IEEE Transactions on Vehicular Technology

A GNN-enabled Multipath Routing Algorithm for Spatial-Temporal Varying LEO Satellite Networks

Journal:	*IEEE Transactions on Vehicular Technology*
Manuscript ID	VT-2023-00888
Suggested Category:	Regular Paper
Date Submitted by the Author:	09-Mar-2023
Complete List of Authors:	Huang, Yunxue; Beijing Jiaotong University, Yang, Dong; Beijing Jiaotong University Feng, Bohao; Beijing Jiaotong University, Tian, Aleteng; Beijing Jiaotong University Dong, Ping; Beijing Jiaotong University, Yu, Shui; University of Technology Sydney Zhang, Hongke; Beijing Jiaotong University
Keywords:	GNN-enabled Multipath Routing (GMR), Link Disjoint Multipath Routing (LDMR), GNN-based MultiPath Traffic Engineering (GNN-MPTE), LEO Satellite Networks (LSNs)

A GNN-enabled Multipath Routing Algorithm for Spatial-Temporal Varying LEO Satellite Networks

Yunxue Huang, Dong Yang, *Member, IEEE*, Bohao Feng, *Member, IEEE*, Aleteng Tian,
Ping Dong, *Member, IEEE*, Shui Yu, *Fellow, IEEE*, and Hongke Zhang, *Fellow, IEEE*

***Abstract*—Low Earth Orbit (LEO) satellite networks have recently been regarded as a promising solution to provide ubiquitous user access and flexible content delivery. To meet the increasing demands of satellite services, multipath routing is leveraged to transmit each flow via multiple transmission paths concurrently. However, dynamically changed satellite topologies have brought challenges to multipath route planning and traffic splitting among different paths. Thus, in the paper, we propose a GNN-enabled Multipath Routing (GMR) scheme to maximize the network efficiency. Particularly, we firstly formulate the mutipath routing issue as a stochastic optimization problem under the bandwidth limitation and flow conservation constraint, and further present a Link Disjoint Multipath Routing (LDMR) scheme for the central route calculation, where available transmission paths and uneven traffic density are jointly considered to maximize the link utilization. Then, to balance the load distribution of different paths, a GNN-based MultiPath Traffic Engineering (GNN-MPTE) algorithm is designed for dynamic flow splitting based on estimated path quality. Finally, we implement the proposed GMR scheme and corresponding algorithms in Network Simulator NS-3 and make comparisons with other benchmarks. Simulations results demonstrate that the proposed GNN-MPTE can be extended to arbitrary inclined/polar orbit constellations without repetitive training, and significantly prompt the transmission quality including average delay, throughput, and flow completion ratio.**

***Index Terms*—GNN-enabled Multipath Routing (GMR), Link Disjoint Multipath Routing (LDMR), GNN-based MultiPath Traffic Engineering (GNN-MPTE), LEO Satellite Networks (LSNs).**

I. Introduction

Satellite networks are envisioned as a promising technique of the future 6G era, for a sake of its unique features of flexible deployment and wide coverage capacity [1], [2]. In recent years, with tremendous advancements of communication technologies and sharp reduction of satellite manufacturing cost, the space information network based on Low Earth Orbit (LEO) mega satellite constellations has attracted wide attention of industry and academic, aiming to provide ubiquitous user access and efficient content delivery [3]-[5].

To meet increasing satellite services, multipath routing is regarded as the most effective way [6], with significant advantages such as increased reliability, reduced packet drop and resilience to failure, compared with the single-path routing. However, dynamically changed topology structures in mega satellite constellations have brought challenges to multipath route planning and traffic splitting among different paths. Although many distributed solutions take consideration of congested state of neighbor nodes to obtain candidate paths, they suffer from frequent route updates and heavy computation burden [7]-[9]. To effectively reduce the pressure of satellite processing, the path calculation is centrally performed in the more powerful terrestrial controller by introducing the Software Defined Network (SDN) [10] technique. Besides, since LEO satellites are with high density Inter-Satellite Links (ISLs) and abundant available paths, planning multiple link-disjoint transmission paths for each flow concurrently is a proper way to improve network transmission performance, where the link utilization of overall network and uneven flow density derived from global population distribution are jointly considered for fair resource occupancy.

On the other hand, due to a lack of accurate traffic pattern for LEO Satellite Networks (LSNs), how to make fine-grained traffic splitting decisions between multiple paths is another challenge to be addressed. Recently, the Deep Reinforcement Learning (DRL) shows great potential in learning temporal and spatial traffic characteristics for making Traffic Engineering (TE) decisions, and there is a growing interest to leverage DRL for adaptive traffic distribution instead of static flow segmentation methods [11]-[13]. Nevertheless, DRL-based models are required to aggregate sequential topology snapshots for decision-making in such dynamic satellite environment, with high-dimension and sparse state space, where the dimensions of the model dramatically increase with the expansion of the constellation size and traffic demands, leading to poor scalability. Fortunately, the Graph Neural Network (GNN) is tailored to realize relationship inference and combinatorial generalization over graph-structured data [14], which can be applied to satellite constellations of different sizes without additional model modification. Consequently, it is possible to implement a model-free traffic allocation scheme for efficient and continuous flow control in LSNs.

To this end, in this study, we propose a GNN-enabled Multipath Routing (GMR) scheme for LSNs, aiming to significantly improve network efficiency. Particularly, considering time-varying ISL characteristics and increasing traffic volume [15], the multipath routing issue is formulated as a stochastic optimization problem to maximize the network utility under the bandwidth limitation and flow conservation constraint. To obtain the optimal path solution, the optimization problem is

This paper is supported by the National Natural Science Foundation of China under Grant No. 62271040, and Open Research Projects of Zhejiang Lab under Grant No. 2022QA0AB06. *(Corresponding author: D. Yang.)*

Y. Huang, D. Yang, B. Feng, A. Tian, P. Dong, H. Zhang are with the Institute of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China. E-mail: {20111029, dyang, bhfeng, 20111021, pdong, hkzhang}@bjtu.edu.cn.

S. Yu is with the School of Computer Science, University of Technology Sydney, NSW, 2007, Australia. E-mail: shui.yu@uts.edu.au.

firstly decomposed into two subproblems, namely the multi-path route planning and traffic splitting. Further, we present a Link Disjoint Multipath Routing (LDMR) algorithm to compute multiple reachable paths for all source and destination pairs, where abundant available paths are fully utilized. Then, a GNN-based MultiPath Traffic Engineering (GNN-MPTE) algorithm is designed to dynamically distribute appropriate traffic proportion to candidate paths based on collected link state and traffic matrices. Finally, we implement the proposed GMR scheme and corresponding algorithms in NS3 with the Shortest Path First (SPF) [47], Equal Cost Multiple Path (ECMP) [19], DRL-TE [30] and Deep Deterministic Policy Gradient-based TE (DDPG-TE) benchmarks [48] as comparisons. Extensive simulations are performed and corresponding results present that the proposed GNN-MPTE can be extended to arbitrary inclined/polar orbit constellations without retraining, and prompt the link fairness and transmission quality including the average throughput, delay, and flow completion ratio.

The contributions of this paper are summarized as follows:

- We formulate the multipath routing issue into a stochastic optimization problem, aiming to maximize the utility of overall network while satisfying both bandwidth limitation and flow conservation constraint.
- For offline route planning of the GMR scheme, a LDMR algorithm is presented to seek multiple link-disjoint paths for any source and destination pair, with the link occupancy and traffic distribution considered.
- For online traffic splitting of the GMR scheme, a GNN-MPTE algorithm is proposed to dynamically adjust the volume of subflows passing through different alternative paths, where the correlation between the topology connectivity and multiple paths is deeply integrated to handle arbitrary inclined/polar orbit constellations.

The remainder of this paper is organized as follows. Section II depicts related work of route planning, RL-based TE, and GNNs. Then, the proposed system model and formulated optimization problem are presented in the next section, followed by descriptions of the LDMR and GNN-MPTE algorithms. The last two sections show the performance evaluation and conclusion respectively. Useful notations in the paper are summarized as Table I.

II. Related Work

A. Route Planning schemes

For traditional static networks, a mass of researches have focused on improvements of routing protocols, which calculate the global routing policy based on relevant link properties. However, due to the relative movements between satellite nodes, satellite networks cannot straightforwardly adopt these approaches for the routing optimization. In [16], the authors propose a Virtual Topology (VT) strategy to partition the overall period into a number of timeslots, where the SPF algorithm is leveraged to seek optimal paths between all source and target node pairs. In [17], the authors propose a lightweight routing metric as the pathloss metric to select the ISLs with high data rate, thereby improving network transmission capacity. With the scale of satellite constellation increases, many adaptive routing algorithms are developed, where satellites dynamically exchange state packets to update paths. In [9], the authors propose a Path-quality aided and Lifetime-aware Dynamic Routing (PLDR) scheme, where each node autonomously selects next hop according to the real-time topology. In [18], the authors propose a dynamic routing mechanism for LSNs to effectively handle regular link handovers and unexpected link failure.

However, these single-path mechanisms are vulnerable to traffic congestion and resource waste, as packets sent to the same target node are transmitted to the only intermediate node even if it is with heavy load. To this end, the ECMP routing protocol is proposed for load balancing, which equally splits the end-to-end traffic and forwards sub-flows via multiple transmission paths with the same routing cost [19]. In [20], the authors raise a Network Coding-based Multipath Cooperative Routing (NCMCR) method to maximize the transmission capacity, where each flow is delivered along multiple link-disjoint paths concurrently. Nevertheless, existing multipath routing approaches take no consideration of adaptive traffic splitting ratio between different forwarding paths, leading to severe path collision between different elephant flows and sharp packet loss.

B. Reinforcement Learning based Traffic Engineering schemes

The rapid development of communication techniques and unprecedented growth of intelligent devices make a tremendous surge of network traffic volume [21]. To provide better service quality, fine-grained TE solutions are required for the path control and traffic steering. In recent years, Reinforcement Learning (RL) is envisioned as an intelligent tool to facilitate the best decision-making by direct interactions with the environment [22]. In [23], the authors propose a Reinforcement Learning and Software-Defined Networking Intelligent Routing (RSIR) scheme to obtain optimal paths according to related link state, where the Q-learning algorithm is regarded as the function approximator to assess the value of each action. However, as the size of state/action space grows, the Q-table is required to store numerous state-action pairs, leading to the memory explosion.

