On-Demand and Scalable Topology Control Service for LEO Satellite Network Evolving

Long Chen, *Member, IEEE*, Yi Ching Chou, Haoyuan Zhao, Hengzhi Wang, *Member, IEEE*, Feng Wang, *Senior Member, IEEE*, Hao Fang, Sami Ma, Feilong Tang, *Senior Member, IEEE*, Linghe Kong, *Senior Member, IEEE*, and Jiangchuan Liu, *Fellow, IEEE*

***Abstract*—Inter-Satellite Links (ISLs) are pivotal for delivering global connectivity services and optimizing resource utilization in 6G and beyond. However, delivering effective topology control services through ISL provisioning faces critical challenges in *sustainability* and *reliability*. Reducing ISLs can conserve energy and extend satellite battery life for Low-Earth-Orbit (LEO) satellites where replacing batteries is impractical. Conversely, increasing ISLs can enhance service reliability but may lead to uneven traffic distribution, overloading nodes, and accelerating battery degradation, ultimately degrading the quality of 6G services. To tackle this dilemma, we propose TASRI—a service-oriented framework for *Traffic-Aware, Sustainable, and Reliable ISL provisioning*. TASRI provides a dynamic topology control service by partitioning network topologies into logical zones, enabling flexible ISL activation and deactivation to adapt to varying service demands, ensuring efficient resource utilization and dynamic service orchestration. Using a sustainability-oriented weight model, we formulate the topology control service optimization problem and introduce a scalable on-demand topology evolving algorithm with a bounded approximation ratio. Extensive real-world deployment-based simulation results show that, compared to the state-of-the-art, our TASRI can substantially reduce battery life consumption, while achieving comparable reliability and excellent scalability with considerably fewer ISLs or ISL handovers.**

***Index Terms*—Topology control service, on-demand ISL provisioning, LEO satellite networks.**

I. Introduction

LOW-EARTH-ORBIT (LEO) satellite networks are becoming indispensable service infrastructures for the 6G networks and beyond, due to their global coverage. By leveraging laser Inter-Satellite Links (ISLs), LEO satellites enable direct communication with up to 30% shorter delays compared to terrestrial cable-based transmission [2]. ISLs also reduce handovers compared to the classic bent-pipe scheme, which relies solely on ground-satellite links and suffers from limited visible windows [3]. *This positions the topology control based on ISL provisioning as a critical service* that directly impacts space resource utilization efficiency, much like how backbone topology management service influences Internet routing and congestion control strategies.

However, achieving efficient topology control service by leveraging ISL provisioning also brings many challenges compared to terrestrial networks [4]. First, unlike the backbone of terrestrial networks which are usually composed of fiber optic cables and may rarely change after being established, wireless ISLs in satellite networks offer greater flexibility by enabling adaptive activation and deactivation. This flexibility, while essential for service-oriented architectures in space networks, introduces higher complexity, especially in the dynamic 3-dimensional satellite environment.

Second, satellite networks, especially LEO satellite networks, have limited resources [5] and face significant sustainability[1] challenges [6]. Replacing or updating battery cells after launching a satellite requires significant effort, if not entirely impossible. Unreasonable energy consumption caused by link maintenance and traffic forwarding can accelerate the deterioration of battery life, reducing the operational lifespan of the satellite. This, in turn, increases the need to launch replacement satellites, raising both costs and the risk of space debris. For example, SpaceX reported that 5% of its first batch of Starlink satellites stopped responding [7], possibly due to hardware issues or battery cell outages. Therefore, providing efficient topology control services under these constraints is essential to ensuring the long-term operation of the network.

In addition, as a form of wireless communication, ISLs are more susceptible to errors compared to terrestrial cables. Coupled with the uneven spatial-temporal distribution of traffic in space service computing systems, this necessitates a traffic-aware and dynamic approach to ISL provisioning. On the one hand, a sparse ISL topology can lead to service interruptions if a link fails, particularly problematic during high-traffic periods. On the other hand, an overly dense topology consumes excessive energy for link maintenance, undermining sustainability. By activating additional ISLs between satellites with lower

Received 28 December 2024; revised 5 May 2025; accepted 17 May 2025. Date of publication 4 June 2025; date of current version 8 August 2025. This work was supported in part by the NSERC Discovery Grant and in part by the National Natural Science Foundation of China under Grant 62302314, Grant 62172275, Grant 62472176, Grant W2412089, and Grant 61832013. *(Long Chen and Yi Ching Chou contributed equally to this work.) (Corresponding author: Jiangchuan Liu.)*

Long Chen, Yi Ching Chou, Haoyuan Zhao, Hao Fang, Sami Ma, and Jiangchuan Liu are with the School of Computing Science, Simon Fraser University, British Columbia V5A 1S6, Canada (e-mail: longchen.cs@ieee.org; ycchou@sfu.ca; hza127@sfu.ca; fanghaof@sfu.ca; masamim@sfu.ca; jcliu@sfu.ca).

Hengzhi Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518052, China (e-mail: whz@szu.edu.cn).

Feng Wang is with the Department of Computer and Information Science, University of Mississippi, Oxford, MS 38677 USA (e-mail: fwang@cs.olemiss.edu).

Feilong Tang and Linghe Kong are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: tang-fl@cs.sjtu.edu.cn; linghe.kong@sjtu.edu.cn).

Digital Object Identifier 10.1109/TSC.2025.3576690

[1]In this paper, sustainability refers to the lifespan of the onboard batteries of satellites.

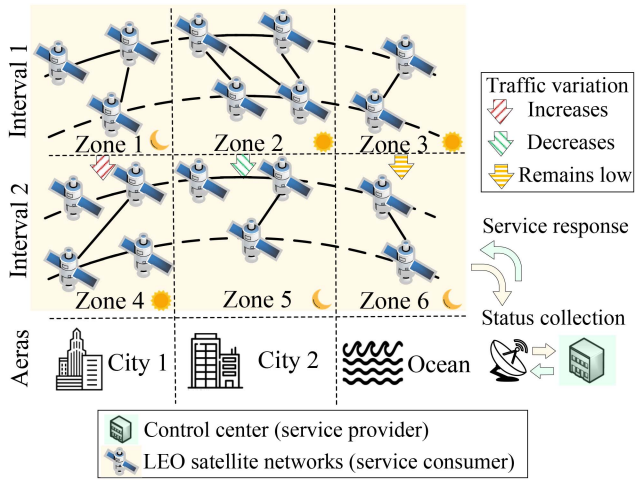


Fig. 1. Delivering on-demand and scalable topology control service over the zone-wise topology across spatial and temporal dimensions.

depth-of-discharge (DoD) when traffic increases, the network can both balance load and improve reliability by providing more alternate paths. Conversely, under lighter loads, deactivating ISLs at nodes with higher DoD conserves energy and extends satellite lifespan. Therefore, *the ISL provisioning-based topology control must incorporate traffic awareness to ensure both sustainable and reliable service delivery.*

Yet, the topology control service in satellite networks is still in its nascent stage. Current research and industrial products predominantly adopt the classic +Grid topology [8]. Specifically, within the same constellation, a satellite is assumed to be capable of establishing a maximum of two permanent intra-plane ISLs and two temporary inter-plane ISLs. In particular, the two inter-plane ISLs will be temporarily deactivated due to the antenna adjustment speed being unable to handle the relatively high angular speed. While there are various dynamic satellite topology control services [4], [8], [9], few of them consider traffic and jointly address sustainability and reliability issues.

Different from existing works, we for the first time propose a *Traffic-Aware, Sustainable and Reliable ISL provisioning (TASRI)* framework to deliver topology control service for LEO satellite networks. As illustrated in Fig. 1, the ground-based control center serves as the *service provider*, delivering topology control services by collecting satellite network status—such as load and battery usage—via ground stations. It dynamically partitions the network topology into logical zones and determines the activation or deactivation of ISLs within each zone. The LEO satellite networks, acting as *service consumers*, then reconfigure the topology based on the update strategy received from the service provider.

More specifically, after initially constructed in zone 1, zone 2 and zone 3 above City 1, City 2 and the Ocean at interval 1, respectively, the network topology evolves to zones 4, 5, 6 at interval 2 as the satellites are moving along their orbits with the joint considerations of service needs, sustainability, and reliability. Specifically, more ISLs can be established in zones 2 and 4 to meet the increased service demand and enhance reliability, while zones 1, 3, 5 and 6 have fewer ISLs, reducing link maintenance energy consumption and improving sustainability. In this way, we can enable the dynamic control of the zone-wise satellite topology across spatial and temporal dimensions. To achieve this, two key problems need to be addressed:

1) How do we accurately model the cost of ISL provisioning in dynamic satellite networks composed of unreliable ISLs?
2) How can we efficiently obtain an evolving, sustainable and reliable topology that adapts to traffic changes?

In this paper, we strive to tackle these problems and the main contributions are summarized as follows:

1) We propose a *Traffic-Aware, Sustainable, and Reliable ISL provisioning (TASRI)* framework, which envisions an *on-demand topology management service* for LEO satellite networks. This service adaptively (de)activates ISLs in response to dynamic traffic conditions, aiming to prolong battery life and maintain reliability.
2) Building upon TASRI, we develop a sustainability-oriented weight model tailored to LEO networks and formulate the ISL provisioning problem. We prove its NP-hardness and propose a scalable *on-demand topology evolving* algorithm. Rigorous analysis shows a bounded approximation ratio, ensuring both the feasibility and efficiency of the proposed approach in practice.
3) We conduct extensive real-world deployment-based simulations, which show that TASRI has less life consumption, and achieves better scalability, almost the same excellent reliability and end-to-end delay with considerably fewer ISLs or ISL handovers compared to the four most related schemes in the state-of-the-art.

The remainder of this paper is structured as follows: Section II briefly presents our motivations and reviews the related work. Section III introduces the system model of our TASRI framework and formulate the problem. In Section IV, we propose the algorithm with a rigorous theoretical analysis, followed by the performance evaluation in Section V. Finally, we conclude this paper in Section VI.

II. Motivations and Related Work

A. Energy Efficiency Does Not Ensure Sustainability

Optimizing energy efficiency does not necessarily improve sustainability, because the same amount of energy consumption can have different impacts depending on both *how* it is distributed and *where* it begins in terms of initial DoD.

First, the distribution of energy consumption matters. Suppose all satellites start with the same initial DoD, denoted as $\mathcal{D}_v(\tau) = \frac{B_v^{usage}(\tau)}{B_v^{\max}}, \forall v \in V$, where $B_v^{usage}(\tau)$ is the battery usage and $B_v^{\max}$ is the battery capacity. As shown in Fig. 2, if a single node A increases its DoD from 0% to 40%, the life consumption is 0.1325. In contrast, distributing the same energy consumption across two nodes (each from 0% to 20%) reduces the total life consumption to 0.0916.[2]

Along this direction, one of the key factors to optimize this distribution is to consider *traffic load*. When traffic increases, it's essential to activate more ISLs to distribute the traffic evenly across different paths. This helps to avoid overloading a single path, which could cause a rapid increase in the DoD and, consequently, a sharp rise in life consumption. On the other hand, when traffic decreases, unnecessary ISLs can be deactivated to reduce the energy overhead associated with maintaining these

[2]Calculated using the life consumption model in (16) with $A = 0.8$ from [6].

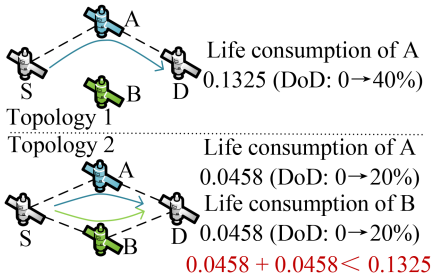


Fig. 2. Energy consumption concentrated on a single node has a greater impact on sustainability than when distributed across multiple nodes.

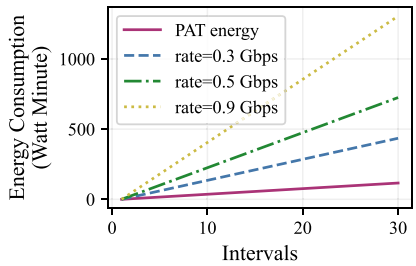


Fig. 3. Energy consumption comparison (ISL provisioning once during each interval, ISL capacity=1 Gbps): Pointing, Acquisition, Tracking (PAT) v.s. transmission.

links. Furthermore, when deactivating ISLs, it is important to ensure that ISLs carrying active flows are not improperly shut down, especially in response to fluctuating traffic load. If such ISLs are closed without careful consideration of traffic patterns, it could lead to unnecessary rerouting, resulting in fluctuations in network performance.

Second, the initial DoD also plays a critical role. Even if the total energy consumption is the same, starting from a higher DoD leads to greater overall life consumption. For instance, increasing the DoD from 60% to 80% has a greater impact than increasing it from 20% to 40% [6].

