns in the satellite network.

Acknowledgment

This work was supported in part by the NSFC under Grant 62101202.

References

[1] L. Zhao, C. Wang, K. Zhao, D. Tarchi, S. Wan, and N. Kumar, “INTERLINK: A Digital Twin-Assisted Storage Strategy for Satellite-Terrestrial Networks,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 3746-3759, Oct. 2022.

[2] N. Yarr and M. Ceriotti, “Optimization of Intersatellite Routing for Real-Time Data Download,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2356-2369, Oct. 2018.

[3] Z. Han, C. Xu, G. Zhao, S. Wang, K. Cheng, and S. Yu, “Time-Varying Topology Model for Dynamic Routing in LEO Satellite Constellation Networks,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3440–3454, Mar. 2023.

[4] J. Li, H. Lu, K. Xue, and Y. Zhang, “Temporal Netgrid Model-Based Dynamic Routing in Large-Scale Small Satellite Networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6009–6021, Jun. 2019.

[5] T. Zhang, H. Li, S. Zhang, J. Li, and H. Shen, “STAG-Based QoS Support Routing Strategy for Multiple Missions Over the Satellite Networks,” *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6912-6924, Oct. 2019.

[6] Z. Zhang, C. Jiang, S. Guo, Y. Qian, and Y. Ren, “Temporal Centrality-Balanced Traffic Management for Space Satellite Networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4427-4439, May 2018.

[7] Y. Huang, X. Jiang, S. Chen, F. Yang, and J. Yang, “Pheromone Incentivized Intelligent Multipath Traffic Scheduling Approach for LEO Satellite Networks,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5889-5902, Aug. 2022

[8] Y. Sun, M. Peng, S. Zhang, G. Lin, and P. Zhang, “Integrated Satellite-Terrestrial Networks: Architectures, Key Techniques, and Experimental Progress,” *IEEE Network*, vol. 36, no. 6, pp. 191-198, Nov. 2022

[9] S. Yuan, Y. Sun, and M. Peng, “Joint Network Function Placement and Routing Optimization in Dynamic Software-defined Satellite-Terrestrial Integrated Networks,” *IEEE Trans. Wireless Commun.*, vol. 23, pp. 5172-5186, May 2024.

[10] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural Message Passing for Quantum Chemistry,” in *Proc. ICML’17*, vol. 70, pp. 1263–1272, Aug. 2017, .

[11] V. Mnih et al., “Human-level Control through Deep Reinforcement Learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[12] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, “Deep Reinforcement Learning Meets Graph Neural Networks: Exploring A Routing Optimization Use Case,” *Computer Communications*, vol. 196, pp. 184–194, Dec. 2022.

[13] J. Y. Yen, “An Algorithm for Finding Shortest Routes from All Source Nodes to A Given Destination in General Networks,” *Quarterly of Applied Mathematics*, vol. 27, no. 4, pp. 526-530, Apr. 1970.

[14] I. K. Son, S. Mao, and S. K. Das, “On Joint Topology Design and Load Balancing in Free-Space Optical Networks,” *Optical Switching and Networking*, vol. 11, pp. 92–104, Jan. 2014.