To this end, DRL algorithm is introduced to implement the Deep Neuron Network (DNN) [24] as the Q-function approximator, where the observations and Q values corresponding to all actions are treated as the input and output, respectively, and the best action can be obtained by maximizing the Q-function. In [25], the authors proposed a RL-based routing algorithm to address the TE issue, aiming to maximize the volume of two-way flows, where the Dueling Double Deep Q Learning (Dueling DDQN) architecture is used to decrease overestimated Q values for better learning stability. Although the DRL approach can handle the discrete action space, it cannot solve the continuous control problems because a full search of greedy policy is infeasible.

Thus, the Actor-Critic (AC) architecture is proposed to solve high-dimensional and continuous action issues [26]. In [27], the authors present a Scalable DRL-based routing architecture (ScaleDeep) to dynamically adjust link weights based on the traffic fluctuation of the key nodes, aiming to significantly reduce the average flow completion time and improve robustness. In [28], the authors propose Traffic Engineering with Explicit Path Control (EPC-TE) to determine the volume of sub-flows transmitted via different paths for better flow steering. In [29], the authors raise a DRL-based routing method for metric updating based on real link load, thereby optimizing the service quality. In [30], the authors propose a DRL-based TE algorithm to distribute any flow to pre-calculated multiple paths based on related link quality, aiming to maximize the network efficiency. However, existing DRL-based researches are coupled with the input state, and are limited to the same topology known in the training, leading to poor scalability.

C. Graph Neural Networks

GNN is designed to handle the graph-structured data, where the corresponding internal network structures are dynamically constructed to capture spatial features among all elements in the given graph [31]. Although GNN has implemented many different architectural variants, most follow the principle of an iterative message-passing process, where edges and nodes of the graph exchange their feature information with each other, and the stable output is generated via the readout phase [32]. Compared with other Neuron Network (NN) structures, the trained GNN model can be directly migrated to different size graphs, and has been applied to several domains.

In communication networks, GNN is combined with DRL for accurate decision-making in route planning and traffic control aspects. In [33], the authors propose a GNN and DRL-based architecture to learn optimal routing policy over arbitrary topologies, where the action represents the bandwidth demands allocated to each candidate path. In [34], the authors integrate the DRL and GNN to make schedule decisions of data processing jobs in data clusters without manual feature engineering. In [35], the authors propose an efficient two-staged approach to optimize the path configuration via real interactions with the environment, with the network topologies and traffic matrices dynamically changed. In [14], the authors propose the RouteNet architecture to reason the relationship among topology connectivity, routing paths, and traffic demands, thereby generating the performance metrics including delay, jitter and packet loss.

Although above approaches have shown significant improvements in network performance, they mostly focus on the single-path transmission between any source and destination pair, leading to serious link congestion and packet loss. To this end, the multipath route planning and dynamic traffic control are required to be jointly considered for flow maximization and quality of service guarantee.

III. System Model And Problem Formulation

The software-defined LEO satellite network scenario is presented for central path calculation and traffic control, followed by the traffic and route models to improve the network throughput in the considered LSN. Then, the multipath route issue is formulated as a stochastic optimization problem under corresponding constraints.

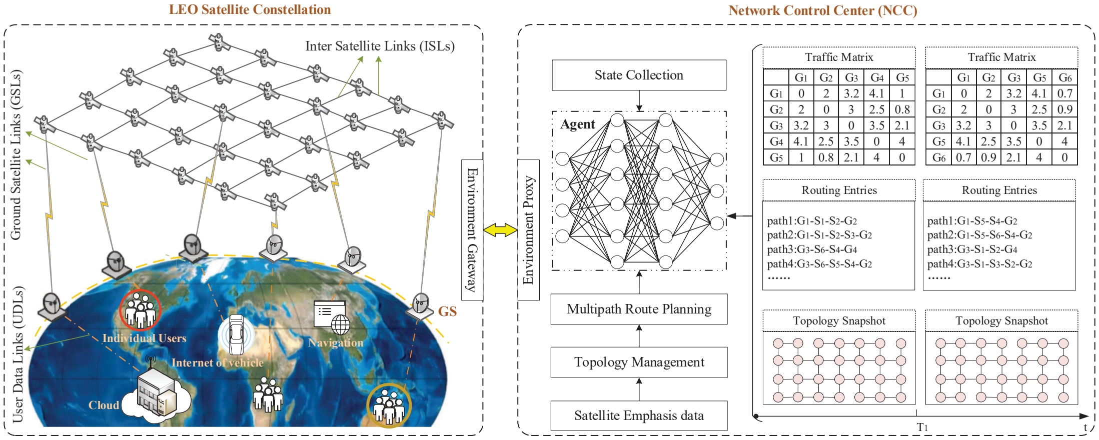
TABLE I:
PARAMETERS USED IN PAPER.

Parameter	Description
T, N	The vector and number of timeslots
$\mathcal{S}, \mathcal{G}$	Satellites and GSs
$\mathbf{V}, \mathbf{E}^{\tau_i}$	Vertexes and edges at the timeslot τ_i
R, W	The path matrix and traffic splitting policy
$p_{m,x}^{\tau_i}$	The x-th path for the m-th pair of nodes at the timeslot τ_i
$w_{m,x}^{\tau_i}$	The traffic splitting ratio for the x-th path between the m-th pair of nodes at the timeslot τ_i
M, K	The number of source-destination pairs and multiple paths
P_{τ_i}	The set of source-destination GSs
$pd_{s,v}^{\tau_i}$	The propagation delay between s and v at the timeslot τ_i
$l_{s,v}^{\tau_i}, d_{s,v}^{\tau_i}$	The inter-satellite distance and total delay at the timeslot τ_i
$UD_a, \mathrm{F}_{a,b}^{\tau_i}$	The load of the zone a, and the traffic between zone a and b
$\mathrm{F}_{\mathcal{G}_a,\mathcal{G}_b}^{\tau_i}$	The traffic volume between GSs a and b
$f_{e_j}^{\tau_i}$	The throughput through the edge e_j at the timeslot τ_i
$\bar{d}, \bar{f}$	The average throughput, delay
$\delta_{m,x}^{\tau_i}$	The packet loss rate for the x-th path between the m-th pair of nodes at the timeslot τ_i
ϱ, ϵ	on/off shape and scale

A. The Considered Scenario

A satellite multipath routing scenario is presented in Fig. 1, where the SDN technique separates the overall network into logically independent data and control planes. The data plane is formed by an LEO satellite constellation and several Ground Stations (GSs), where N_s LEO satellites can connect with their intra-orbit and inter-orbit four satellites, and N_g GSs can be configured to connect to the nearest LEO satellite at the range of their elevation angle. These underlaying switch nodes can provide diversified user access and data forwarding anytime and anywhere [36], where each user terminal can access the LSN by User Data Links (UDLs) for necessary end-to-end traffic delivery, especially for the reconstruction of disaster areas. The Network Control Center (NCC) is regarded as the central controller for the unified topology management, state collection, path control and traffic scheduling.

To enhance the network throughput and link utilization, we propose the GMR scheme for such software-defined LSN with several modules added. Particularly, the topology management module is responsible for the topology division with an equal time length. Based on the derived topology snapshot, the multipath route planning module computes multiple available paths for all source and destination pairs. The environment proxy and gateway are responsible for message serialization and de-serialization between different programming languages. By interaction with each other, they are capable of forwarding the observation state and action to the agent and LSN, respectively. Based on newly received traffic matrix and link state, the agent generates the traffic splitting decision of different paths. Consequently, when all LEO satellites and GSs have achieved



Traffic Matrix

	G1	G2	G3	G4	G5
G1	0	2	3.2	4.1	1
G2	2	0	3	2.5	0.8
G3	3.2	3	0	3.5	2.1
G4	4.1	2.5	3.5	0	4
G5	1	0.8	2.1	4	0

Traffic Matrix

	G1	G2	G3	G5	G6
G1	0	2	3.2	4.1	0.7
G2	2	0	3	2.5	0.9
G3	3.2	3	0	3.5	2.1
G5	4.1	2.5	3.5	0	4
G6	0.7	0.9	2.1	4	0

Routing Entries

path1:G1-S1-S2-G2
path2:G1-S1-S2-S3-G2
path3:G3-S6-S4-G4
path4:G3-S6-S5-S4-G2
……

Routing Entries

path1:G1-S5-S4-G2
path2:G1-S5-S6-S4-G2
path3:G3-S1-S2-G4
path4:G3-S1-S3-S2-G2
……

Fig. 1: The GNN-enable Multipath Routing scheme for software-defined LEO satellite networks.

success networking, the NCC is able to dynamically update corresponding path and traffic configurations with a global view of real network state, which are also informed to corresponding switch nodes to locally load via the control paths. After receiving user requests, these access nodes decapsulate data packets and perform efficient message distribution based on newly received control policies. When the network state changes, the agent dynamically updates the traffic splitting ratio between all candidate paths.

B. The Traffic Model

Since each LEO satellite provides random accesses for Internet users within its coverage, the traffic requirement between any two relay satellites depends on the population distribution of their coverage areas. However, with the periodical movement of satellites, inter-satellite traffic suffers from the dynamical variation. To this end, the surface of earth is equally partitioned into Z logical zones via the Virtual Node (VN) method [37], which are regarded as virtual nodes. At any timeslot, the virtual node can be mapped to the nearest satellite for the traffic transmission. When the former satellite moves away, the traffic is switched to the successor. According to [38], the traffic between any two logical zones can be denoted by Eq. (1),

$$f_{a,b}^{\tau_i} = \mathcal{F} \cdot \frac{(UD_a \cdot UD_b)^{\alpha} / l_{a,b}^{\beta}}{\sum\limits_{a} \sum\limits_{b} (\mathcal{F} \cdot \frac{(UD_a \cdot UD_b)^{\alpha}}{l_{a,b}^{\beta}})}, \forall a, b \in [1, Z] \tag{1}$$

where $\mathcal{F}$ is the total network traffic, θ and φ denote the experience factor, $l_{a,b}$ refers to the center distance between any two zones, UD_a and UD_b represent the network user density of the zone a and b, respectively.