B. Energy-Efficient Routing is Not Enough

Routing and traffic engineering are widely viewed as effective ways to improve energy efficiency in satellite networks. For instance, Yang et al. [6] proposed the GreenSR algorithm to prolong satellite lifespan through energy-aware path selection. However, LEO satellite networks offer an additional dimension of flexibility via wireless ISLs, which can be dynamically activated or deactivated. Relying solely on energy-efficient routing in a *fixed* topology is therefore insufficient for two main reasons:

- *Maintaining all possible ISLs is energy-intensive:* Although establishing redundant of ISLs (e.g., using time-expanded graphs [10]) allows routing algorithms to find efficient paths, it requires considerable energy and degrades overall sustainability.
- *A fixed subset of ISLs limits the solution space:* Routing in a static topology can only leverage existing links [11], risking a suboptimal global solution. For example, in Fig. 2, Topology 1 forces the path from S to D to pass through A exclusively, whereas Topology 2 enables both S-A-D and S-B-D, potentially leading to more balanced and sustainable routing.

In summary, while routing and traffic engineering can optimize energy usage along established paths [12], [13], they remain constrained by the available ISLs. Dynamically adjusting ISL provisioning expands the optimization space and raises the theoretical upper bound for sustainability.

C. The Overhead of ISL Provisioning is Acceptable

For dynamic ISL activation, on-demand ISL provisioning relies on precise and efficient Pointing, Acquisition, and Tracking (PAT) systems. Existing research indicates that ISL setup time can be reduced to the order of milliseconds [14], making on-demand ISL provisioning practical in highly dynamic environments.

Furthermore, although ISL provisioning incurs additional energy overhead due to the PAT process, we found that limiting the provisioning frequency can keep this energy overhead within acceptable limits. As shown in Fig. 3, we examined the energy consumption of activating or deactivating a single ISL *once* within each interval and compared it with the energy consumed by transmission. The PAT delay is set to 100 ms [14], with the PAT power equal to the maximum power of the laser transceiver called CLICK, 40 watts [15]. We use the transmission energy consumption model and set the battery capacity (5,000 Watt Minute) and the maximum ISL bandwidth (1 Gbps) based on [6]. The results indicate that the energy overhead from PAT remains within an acceptable range compared to transmission costs.

D. Summary of Related Work

Current solutions related to topology control service based on ISL provisioning primarily employ the +Grid topology, wherein each satellite can establish a maximum of four ISLs [16], [17]. On the other hand, dynamic ISL provisioning schemes can be categorized into the following two groups.

1) Traffic-Aware ISL Provisioning Service: Most traffic-aware schemes aim to balance load or reduce congestion by dynamically switching inter-satellite links (ISLs). Bhattacherjee and Singla [4] introduced a motif-based topology, where each motif consists of three ISLs sharing a common satellite. Building on this, our TASRI approach addresses city hotspots caused by uneven traffic and mitigates battery aging by dynamically managing ISL use. This approach also conserves energy by switching off ISLs in remote, low-traffic areas, thus extending satellite lifespan. Balancing load improves network performance, reduces congestion, and lowers energy consumption [18], [19], but must be done with careful attention to avoid suboptimal topologies that lead to detours, increased energy use, and accelerated battery aging.

Other studies have explored different ISL configurations to enhance performance. ×Grid [20] facilitates inter-satellite handoffs to minimize detours and packet loss, while Wang et al. [8] expanded the +Grid structure with additional intra- and inter-orbit ISLs to balance load, though this increases energy consumption. Ji et al. [21] proposed a geometric structure for mega constellations that balances network scale and traffic control delays, though energy efficiency remains a critical concern, especially given the high costs of hardware or battery upgrades. Liu et al. [22] emphasized energy-aware design by modeling link availability and traffic demand via algebraic connectivity.

Several studies have proposed methods for optimizing ISL deployment based on traffic distribution. Wang et al. [23] introduced a method to remove ISLs for better bandwidth efficiency, but this can lead to load imbalances that accelerate battery degradation. Similarly, Wang et al. [24] suggested activating extra encountering ISLs to reduce latency, overlooking their additional energy cost. Han et al. [25] proposed a Non-Grid-Mesh topology to improve communication performance, but did not consider the impact of ISL energy consumption on satellite battery life. Lan et al. [26] focused on quasi-torus topology to maximize network capacity, yet did not account for ISL-related energy consumption. Yang et al. [27] optimized terminal acquisition probability and channel gain to improve laser ISL stability and reduce latency, but ignored the energy impact on battery life. Guo et al. [28] developed an algorithm to optimize throughput capacity through ISL utilization, neglecting the effects on satellite battery lifespan.

Lastly, Lai et al. [29] introduced MegaReduce, a constellation design framework to minimize deployment costs by optimizing redundant paths and latency constraints, whereas our approach specifically targets dynamic ISL scheduling within existing constellations. Wang et al. [30] proposed a federated reinforcement learning approach to optimize ISL scheduling, which reduces hop count and active links; however, their method may incur significant synchronization overhead in large-scale networks.

In summary, energy efficiency must be carefully considered to avoid suboptimal topologies that lead to detours, increased energy use, and accelerated battery aging.

2) Energy-Efficient ISL Provisioning Service: Energy-efficient schemes often utilize dynamic ISLs to reduce power consumption. Chen et al. [9] proposed a weighted sum of propagation delay and energy costs (for ISL maintenance and switching), constructing a minimum spanning tree from historical data. However, tree-based designs limit paths between nodes, risking network disruption if links fail. Their focus on energy alone overlooks battery health factors like depth of discharge ratios, which, if high, can hasten battery deterioration. Radhakrishnan et al. [31] analyzed energy efficiency in inter-satellite communication, while Yang et al. [6] introduced an adaptive approach for the +Grid structure, reducing load by putting selected nodes to sleep. However, frequent link handovers in this method can disrupt network flows. Other schemes targeting single-hop efficiencies [32], [33] struggle to scale in large LEO constellations, and UltraDense [34] reduces the number of satellites at the cost of severe traffic imbalances and accelerated battery aging.

Several studies focus on topology optimization to balance energy use and performance. Hu et al. [35] introduced the AlphaSat algorithm, a two-phase topology optimization method that reduces latency and optimizes power consumption, but without considering the impact on battery lifespan. Similarly, Chen et al. [36] proposed a pruned inter-plane ISL topology framework to explore routing and cost-saving potential through optimization, but did not evaluate the effect on satellite battery life. Higashimori et al. [37] proposed a multilayer topology control framework to enhance traffic handling in non-uniform conditions. While effective, this architecture requires cross-operator coordination and may raise privacy concerns, whereas our approach simplifies the problem by focusing on single-layer topology optimization to avoid such complexities.

Guo et al. [38] proposed a distributed inter-layer topology optimization method for LEO double-layered satellite networks, which maximizes observation benefits through coordination between remote-sensing and communication layers. However, this inter-layer framework introduces coordination overhead. Wang et al. [39] presented a federated learning-based ISL scheduling method that reduces energy consumption and hop count through asynchronous hierarchical learning, but their focus is on energy savings from link deactivation without considering service quality impacts or long-term battery degradation. Lan et al. [40] proposed an algorithm to remove redundant ISLs while maintaining capacity, but their capacity-centric approach may compromise network connectivity reliability.

To conclude, *most existing studies prioritize improving energy utilization, often overlooking the extension of battery lifespan for long-term sustainability*. In contrast, the distinctive aspect of our TASRI lies in the dynamic construction of a *traffic-aware, sustainable,* and *reliable* satellite network topology, in a *scalable* way, which sets it apart from existing works.

III. System Model of TASRI

A. Satellite Topology Control Service Architecture

LEO satellites orbit the Earth periodically at high velocities-for instance, those in Starlink Shell 1 maintain an orbital speed of 7.6 km/s [41]. Despite their rapid motion, the relative positions between satellites within the same orbital plane remain stable. This stability arises because co-orbital satellites are uniformly distributed and maintain identical velocities, enabling consistent intra-orbit ISLs. In contrast, inter-orbit ISLs exhibit dynamic behavior: while the relative positions of paired satellites across different orbits remain approximately fixed, their inter-satellite distance varies continuously due to orbital mechanics, resulting in time-dependent propagation delays [42]. The most severe limitation occurs when a satellite approaches polar regions–the extreme relative velocity between cross-orbit satellites exceeds the steering capability of ISL antennas, necessitating link deactivation to avoid hardware failure [8].

Based on the above discussion, we employ a discrete-time model to divide the period into a series of intervals, denoted as $\mathbb{T} = \{\tau\}$. During each interval, we assume that the topology remains stable. In each interval $\tau \in \mathbb{T}$, the candidate topology is represented as a graph $G(\tau) = (V, E(\tau))$, where the node set V includes all satellites in the LEO satellite networks. The time-dependent edge set $E(\tau)$ consists of all candidate ISLs determined by the *visible conditions* and *hardware limitations*.

The visible conditions stipulate whether a candidate ISL can be established between any two satellites, i.e., $u, v \in V$, which require that 1) the distance between them is within a predefined threshold α [4], and 2) a clear line-of-sight must exist between the two satellites [9]. More specifically, the straight-line segment connecting the centers of two satellites is evaluated against Earth's spherical surface. Any portion of this segment that intersects the Earth, i.e., falls below the geometric horizon imposed by Earth's curvature, is deemed to have an obstructed line-of-sight. Such blocked links are excluded from the candidate ISL set.

Furthermore, the maximum number of ISLs that a satellite can be established, denoted as κ_v, is determined by its hardware limitations. By leveraging smaller ISL terminals with extended communication ranges,[3] a single PAT system can manage multiple ISLs simultaneously [15]. For example, Starlink satellites

[3]https://www.tesat.de/products#laser

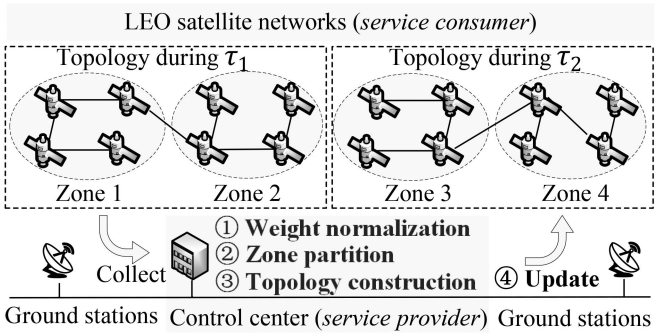


Fig. 4. Satellite topology control architecture.

can establish several ISLs using multiple space lasers[4] and satellites can establish up to 88 permanent and 396 temporary ISLs within a constellation [43].

For ease of exposition, we focus on the provisioning of ISLs, rather than the provisioning of ground-satellite links which is an orthogonal problem to the one studied in this paper. The provisioning of ground-satellite links is usually subject to regulations concerning data privacy [44] and budget, and can be addressed by ground station infrastructure planning [41], [45].

Based on the above discussion, under the satellite topology control service architecture as shown in Fig. 4, our TASRI works as follows: at the beginning of each interval, we run the on-demand topology evolving algorithm in the terrestrial control center, acting as the *service provider*. The LEO satellite networks act as the *service consumer*.

Our algorithm selects a subset of ISLs from all *candidate*s. In other words, without a visual link between two satellites, an ISL cannot be established, but communication is still possible through multi-hop paths. Hence, although the number of satellite pairs without a visual link increases with network scale, it has *limited* impact on topology connectivity. The key factors affecting connectivity are hardware limitations, such as the number of antennas, and the number of visible neighboring satellites.

The algorithm first normalizes the edge weight based on the traffic condition observed in the previous interval. Then, the whole topology is divided into a set of logical zones by mobility. Next, we build the topology by carefully selecting the cross-zone and in-zone ISLs. Finally, each satellite dynamically activates or deactivates ISLs based on the algorithm outputs.

B. Sustainability-Oriented Weight Model

We design the weight models by considering the energy consumption impact on battery lifespan, primarily from the costs of traffic forwarding, link maintenance, and switching.

1) Node Weight Model: We assume that each node has the same unit energy cost for traffic forwarding, denoted as ϵ_T. The node weight is defined as the product of battery life consumption and energy cost for traffic forwarding, expressed as

$$e_v(\tau) = \mathcal{L}(D_v(\tau))\epsilon_T T_v(\tau), \tag{1}$$

where $\mathcal{L}(D_v(\tau)) = D_v(\tau) \times 10^{0.8(D_v(\tau)-1)}$ represents the life consumption of the onboard lithium-ion battery cells [6]. As the weight increases semi-exponentially with the DoD and linearly with $T_v(\tau)$, we aim to select ISLs associated with nodes having smaller node weights, i.e., those with lower DoD and lighter traffic loads during topology construction.

2) Link Weight Model: Before introducing the link weight model, we define the failure probability of each wireless edge[5] $l \in E(\tau)$ as $f_l(\tau) \in [0, 1)$. We focus here on edge failures, although it can be readily extended to the node failure scenarios.

We take the ISL maintenance cost, which quantifies the required energy for Pointing, Acquisition, and Tracking (PAT), as the link weight. Specifically, we define:

$$e_l(\tau) = \frac{d_l(\tau)^2 b_l(\tau)\epsilon_S}{1 - f_l(\tau)}, \tag{2}$$

where $d_l(\tau)$ and $b_l(\tau)$ represent the propagation delay and bandwidth of link l in interval τ, respectively. Here, ϵ_S is the required PAT-system energy to maintain the ISL. By placing $1 - f_l(\tau)$ in the denominator, links with higher failure probabilities incur larger weights. Consequently, our subsequent algorithm, which aims to minimize total weights, naturally avoids unreliable ISLs.