To generate data traffic, N_g populous cities are selected as source and destination nodes, where each logical zone only deploys a GS to conveniently forward packets generated by their located zones [39]. Thus, the traffic between two GSs $\mathcal{G}_i$ and $\mathcal{G}_j$ equals the traffic between their located zones a and b, i.e., $F_{\mathcal{G}_i,\mathcal{G}_j}^{\tau_i} = F_{a,b}^{\tau_i}$. Since not all GSs can directly communicate with LEO satellites in each snapshot, assume that there are M pairs of sources and destinations, which is denoted by $P_{\tau_i} = \{(s_i^g, d_i^g) | i = 1, ..., M\}$. Recent studies have adopted nonpersistent on-off flow to fit user request process. The on/off periods of flows determine average burst time and idle time between two independent subflows, and they all follow the Pareto distribution [40] with the on/off shape ϱ and scale parameters ϵ. Let t denotes the time of on-time or off-time for any flow. Thus, the arrival probability of traffic $prob(t)$ is defined as Eq. (2).

$$prob(t) = \begin{cases} 0, & \text{if t} < \epsilon; \\ \frac{\varrho \epsilon^{\varrho}}{\tau^{\varrho+1}}, & \text{otherwise.} \end{cases} \tag{2}$$

C. The Route Model

The LSN is modeled as a spatial-temporal graph $\mathbb{G} = \{(\mathbf{V}, \mathbf{E}, \mathrm{T}) | \forall i \in [1, N]\}$, where $\mathbf{V} = \mathcal{S} \bigcup \mathcal{G}$ is formed by all LEO satellites and GSs, $\mathbf{E} = \mathbf{E}_S \cup \mathbf{E}_G$ is a dynamically changed link set with satellite movements, T is a vector composed of several timeslots with a fixed span of T_0/h. During the timeslot τ_i, the number of links is denoted by $|\mathbf{E}^{\tau_i}|$. Since there are abundant available paths between any two satellites, K paths between any node pair can be obtained for efficient traffic dispersion during each topology snapshot, which can be denoted by $\mathrm{R} = \{\mathrm{R}_{\tau_i} | \tau_i \epsilon \mathrm{T}\} = \{p_{1,1}^{\tau_i}, ..., p_{1,K}^{\tau_i}, ..., p_{M,K}^{\tau_i}\}$. Here, $p_{m,x}^{\tau_i} = \{s_m^g, s_m^s, s_j, ..., d_m^s, d_m^g\}$ denotes the x-th path between the m-th source and destination nodes, where s_m^g, d_m^g represent the source and destination GSs, and s_m^s, s_m^s denote their access satellites. To depict the correlation between the path and corresponding links, the path is replaced with $p_{m,x}^{\tau_i} = \{e_{j1}^{\tau_i}, e_{j2}^{\tau_i}, ..., e_{j_{n_{m,x}}}^{\tau_i}\}$, where $n_{m,x}$ is the path length. Let $qd_{e_j}^{\tau_i}$ represents the queuing delay depending on the finite queue length L_q of each node and np represents the packet processing delay of each node. Based on the Two-Line Element (TLE) [41] ephemeris data, the position of the satellite s and v at the timeslot τ_i in a cartesian coordinates system are denoted by $(x_s^{\tau_i}, y_s^{\tau_i}, z_s^{\tau_i})$ and $(x_v^{\tau_i}, y_v^{\tau_i}, z_v^{\tau_i})$ respectively, and thus their distance can be derived by Eq. (3).

$$l_{s,v}^{\tau_i} = \sqrt[2]{(x_s^{\tau_i} - x_v^{\tau_i})^2 + (y_s^{\tau_i} - y_v^{\tau_i})^2 + (z_s^{\tau_i} - z_v^{\tau_i})^2}, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{S}, s \neq v \tag{3}$$

Thus, the propagation delay between the satellite s and v at the i-th timeslot can be denoted by Eq. (4),

$$pd_{s,v}^{\tau_i} = l_{s,v}^{\tau_i}/C \tag{4}$$

where C is the light speed in vacuum. Thus, the transmission delay of the x-th path for the m-th source and destination pairs at the timeslot τ_i can be denoted by Eq. (5).

$$d_{m,x}^{\tau_i} = \sum_{j=1}^{n_{m,x}} pd_{e_j}^{\tau_i} + qd_j^{\tau_i} + np \tag{5}$$

Based on bandwidth resource of related paths, each source node can split the flow into K sub-flows passing through K paths concurrently, and the sum of these subflows is required to satisfy the flow conservation constraint Eq. (6),

$$\sum_{x=1}^{K} w_{m,x}^{\tau_i} = 1, \forall m = 1, ..., M \tag{6}$$

where $w_{m,x}^{\tau_i}$ denotes the traffic ratio of the subflow between the m-th node pair passing through the x-th path. The traffic splitting solutions for all subflows can be denoted by $\mathrm{W} = \{w_{1,1}, ..., w_{1,K}, ..., w_{m,K}\}$. As a result, the amount of traffic along the link e_j at the timeslot τ_i is denoted by Eq. (7),

$$f_{e_j}^{\tau_i} = \sum_{m=1}^{M} \sum_{x=1}^{K} \sum_{l=1}^{n} g(p_l, e_j) b_m^{\tau_i} w_{m,x}^{\tau_i} \delta_{m,x}^{\tau_i} \tag{7}$$

where $b_m^{\tau_i}$ denotes the bandwidth demand of the m-th node pairs, $\delta_{m,x}$ is the packet drop ratio of the x-th path, and $g(p_l, e_j) = 1$ represents the path p_l passes through the link e_j, otherwise, $g(p_l, e_j) = 0$. Let BW denote the maximum bandwidth of ISLs. Therefore, the occupancy capacity of each ISL should not exceed the bandwidth limitation Eq. (8).

$$f_{e_j}^{\tau_i} \le BW, \forall e_j \in p_{m,k}^{\tau_i}, e_j \in \mathbf{E}_S \tag{8}$$

Additionally, the throughput between the m pair of nodes can be denoted by Eq. (9).

$$f_{m,k}^{\tau_i} = max\{f_{e_j}^{\tau_i}\}, \forall e_j \in p_{m,k}^{\tau_i} \tag{9}$$

D. Problem Formulation

Based on the $\alpha-$fairness model [30], the utility function for LSNs is redefined as Eq. (10) to evaluate the overall network efficiency.

$$U(\bar{f}_{\tau_i}, \bar{d}_{\tau_i}) = \sigma_1 \log(\bar{f}_{\tau_i}) - \sigma_2 \log(\bar{d}_{\tau_i}) \tag{10}$$

Here, σ_1 and σ_2 represent the related importance of the average throughput and delay, $\bar{f}_{\tau_i}$ and $\bar{d}_{\tau_i}$ are average throughput and delay at current timeslot, which can be denoted by Eq. (11) and Eq. (12),

$$\bar{f}_{\tau_i} = \sum_{m=1}^{M} \sum_{k=1}^{|f_m|} \sum_{y=1}^{|p_k|} \varsigma_y \nabla_{packet}/\tau_i, \forall i \in [1, h] \tag{11}$$

$$\bar{d}_{\tau_i} = \sum_{m=1}^{M} \sum_{k=1}^{|f_m|} \sum_{y=1}^{|p_k|} d_{m,k}^{\tau_i} \varsigma_y / \sum_{m=1}^{M} \sum_{k=1}^{|f_m|} \sum_{y=1}^{|p_k|} \varsigma_y, \forall i \in [1, h] \tag{12}$$

where $|f_m|$ represents the flow amount between the m-th node pair, $|p_k|$ depicts the count of packets for the k-th flow, ∇_{packet} is the byte volume of each packet, and $\varsigma_y = 0, 1$ is a binary variable to measure whether the y-th packet is successfully delivered. The multipath routing optimization problem is formulated as follows.

$$\begin{aligned} P0: \quad & \max_{R,W} U(\bar{f}, \bar{d}), \forall \tau_i \in \mathrm{T} \\ s.t. \quad & C1: \bar{d} = \frac{1}{N} \sum_{i=1}^{N} \bar{d}_{\tau_i} \\ & C2: \bar{f} = \frac{1}{N} \sum_{i=1}^{N} \bar{f}_{\tau_i} \\ & C3: f_{e_j}^{\tau_i} \le BW, \forall e_j \in p_{m,x}^{\tau_i}, e_j \in \mathbf{E}_S \\ & C4: \sum_{x=1}^{K} w_{m,x}^{\tau_i} = 1, \forall m \in [1, M] \\ & C5: d_{m,x}^{\tau_i} \le d_{m,y}^{\tau_i}, \forall 1 \le x \le y \le K, \forall m \in [1, M] \end{aligned} \tag{13}$$

Here, C1 and C2 denote the average delay and throughput in the LSN respectively, C3 represents that the throughput for the link e_j at the timeslot τ_i cannot exceed the bandwidth constraint, C4 depicts that the total traffic amount between any node pair equals the sum of subflows along mulitple paths, C5 ensures the delay constraints for all candidate paths between each node pair. Considering the spatial-temporal changed traffic requests and network topologies, the problem $P0$ is regarded as a stochastic optimization problem.

IV. Multipath Route Optimization Algorithm

The multipath route optimization problem is further decomposed into two subproblems in this section, namely multipath route planning and traffic splitting. To this end, we propose the LDMR algorithm to compute multiple paths with the acceptable latency, and the GNN-MPTE algorithm to generate the traffic splitting ratio based on collected network state, aiming to enhance the network utility.

A. The Multipath Route Planning Subproblem

The multipath route planning problem can be formulated as Eq. (14), aiming to seek the shortest-delay path and sub-optimal delay path.

$$\begin{aligned} P1: \quad & \min_{R} d_{m,x}^{\tau_i}, \forall \tau_i \in \mathcal{T}, \forall m \in [1, M], \forall x \in [1, K] \\ s.t. \quad & d_{m,x}^{\tau_i} \le d_{m,y}^{\tau_i}, \forall 1 \le x \le y \le K \end{aligned} \tag{14}$$

As given in Algorithm 1, the operation process of the proposed LDMR algorithm is described as follows. To guarantee optimal delay of candidate paths, each element of the weight matrix is initialized as link delay, the path set R and the occupancy frequency of all edges are reset as empty set and zero, respectively. Based on initial weight setting, the NCC firstly calculates the shortest-delay path for any node pair via the Dijkstra Shortest Path (DSP) method.