The term $d_l(\tau)^2\, b_l(\tau)$ illustrates how PAT energy scales with both *distance* and *bandwidth*. In free space, electromagnetic waves spread out spherically, which causes path loss to grow with the square of the distance. Thus, maintaining a fixed received power level at greater distances requires transmit power to increase proportionally to the square of the distance [46]. Because $d_l(\tau)$ is *linearly related* to physical distance via the speed of light, we use $d_l(\tau)^2$ to capture this effect on path loss. Meanwhile, a higher link bandwidth $b_l(\tau)$ demands additional transmit power to overcome increased thermal noise power, as prescribed by the thermal noise model and Shannon's capacity theorem [47]. Thus, the $d_l(\tau)^2\, b_l(\tau)$ product serves as a *combined scaling factor* for the energy needed to keep the link active.

C. Mobility-Aware Logical Zones

As discussed earlier, ISL provisioning needs to be traffic-aware, adjusting based on fluctuations in traffic to balance maintenance costs and system performance. When traffic decreases, reducing ISLs can help lower maintenance costs, while increasing ISLs during traffic spikes can enhance sustainability by mitigating the impact on battery life and improving transmission reliability. Given the global coverage and periodic motion of LEO satellites, satellites in close proximity often experience *similar* traffic patterns. For instance, satellites over densely populated urban areas typically carry heavier loads than those over oceans. Hence, ISL provisioning does not need to be handled for each satellite individually. Instead, the set of satellite V, can be divided into logical zones to improve provisioning efficiency.

We start by categorizing the zones into two types: *hot* zones and *cold* zones. This provides a high-level segmentation of satellite areas based on general traffic patterns. Specifically, this process begins in a *bottom-up* fashion, where each satellite initially belongs to its own zone, and zones are gradually merged based on their characteristics. If a satellite is located over *oceans*, it belongs to a cold zone. For a satellite over *land*, its zone type is determined by comparing its current load with historical data over a certain period. We introduce an adjustable parameter ξ, representing the number of historical intervals of interest. If $o_v^\tau \geq \frac{\sum_{i=1\ldots\xi} o_v^{\tau-i}}{\xi}$, the satellite belongs to a hot zone; otherwise,

[4] https://www.starlink.com/technology

[5] In this paper, both edges, links, and ISLs have the same meaning and are used interchangeably when no ambiguity arises.

it belongs to a cold zone, where o_v^τ is the queue occupancy ratio of v at the beginning of τ. We have

$$H(\tau) = \{h\}, h = \{v | v \in V, o_v^\tau \geq \frac{\sum_{i=1\ldots\xi} o_v^{\tau-i}}{\xi}, v \text{ is over the land}\}, \quad (3)$$

$$C(\tau) = \{c\}, c = \{v | v \in V, o_v^\tau < \frac{\sum_{i=1\ldots\xi} o_v^{\tau-i}}{\xi}, v \text{ is over the land}\} \cup \{v | v \in V, v \text{ is over the ocean}\} \quad (4)$$

$$V = \{h \cup c | \forall h \in H(\tau), c \in C(\tau)\}, \quad (5)$$

$$h \cap c = \emptyset, \forall h \in H(\tau), c \in C(\tau), \quad (6)$$

where $H(\tau)$ (resp. $C(\tau)$) is the set of hot (resp. cold) zones.

After the coarse-grained classification, we proceed with merging zones. For detailed steps on the merging process, please refer to Section IV-C. Once the merging is complete, it is obvious that the number of zones, denoted as $|H(\tau)|$ for hot zones and $|C(\tau)|$ for cold zones, as well as the nodes within each zone, change over time. This is due to the dynamic nature of the topology and traffic patterns.

We define k_{uv} as the number of edge-disjoint paths between u and v. On a *more detailed level*, we introduce an adjustable variable $\beta \in (0, 1)$ to *constrain the topological properties* of each zone after ISL provisioning. Specifically, if both nodes are in the *same* hot zone, i.e., $u, v \in h \in H(\tau)$, k_{uv} ranges from $\beta k_{uv}^{\max}$ to $k_{uv}^{\max}$. Conversely, if both nodes are in the *same* cold zone, i.e., $u, v \in c \in C(\tau)$, k_{uv} ranges from 1 to $\beta k_{uv}^{\max}$. Note that $k_{uv}^{\max}$ is the maximal number of edge-disjoint paths obtained by the candidate topology $G(\tau)$. If a zone contains only one node, there is no need to establish a path within it. We have :

$$k_{uv} \in [\beta k_{uv}^{\max}, k_{uv}^{\max}], \forall u, v \in h, \quad (7)$$

$$k_{uv} \in [1, \beta k_{uv}^{\max}], \forall u, v \in c. \quad (8)$$

After constructing the zones, we perform ISL provisioning for each zone individually to meet the requirements of (7) and (8). The value of k_{uv} is a range rather than a fixed number because we aim to provide greater flexibility in ISL provisioning, allowing for improvements in both reliability and sustainability. During the provisioning process, k_{uv} can be adjusted based on the quality of the ISLs, enabling adaptive activation and deactivation of links. For more details, please refer to Section IV-D.

D. Problem Formulation

Here we formulate the problem in our TASRI framework as an optimization problem to minimize the sum of edge and node weights in each interval as

$$\textbf{minimize} \quad \sum_{l \in E(\tau), v \in V} x_l(\tau) e_l(\tau) + z_v(\tau) e_v(\tau) \quad (9)$$

$$s.t. \quad (1) - (8),$$

$$y_l^v(\tau), z_v(\tau) \in \{0, 1\}, \forall l \in E(\tau), v \in V, \quad (10)$$

$$z_v(\tau) = \min \left(1, \sum_{l \in E(\tau), x_l(\tau)=1} y_l^v(\tau)\right), \forall v \in V, \quad (11)$$

$$\sum_{l \in E(\tau), x_l(\tau)=1} y_l^v(\tau) \leq \kappa_v, \forall v \in V, \quad (12)$$

where $x_l(\tau) \in \{0, 1\}, \forall l \in E(\tau)$ is a binary variable to indicate whether the edge l is selected in the current interval ($x_l(\tau) = 1$) or not. The optimization goal in (9) aims to select a subset of edges and nodes to minimize the total weights. We use two binary variables $y_l^v(\tau)$ and $z_v(\tau)$ in (10) to indicate whether satellite v is incident to a selected edge l ($y_l^v(\tau) = 1$) and to any edge in $E(\tau)$ ($z_v(\tau) = 1$), respectively. The constraint (11) ensures that the node energy consumption is considered at most once in the optimization goal in (9). Finally, constraint (12) represents the hardware limitations of any node $v \in V$, where each satellite $v \in V$ can establish at most κ_v ISLs and κ_v captures the maximum number of *active* terminals on each satellite. In our model, we do *not* consider reorienting laser terminals, as such mechanical adjustments are time-consuming and unsuitable for highly dynamic topologies. Instead, each terminal is assumed to be pre-aligned. This simplification is consistent with existing +Grid designs, where multiple terminals face fixed directions and can be (de)activated as needed.

Lemma 1: Given a graph $G(\tau) = (V, E(\tau))$, finding a subgraph $G'(\tau) = (V, E'(\tau))$ where $E'(\tau) \subseteq E(\tau)$ with k-edge-disjoint paths between u and v, $\forall u, v \in V$, is equivalent to finding a k-edge-connected graph on $G(\tau)$.

Proof: We first prove that if a graph has k-edge-disjoint paths for each pair of nodes, it is k-edge-connected. We build an edge set $X \subseteq E(\tau)$ with $|X| < k$. After removing all edges in X, there still exists at least one path for each pair of nodes. Thus, the graph is k-edge-connected.

We then prove that if a graph is k-edge-connected, then it has k-edge-disjoint paths for each pair of nodes. Let the capacity of all edges in the graph be 1. We consider two arbitrary distinct nodes u and v in the graph. Based on the max-flow min-cut theorem [48], the maximum flow between u and v is equal to the minimum number of edges whose removal disconnects them. Since the graph is k-edge-connected, the minimum number of edges whose removal disconnects u and v is k. Hence, the maximum flow between u and v is k. We can easily obtain k edge-disjoint paths between any pair of nodes by decomposing the maximum flow between them into k edge-disjoint paths based on the Ford-Fulkerson method [49]. □

Theorem 1: The formulated problem is NP-hard.

Proof: We consider a special case of the formulated problem in which we ignore the node weight and edge weight normalization. We assume that the number of edge-disjoint paths k_{uv} for all pairs of nodes have the same value, denoted as k_0. Considering the large number of nodes in the LEO satellite network such as Starlink, we can assume $k_0 \geq 2$. Based on Lemma 1, the special case of the formulated problem is equivalent to finding the minimum k-edge-connected subgraph with $k = k_0 \geq 2$, which is NP-hard [50]. The above reduction is done in polynomial time, which completes the proof. □

IV. On-Demand Topology Evolving Algorithm

A. Overview

We propose the On-Demand Topology Evolving (ODTE) algorithm in this section. It solves the formulated problem, which aims to minimize the sum of weights, by (de)activating a subset of ISLs during each interval. However, ISL provisioning can disrupt traffic transmission; for example, an ISL in the transmission path of a flow might be deactivated. To address this, as shown in Fig. 5, we first propose a traffic-aware *edge weight*

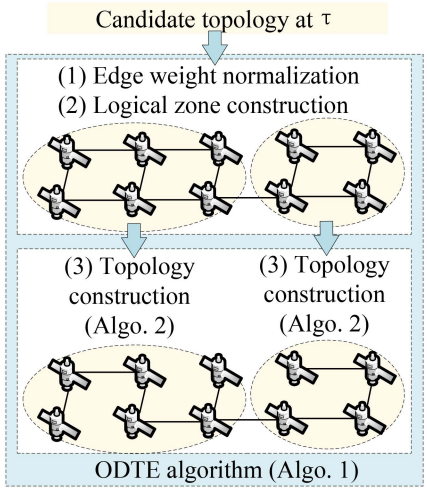


Fig. 5. On-demand topology evolving framework.

normalization model that dynamically updates edge weights by estimating the duration of ongoing flows. The longer the flow duration, the smaller the edge weight, making it more likely that the ISL with active flows will remain active in the next interval. For example, if a flow lasts across two intervals and the topology changes in the next interval such that the previously active path is no longer available, the flow may switch to a new path. This can cause retransmissions or require more complex protocols to minimize packet loss and prevent performance degradation.

To *efficiently* address the problem, we partition the large-scale satellite network into logical zones instead of considering individual satellites, and propose a *logical zone construction* method (as shown in (2) in Fig. 5). Each zone comprises satellites with similar traffic patterns, allowing us to focus on ISL provisioning within each zone.

Finally, we propose a sustainable and reliable *topology construction* algorithm, depicted as (3) in Fig. 5. This algorithm balances sustainability and reliability by reducing the number of active ISLs, while ensuring that the number of edge-disjoint paths between any pair of nodes in the zone meets an adaptive threshold. This threshold increases in response to ISL failures and decreases otherwise.

B. Traffic-Aware Edge Weight Normalization

In LEO satellite networks, we can obtain the routing path of a flow in the terrestrial control center, where the ground station periodically collects resource status from satellites, determines the routing paths, and orchestrates resources [51]. We denote the flow set in the previous interval as $F(\tau-1)$. Given any flow $f \in F(\tau-1)$, its routing path contains a set of edges denoted as $E_f(\tau-1)$. We first find the bottleneck bandwidth along the path as

$$\overline{b}_f = \min_{l \in E_f(\tau-1)} b_l(\tau), \tag{13}$$

where $b_l(\tau)$ is the bandwidth of the edge l.

We represent the estimated number of intervals that a flow f exists in the edge l as τ_l^f, which can be obtained by

$$\tau_l^f = \frac{T_f}{\overline{b}_f|\tau|} \times o_u^\tau \times o_v^\tau, \forall l \in E_f(\tau-1), \tag{14}$$

Algorithm 1. On-Demand Topology Evolving (ODTE).