Based on updated random weight matrix and topology connectivity, the NCC further searches other sub-optimal delay transmission paths for all traffic requests in the descending order of the flow volume, as elephant flows with higher throughput are more likely to result in congestion in bottleneck

Algorithm 1: The Link-Disjoint Multipath Routing (LDMR)

```
   Input: the graph $\mathbb{G}$, the vector of source-destination
          pairs $P$, upper and lower bound of the random
          weight $r_1$, $r_2$, $r_3$
   Output: the vector of candidate paths $\mathbf{R}$
 1 Initialization: $\mathbf{M}$, $\mathbf{R} = \varnothing$, $U_e^{\tau_i} = 0, \forall e \in \mathbf{E}, \forall \tau_i \in \mathrm{T}$;
 2 for $i = 1 : N$ do
 3     ▷ Compute the shortest-delay path $\mathbf{R}^{\tau_i}$ via DSP;
 4     ▷ Compute other candidate paths;
 5     for $(s, d)$ in $P_{sort}^{\tau_i}$ do
 6         for $x = 1 : K - 1$ do
 7             ▷ Update the occupancy frequency of all
               edges;
 8             for $(a, b)$ in $\boldsymbol{R}^{\tau_i}$ do
 9                 $U_{a,b}^{\tau_i} \leftarrow U_{a,b}^{\tau_i} + 1$;
10             end
11             ▷ Update the link metric matrix $\mathbf{M}^{\tau_i}$;
12             for $(a, b)$ in $\boldsymbol{E}^{\tau_i}$ do
13                 if $U_{a,b}^{\tau_i} < N_{th}^{e}$ then
14                     $m_{a,b}^{\tau_i} \leftarrow rand(r_1, r_2)$;
15                 else
16                     $m_{a,b}^{\tau_i} \leftarrow rand(r_2, r_3)$;
17                 end
18             end
19             for $e$ in $\boldsymbol{R}_{s,d}^{\tau_i}$ do
20                 Remove the edge from $\mathbf{E}^{\tau_i}$;
21             end
22             Compute the feasible path $\mathbf{R}_{s,d}^{\tau_i}$ via DSP;
23         end
24     end
25 end
```

links, compared with micro flows. Take the backup path calculation between the node s and d for example, the NCC updates the link occupancy frequency based on Eq. (15) and the random weight matrix derived from pre-defined upper and lower bound,

$$U_{a,b}^{\tau_i} \leftarrow U_{a,b}^{\tau_i} + 1, (a, b) \in R^{\tau_i} \tag{15}$$

where $U_{a,b}^{\tau_i}$ represents the occupancy number of the link between the node a and b at the timeslot τ_i and $(a, b) \in R^{\tau_i}$ depicts current link appears the set of shortest paths R^{τ_i}. Lines 13-19 show the updating process of the weight matrix based on pre-defined upper and lower bound. Let N_{th}^{e} denotes the threshold of the link utilization frequency. If the usage frequency $U_{a,b}^{\tau_i}$ for any link is less than the threshold N_{th}^{e}, the NCC randomly assigns an integer between r_1 and r_2 as the new link metric $m_{a,b}^{\tau_i}$, otherwise, the link metric is set as a random integer between r_2 and r_3. Besides, the NCC removes some links from the topology, which exist in the shortest-delay path $R_{s,d}^{\tau_i}$, aiming to avoid link conflict among different paths. Afterwards, the NCC computes a feasible path for current node pair (s, d) via the DSP method with idle links fully used. Similarly, repeat above process until K paths between all source and destination nodes have been obtained.

B. The Traffic Splitting Subproblem

Based on the certain routing decision, the problem $P0$ can be transformed into a traffic splitting problem among multiple paths, which can be formulated as Eq. (16), aiming to maximize the network utility.

$$\begin{aligned} P2: \quad & \max_{W} U(\bar{f}, \bar{d}), \forall \tau_i \in \mathrm{T} \\ s.t. \quad & Eq.(6), (8) \end{aligned} \tag{16}$$

Since the traffic splitting decision lies on current network state and traffic demands without requiring historical experience, the problem $P2$ can be transformed as a Markov Decision Process (MDP). As presented in Fig. 2, a GNN-MPTE framework formed by the environment and a GNN-based DRL agent is introduced to address the MDP, where the LSN is regarded as the environment, the agent deployed on the NCC generates the traffic splitting policy based on given state, and the GNN is leveraged as the function approximator for handling dynamic topologies. At each step of the MDP, the agent generates a traffic splitting decision based on newly collected state and receives immediate reward and latest state via interactions with current environment. Throughout the recurrent training, the agent optimizes the decision-making task, with the objective of maximizing the cumulative reward. The detailed descriptions of state, action, and reward are presented below.

1) State: During each topology snapshot, the NCC periodically collects the traffic matrix of all traffic requests $\mathbf{TM}_t = [b_1, ..., b_i, ..., b_M]$, and the throughput of all links $\mathbf{F}_t = [f_1, f_2, ..., f_{|\mathbf{E}_t|}]$, where b_i is the bandwidth demand between the i-th pair of nodes, and f_j is the occupied capacity of the j-th links. Thus, the remaining bandwidth of any link can be denoted by Eq. (17),

$$c_i = B - f_i, i \in [1, |\mathbf{E}_t|] \tag{17}$$

and the available capacity of all links are denoted by $\mathbf{C}_t = [c_1, c_2, ..., c_{|\mathbf{E}_t|}]$. Besides, multiple paths between all pairs of source and destination nodes derived by the LDMR algorithm and topology connectivity at current snapshot are also informed to the agent, aiming to evaluate the quality level of different paths. Let p_{max} represents the maximum length of paths between all node pairs. Thus, the j-th path between the i-th pair of nodes can be denoted by a set of links with the equal length, which can be expressed as $p_{i,j} = \{e_a, e_b, ..., e_{p_{max}}\}$. If the length of any path is less than the maximum, its path can be extended by zero-padding. Accordingly, the path matrix can be denoted by $\mathbf{P}_t = [p_{1,1}, ..., p_{1,K}, ..., p_{M,K}]$ with the size of $(M \times K) \times p_{max}$. Consequently, the state vector at the t-th step can be denoted by Eq. (18), where S represents the state space of the considered LSN.

$$s_t = [\mathbf{C}_t, \mathbf{TM}_t, \mathbb{G}_t, \mathbf{P}_t], s_t \in S \tag{18}$$

2) Action: Based on above observation state, the agent centrally makes the traffic splitting decision to disperse the traffic between any source and destination pair to pre-computed multiple paths. Thus, the action is defined as a vector formed by the flow splitting ratio of K candidate paths between all node pairs, which can be denoted by Eq. (19),

$$\mathbf{a}_t = [w_{1,1}^{t}, ..., w_{1,K}^{t}, ..., w_{M,K}^{t}] \tag{19}$$

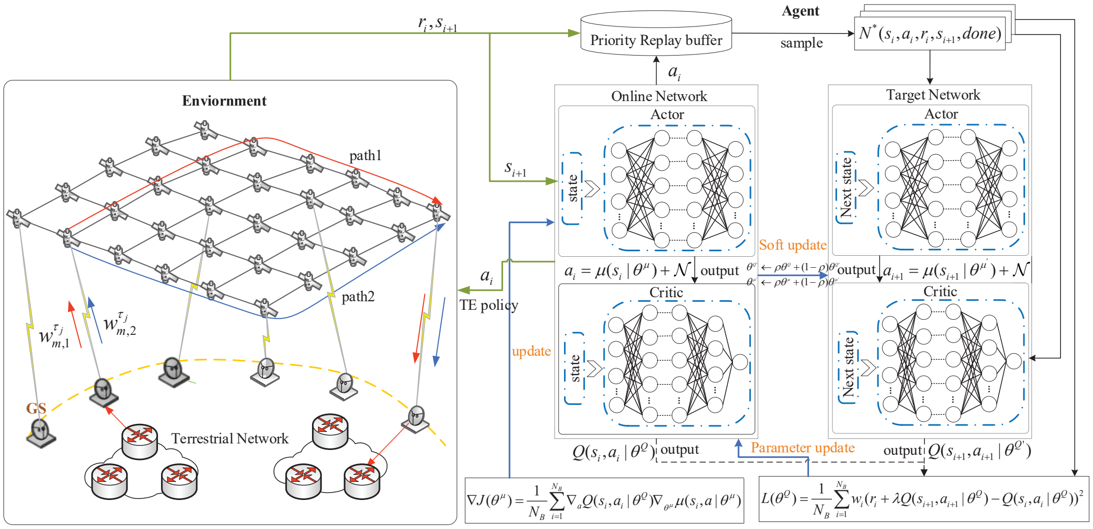

Agent
Priority Replay buffer
sample
r_i, s_{i+1}
$N^*(s_i,a_i,r_i,s_{i+1},done)$
a_i
Enviornment
path1
path2
$w_{m,1}^{\tau_j}$
$w_{m,2}^{\tau_j}$
GS
Terrestrial Network
s_{i+1}
Online Network
Actor
state
Target Network
Next state
Soft update
$a_i = \mu(s_i|\theta^\mu)+\mathcal{N}$
output
$a_{i+1} = \mu(s_{i+1}|\theta^\mu)+\mathcal{N}$
$\leftarrow \rho\theta^\mu + (1-\rho)\theta^{\mu'}$
$\leftarrow \rho\theta^Q + (1-\rho)\theta^{Q'}$
TE policy
update
Critic
$Q(s_i,a_i|\theta^Q)$
Parameter update
$Q(s_{i+1},a_{i+1}|\theta^{Q'})$
$\nabla J(\theta^\mu) = \frac{1}{N_B}\sum_{i=1}^{N_B}\nabla_a Q(s_i,a_i|\theta^Q)\nabla_{\theta^\mu}\mu(s_i,a|\theta^\mu)$
$L(\theta^Q) = \frac{1}{N_B}\sum_{i=1}^{N_B} w_i(r_i+\lambda Q(s_{i+1},a_{i+1}|\theta^Q)-Q(s_i,a_i|\theta^Q))^2$

Fig. 2: The framework and workflow of the proposed GNN-MPTE algorithm.

where the traffic splitting ratio among K paths for any a pair of source and destination nodes should satisfy Eq. (6). Consequently, the size of the action space is expressed as $1 \times (K \times M)$.

3) Reward: After execution of the action a_t, the agent can obtain a delayed feedback r_t to assess the effectiveness of the given action under the state s_t. In this paper, the immediate reward is defined as Eq. (20),

$$r_t = \sum_{m=1}^{M}\sum_{l=1}^{K} U(x_{m,l}, y_{m,l}) \tag{20}$$

where $x_{m,l}$ and $y_{m,l}$ represent average transmission delay and throughput for the l-th path between the m pair of nodes, separately. Consequently, the agent will learn the traffic configuration policy between multiple paths based on the preference of the lower delay and higher throughput.