Input: $G(\tau)=(V,E(\tau)), F(\tau-1), \{e_l(\tau)|\forall l \in E(\tau)\}, \{e_v(\tau)|\forall v \in V\}, \{y_l^v(\tau)|\forall l \in E(\tau), v \in V\}$
Output: $G'(\tau)=(V,E'(\tau)), E'(\tau) \subseteq E(\tau)$

1: $V'=V, H(\tau)=\{h_v|\forall v \in V'\}, C(\tau)=\{c_v|\forall v \in V'\}$
2: **for** $l \in E(\tau), v \in V$ **do** *//Edge weight normalization*
3: **if** v is incident to l **then**
4: $e_l = e_l + \frac{e_v}{\sum_{l \in E(\tau)} y_l^v(\tau)}$
5: $\mathcal{T}_l = \max_{\forall f \in F(\tau-1), l \in E_f(\tau-1)} \tau_l^f$
6: $e_l(\tau) = e_l(\tau)/\mathcal{T}_l$
7: **while** $V' \neq \emptyset$ **do** *//Logical zone construction*
8: $v \leftarrow V', h_v = \{v\}$ or $c_v = \{v\}$
9: **for** $u \in \mathcal{N}_v(\tau)$ **do**
10: $u \in h_u$ or $u \in c_u$
11: **if** $v \in h_v, u \in h_u, h_v \neq h_u, |h_v|+|h_u| \leq \gamma_H$ **then**
12: $H(\tau) = H(\tau) - \{h_v\} - \{h_u\}, h_+ = h_v \cup h_u$
13: $H(\tau) = H(\tau) \cup \{h_+\}, V' = V' - h_+$
14: **if** $v \in c_v, u \in c_u, c_v \neq c_u, |c_v|+|c_u| \leq \gamma_C$ **then**
15: $C(\tau) = C(\tau) - \{c_v\} - \{c_u\}, c_+ = c_v \cup c_u$
16: $C(\tau) = C(\tau) \cup \{c_+\}, V' = V' - c_+$
17: **for** $h \in H(\tau)$ **do** *//Expanding*
18: **if** $\exists v \in h \in H(\tau), u \in \mathcal{N}_v(\tau), u \in c \in C(\tau)$ **then**
19: $H(\tau) = H(\tau) - \{h\}, h_+ = h \cup \{u\}, H(\tau) = H(\tau) \cup \{h_+\}$
20: $C(\tau) = C(\tau) - \{c\}, c_- = c - \{u\}, C(\tau) = C(\tau) \cup \{c_-\}$
21: **for** $V_A(\tau) \in H(\tau) \cup C(\tau)$ **do** *//Topology construction*
22: **for** $v \in V_A(\tau), V_{A'}(\tau) \in H(\tau) \cup C(\tau) \backslash V_A(\tau)$ **do**
23: $l = \arg\min_{l \in E(\tau), \forall u \in V_{A'}(\tau), y_l^v(\tau)=y_l^u(\tau)=1} e_l$
24: $E'(\tau) = E'(\tau) \cup \{l\}$
25: $A = (V_A(\tau), E_A(\tau))$
26: Run Algorithm 2 to get $A' = (V_A(\tau), \overline{E}_A(\tau))$
27: $E'(\tau) = E'(\tau) \cup \overline{E}_A(\tau)$

where T_f is the amount of traffic to be transmitted and $|\tau|$ is the length of τ. u and v are two endpoints of l.

The first term provides an initial estimate by considering the bottleneck bandwidth and T_f. The last two terms account for the influence of other flows, indicating that higher loads lead to a longer expected duration of flow f on l.

Edge weight normalization: As shown in Algorithm 1, we first amortize the cost of each node into its incident edge in lines 2–4. The denominator in line 4 calculates the degree of the node, i.e., the number of edges that are incident to the node. Then, given any edge $l \in E(\tau)$, we find the maximum estimated duration of a flow that passes through l, denoted as $\mathcal{T}_l$ in line 5. Finally, we divide the edge weight by $\mathcal{T}_l$ in line 6, so that the edge l is more likely to be chosen in the current interval and the transmission interruption is minimized.

C. Mobility-Aware Logical Zone Construction

We first initialize a zone for each node where the node is the only node in its zone, and the type of the zone is based on the location of the node. Then, the logical zones are built in two steps, *merging* and *expanding*.

Merging: We denote the neighbor set of v as $\mathcal{N}_v(\tau)$. Two symbols, h_v and c_v, are used to represent the hot or cold

zone that contains the node v, respectively. As shown in lines 7–16, we randomly choose a node v from V', where V' is initialized from V. A hot zone h_v or a cold zone c_v is then created to contain node v, with the zone type depending on the location of node v. Then, we consider each neighbor $u \in \mathcal{N}_v(\tau)$ of v. If v and u are in different zones but the zone type is the same, we merge their zones into one zone with the same zone type. Otherwise, we continue to consider another neighbor until all nodes in $\mathcal{N}_v(\tau)$ are considered. For example, if $h_v \neq h_u$, we have $H(\tau) = H(\tau) - \{h_v\} - \{h_u\}, h_+ = h_v \cup h_u$ and $H(\tau) = H(\tau) \cup \{h_+\}$. Finally, we remove the nodes in the merged zone from V'. We repeat the above process until V' is empty, which indicates all satellites during τ are considered and the logical zone sets $H(\tau)$ and $C(\tau)$ are constructed. Considering the large scale of LEO satellite networks, we use two thresholds, γ_H and γ_C, to restrict the size of merged hot and cold zones in lines 11 and 14, respectively, accelerating the logical zone topology construction in the highly dynamic environment.

Expanding: A dish located on the coast may select a LEO satellite node as the access point, where the coordinate of the node is in the ocean but very close to land. The node should be treated as in a hot zone. To address this, we expand each hot zone in lines 17–20 by moving the adjacent node that is in a cold zone to the hot zone.

D. Sustainable and Reliable Topology Construction

Our goal here is to construct a sustainable and reliable topology $G'(\tau) = (V, E'(\tau))$ by selecting a subset of edges from $E(\tau)$. We construct the topology by taking each logical zone $A = (V_A(\tau), E_A(\tau))$ as input, where $V_A(\tau) \in H(\tau) \cup C(\tau)$ and $E_A(\tau)$ is a set of edges between the nodes $u, v \in V_A(\tau)$ within the zone A.

Cross-zone edge selection: As shown in lines 22–24 in Algorithm 1, given any satellite v in a logical zone A and another neighbor logical zone A', we add the cross-zone edge with the minimum weight that is incident to v and u in A' (where $y_l^v(\tau) = y_l^u(\tau) = 1$) to $E'(\tau)$. This process is repeated until all logical zones are considered.

In-zone edge selection: The problem is a special case of the classic k-edge-connected graph with a minimum sum of weights, where we require there exist at least k_{uv} edge-disjoint paths between any pair of nodes in the same zone, i.e., $u, v \in V_A(\tau)$, and k_{uv} is an adaptive variable defined in (7) and (8). In a nutshell, for each logical zone A, we first find $\overline{G}_A(\tau) = (V_A(\tau), \overline{E}_A(\tau))$ with the minimum weights based on $\{e_l(\tau)|\forall l \in E(\tau)\}$, which guarantees that there exist at least $\overline{k} = \min_{u,v \in V_A(\tau)} k_{uv}$ edge-disjoint paths between any pair of nodes, which will be discussed in *Construction of* $\overline{E}_A(\tau)$ and *Construction of* $\overline{G}_A(\tau)$ later. Then, we greedily add edges in $\overline{E}_A(\tau)$ so as to satisfy the constraints (7) and (8) for any pair of nodes, which will be discussed in *Update of* $\overline{G}_A(\tau)$ later.

More specifically, as shown in Algorithm 2, we take the logical zone A, the weight set of edges, as well as the connectivity requirements, including k_{uv} and $E_{uv}^{\max}(\tau)$, as inputs. The latter represents the edge set that contains all edges to construct $k_{uv}^{\max}$-edge joint paths, where we can easily obtain $k_{uv}^{\max}$ using the classic max-flow min-cut theorem [48], by duplicating each edge into two directed edges with unit capacity. Note that the k_{uv} is *initialized* as $\beta k_{uv}^{\max}$ (resp. 1) in hot zones (resp. cold zones). *To jointly improve sustainability and reliability, it varies with the ranges defined in (7) and (8).* In detail, if an edge of any path between u and v fails in the previous interval, i.e., $\tau - 1$, and it is selected by some traffic, we update it by $k_{uv} = \min(k_{uv} + 1, k_{uv}^{\max})$ for hot zones and $k_{uv} = \min(k_{uv} + 1, \beta k_{uv}^{\max})$ for cold zones. Otherwise, we have $k_{uv} = \max(k_{uv} - 1, \beta k_{uv}^{\max})$ (resp. $k_{uv} = \max(k_{uv} - 1, 1)$) for hot zones (resp. cold zones).

Algorithm 2: Sustainable and Reliable Topology Construction.

Input: $A, \{e_l(\tau)|\forall l \in E_A(\tau)\}, \{k_{uv}, E_{uv}^{\max}(\tau)|u, v \in V_A(\tau)\}$
Output: $A' = (V_A(\tau), \overline{E}_A(\tau))$

1: $\overline{k} = \min_{u,v \in V_A(\tau)} k_{uv}$ //*Build* $\overline{E}_A(\tau)$
2: $\overline{E}(\tau) = \emptyset, \tilde{E}_A(\tau) = E_A(\tau)$
3: **while** $\exists v \in V_A(\tau)$ is incident to $< \overline{k}$ edges in $\overline{E}(\tau)$ **do**
4: $\quad l = \arg\min_{l \in \tilde{E}_A(\tau)} e_l(\tau)$
5: $\quad \overline{E}(\tau) = \overline{E}(\tau) \cup \{l\}, \tilde{E}_A(\tau) = \tilde{E}_A(\tau) - \{l\}$
6: $E_0(\tau) = \emptyset, \overline{E}_A(\tau) = E_A(\tau)$
7: Sort $E_A(\tau) \backslash \overline{E}(\tau)$ in non-increasing order of e_l
8: **for** $l \in E_A(\tau) \backslash \overline{E}(\tau)$ **do** //*Build* $\overline{G}_A(\tau)$
9: $\quad$ **if** l is critical in $\overline{G}_A(\tau) = (V_A(\tau), \overline{E}_A(\tau))$ **then**
10: $\quad\quad E_0(\tau) = E_0(\tau) \cup \{l\}$
11: $\quad$ **else**
12: $\quad\quad \overline{E}_A(\tau) = \overline{E}_A(\tau) - \{l\}$
13: We get $\overline{E}_A(\tau)$ that is equal to $E_0(\tau) \cup \overline{E}(\tau)$
14: $K(\tau) = \{\overline{k}_{uv} = k_{uv} - \overline{k} | \overline{k}_{uv} \neq 0, \forall u, v \in V_A(\tau)\}$ //*Update* $\overline{G}_A(\tau)$
15: **while** $K(\tau) \neq \emptyset$ **do**
16: $\quad E_{uv}(\tau) = E_A(\tau), G_{uv}(\tau) = (V_A(\tau), E_{uv}(\tau))$
17: $\quad u, v = \arg\min_{u,v \in V_A(\tau)} \overline{k}_{uv}$
18: $\quad E_{uv}(\tau) = E_{uv}(\tau) - \{l | l \in E_{uv}^{\overline{k}}(\tau)^6\}$
19: $\quad$ **if** Cannot find $\overline{k}_{uv}$ edge-disjoint paths on $G_{uv}(\tau)$ by the max-flow min-cut theorem **then**
20: $\quad\quad E_{uv}(\tau) = E_{uv}(\tau) \cup E_{uv}^{\max}(\tau)$
21: $\quad\quad$ Find $\overline{k}_{uv}$ edge-disjoint paths on $G_{uv}(\tau)$ again
22: $\quad \overline{E}_A(\tau) = \overline{E}_A(\tau) \cup \{l | l \in E_{uv}^{\overline{k}_{uv}}(\tau)^6\}$
23: $\quad K(\tau) = K(\tau) - \{\overline{k}_{uv}\}$, update $K(\tau)$ by $\overline{E}_A(\tau)$

Construction of $\overline{E}_A(\tau)$*:* We first get $\overline{k}$ in line 1, which is the minimum k_{uv} among the pairs of nodes in $V_A(\tau)$. Then, we build the minimum $\overline{G}_A(\tau)$ by finding the minimum k-edge-connected graph on A in lines 3–5. Specifically, we greedily find an edge set $\overline{E}(\tau) \subseteq E_A(\tau)$ with minimum cardinality, where each node $v \in V$ is incident to at least $\overline{k}$ edges in $\overline{E}(\tau)$. In line 6, we initialize an empty edge set $E_0(\tau)$ and a graph $\overline{G}_A(\tau) = (V_A(\tau), \overline{E}_A(\tau))$ with $\overline{E}_A(\tau) = E_A(\tau)$, where $\overline{G}_A(\tau)$ is a k-edge-connected graph.

Definition 1: Given a graph $G(\tau) = (V, E(\tau))$, a constant $\overline{k}$ and two endpoints $u, v \in V$ of edge $l \in E(\tau)$, we say l is critical if there exist $\overline{k}$ and at most $\overline{k} - 1$ edge-disjoint path between u and v before and after l is removed. Otherwise, l is not critical.

Construction of $\overline{G}_A(\tau)$*:* We then carefully remove all non-critical edges, which are defined in Definition 1, in $\overline{G}_A(\tau)$ to minimize the weights. As shown in lines 7–13 of Algorithm 2, we consider all edges $l \in E_A(\tau) \backslash \overline{E}(\tau)$ in an arbitrary order. If l is critical, we keep it in $\overline{E}_A(\tau)$ and add it into $E_0(\tau)$. Otherwise, we simply remove it from $\overline{E}_A(\tau)$. After this, we get $\overline{E}_A(\tau) = E_0(\tau) \cup \overline{E}(\tau)$ and obtain the minimum k-edge-connected graph $\overline{G}_A(\tau) = (V_A(\tau), \overline{E}_A(\tau))$.

Update of $\overline{G}_A(\tau)$: Since there may exist node pairs, i.e., u and v, with $k_{uv}>\overline{k}$, we initialize a set $K(\tau)=\{\overline{k}_{uv}=k_{uv}-\overline{k}|\overline{k}_{uv}\neq 0, \forall u,v\in V_A(\tau)\}$ in line 14. While there still exist elements in $K(\tau)$, we initialize a graph $G_{uv}(\tau)=(V_A(\tau),E_{uv}(\tau))$ in line 16, where $E_{uv}(\tau)=E_A(\tau)$. We consider the node pair with the minimum $\overline{k}_{uv}$ in line 17 and remove all edges in $E^{\overline{k}}_{uv}(\tau)$[6] in line 18.

It is possible that the removed edges in $E^{\overline{k}}_{uv}(\tau)$ belong to more than two pairs of nodes in the final topology that have k_{uv} edge-disjoint paths. Thus, we first use the max-flow min-cut theorem to check whether $\overline{k}_{uv}$ edge-disjoint paths still exist after the removal in line 19. If it does not, we add all edges in $E^{\max}_{uv}(\tau)$ to $E_{uv}(\tau)$ and find the $\overline{k}_{uv}$ edge-disjoint paths with minimum weights from $G_{uv}(\tau)$ again in lines 20–21. Otherwise, we add edges $l\in E^{\overline{k}_{uv}}_{uv}(\tau)$[6] to $\overline{E}_A(\tau)$ in line 22. The added edges may also influence the connectivity of $\overline{G}_A(\tau)$. Thus, we update $K(\tau)$ in line 23 by removing the minimum element and recalculate $\overline{k}_{uv},\forall u,v\in V_A(\tau)$ based on $\overline{E}_A(\tau)$.

Note that $E^{\max}_{uv}(\tau)$ contains all edges of $k^{\max}_{uv}$ edge-disjoint paths ($E^{\overline{k}_{uv}}_{uv}(\tau)\subset E^{\max}_{uv}(\tau)$), which ensures that we can find $\overline{k}_{uv}$ edge-disjoint paths in line 21. The above process in lines 15–23 is repeated until $K(\tau)$ is empty and we obtain the sustainable and reliable topology in logical zone A. After all logical zones in $H(\tau)\cup C(\tau)$ are considered, we arrive at the final topology.

Sustainability gains: while we ensure the existence of k_{uv} edge-disjoint paths between any pair of nodes, our TASRI primarily gains sustainability from three aspects.

First, we formulate the node and edge weights predicated on depth-of-discharge and energy consumption, both of which significantly influence battery lifespan. By utilizing these weights as inputs, our ODTE algorithm minimizes their sum by carefully selecting only a subset of candidate ISLs in each interval. *Second*, the provision of ample edge-disjoint paths also allows applications to choose a route that incorporates nodes with the lowest depth-of-discharge ratio, enhancing sustainability at the application level. *Finally*, the traffic-aware path number update approach, such as $k_{uv}=\max(k_{uv}-1,1)$, further mitigates link maintenance cost by establishing fewer paths. Before we further examine the sustainability gain of our TASRI in Section V via extensive evaluations, we first provide a theoretical analysis of our solution in the next subsection.

E. Theoretical Analysis

Theorem 2: The Algorithm 2 builds a k-edge-connected graph $\overline{G}_A(\tau)$ in lines 7–13.

Proof: Since $\overline{E}_A(\tau)$ is initialized as $E_A(\tau)$, we can see that $\overline{G}_A(\tau)$ is k-edge-connected at the beginning. After removing some non-critical edges from $\overline{E}_A(\tau)$, the graph $\overline{G}_A(\tau)$ is still k-edge-connected. Thus, we need to show that $E_0(\tau)\cup\overline{E}(\tau)\subseteq\overline{E}_A(\tau)$ and $\overline{E}_A(\tau)\subseteq E_0(\tau)\cup\overline{E}(\tau)$.

On the one hand, since initially we let $\overline{E}_A(\tau)=E_A(\tau)$ and only considers the edges in $E_A(\tau)\backslash\overline{E}(\tau)$, we have $l\in\overline{E}_A(\tau),\forall l\in\overline{E}(\tau)$. Thus, we get $\overline{E}(\tau)\subseteq\overline{E}_A(\tau)$. During the construction of $\overline{E}_A(\tau)$, if the edge is critical, we put it into $E_0(\tau)$ and keep it in $\overline{E}_A(\tau)$. So, we have $l\in\overline{E}_A(\tau),\forall l\in E_0(\tau)$ and $E_0(\tau)\subseteq\overline{E}_A(\tau)$. Combining the above two observations, we get $E_0(\tau)\cup\overline{E}(\tau)\subseteq\overline{E}_A(\tau)$.

On the other hand, we prove $\overline{E}_A(\tau)\subseteq E_0(\tau)\cup\overline{E}(\tau)$ by contradiction. We assume $\exists l\in\overline{E}_A(\tau), l\notin E_0(\tau)\cup\overline{E}(\tau)$. Thus, we have $l\notin\overline{E}(\tau)$ and $l\in E_A(\tau)\backslash\overline{E}(\tau)$. Based on the assumption, we have $l\notin E_0(\tau)$ which means that l is non-critical. From Algorithm 2, l should be removed from $\overline{E}_A(\tau)$ which contradicts with the assumption that $l\in\overline{E}_A(\tau)$. □

Theorem 2, together with the centralized algorithmic logic implemented in lines 16–22 of Algorithm 2, formally guarantees the existence of k_{uv} edge-disjoint paths between any node pair $(u,v)\in V_A(\tau)$ through rigorous graph-theoretic construction, without relying on a supervisory mechanism. This theoretical assurance forms a solid foundation for topology reliability. Furthermore, the empirical evaluation will demonstrate the topology's operational robustness under dynamic network conditions.

Theorem 3: The time complexity of the ODTE algorithm is $O(|E|(|V|+|F_M|)+|V^M_A|^5|E^M_A|^2)$, where $|V^M_A|=\max_{A\in H\cup C}|V_A|$, $|E^M_A|=\max_{A\in H\cup C}|E_A|$ and $|F_M|$ is the maximum number of flows.

Proof: We omit τ in some notations for brevity in the proof. In the edge weight normalization stage, we need $O(|V||E|)$ to determine the degree of nodes. It takes $O(|E|)$ and $O(|E||F_M|)$, where $|F_M|$ is the maximum number of flows of all intervals, to amortize the node weight and effects of traffic to the edge weight, respectively.

In the logical zone construction stage, it needs to consider every node in V in the worst case, which costs $O(|V|)$. In the topology construction stage, the algorithm can be run in each logical zone simultaneously. It takes $O(|E|)$ to select the edges that connect different logical zones.

Then we run Algorithm 2 for each zone in $H\cup C$. To build $\overline{E}$, we need at most $O(|V_A|^3|E_A|^2)$ and $O(|E_A|)$ iterations to find $\overline{k}$ and $\overline{E}_A$, respectively. To build $\overline{G}_A(\tau)$, it takes $O(\overline{k^M}^3|V^M_A|^2+|E^M_A|^{1.5}(\log|V^M_A|^2))$ computations based on [52], where $|V^M_A|=\max_{A\in H\cup C}|V_A|$, $|E^M_A|=\max_{A\in H\cup C}|E_A|$ and $\overline{k^M}=\max_{u,v}k_{uv}$.

Finally, to update $\overline{G}_A(\tau)$ until the connectivity requirements are satisfied, we need to consider every element in K with the maximum size of $O(|V_A|^2-1)$. We can sort K to find the minimum $\overline{k}_{uv}$ costs in line 17, which costs $O(|V_A|\log|V_A|)$. The edge-disjoint path calculation and update of K takes $O(|V_A||E_A|^2)+O((|V_A|^2-1)*|V_A||E_A|^2)=O(|V_A|^3|E_A|^2)$ iterations in total. Based on the above analysis, the update of $\overline{G}_A(\tau)$ needs $O(|V^M_A|^5|E^M_A|^2)$ iterations. To sum up, the ODTE algorithm costs $O(|E|(|V|+|F_M|)+O(|V_A|^3|E_A|^2)+O(|E_A|)+O(\overline{k^M}^3|V^M_A|^2+|E^M_A|^{1.5}(\log|V^M_A|^2))+(|V_A|^5|E_A|^2)=O(|E|(|V|+|F_M|)+\overline{k^M}^3|V^M_A|^2+|E^M_A|^{1.5}(\log|V^M_A|^2)+|V^M_A|^5|E^M_A|^2)=O(|E|(|V|+|F_M|)+|V^M_A|^5|E^M_A|^2)$. □

Despite that our algorithm's complexity involves the fifth power of the node quantity within a zone, it still has *good scalability* since the topology construction is designed for *parallel* execution across all zones. This design, coupled with the zone size restrictions for each type as detailed in Algorithm 1, i.e., γ_H and γ_C, upholds the high efficiency of the algorithm.

[6] $E^{\overline{k}}_{uv}(\tau)\subseteq E_{uv}(\tau)$ (resp. $E^{\overline{k}_{uv}}_{uv}(\tau)\subseteq E_{uv}(\tau)$) is the edge set of $\overline{k}$ (resp. $\overline{k}_{uv}$) edge disjoint paths between u and v, $\forall u,v\in V$.

Theorem 4: The approximation ratio of the ODTE algorithm is $(1+\frac{2}{\overline{K}+1})\overline{A}$, where $\overline{K}=\min_{u,v\in V_A(\tau),V_A(\tau)\in H(\tau)\cup C(\tau)} k_{uv}$ is the minimum number of edge-disjoint paths among all zones and $\overline{A}$ is the number of zones.

Proof: The edge weight normalization and logical zone construction can accurately get the edge weight and mobility-based zones, respectively. Similarly, the selection of cross-zone ISLs does not cause any gap between the final output and optimal value. During the in-zone ISL construction as shown in Algorithm 2, the building of $\overline{E}_A(\tau)$ and $\overline{G}_A(\tau)$ has an approximation ratio of $1+\frac{2}{\overline{k}+1}$ based on [52], where $\overline{k}$ is the number of edge-disjoint paths. Then, in the $\overline{G}_A(\tau)$ update stage, the worst case is when $K(\tau)=\emptyset$, which means that no pairs $u,v\in V_A(\tau)$ need to find additional edge-disjoint paths. Thus, the approximation ratio is $(1+\frac{2}{\overline{K}+1})\overline{A}$, where $\overline{K}=\min_{u,v\in V_A(\tau),V_A(\tau)\in H(\tau)\cup C(\tau)} k_{uv}$ and $\overline{A}$ is the number of zones. □

V. Performance Evaluations

A. Simulation Setup

1) Parameter Settings: We use the Starlink first shell constellation in our simulations, which has 72 orbits and each orbit has 22 satellites operating at 550 km altitude with 53° inclination angle [53]. The maximum distance for an ISL between two satellites operating at 550km altitude is 5,014 km [4]. The maximum number of ISLs per satellite κ_v is 8 [43]. Each ISL has a failure probability between 0 and 0.05 [54], and each zone can have at most 30 satellites to increase scalability. We set the traffic forwarding factor ϵ_T as 0.4 to quantify the effects on depth-of-discharge and the amount of forwarded traffic on the node weight. We set $d_l^{PAT}(\tau)$ for all links as the average PAT delay 100 ms based on [14]. Based on [15], we take the peak power of the laser transceiver as 40 watts. The PAT-related energy consumption for a single ISL switching is 40 watts $\times$ 0.1 s = 0.0067 watt minutes. Hence, we set the cost of switching ISL ϵ_S as 0.067. We set ξ, the number of interested intervals to initialize zones, as 5. We set the interval at which the ground-based control center collects status and updates configuration as 1 minute based on [55] and [56].

The flows are generated based on Poisson distribution. Hot zones (resp. cold zones) have 20 (resp. 1) expected flow(s) per interval. The number of expected flows is adjusted for hot zones to represent different traffic intensities. Based on the flow intensity, we set the reliable constant β as 0.2 to distinguish different zones. We assume each flow contains the same amount of traffic, thus consuming the same amount of traffic forwarding energy per interval for each satellite. Each simulation is run for 30 intervals.

We use the python library *PyEphem*[7] and the Two-Line Element (TLE) set of Starlink and other satellite networks to build the simulations. We generate the TLE sets for the satellite networks using the code from Hypatia [3] based on parameters, such as their altitude, number of orbits, number of satellites per orbit, and inclination angle. The TLE sets describe the orbital characteristics of the satellite networks, which can be used to simulate the satellite mobility.

[7]https://rhodesmill.org/pyephem/

2) Comparison Schemes and Performance Metrics:

- E+Grid [8] extends the classic +Grid topology, in which each satellite statically establishes two intra-orbit ISLs with the two closest neighboring satellites in the same orbit and six inter-orbit ISLs with three closest neighboring satellites from two adjacent orbits.
- LAEE [9] considers edge weights, which is a weighted sum of propagation delay and energy consumption in terms of ISL maintenance and switching, from the previous interval and dynamically builds a topology by forming a spanning tree with the minimum weight.
- $\times$Grid [20] takes into account the switching between the ground station and satellites to construct the topology, aiming to ensure that after the switch, the new path includes ISL without requiring detours.
- UltraDense [34] aims to minimize the number of satellites to achieve full ground station coverage. In other words, it maximizes the number of shared nodes on the paths between any two ground stations.

We examine various factors including sustainability, reliability, traffic adaptation, algorithm overhead, and scalability. All compared schemes are executed directly on the whole graph without the awareness of hot and cold zones.

B. Results and Analysis

1) Sustainability: The sum of weights is an important metric for measuring sustainability because it is tailored for LEO satellite networks by customizing the importance of different factors, including the depth-of-discharge of the battery, link maintenance energy consumption and node switch energy consumption, providing a comprehensive assessment and facilitating continuous improvement in various aspects of the LEO satellite networks.