The cumulative reward depicts the potential values until the overall game ends, which can be denoted by Eq. (21).

$$G_t = \sum_{k=0}^{\infty} \lambda^k r_{t+k+1} \tag{21}$$

Here, $\lambda \in [0, 1]$ is the discount factor to balance the instant reward and future reward. Since there are several possible actions in a given state, the average long-term gains of each state and state-action pair can be denoted by Eq. (22) and Eq. (23), respectively.

$$V^\mu(s) = \mathbb{E}_\mu(G_t|S_t = s) \tag{22}$$

$$Q^\mu(s,a) = \mathbb{E}_\mu(G_t|S_t = s, a_t = a) \tag{23}$$

The target of the DRL agent is to obtain the optimal policy μ^* to maximize the action-value function. To guide the decision-making process of the agent, the updating process of the action-value function can be denoted by Eq. (24) in the Bellman form. When the DRL model has converged to a stable level, the optimal action can be denoted by Eq. (25).

$$Q^\mu(s,a) = \mathbb{E}_\mu[r_t + \lambda Q^\mu(s_{t+1}, a_{t+1})|S_t = s, a_t = a]) \tag{24}$$

$$a^{\mu^*} = \underset{a}{argmax}\, Q^{\mu^*}(s,a) \tag{25}$$

C. The Training Process of the GNN-MPTE Algorithm

Algorithm 2 presents the sequential training procedure of the proposed GNN-MPTE. In the initialization stage, the online actor network $\mu(s|\theta^\mu)$ and the critic network $Q(s,a|\theta^Q)$ are initialized, where θ^μ and θ^Q denote the hyper-parameters of both networks respectively, and their target networks are similarly expressed as $\mu^{'}(s|\theta^{\mu^{'}})$ and $Q^{'}(s,a|\theta^{Q^{'}})$ by coping the same parameters with the online networks. The Replay buffer B is set as a empty vector with the size of N_B. During each episode, the environment is reset as the initialized state s_0. The proposed algorithm adopts the experience generation and policy learning two steps to train the neural networks.

When given the observed state s_t, the action can be obtained based on Eq. (26),

$$a_t = \mu(s_t|\theta^\mu) + \mathcal{N} \tag{26}$$

where $\mathcal{N}$ represents the added Gaussian noise for higher exploration efficiency. Then, the agent receives the instant reward and successor state after interacting with the environment. Before the training step, each sample $(s_t, a_t, r_t, s_{t+1}, done)$ is treated as the maximal priority p_{max}, and stored into the replay buffer, where $done \in \{0, 1\}$ denotes whether an episode of the model training has been finished.

When the number of experience stored in the replay buffer $|B|$ exceeds the pre-defined threshold B_{max}, a mini-bath of transitions $(s_i, a_i, r_i, s_{i+1}, done)$ are sampled from the replay buffer for the model training. Based on the state s_{i+1}, a ceratin action a_{i+1} can be obtained via the target actor network. Consequently, the target Q value can be expressed as Eq. (27) via the target critic network.

$$y_i = r_i + \lambda \cdot (1 - done) \cdot [Q(s_{i+1}, a_{i+1}|\theta^Q)] \tag{27}$$

Similarly, the online critic network returns current estimation value $Q(s_i, a_i)$ for the state-action pair (s_i, a_i). As a result, the Temporal-Difference (TD) error between the target and current Q values can be denoted by Eq. (28).

$$\delta_i = y_i - Q(s_i, a_i|\theta^Q) \tag{28}$$

Algorithm 2: The Training Process of the GNN-MPTE Algorithm

```
Initialization: online critic and actor networks, both target networks, and replay buffer $B$;
for $episode = 1 : L$ do
    Reset the initial observation state as $s_0$;
    for $step\ t = 1 : T$ do
        Compute the action $a_t$ according to Eq. (26);
        Receive the immediate reward $r_t$ and new state $s_{t+1}$;
        Store $(s_t, a_t, r_t, s_{t+1}, done)$ into the replay buffer with the maximal priority $p_{max}$;
        Update the observe state $s_t \leftarrow s_{t+1}$;
        if $|B| > B_{max}$ then
            for $i = 1 : N_B$ do
                Sample a transition $(s_i, a_i, r_i, s_{i+1}, done)$ from the replay buffer according to Eq. (29);
                Calculate the important-sampling weight via Eq. (31);
                Derive target value $y_i$ from the target critic network according to Eq. (27);
                Derive the Q value $Q(s_i, a_i)$ from the critic network;
                Update the probability of the sampling transition via Eq. (30);
            end
            Calculate the loss for the online critic network via Eq. (32);
            Update the parameter for the online critic network via Eq. (33);
            Calculate the loss for the online actor network via Eq. (34);
            Update the parameter for the online actor network via Eq. (35);
            Update the weight parameters for target actor and critic networks based on Eq. (36) and Eq. (37), respectively;
        end
    end
end
```

Based on the absolute error, the priority of each transition sample can be updated by Eq. (29),

$$p_i =| \delta | + \xi \tag{29}$$

where ξ is a small positive value to prevent some experience with the priority of zero not being visited. Accordingly, the probability of the sampling transition is updated via Eq. (30),

$$prob(i) = \frac{p_i^\alpha}{\Sigma_{i=k} p_k^\alpha} \tag{30}$$

where α controls the importance of the experience. Afterwards, the important-sampling weight of each sample is updated via Eq. (31), where β represents a coefficient.

$$\omega_i = (\frac{prob(i)}{prob_{min}})^{-\beta} \tag{31}$$

Algorithm 3: The Message Passing Process of the GNN-MPTE algorithm

```
Input: the graph $\{(\mathcal{S}, \mathbf{E}, \mathbf{C})\}$, the traffic matrix and a set of candidate paths
Output: the vector of the path quality
Initialization: $\mathbf{h}_{e_i}^t$,$\mathbf{h}_{p_x}^t$,$\mathbf{P}$;
for $t = 1 : h$ do
    for $p_x$ in $\boldsymbol{P}$ do
        for $e_i$ in $p_x$ do
            $\mathbf{m}_{p_x}^{t+1} \leftarrow M(\mathbf{h}_{p_x}^t, \mathbf{h}_{e_i}^t)$;
        end
    end
    $\mathbf{h}_p^{t+1} \leftarrow U(\mathbf{h}_p^t, \mathbf{m}_p^{t+1})$;
    for $e_i$ in $\boldsymbol{E}$ do
        for $p_x$ in $\boldsymbol{P}$ do
            if $e_i$ in $p_x$ then
                $\mathbf{m}_{e_i}^{t+1} \leftarrow \mathbf{m}_{e_i}^t \oplus \mathbf{m}_{p_x}^t$;
            end
        end
        $\mathbf{h}_{e_i}^{t+1} \leftarrow U(\mathbf{h}_{e_i}^t, \mathbf{m}_{e_i}^{t+1})$;
    end
end
$\hat{y} \leftarrow R(\mathbf{h}_p^t)$;
```

As a result, the loss of the online critic network can be denoted by Eq. (32).

$$L_q = 1/N_B \cdot \sum_{i=1}^{N_B} \omega_i \cdot \delta_i^2 \tag{32}$$

With the help of the gradient descending approach, corresponding parameter for the online critic network θ^Q can be updated by Eq. (33),

$$\theta^Q \leftarrow \theta^Q + \eta^Q \cdot \delta_i \nabla_{\theta^Q} Q(s, a|\theta^Q) \tag{33}$$

where η^Q is the learning rate, and $\nabla_{\theta^Q} Q(s, a|\theta^Q)$ represents the gradient of the Q function. Since the critic network guides the policy preference of the actor network, the loss of current actor network can be denoted by Eq. (34).

$$L_\mu = -Q(s, a|\theta^Q) \tag{34}$$

According to the chain derivative rule, the parameter updating for the online actor network can be denoted by Eq. (35), where η^μ represents the learning rate.

$$\begin{aligned} &\theta^\mu \leftarrow \theta^\mu + \eta^\mu \cdot \nabla_{\theta^\mu} J, \\ &\approx \theta^\mu + \eta^\mu \cdot \mathbb{E}[\nabla_{\theta^\mu} Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t, \theta^\mu)}], \\ &= \theta^\mu + \eta^\mu \cdot \mathbb{E}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=a_t} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s_t}] \end{aligned} \tag{35}$$

The corresponding parameters for the target actor and critic networks are soft-updated by Eq. (36) and Eq. (37), where ρ is the learning rate.

$$\theta^{\mu^{'}} \leftarrow \rho\theta^\mu + (1-\rho)\theta^{\mu^{'}} \tag{36}$$

$$\theta^{Q^{'}} \leftarrow \rho\theta^Q + (1-\rho)\theta^{Q^{'}} \tag{37}$$

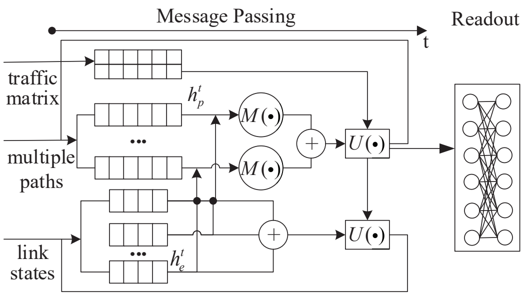


Fig. 3: The MPNN architecture.

D. The Message Passing Architecture

In our proposed DRL agent, the Message Passing Neuron Network (MPNN) is leveraged to directly handle graph-structured topologies with variable sizes, and output the end-to-end path performance estimates. Algorithm 3 presents the message passing process of the proposed GNN model, formed by the initialization, message passing and readout phases.

During the initialization stage, the input observation state can be transformed as related feature vectors respectively, where the hidden state of each edge is obtained by edge embedding with available capacity C_e, each end-to-end path is encoded as a vector of ordered edges, and each element of the traffic matrix represents the bandwidth demand of any source and destination pair. Here, the dimensions of these hidden state vectors are all configurable parameters of the GNN model.