Sum of weights: As shown in Fig. 6(a), our TASRI achieves the *lowest* and *stable* weight because TASRI can dynamically activate or deactivate some ISLs based on the energy consumption of nodes and edges during each interval. In addition, TASRI estimates the duration of existing flows based on the available network resources such as the bandwidth of ISLs and the movement pattern of satellites. As a result, the number of edge-disjoint paths k_{uv} for each end-point pair (u,v) keeps changing and well adapts to the varying traffic conditions. This offers the upper-layer application more opportunity to choose a more sustainable routing path, for example, a path contains nodes with low depth-of-discharge ratios.

On the contrary, as the first interval lacks any flow history, the LAEE scheme experiences a weight spike at the second interval and constructs an underperforming topology. Similarly, both E+Grid and $\times$Grid construct relatively static topologies by incorporating redundant ISLs and altering the direction of ISLs, resulting in a higher weight. Although UltraDense minimizes the number of satellites while ensuring global coverage, it may lead to load concentration on certain nodes, causing their weights to increase rapidly.

Life consumption: Consuming energy at a lower battery level may have a higher negative impact on the battery life [6] where the battery life is the number of cycles the battery can be completely discharged. As shown in Fig. 6(b), compared to other schemes, our TASRI framework significantly reduces life consumption, since it directly considers the depth-of-discharge ratio in the node weight model. On the other hand, in each interval, TASRI also models the link delay and bandwidth as link

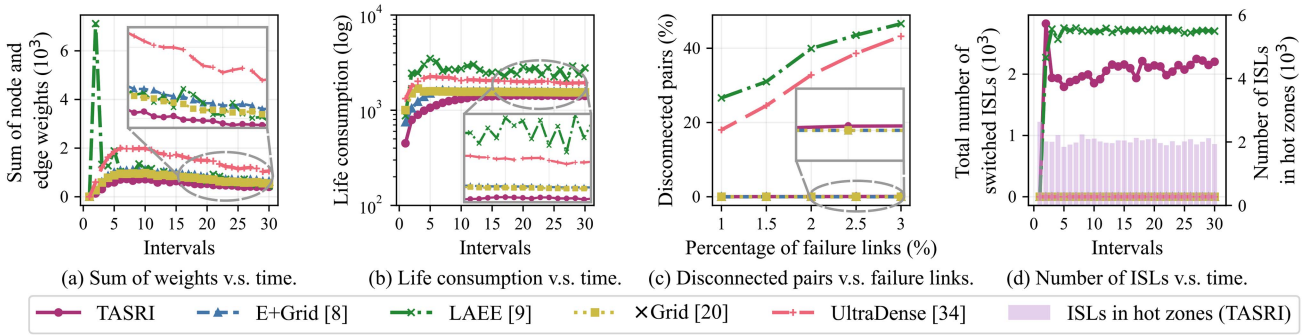


Fig. 6. Performance comparison in terms of weight, sustainability, reliability and number of ISLs.

weight, optimizing the transmission performance of upper-layer applications in routing selection. Over multiple intervals, to reduce the impact of frequent link switching on ongoing traffic flows, it also normalizes the link weight by estimating the duration of the flows. As a result, TASRI efficiently solves the on-demand ISL provisioning problem by minimizing the sum of node weights and link weights.

The LAEE scheme has high variation in life consumption because the minimum-spanning-tree topology has only one edge-disjoint path. For example, suppose there are multiple flows between a pair of nodes. In that case, these flows might choose the same path from a topology constructed by LAEE, resulting in a sharp increase in the depth-of-discharge at each node on the path, which greatly increases the life consumption of these nodes. In contrast, the battery levels of nodes not on the path remain almost unchanged. When the topology changes, the flows might select a new path in the updated topology that includes many nodes with high depth-of-discharge, thereby significantly increasing their life consumption. The relatively static E+Grid and ×Grid schemes do not adapt well to traffic variation. The additional link maintenance costs and uneven node load increase the depth-of-discharge, resulting in higher life consumption. UltraDense's high life consumption is also due to the limited number of nodes, which results in higher and uneven overall load distribution.

2) Reliability: Considering the unstable wireless environment in outer space, one may wonder whether the topology built by our TASRI is robust. In other words, whether pairs of nodes are still connected when some links fail since we only choose a subset of the candidate ISLs to build the topology. We conduct another experiment by randomly removing some edges to simulate unreliable ISLs and count the number of flows that have no path between their end-point pairs.

Fig. 6(c) shows the percentages of disconnected pairs for our TASRI and other schemes at each interval. The topology built by E+Grid always has the best robustness since it activates all candidate ISLs. Besides, ×Grid also performs best since it only changes the direction of ISLs based on E+Grid. The *redundant* links increase the probability that there exists a path between any pair of nodes. Despite having fewer edges, our TASRI has almost the same reliability compared to the E+Grid and ×Grid schemes since TASRI dynamically adjusts the number of edge-disjoint paths between end-point pairs to ensure connectivity.

More specifically, our TASRI incorporates an adaptive mechanism whereby it establishes additional ISLs in the current interval to enhance reliability in the event of an ISL failure in the previous interval. On the other hand, if the link remains stable in the previous interval, our TASRI may decrease the number of edge-disjoint paths during the topology construction in this interval, so that the overall life consumption can be improved. This adaptive mechanism ensures reliable connectivity between any pair of nodes in the error-prone wireless space environment.

Conversely, the LAEE scheme has the worst performance due to its minimum-spanning-tree topology, as a failed ISL can easily break the paths of many end-point pairs. Similarly, UltraDense's greedy approach weakens its fault tolerance when ISL fails.

3) Scalability: We conduct experiments to show the scalability of TASRI. Fig. 7(a) shows that TASRI maintains the lowest life consumption across different satellite numbers, with life consumption increasing linearly as the network scales. TASRI achieves this by dividing the global topology into two types of logical zones, adjusting the number of ISLs based on zone type to minimize redundant ISLs in idle areas like oceans. Additionally, TASRI dynamically adjusts ISLs based on nodes' DoD ratios and network load, avoiding high-consumption nodes. However, in LAEE and UltraDense, some paths may be shared by certain end-to-end pairs, causing uneven load distribution. This leads to a rapid increase in the DoD of nodes on these common paths, resulting in near-exponential growth in life consumption.

Fig. 7(b) shows the average disconnected pairs with different numbers of satellites. Despite having fewer ISLs than E+Grid and ×Grid, TASRI maintains comparable reliability across network scales when ISLs fail. This is because TASRI ensures a specific number of edge-disjoint paths between nodes within each zone, adjusting based on previous link failures. If an ISL fails, TASRI adds ISLs in the next interval to reduce disconnection risk. E+Grid and ×Grid also maintain high reliability with *redundant* ISLs, whereas the tree-like topologies of LAEE and UltraDense are more vulnerable to failures, especially as the network scales up.

Fig. 7(c) shows the average end-to-end delay with different satellite numbers. TASRI, E+Grid, and ×Grid have comparable delays across network scales, while LAEE and UltraDense consistently show the highest delays. TASRI selects a subset of ISLs for its topology but optimizes for propagation delay and node load by incorporating them into the edge and node weights. This allows TASRI to construct efficient topologies with *fewer* ISLs and *lower* energy consumption than E+Grid and ×Grid. In contrast, both E+Grid and ×Grid activate all candidate ISLs, resulting in higher overhead to maintain minimal end-to-end delay. LAEE and UltraDense's tree-like topologies often route through distant common nodes, causing higher delays.

Fig. 7(d) shows the weights with different satellite numbers. TASRI consistently has smaller average weights than E+Grid

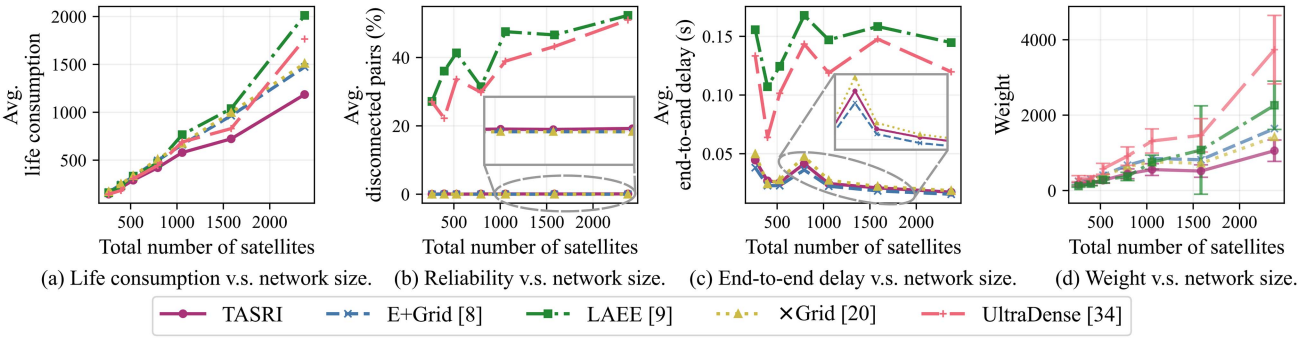


Fig. 7. Performance comparison with different network sizes.

and ×Grid across network scales due to having fewer ISLs. However, TASRI shows larger weight fluctuations because, unlike the more static E+Grid and ×Grid, TASRI dynamically adjusts ISLs at the start of each interval based on load, node mobility, and previous ISL failures, leading to these variations.

As the network scale grows, the average weights of LAEE and UltraDense surpass TASRI. This is due to many end-to-end node pairs sharing ISLs in their topologies, with these shared ISLs increasing rapidly as the network expands. Since node weights depend exponentially on depth-of-discharge, this increase accelerates with network size. Additionally, LAEE and UltraDense show the largest weight fluctuations at larger scales due to significant changes in topology between time slots.

4) Traffic Adaptation: Number of (switched) ISLs. Fig. 6(d) illustrates the number of switched ISLs and ISLs in hot zones for TASRI. The switched ISLs are the ISLs that have a change in their switch-on/off status in the current interval compared to the previous interval. Our TASRI has a smaller number of switched ISLs in comparison to the LAEE scheme since we account for the impact of switching ISLs on flows and associated costs. On the other hand, the E+Grid and ×Grid schemes have zero switched ISLs for all intervals since they build a stable topology without ISL switching. Moreover, our TASRI maintains a stable number of ISLs even though only a subset of candidate ISLs is activated or deactivated in hot zones to satisfy traffic demands.

Weights v.s. traffic intensity: To further illustrate the traffic adaptation of TASRI, we compare the average weight and life consumption under different traffic intensities, which represent the average number of flows in hot zones. As shown in Fig. 8(a), TASRI achieves the lowest average weight due to the adaptive amortization of node weights to edge weights and normalization of edge weights based on traffic flow. In contrast, both E+Grid and ×Grid schemes have higher weights since they have more redundant ISLs. Besides, LAEE and UltraDense also have higher weights since the tree-like topology provides less path diversity for the upper application, causing some nodes to become the common hot spot for many selected paths and exponentially increasing the node weights due to load imbalance.

Life consumption v.s. traffic intensity: According to Fig. 8(b), our TASRI framework has the lowest life consumption under different traffic intensities. This demonstrates the sustainability and flexibility of our TASRI framework in handling varying traffic volumes, especially when the traffic intensity is high. This is because our TASRI uses the depth-of-discharge ratios of the nodes as weights, and on this basis, TASRI models and solves the problem of minimizing both node and link weights. As the load increases, the node weights will exponentially increase due to the increasing depth-of-discharge ratios. Therefore, our TASRI strives to avoid constructing topologies that would rapidly increase these weights, thereby optimizing the overall life consumption of the topology. In contrast, both LAEE and UltraDense schemes show worse sustainability due to the uneven increase in node loads caused by greedily establishing a tree-like topology, resulting in a deeper battery discharge and a higher life consumption.

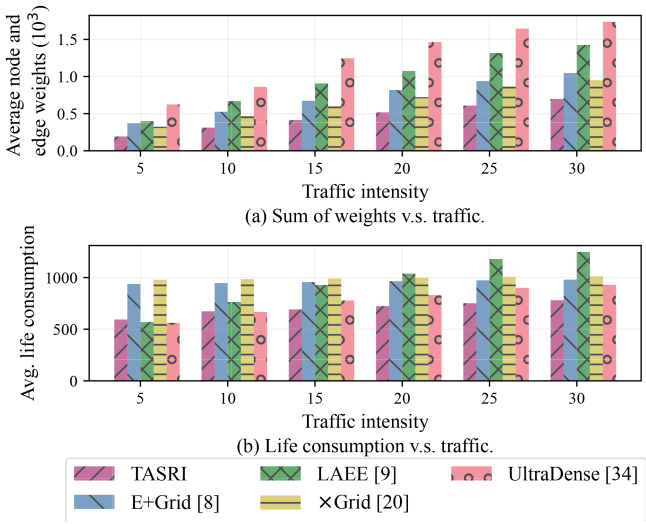


Fig. 8. Weight and life consumption with traffic intensity.

TABLE I
ADAPTATION TO UNEXPECTED CONTROL TRAFFIC

Performance metrics	Unexpected control traffic (%)			
	1%	2%	3%	4%
Avg. life consumption	530.22	561.93	568.00	595.99
Avg. disconnected path (%)	0.00	0.00	0.00	0.00
Avg. E2E delay (ms)	23.58	23.61	22.10	22.14

Adaptation to unexpected control traffic: We incorporated varying proportions of unexpected control traffic in the simulations, as shown in Table I. The results indicate that TASRI's performance metrics remain relatively stable despite these additions. While TASRI does not explicitly account for bursty traffic during the topology construction phase, unexpected traffic influences the battery's DoD, which in turn affects the node weight at the start of the next interval. TASRI, designed to minimize weight, inherently adapts to such changes. Moreover,

TABLE II
AVERAGE PERFORMANCE METRICS IN NEAR-POLAR CONSTELLATIONS (PARENTHESES SHOW % WORSE THAN TASRI)

Constellations	Schemes	Average life consumption	Average disconnect paths (%)	Average end-to-end delay (ms)
Iridium	TASRI	35.79 (+0.00%)	0.00 (+0.00%)	41.1 (+0.00%)
	E+Grid [8]	41.78 (+16.73%)	0.00 (+0.00%)	39.67 (-3.48%)
	LAEE [9]	36.57 (+2.18%)	10.17 (+10.17%)	105.91 (+157.69%)
	×Grid [20]	42.46 (+18.64%)	0.00 (+0.00%)	40.84 (-0.63%)
	UltraDense [34]	37.51 (+4.9%)	9.54 (+9.54%)	79.33 (+93.02%)
OneWeb	TASRI	327.06 (+0.00%)	0.00 (+0.00%)	24.6 (+0.00%)
	E+Grid [8]	373.61 (+14.23%)	0.00 (+0.00%)	23.3 (-5.29%)
	LAEE [9]	406.08 (+24.16%)	33.09 (+33.09%)	120.61 (+390.28%)
	×Grid [20]	379.17 (+15.93%)	0.00 (+0.00%)	23.9 (-2.85%)
	UltraDense [34]	342.13 (+4.6%)	24.3 (+24.3%)	92.7 (+276.83%)

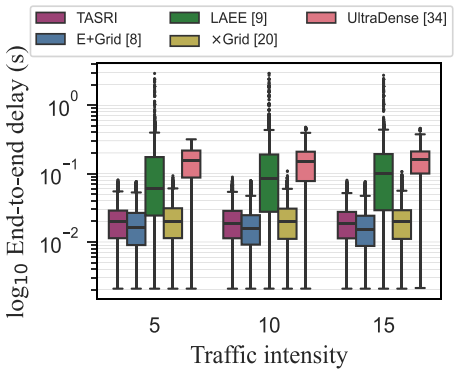


Fig. 9. End-to-end delay v.s. traffic intensity.

TASRI ensures that the number of edge-disjoint paths between any two nodes remains above a predefined threshold, effectively mitigating congestion caused by traffic surges.

5) Adaptation to Near-Polar Constellations: As shown in Table II, we evaluated the performance of TASRI under two additional near-polar constellations, Iridium and OneWeb, with inclination angles of 86.4° and 87.9°, respectively. The results indicate that TASRI still outperforms other schemes in terms of life consumption while maintaining comparable connectivity performance to E+Grid and ×Grid. Despite the slightly increased End-to-End (E2E) delay compared to E+Grid and ×Grid, the trade-off is still acceptable given the significant benefits in battery life consumption.

6) Overhead, Convergence, and Service Continuity: Overhead in end-to-end delay. The topology with a subset of candidate ISLs tends to have a higher E2E delay due to longer paths. However, as shown in Fig. 9, our TASRI framework achieves a stable and similar E2E delay compared to the E+Grid and ×Grid schemes despite having much fewer redundant ISLs. This is attributable to our traffic-aware TASRI framework, which is proficient in choosing critical ISLs based on movement patterns of the satellites and features of flows. More specifically, our TASRI incorporates the propagation delay in the link weight model and the load condition of nodes in the node weight model. On the other hand, our TASRI also estimates the duration of flows so that the interruption effects of ISL switching on existing flows can be also normalized into the link weight model. Finally, the sum of the link and node weight is minimized by TASRI, resulting in a relatively stable and good end-to-end delay performance as shown in the figure.

TABLE III
PERCENTAGES OF SWITCHED ISLS AND CONTROL PLANE OVERHEAD PER INTERVAL. GSL CAPACITY: 2 GBPS [55]

Number of satellites	1056	1584	2376
Switched ISLs (%)	20.18%	21.16%	21.47%
GSL bandwidth utilization (%)	0.011%	0.016%	0.024%
TASRI runtime (s)	1.05	1.22	1.23

Conversely, due to its short-sighted traffic consideration, the LAEE scheme shows an unstable E2E delay, causing frequent ISL changes and traffic interruptions. Besides, both LAEE and UltraDense have a higher end-to-end delay compared to the other two schemes, because the tree-like topology may require even closely located points to detour a long distance to reach each other. For example, the common parent node of two endpoints might be the root of the tree.

Switched ISLs and control plane overhead: As shown in Table III, the proportion of switched ISLs remains around 20% across different network sizes. This is because, while ISLs are dynamically activated and deactivated, we set the switching frequency to once per minute, which aligns with typical real-world ISL switch frequencies (e.g., [55] reports frequencies between 11.4 and 208.7 seconds). Furthermore, we assume the control signaling of TASRI uses 128-byte packets. In a single ground station scenario with a Ground-Satellite Link (GSL) bandwidth of 2 Gbps, the GSL bandwidth utilization is negligible ($< 0.025\%$), and it is even smaller in multi-ground station setups. The runtime on a 3.0GHz CPU (AMD EPYC 7313) is within 1.25 seconds, meaning the computation overhead in the control plane is almost imperceptible.

ODTE algorithm gap to optimality: Let us take a closer look at the ODTE algorithm, which is the core of TASRI. To obtain the optimal solution for the original problem, we employed the PuLP,[8] a linear and mixed integer programming solver written in Python. Due to the NP-hard nature of the problem and the solver's limited speed, we confined optimal solution tests to smaller-scale scenarios. As illustrated in Fig. 10, when the number of nodes exceeds 16, the optimal solution's runtime becomes prohibitive, exceeding even the 60-second interval duration.

Meanwhile, Fig. 11 demonstrates that ODTE consistently achieves performance very close to the optimal value, often outperforming the theoretical worst-case bound (gray dashed line). Two specific occurrences underscore both its efficiency and its robustness: first, the small gap between the ODTE

[8]https://github.com/coin-or/pulp

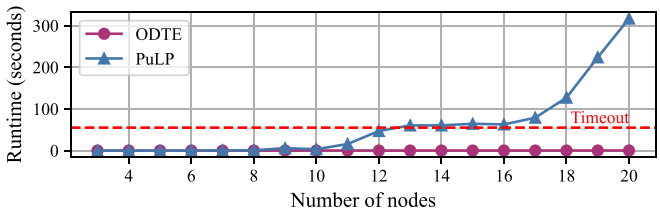


Fig. 10. The runtime of ODTE and the optimal solution.

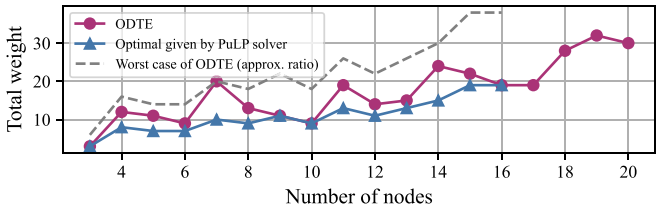


Fig. 11. The gap between ODTE and the optimal solution.

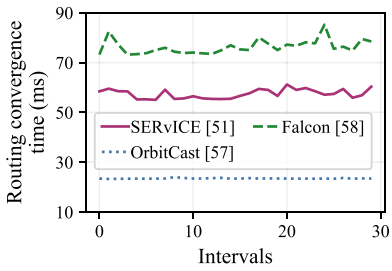


Fig. 12. Routing convergence time after reconfiguration.

curve and the optimal-solution baseline quantifies the minimal overhead of using a polynomial-time heuristic instead of an exact NP-hard solver—showing that, in practice, ODTE delivers almost the best possible topology with tractable runtime; second, the sizable margin by which ODTE exceeds the worst-case guarantee highlights its dependable behavior under extreme conditions, assuring designers that even in the least favorable scenarios, ODTE will not collapse to the pessimistic bound. Together, these two occurrences establish ODTE as a practically deployable algorithm that not only minimizes operational cost through near-optimal designs but also maintains strong performance guarantees under stress.

Routing convergence time: Routing convergence time refers to the duration from the moment a topology change triggers the routing algorithm to begin its computation until all nodes have completed their routing updates. It is a key metric reflecting the cost of topology reconfiguration. An effective on-demand ISL provisioning scheme should ensure that the convergence time remains stable and relatively small. In our study, we selected three routing algorithms: SERvICE [51], ORBITCAST [57], and Falcon [58], which represent centralized, distributed, and centralized multi-path routing approaches, respectively. As shown in Fig. 12, the convergence times of these algorithms were observed to remain stable, with a maximum value not exceeding 90 ms. This is because our proposed TASRI scheme constrains ISL provisioning to occur only at the beginning of each interval, ensuring that the topology remains stable during the interval, thus avoiding slower routing convergence due to frequent reconfigurations. Furthermore, during edge weight normalization, longer flow durations result in lower edge weights, reducing the likelihood of ISL closure and minimizing re-routing due to topology changes.

Traffic demand satisfaction: While our work primarily focuses on topology construction rather than Quality-of-Service (QoS) based routing, the proposed scheme inherently accounts for flow continuity requirements. The topology construction process considers the expected duration of active flows when making ISL activation decisions, aiming to minimize service disruptions. Specifically, during ISL deactivation, we prioritize preserving ISLs carrying existing flows to avoid unnecessary rerouting. This approach is evaluated by measuring the percentage of uninterrupted flows as an indirect metric of traffic demand satisfaction. Experimental results demonstrate that our method improves this metric from 9.15% to 24.05% compared to LAEE, indicating better preservation of active flows. This design philosophy naturally bridges topology construction with QoS considerations, though QoS-based routing optimization remains an interesting direction for future research.

VI. Conclusion

In this paper, we propose a novel *Traffic-Aware, Sustainable, and Reliable ISL provisioning (TASRI)* framework to deliver on-demand and scalable topology control service for LEO satellite networks. To implement this service-oriented framework, we designed a sustainability-oriented weight model tailored for the satellite topology control service, formulated the service optimization problem, and proposed a scalable *on-demand topology evolving (ODTE)* algorithm with rigorous theoretical guarantees. Extensive simulation results demonstrate that TASRI significantly improves sustainability, while maintaining excellent reliability, end-to-end delay, and superior scalability compared to state-of-the-art schemes, with considerably fewer ISLs or ISL handovers.

In the future, satellite networks will continue to grow in scale, and the range of supported services will become increasingly diverse. The on-demand perspective introduced by TASRI can be extended to other key aspects, such as latency reduction, and coverage improvement. This flexibility enables continuous topology optimization to meet evolving service demands and resource constraints, supporting the development of more robust and efficient satellite network infrastructures.

Acknowledgment

This paper extends our work published in IEEE/ACM IWQoS'24 [1].

References

[1] L. Chen et al., "TASRI: Toward traffic-aware, sustainable and reliable isl provisioning for LEO satellite constellation networking," in *Proc. IEEE/ACM 31st Int. Symp. Qual. Serv.*, 2024, pp. 1–6.

[2] H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "A survey on nongeostationary satellite systems: The communication perspective," *IEEE Commun. Surv. Tut.*, vol. 25, no. 1, pp. 101–132, Aug. 2023.

[3] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the "internet from space" with hypatia," in *Proc. ACM 20th Internet Meas. Conf.*, 2020, pp. 214–229.

[4] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *Proc. ACM 15th Int. Conf. Emerg. Netw. Exp. Technol.*, 2019, pp. 341–354.

[5] K. An, M. Lin, J. Ouyang, and W.-P. Zhu, "Secure transmission in cognitive satellite terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 3025–3037, Nov. 2016.

[6] Y. Yang, M. Xu, D. Wang, and Y. Wang, "Towards energy-efficient routing in satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3869–3886, Dec. 2016.

[7] J. Foust, "Starlink failures highlight sustainability concerns in LEO satellite networks," 2019. [Online]. Available: https://spacenews.com/starlink-failures-highlight-space-sustainability-concerns/

[8] J. Wang, L. Li, and M. Zhou, "Topological dynamics characterization for LEO satellite networks," *Comput. Netw.*, vol. 51, no. 1, pp. 43–53, Jan. 2007.

[9] L. Chen et al., "Load-adaptive and energy-efficient topology control in LEO mega-constellation networks," in *Proc. IEEE 23rd Glob. Commun. Conf.*, 2022, pp. 1552–1557.

[10] K. Shi, X. Zhang, S. Zhang, and H. Li, "Time-expanded graph based energy-efficient delay-bounded multicast over satellite networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10380–10384, Sep. 2020.

[11] X. Zhang et al., "Energy-efficient computation peer offloading in satellite edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3077–3091, Apr. 2024.

[12] M. Marchese and F. Patrone, "E-CGR: Energy-aware contact graph routing over nanosatellite networks," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 3, pp. 890–902, Sep. 2020.