Then, the algorithm executes h iterative message passing steps as shown in Fig. 3, where each link and path exchange information with each other by the message function $M(\cdot)$ and update function $U(\cdot)$, until they reach their respective stable states. The message function is used to aggregate the information of each forwarding path between any node pair and the hidden state of all corresponding links on current path to generate a new message $m_{p_x}^{t+1}$, which can be denoted by Eq. (38),

$$\mathbf{m}_{p_x}^{t+1} = \sum_{e_i \in p_x} M_t(\mathbf{h}_{p_x}^t, \mathbf{h}_{e_i}^t) \tag{38}$$

where h_{e_i} represents the hidden state of the edge e_i, and $\mathbf{h}_{p_x}^t$ is the hidden state of the path p_x. Based on these hidden states, the messages for all routing paths and all links are aggregated via an element-wise sum. Afterwards, the update function leverages the Gated Recurrent Unit (GRU) to capture sequential behavior for the hidden states of links and routing paths. In particular, the hidden state for the x-th path can be updated by Eq. (39) based on the aggregated message and encoded traffic demands.

$$\mathbf{h}_p^{t+1} = GRU(\mathbf{h}_p^t, \mathbf{m}_{p_x}^{t+1}) \tag{39}$$

Similarly, based on the corresponding path-level message, the link state can be updated by Eq. (40).

$$\mathbf{h}_e^{t+1} = GRU(\mathbf{h}_e^t, \sum_{k \in N_p} \mathbf{m}_{p_x}^{t+1}) \tag{40}$$

Consequently, the message of each link will include the states of neighbor links and corresponding paths passing through current link, and the message of each transmission path also gathers the states of corresponding links formed by this path and other backup paths. Finally, the readout function outputs the quality of each path by implementing a fully-connected DNN, which can be denoted by Eq. (41).

$$\hat{y} = R(\mathbf{h}_{p_x}^t), x = 1, ..., K \times M \tag{41}$$

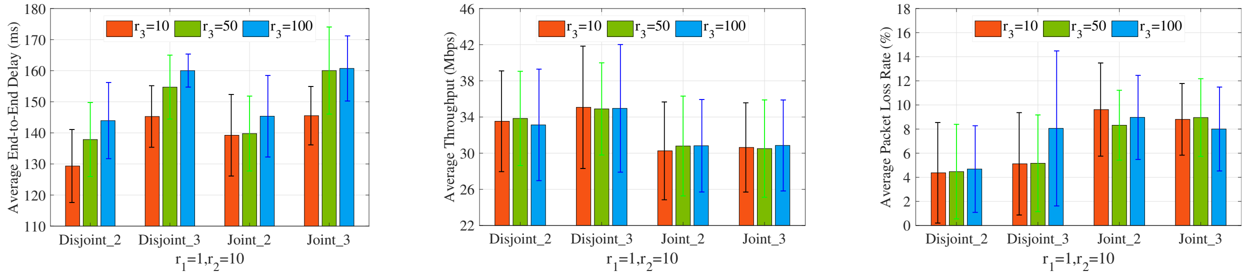
E. The Workflow of the Proposed GMR Scheme

The workflow of the proposed GMR scheme contains two stages, namely offline multipath route planning and online traffic splitting between different transmission paths. At the offline phase, the NCC pre-calculates the optimal path solution for each elephant flow and micro flow based on collected link delay and topology connectivity. Specifically, the time-varying network is divided into a series of discrete topology snapshots via the topology management model, as the trajectory of satellites is periodical and predictable. During each timeslot, in descending order of end-to-end traffic volume, the multipath route planning model pre-calculates K link-disjoint available paths for all traffic requests, and then distributes the forwarding paths to corresponding switch nodes before switching to the next topology snapshot.

During the online phase, the NCC regularly adjusts the volume of subflows passing through different paths, based on the collected traffic matrix and remaining link bandwidth derived from Eq. (17). Firstly, the state collection module deployed in each switch node periodically monitors the link transmission throughput, path occupancy and traffic requirements in such LSNs, which are sent to the agent for the intelligent resource scheduling. Then, the agent generates the traffic splitting ratio between pre-calculated candidate paths and informs the related source access satellites to adjust the traffic transmission rate. Finally, when receiving any traffic request, switch nodes locally load corresponding paths and traffic splitting configurations for efficient data delivery.

In addition, some key techniques are leveraged for the system optimization as follows. The actor and critic networks are applied to the continuous traffic schedule, where the former is leverage to directly map the observed state to the action, the latter is regarded as a function approximator to assess the value of any action-state pair. Besides, the target actor and critic networks with parameters $\theta^{\mu'}$ and $\theta^{Q'}$ are introduced to decrease the relevance between current and estimated values, which are with the same network structures with the related online networks. Besides, both target critic and actor networks are updated by slowly tracking on the related online critic and actor networks, separately [42]. Therefore, the update speed of the target networks is reduced, thereby significantly improving the model stability.

Secondly, the Prioritized Experience Replay (PER) technique is leveraged to accelerate the training of the model. The prior works sample transitions from the replay buffer at the same probability regardless of their significance. However, experience with higher absolute error is more effective than others with the lower error for the gradient updating. To this end, the PER approach distinguishes different samples stored in the replay buffer based on their corresponding absolute



(a) Average End-to-End Delay. (b) Average Throughput. (c) Average Packet Loss Rate.

Fig. 4: The system performance for the four types of multiple paths derived by the LDMR algorithm as the upper bound of random weight r_3 changes.

TABLE II:
SIMULATION PARAMETERS

Network Parameter	Value	Learning Parameter	Value
T_0	1200s	N_B, B_{max}	32,100
N	20	T, h	500,3
N_g	24	α, β, λ	$0.6, 0.4, 0.9$
r_1, r_2	$1, 10$	ξ, ρ	$0.01, 0.001$
L_q, ∇_{packet}	$100, 1KB$	η^Q, η^μ	$0.001, 0.001$

error. Therefore, these samples with higher absolute error are frequently reused for the faster convergence of the model.

Thirdly, as the input topology and traffic matrices are time-varying and sparse, the GNN is leveraged to handle corresponding routing configurations in variable-size graph-structured topologies. Comparing with general DNN architectures, the GNN has better generalization ability for dynamic network structures with different scale, since it can aggregate characteristics between different elements in a graph without a specification for dimensions of the input.

V. Performance Evaluation

Based on different satellite constellation topologies and traffic matrices, extensive simulations are performed to assess the performance of the proposed LDMR and GNN-MPTE algorithms.

A. Simulation Setup

We leverage NS3 [43] and ns3gym toolkit [44] to conduct the LSN, aiming to support central route calculation and intelligent traffic control. The proposed GNN-MPTE algorithm is implemented by using Pytorch 1.14 based on Python3.7, where the hidden states of any traffic demand and link are with 32 neurons, and the activation function adopts the ReLU. To test the generalization performance of the different network scale for the proposed algorithm, two different sizes of satellite constellations are implemented, where the former GlobalStar contains 24 satellites at an altitude of 1400km with the inclination of 55°, and the latter Iridium includes 66 satellites at an altitude of 780km with the inclination of 90°. Besides, two datasets are generated based on uneven population distribution [45], where the number of the source and destination pairs are 15 and 30 respectively, and the on/off shape and scale parameters are 1.5 and 500ms [4].

Firstly, we conduct the experiment to estimate the performance of the proposed LDMR method with the number and jointness of multiple paths changed, where each source node forwards end-to-end traffic to pre-calculated multiple paths with equal proportions, to evaluate average throughput, packet loss rate and delay. On the other hand, we train the proposed GNN-MPTE model offline in GlobalStar constellation and the first ten-timeslot traffic matrices. Then, the trained model is directly deployed on the NCC, and the last ten-timeslot traffic of both datasets are all thrown as the test samples, with key performance indicators containing average throughput, flow finish rate, delay, and Jain fairness Index [46] estimated. Other parameters in the paper are demonstrated in Table II.

Note that the average throughput refers to the number of successfully transmitted bytes, the average flow completion ratio represents the number of flows that have been completed to the total ones, the average delay estimates the mean end-to-end delay of successfully delivered packets without consideration of dropped packets, the average Jain fairness Index denotes the fairness of the system utilization, which can be denoted by Eq. (42).

$$\bar{J}_{\tau_i} = (\sum_{j=1}^{|\mathbf{E}^{\tau_i}|} B/f_{e_j}^{\tau_i})^2/[|\mathbf{E}^{\tau_i}| \sum_{i=1}^{|\mathbf{E}^{\tau_i}|} (B/f_{e_j}^{\tau_i})^2] \tag{42}$$

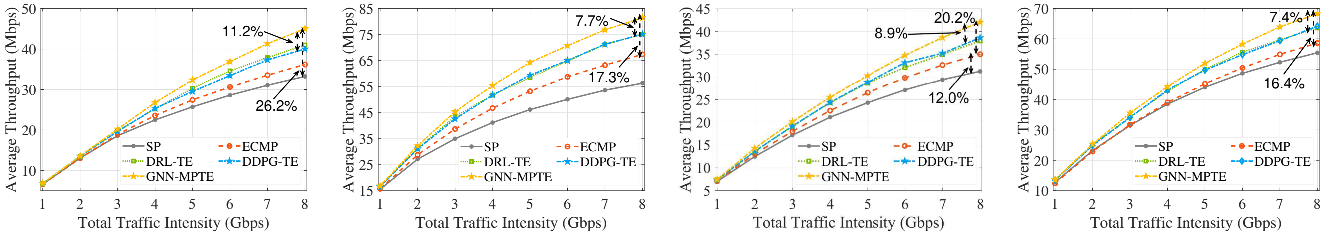
Additionally, the following benchmarks are also implemented as comparisons to present the performance of the GNN-MPTE algorithm.

1) Shortest Path First Algorithm (SPF) [47]: The strategy only adopts average transmission delay as the link metric with the objective of the minimum total delay.

2) Equal Cost Mutiple Paths Algorithm (ECMP) [19]: It splits total traffic to multiple paths of the equal cost for maximizing the available capacity.

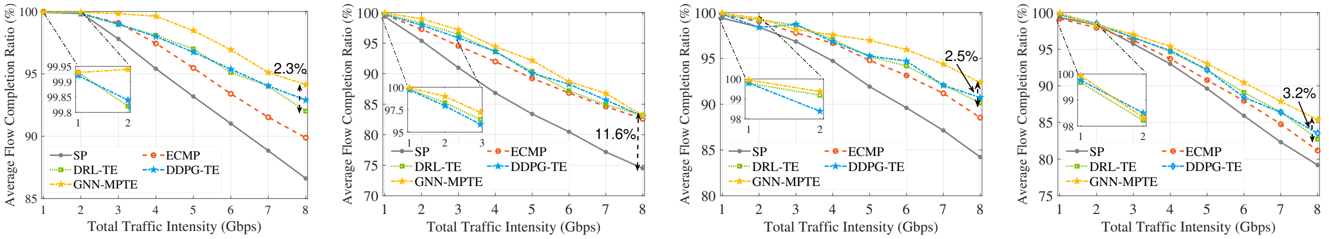
3) DRL-TE [30]: This method disperses any data flow to multiple paths based on the respective link quality in terms of the remaining bandwidth and delay. It adopts the AC-based PER method to optimize the general DRL framework, thereby accelerating the convergence of the model.

4) DDPG-TE [48]: The approach employs DDPG algorithm to dynamically allocate the proportion of traffic via different transmission paths. Since the dimensions of state



(a) (b) (c) (d)

Fig. 5: The Average Throughput of different satellite constellations based on given two traffic datasets vs the total traffic intensity. (a) The Average Throughput based on the dataset1 and GlobalStar constellation. (b) The Average Throughput based on the dataset2 and GlobalStar constellation. (c) Model Generalization based on the dataset1 and Iridium constellation. (d) Model Generalization based on the dataset2 and Iridium constellation.