[13] J. Liu, B. Zhao, Q. Xin, J. Su, and W. Ou, "DRL-ER: An intelligent energy-aware routing protocol with guaranteed delay bounds in satellite mega-constellations," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2872–2884, Oct.-Dec. 2021.

[14] D. Bhattacharjee, A. U. Chaudhry, H. Yanikomeroglu, P. Hu, and G. Lamontagne, "Laser inter-satellite link setup delay: Quantification, impact, and tolerable value," in *Proc. IEEE 24th Wireless Commun. Netw. Conf.*, 2023, pp. 1–6.

[15] P. Grenfell, A. Aguilar, K. Cahoy, and M. Long, "Pointing, acquisition, and tracking for small satellite laser communications," in *Proc. AIAA/USU 32nd Conf. Small Satellites*, 2018, pp. 1–7.

[16] "Iridium: An overview," [Online]. Available: https://www.iridium museum.com/

[17] G. Song, M. Chao, B. Yang, and Y. Zheng, "TLR: A traffic-light-based intelligent routing strategy for NGEO satellite IP networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3380–3393, Jun. 2014.

[18] Y. Li, Y. Wang, Q. Zhang, and Z. Yang, "TCDS: A time-relevant graph based topology control in triple-layer satellite networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 424–428, Mar. 2020.

[19] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of inter-satellite link paths for LEO mega-constellation networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2743–2755, Mar. 2021.

[20] J. Mclaughlin, J. Choi, and R. Durairajan, "×Grid: A location-oriented topology design for LEO satellites," in *Proc. 1st ACM Workshop LEO Netw. Commun.*, 2023, pp. 37–42.

[21] S. Ji, D. Zhou, M. Sheng, and J. Li, "Mega satellite constellation system optimization: From a network control structure perspective," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 913–927, Feb. 2022.

[22] X. Liu, X. Chen, L. Yang, Q. Chen, J. Guo, and S. Wu, "Dynamic topology control in optical satellite networks based on algebraic connectivity," *Acta Astronautica*, vol. 165, pp. 287–297, Dec. 2019.

[23] W. Wang, Y. Zhao, Y. Zhang, X. He, Y. Liu, and J. Zhang, "Intersatellite laser link planning for reliable topology design in optical satellite networks: A networking perspective," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2612–2624, Sep. 2022.

[24] X. Wang, W. Li, S. Han, M. Yang, and Z. Jiang, "Enabling high-connectivity LEO satellite networks via encountering inter-satellite links," in *Proc. IEEE 24th Glob. Commun. Conf.*, 2023, pp. 4883–4889.

[25] K. Han et al., "Non-grid-mesh topology design for megaLEO constellations: An algorithm based on NSGA-III," *IEEE Trans. Commun.*, vol. 72, no. 5, pp. 2881–2896, May 2024.

[26] T. Lan, D. Zhou, M. Sheng, and J. Li, "Capacity analysis of LEO mega-constellation networks," *IEEE Trans. Commun.*, vol. 10, pp. 18420–18433, 2025.

[27] J. Yang, B. Li, K. Fan, L. An, and Q. Zhang, "Analysis of laser inter-satellite links and topology design for mega-constellation networks," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34554–34566, Nov. 2024.

[28] L. Guo, J. Liu, M. Sheng, and J. Li, "Constellation topology design for maximum capacity of LEO satellite networks," *IEEE Trans. Commun.*, to be published, doi: 10.1109/TCOMM.2024.3502417, Nov. 2024.

[29] Z. Lai et al., "Your mega-constellations can be slim: A cost-effective approach for constructing survivable and performant LEO satellite networks," in *Proc. IEEE 43rd Int. Conf. Comput. Commun.*, 2024, pp. 521–530.

[30] G. Wang, F. Yang, J. Song, and Z. Han, "Federated reinforcement learning with constellation collaboration for dynamic laser inter-satellite link scheduling," in *Proc. IEEE 41st Int. Conf. Commun.*, 2024, pp. 3815–3820.

[31] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Commun. Surv. Tut.*, vol. 18, no. 4, pp. 2442–2473, Fourth Quarter, 2016.

[32] G. Sümen, G. K. Kurt, and A. Görçin, "A novel LFM waveform for terahertz-band joint radar and communications over inter-satellite links," in *Proc. IEEE 23rd Glob. Commun. Conf.*, 2022, pp. 6439–6444.

[33] V. Sharma and A. Kaur, "Optimization of ISL in hybrid ofdm-isowc transmission system," in *Proc. IEEE 2nd Int. Conf. Adv. Commun., Netw., Comput.*, 2013, pp. 25–28.

[34] R. Deng, B. Di, H. Zhang, L. Kuang, and L. Song, "Ultra-dense LEO satellite constellations: How many LEO satellites do we need?," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4843–4857, Aug. 2021.

[35] H. Hu, Y. Lyu, K. Song, R. Fan, C. Zhan, and J. Yang, "An efficient two-stage networking topology design for mega-constellation of low earth orbit satellites," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2025.3540671.

[36] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, and X. Chen, "3-ISL topology: Routing properties and performance in LEO mega-constellation networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 61, no. 2, pp. 4961–4972, Apr. 2025.

[37] K. Higashimori and T. Inoue, "Dynamic topology control of LEO satellite networks for non-uniformly distributed traffic," in *Proc. IEEE 25th Glob. Commun. Conf.*, 2024, pp. 3027–3032.

[38] N. Guo, L. Liu, and X. Zhong, "Task-aware distributed inter-layer topology optimization method in resource-limited LEO-LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3572–3585, Apr. 2024.

[39] G. Wang, F. Yang, J. Song, and Z. Han, "Dynamic laser inter-satellite link scheduling based on federated reinforcement learning: An asynchronous hierarchical architecture," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 14273–14288, Oct. 2024.

[40] T. Lan, D. Zhou, M. Sheng, and J. Li, "Inter-satellite link planning for high capacity in LEO mega-constellations," in *Proc. 41st IEEE Int. Conf. Commun.*, 2024, pp. 4451–4456.

[41] Starlink Insider, "Starlink ground station locations: An overview," 2025. [Online]. Available: https://starlinkinsider.com/starlink-gateway-locations/

[42] L. Chen, F. Tang, Z. Li, L. T. Yang, J. Yu, and B. Yao, "Time-varying resource graph based resource model for space-terrestrial integrated networks," in *Proc. IEEE 40th Int. Conf. Comput. Commun.*, 2021, pp. 1–10.

[43] A. U. Chaudhry and H. Yanikomeroglu, "Laser intersatellite links in a starlink constellation: A classification and analysis," *IEEE Veh. Technol. Mag.*, vol. 16, no. 2, pp. 48–56, Jun. 2021.

[44] European Parliament and Council of the European Union,"Gdpr: General principle for transfers," 2016. [Online]. Available: https://gdpr-info.eu/art-44-gdpr/

[45] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[46] J. Liang et al., "Free-space optical (FSO) satellite networks performance analysis: Transmission power, latency, and outage probability," *IEEE Open J. Veh. Technol.*, vol. 5, pp. 244–261, 2023.

[47] R. Wang, M. A. Kishk, and M.-S. Alouini, "Ultra-dense LEO satellite-based communication systems: A novel modeling technique," *IEEE Commun. Mag.*, vol. 60, no. 4, pp. 25–31, Apr. 2022.

[48] G. Dantzig and D. R. Fulkerson, "On the max flow min cut theorem of networks," *Linear Inequalities Related Syst.*, vol. 38, pp. 225–231, 2003.

[49] E. L. Johnson, "Networks and basic solutions," *Operations Res.*, vol. 14, no. 4, pp. 619–623, Aug. 1966.

[50] M. R. Garey and D. S. Johnson, *Computers and Intractability*, vol. 174. San Francisco, CA, USA: Freeman, 1979.

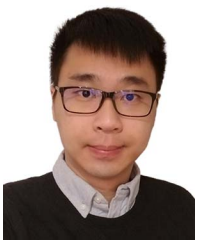
[51] T. Li, H. Zhou, H. Luo, and S. Yu, "SERvICE: A software defined framework for integrated space-terrestrial satellite communication," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 703–716, Mar. 2018.

[52] J. Cheriyan and R. Thurimella, "Approximating minimum-size k-connected spanning subgraphs via matching," *SIAM J. Comput.*, vol. 30, no. 2, pp. 528–560, Aug. 2000.

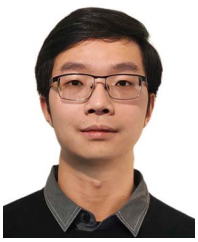
[53] G. Giuliari et al., "ICARUS: Attacking low earth orbit satellite networks," in *Proc. USENIX 26th Annu. Tech. Conf.*, 2021, pp. 64–69.

[54] Y. Li and Y. Chen, "Propagation modeling and analysis for terahertz inter-satellite communications using FDTD methods," in *Proc. IEEE 38th Int. Conf. Commun. Workshops*, 2021, pp. 1–6.
[55] B. Tao, M. Masood, I. Gupta, and D. Vasisht, "Transmitting, fast and slow: Scheduling satellite traffic through space and time," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–15.
[56] L. Chen, F. Tang, and X. Li, "Mobility-and load-adaptive controller placement and assignment in LEO satellite networks," in *Proc. IEEE 40th Int. Conf. Comput. Commun.*, 2021, pp. 1–10.
[57] Z. Lai et al., "OrbitCast: Exploiting mega-constellations for low-latency earth observation," in *Proc. IEEE 29th Int. Conf. Netw. Protoc.*, 2021, pp. 1–12.
[58] M. Lyu, Q. Wu, Z. Lai, H. Li, Y. Li, and J. Liu, "FALCON: Towards fast and scalable data delivery for emerging earth observation constellations," in *Proc. IEEE 42nd Int. Conf. Comput. Commun.*, 2023, pp. 1–10.

Long Chen (Member, IEEE) received the BE and PhD degrees in software engineering from Shanghai Jiao Tong University (SJTU), China, in 2016 and 2022. Currently, he is a postdoctoral researcher in the School of Computing Science with Simon Fraser University, Canada. His research interests include software defined networking, space-terrestrial integrated networks, and machine learning.

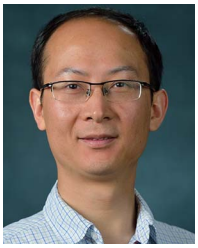


Yi Ching Chou received the BBA degree in finance and the BSc degree in computer science from Simon Fraser University, BC, Canada. He is currently working toward the PhD degree in computer science with Simon Fraser University and received an NSERC CGS M award. His research interests include satellite communications, computer networking, and edge computing.



Haoyuan Zhao received the BSc degree in computing science degrees from Simon Fraser University, British Columbia, Canada. He is currently working toward the PhD degree in computing science with Simon Fraser University. His research interests include satellite networking and multimedia systems.

Hengzhi Wang (Member, IEEE) received the BS and PhD degrees in software engineering from Jilin University in Changchun, China, in 2017 and 2023. He is currently serving as an assistant professor in the College of Computer Science and Software Engineering with Shenzhen University in Shenzhen, China. During his doctoral studies, he also worked as a visiting PhD student at Simon Fraser University. His research interests primarily focus on spatial crowdsourcing and federated learning.



Feng Wang (Senior Member, IEEE) received the bachelor's and master's degrees from Tsinghua University, China in 2002 and 2005, respectively, and the PhD degree from Simon Fraser University, Canada, in 2012. He is currently an associate professor in the Department of Computer and Information Science with the University of Mississippi, USA. His research interests include satellite networks, wireless sensor networks, and cloud/edge computing.

Hao Fang received the BSc (Hons.) degree with distinction in Computing Science from Simon Fraser University, BC, Canada, in 2022. He is currently the PhD degree in computing science with Simon Fraser University. His research areas include satellite communications and networking, particularly with multimedia systems.

Sami Ma received the BSc degree with distinction in computing science from Simon Fraser University, BC, Canada, in 2019. Currently, he is continuing doctoral studies in Computing Science at Simon Fraser University, Canada. His research interests include satellite networks, internet architecture and protocols, and deep learning.

Feilong Tang (Senior Member, IEEE) received the PhD degree in computer science from Shanghai Jiao Tong University (SJTU), China, in 2005. Now, he is a professor in the Department of Computer Science and Engineering, SJTU, China. He was the JSPS research fellow in Japan, and received the Distinguished Pu-Jiang Scholars Award. His research interests focus on space-terrestrial integrated networks and mobile cognitive networks. He is the IET Fellow.

Linghe Kong (Senior Member, IEEE) received the BE degree in automation from Xidian University, in 2005, the master's degree in telecommunication from TELECOM SudParis, in 2007, and the PhD degree in computer science with Shanghai Jiao Tong University, in 2012. He is currently a research professor in Shanghai Jiao Tong University. Before that, he was a postdoctoral fellow with Columbia University and McGill University. His research interests include Internet of things, wireless communications, and mobile computing.

Jiangchuan Liu (Fellow, IEEE) received the BEng (cum laude) degree in computer science from Tsinghua University, Beijing, China, in 1999, and the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2003. He is currently a full professor (with University Professorship) wth the School of Computing Science, Simon Fraser University, BC, Canada. He is a fellow of the Canadian Academy of Engineering and the NSERC E.W.R. Steacie Memorial fellow.