(a) (b) (c) (d)

Fig. 6: The Average Flow Completion Ratio of different satellite constellations based on given two traffic datasets vs the total traffic intensity. (a) The Average Flow Completion Ratio based on the dataset1 and GlobalStar constellation. (b) The Average Flow Completion Ratio based on the dataset2 and GlobalStar constellation. (c) Model Generalization based on the dataset1 and Iridium constellation. (d) Model Generalization based on the dataset2 and Iridium constellation.

and action space are tightly coupling to the scale of input topologies and traffic matrices, the model cannot be extended to arbitrary satellite constellations. The hyperparameters of DDPG-TE follows the default setting of [48].

B. Simulation Results and Analysis

1) The routing performance of different multiple paths: Fig. 4(a)-(c) present the system performance for the four types of multiple paths derived by the LDMR algorithm based on different random weights. It can be seen that adopting two disjoint multiple paths has the best routing performance, compared with other three kinds of multipath patterns. The reason is that with the number of total paths and joint paths increased, more bottleneck links will appear, leading to degradation of throughput and increase of delay. For the two disjoint paths, the network throughput reaches the maximum when the parameter r_3 equals fifty, as displayed in Fig. 4(b). To maximize the network throughput, the LDMR algorithm sets the r_3 as fifty to compute two disjoint paths for each pair of source and destinations in this paper.

2) Average Throughput: Fig. 5 depicts the average throughput for the five algorithms based on the two different topologies and traffic datasets, where each point-plot denotes the mean result of 10 consecutive topology snapshots based on the same traffic intensity and satellite constellation. There are two important conclusions observed from above results. Firstly, the average throughput of the proposed GNN-MPTE gradually grows with the rise of the input traffic intensity. The reason is that more traffic is distributed to corresponding satellites to perform packet forwarding. In particular, as depicted in Fig. 5(a), the proposed GNN-MPTE increases the throughput by 11.2% and 26.2% when the volume of total traffic is $8GB$, compared with the DRL-TE and SPF benchmarks. Similarly, as shown in Fig. 5(b), the proposed algorithm increases the throughput by 7.7% and 17.3%, compared with the DDPG-TE and ECMP.

Secondly, with the topology structure changed, the proposed GNN-MPTE still outperforms the other benchmarks, where the former applies the same model, and the other learning-enabled methods should re-train the model based on new topology. This is because the proposed GNN-MPTE can be extended to other topology structures not seen in the training. In Fig. 5(c), it can be observed that the proposed GNN-MPTE increases average throughput by 8.9% and 20.2% when the volume of total traffic is $8GB$, compared with the DRL-TE and SPF benchmarks. Similarly, as displayed in Fig. 5(d), the proposed algorithm increases the throughput by 7.4% and 16.4%, compared with the DDPG-TE and ECMP.

3) Average Flow Completion Ratio: Fig. 6 presents the average flow completion ratio for the five approaches in the two satellite constellations, with the number of total source and destination pairs changed. In general, as higher traffic intensity can result in partial link congestion, the average flow completion ratio of the five schemes displays a downward

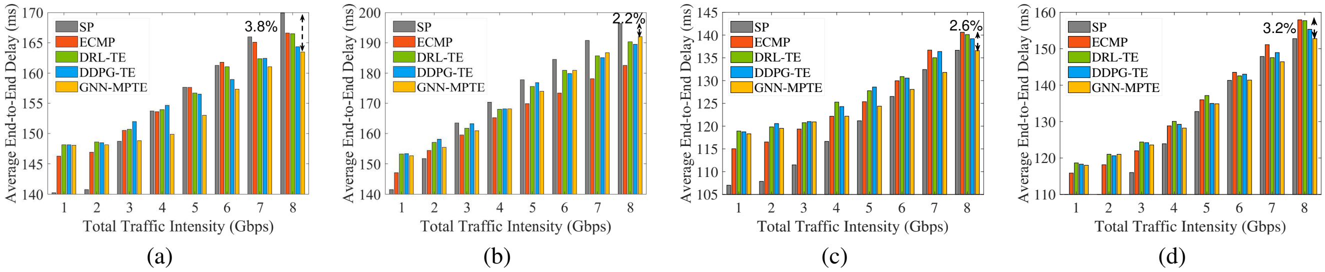


Fig. 7: The Average Delay of different satellite constellations based on given two traffic datasets vs the total traffic intensity. (a) The Average Delay based on the dataset1 and GlobalStar constellation. (b) The Average Delay based on the dataset2 and GlobalStar constellation. (c) Model Generalization based on the dataset1 and Iridium constellation. (d) Model Generalization based on the dataset2 and Iridium constellation.

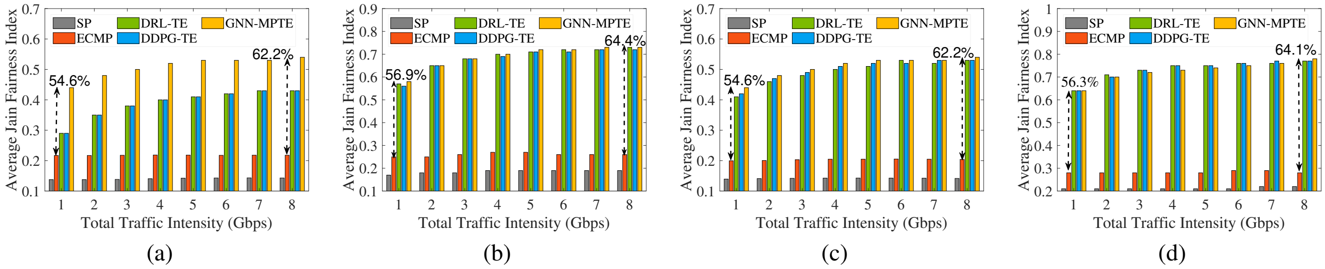


Fig. 8: The Average Jain Fairness Index of different satellite constellations based on given two traffic datasets vs the total traffic intensity. (a) The Average Jain Fairness Index based on the dataset1 and GlobalStar constellation. (b) The Average Jain Fairness Index based on the dataset2 and GlobalStar constellation. (c) Model Generalization based on the dataset1 and Iridium constellation. (d) Model Generalization based on the dataset2 and Iridium constellation.

trend with the growth of the volume of total traffic. However, the proposed scheme in Fig. 6(a) increases the flow completion ratio by 11.7% when the total traffic reaches the upper bound, compared with that in Fig. 6(b). In addition, the proposed scheme also outperforms the other benchmarks, even if it is extended to a new larger-size topology structure. Specifically, as shown in Fig. 6(c) and (d), when the volume of total traffic is $8GB$, the proposed GNN-MPTE increases the flow completion ratio by 2.5% and 3.2% respectively, compared with the DRL-TE baseline. As a result, the proposed algorithm has better generalization ability over satellite constellations with different scales.

4) Average Delay: Fig. 7 displays the delay variation trend over the increase of input traffic intensity in two different constellations and traffic matrices, where each bar represents the mean value over 10 consecutive topology snapshots with the same traffic intensity. As depicted in Fig. 7(a) and (b), the proposed scheme decreases average delay by 3.8% and 2.2% respectively, compared with the SPF, when the total traffic reaches the maximum. As presented in Fig. 7(c) and (d), compared with the ECMP, the proposed scheme can decrease average delay by 2.6% and 3.2% respectively, when the total traffic equals the $8GB$. The reason is that the SPF transmits data packets along the shortest-delay path, and cannot dynamically change paths to avoid link congestion caused by the increased traffic, the ECMP always equally partitions the traffic to equal-cost transmission paths without considering link occupancy. Different from them, the proposed GNN-MPTE can leverage multiple link-disjoint transmission paths currently and dynamically adjust the load distribution based on real link utilization state.

5) Average Jain Fairness Index: Fig. 8 shows the average Jain Fairness Index of the given two traffic matrices in Globalstar and Iridium constellations, where the larger Jain Fairness Index reflects the more fairness of link occupancy. Obviously, compared with traditional schemes, these learning-based schemes can balance the link utilization of the overall network, due to dynamic adjustment of traffic allocation. Particularly, as displayed in Fig. 8(a) and (b), compared with the ECMP benchmark, the proposed GNN-MPTE can dramatically enhance the Jain fairness Index by 54.6% and 56.9% at the lowest traffic intensity, and improve the Jain fairness Index by 62.2% and 64.4% at the upper traffic intensity. Besides, the proposed GNN-MPTE can flexibly adapt to other topology structures. Compared with the ECMP, as depicted in Fig. 8(c), the proposed algorithm increases the Jain fairness Index by 54.6% and 62.2% at the minimum and maximum traffic intensity. Similarly, it can be observed from Fig. 8(d) that the proposed algorithm increases the Jain fairness Index by 56.3% and 64.1% at above two points.

VI. Conclusion

This paper is centered on the multipath route planning for LSNs and fine grained traffic splitting for distinct transmission paths, and further propose the GMR scheme to improve the network throughput and user QoE. Firstly, we design the LDMR scheme for central routing calculation between any two LEO satellites, with the uneven traffic distribution and link utilization jointly considered. Then, we present the GNN-MPTE algorithm for optimal traffic splitting between different paths, where the GNN is leveraged to handle variable-size satellite constellations and traffic matrices. Finally, we implement the proposed algorithms in NS-3 with other benchmarks as comparisons. Related simulation results demonstrate that the proposed GNN-MPTE can be extended to arbitrary inclined/polar orbit satellite constellations without retraining, and effectively enhance the link fairness of LSNs and transmission performance in terms of average throughput, flow completion ratio, and delay.

References

[1] X. Zhu and C. Jiang, "Creating Efficient Integrated Satellite Terrestrial Networks in the 6G Era," *IEEE Wireless Communications*, vol. 29, no. 4, pp. 154-160, 2022.

[2] P. Kumar, S. Bhushan, D. Halder, et al., "fybrrLink: Efficient QoS-Aware Routing in SDN Enabled Future Satellite Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2107-2118, 2022.

[3] B. Feng, Y. Huang, A. Tian, et al., "DR-SDSN: An Elastic Differentiated Routing Framework for Software-Defined Satellite Networks," *IEEE Wireless Communications*, vol. 29, no. 6, pp. 80-86, 2022.

[4] X. Li, F. Tang, L. Chen, et al., "A State-Aware and Load-Balanced Routing Model for LEO Satellite Networks," *IEEE Global Communications Conference*, 2017: pp. 1-6.

[5] X. Zhu and C. Jiang, "Integrated Satellite-Terrestrial Networks Toward 6G: Architectures, Applications, and Challenges," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 437-461, 2022.

[6] S. K. Singh, T. Das, and A. Jukan, "A Survey on Internet Multipath Routing and Provisioning," *IEEE Communicaitons Surveys & Tutorials*, vol. 17, no. 4, pp. 2157–2175, 2015.

[7] Y. Huang, X. Jiang, S. Chen, et al.,"Pheromone Incentivized Intelligent Multipath Traffic Scheduling Approach for LEO Satellite Networks" *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5889-5902, 2022.

[8] Z. Zhang, C. Jiang, S. Guo, et al., "Temporal Centrality Balanced Traffic Management for Space Satellite Networks," *IEEE Transaction Vehical Technology*, vol. 67, no. 5, pp. 4427–4439, 2018.

[9] J. Yang, D. Li, X. Jiang, et al., "Enhancing the Resilience of Low Earth Orbit Remote Sensing Satellite Networks," *IEEE Network*, vol. 34, no. 4, pp. 304–311, 2020.

[10] B. Feng, Z. Cui, Y. Huang, et al., "Elastic Resilience for Software-Defined Satellite Networking: Challenges, Solutions, and Open Issues," *IT Professional*, vol. 22, no. 6, pp. 39-45, 2020.

[11] P. Sun, Z. Guo, J. Li, et al., "Enabling Scalable Routing in Software-Defined Networks With Deep Reinforcement Learning on Critical Nodes", *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 629-640, 2022.

[12] A. Tian, B. Feng, H. Zhou, et al., "Efficient Federated DRL-Based Cooperative Caching for Mobile Edge Networks," *IEEE Transactions on Network and Service Management*, doi: 10.1109/TNSM.2022.3198074.

[13] X. Huang, T. Yuan, G. Qiao, et al., "Deep Reinforcement Learning for Multimedia Traffic Control in Software Defined Networking", *IEEE Network*, vol. 32, no. 6, pp. 35-41, 2018.

[14] K. Rusek, J. Suárez-Varela, P. Almasan, et al., "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260-2270, 2020.

[15] B. Feng, A. Tian, S. Yu, et al., "Efficient Cache Consistency Management for Transient IoT Data in Content-Centric Networking," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12931-12944, 2022.

[16] J. Huang, Y. Su, L. Huang, et al., "An Optimized Snapshot Division Strategy for Satellite Network in GNSS," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2406-2409, 2016.

[17] J. W. Rabjerg, I. Leyva-Mayorga, B. Soret, et al., "Exploiting Topology Awareness for Routing in LEO Satellite Constellations," *2021 IEEE Global Communications Conference*, 2021: pp. 1-6.

[18] T. Pan, T. Huang, X. Li, et al., "OPSPF: Orbit Prediction Shortest Path First Routing for Resilient LEO Satellite Networks," *IEEE International Conference on Communications*, 2019: pp. 1-6.

[19] M. Chiesa, G. Kindler, and M. Schapira, "Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective," *IEEE Transactions on Network*, vol. 25, no. 2, pp. 779–792, 2017.

[20] F. Tang, H. Zhang, and L. T. Yang, "Multipath Cooperative Routing with Efficient Acknowledgement for LEO Satellite Networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 179-192, 2019.

[21] B. Feng, H. Zhou, G. Li, et al., "Enabling Machine Learning with Service Function Chaining for Security Enhancement at 5G Edges," *IEEE Network*, vol. 35, no. 5, pp. 196-201, 2021.

[22] G. Li, H. Zhou, B. Feng, et al., "Efficient Provision of Service Function Chains in Overlay Networks using Reinforcement Learning," *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, pp. 383-395, 2022.

[23] D. M. Casas-Velasco, O. M. C. Rendon and N. L. S. da Fonseca, "Intelligent Routing Based on Reinforcement Learning for Software-Defined Networking," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 870-881, 2021.

[24] Z. M. Fadlullah, F. Tang, B. Mao, et al., "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[25] Y. -R. Chen, A. Rezapour, W. -G. Tzeng, et al., "RL-Routing: An SDN Routing Algorithm Based on Deep Reinforcement Learning," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3185-3199, 2020.

[26] T. P. Lillicrap , J. J. Hunt, A. Pritzel, et al., "Continuous Control with Deep Reinforcement Learning," *International Conference on Learning Representations, ICRL*, 2016: pp. 1–6.

[27] P. Sun, Z. Guo, J. Li, et al., "Enabling Scalable Routing in Software-Defined Networks with Deep Reinforcement Learning on Critical Nodes," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 629-640, 2022.

[28] Z. Luan, L. Lu, Q. Li, et al., "EPC-TE: Explicit Path Control in Traffic Engineering with Deep Reinforcement Learning," in *IEEE Global Communications Conference*, 2021: pp. 1-6.

[29] G. Kim, Y. Kim and H. Lim, "Deep Reinforcement Learning-Based Routing on Software-Defined Networks," *IEEE Access*, vol. 10, pp. 18121-18133, 2022.

[30] Z. Xu, J. Tang, J. Meng, et al., "Experience-driven Networking: A Deep Reinforcement Learning based Approach," *IEEE Conference on Computer Communications*, 2018: pp. 1-6.

[31] W. Jiang, "Graph-based Deep Learning for Communication Networks: A Survey", *Computer Communications*, vol. 185, pp. 40-54, 2022.

[32] J. Gilmer, S. S. Schoenholz, P. F. Riley, et al., "Neural Message Passing for Quantum Chemistry", *International Conference on Machine Learning, PMLR*, 2017: pp. 1263–1272.

[33] P. Almasan, J. Suárez-Varela, A. Badia-Sampera, et al., "Deep Reinforcement Learning Meets Graph Neural Networks: Exploring a Routing Optimization Use Case", 2019, arXiv preprint arXiv:1910. 07

[34] H. Mao, M. Schwarzkopf, S.B. Venkatakrishnan, et al., "Learning Scheduling Algorithms for Data Processing Clusters", *Proceedings of the ACM Special Interest Group on Data Communication, in: SIGCOMM '19, Association for Computing Machinery*, 2019: pp. 270–288.

[35] P. Almasan, S. Xiao, X. Cheng, et al. "ENERO: Efficient Real-Time WAN Routing Optimization with Deep Reinforcement Learning", *Computer Network*, vol. 214, 2022, https://doi.org/10.1016/j.comnet.2022.109166.

[36] H. Guo, J. Li, J. Liu, et al., "A Survey on Space-Air-Ground-Sea Integrated Network Security in 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 53-87, 2022.

[37] Q. Chen, J. Guo, L. Yang, et al., "Topology Virtualization and Dynamics Shielding Method for LEO Satellite Networks," *IEEE Communications Letters*, vol. 24, no. 2, pp. 433-437, 2020.

[38] P. Liu, H. Chen, S. Wei, et al., "Hybrid-Traffic-Detour based Load Balancing for Onboard Routing in LEO Satellite Networks," *IEEE China Communications*, vol. 15, no. 6, pp. 28-41, 2018.

[39] S. Kassing, D. Bhattacherjee, Andr´e Baptista guas, et al. "Exploring the Internet from Space with Hypatia," *ACM Internet Measurement Conference*, 2020: pp. 214-229.

[40] J. Du, C. Jiang, J.Wang, et al., "Resource Allocation in Space Multi Access Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 2, pp. 598–618, 2017.

[41] "NASA HSF: Definition of Two-line Element Set Coordinate System," 2011, [Online]. Available: https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html.

[42] N. C. Luong, D. T. Hoang, S. Gong, et al., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, 2019.

[43] "ns-3 Network Simulator," 2020, [Online]. Available: https://www.nsnam.org/.

[44] "ns3-gym," 2022, [Online]. Available: https://github.com/tkn-tub/ns3-gym.

[45] "The Internet Usage Statistics," 2021, [Online]. Available: https://www.internetworldstats.com/stats.htm.

[46] "The Jain's Fairness Index," [Online]. Available: https://en.wikipedia.org/wiki/Fairness_measure.

[47] G. Song, M. Chao, B. Yang, et al., "TLR: A Traffic-Light-Based Intelligent Routing Strategy for NGEO Satellite IP Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3380-3393, 2014.

[48] B. Chen, P. Sun, P. Zhang, et al., "Traffic Engineering based on Deep Reinforcement Learning in Hybrid IP/SR Network," *China Communications*, vol. 18, no. 10, pp. 204-213, 2021.

Yunxue Huang received the B.S. degree from the School of Electronic and Information Engineering, Beijing Jiaotong University in 2018. She is currently pursuing a Ph.D. degree at the National Engineering Research Center for Advanced Network Technologies, Beijing Jiaotong University. Her research interests contain software-defined network and satellite networks.

Dong Yang (Member, IEEE) received the B.S. degree from Central South University, Hunan, China, in 2003, and the Ph.D. degree in communications and information science from Beijing Jiaotong University in 2009. From March 2009 to June 2010, he was a Post-Doctoral Research Associate with Jönköping University. In August 2010, he joined the School of Electronic and Information Engineering, Beijing Jiaotong University. Since 2017, he has been a Full Professor of communication engineering with Beijing Jiaotong University. His research interests include network architecture, wireless sensor networks, industrial networks, and the Internet of Things.

Bohao Feng (Member, IEEE) is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. His research interests include software-defined network, network functions virtualization, service function chaining, mobile internet, and satellite networks. He has been served as a TPC for a number of international conferences such as IEEE ICC and IEEE GLOBECOM.

Aleteng Tian received his B.S. degree in communication and information systems from Beijing Jiaotong University, where he is currently pursuing the Ph.D. degree at the National Engineering Research Center for Advanced Network Technologies, Beijing Jiaotong University. His current research interests include software-defined network, mobile edge caching, and satellite networks.

Ping Dong (Member, IEEE) is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include mobile Internet, software-defined networking, and network security. He also serves as a technical reviewer for several journals, including IEEE TVT, IEEE WCM, IEEE Network, and IEEE Communication Letters.

Shui Yu (Fellow, IEEE) is currently a Professor with the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include big data, security and privacy, and mathematical modeling. He is currently serving a number of prestigious editorial boards, including IEEE CST, IEEE IoT, IEEE CL, etc. He served as a Distinguished Lecturer of IEEE Communication Society (2018-2021). He is a Distinguished Visitor of IEEE Computer Society, a voting member of IEEE ComSoc Educational Services board, and an elected member of Board of Governor of IEEE Vehicular Technology Society.

Hongke Zhang (Fellow, IEEE) is a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, and the Director of the National Engineering Research Center for Advanced Network Technologies, Beijing, China. He has authored more than ten books and the holder of more than 70 patents. His research has resulted in many papers, books, patents, systems, and equipment in the areas of communications and computer networks.