Dear Editors and Reviewers:

We would like to submit the manuscript "Enabling Practical and Predictable Load Balancing for Low Earth Orbit Mega-Constellations with Matchmaker" for consideration for publication in the IEEE Journal on Selected Areas in Communications - Advances in Internet Routing and Addressing. Part of this paper will be presented at the IEEE International Conference on Communications (ICC) in June 2024. The conference version is attached at the end of this file.

We would like to emphasize that there are many differences between this submission and the conference version. The key differences are summarized as follows.

- We enriched Section II by discussing the handover overhead of user-satellite assignment and how our proposed solution overcomes this limitation.
- For better readability, we added Table 1 to introduce the table of notations used in this paper.
- We formulated a new optimization problem named Handover-aware User-Satellite Assignment (HUSA), which jointly considers load balancing performance and user-satellite handover overhead.
- We added Section V-A to provide rigorous proof of the NP-hardness of the formulated HUSA problem.
- To solve this problem efficiently, we added Section V-B to introduce our new rounding-based heuristic algorithm.
- We added Section V-C to provide a detailed performance analysis of the rounding-based heuristic algorithm, emphasizing the satisfaction of all constraints and the approximation factor for satellite load constraints.
- We added Section V-D to give the time complexity analysis of the proposed rounding-based heuristic algorithm.
- We added Section V-E to introduce our proposed Matchmaker, which involves user terminal selection, user-satellite assignment, and two-step routing.
- We enriched Section VI by adding extensive simulation results on load balancing performance, handover overhead, user-experienced latency, and scalability performance under the current operational Starlink LEO mega-constellation.
- We also improved the background knowledge of our paper by introducing and analyzing the following recent works.

[1] S. Dou, S. Zhang, and K. L. Yeung, “Achieving Predictable and Scalable Load Balancing Performance in LEO Mega-Constellations,” in Proc. of the IEEE International Conference

on Communications (ICC), 2024, pp. 1–6.

[3] T. Ma, B. Qian, X. Qin, X. Liu, H. Zhou, and L. Zhao, "Satellite-terrestrial integrated 6G: An ultra-dense LEO networking management architecture," IEEE Wireless Communications, 2022.

[12] Y. Yang, M. Xu, D. Wang, and Y. Wang, "Towards energy-efficient routing in satellite networks," IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3869–3886, 2016.

[13] R. d. N. M. Macambira, C. B. Carvalho, and J. F. de Rezende, "Energy-efficient routing in LEO satellite networks for extending satellites lifetime," Computer Communications, vol. 195, pp. 463–475, 2022.

[20] P. Yue, J. An, J. Zhang, J. Ye, G. Pan, S. Wang, P. Xiao, and L. Hanzo, "Low earth orbit satellite security and reliability: Issues, solutions, and the road ahead," IEEE Communications Surveys & Tutorials, 2023.

[23] X. Cao and X. Zhang, "SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks," in Proc. of the IEEE International Conference on Computer Communications (INFOCOM), 2023, pp. 1–10.

[24] H.-L. Maattanen, B. Hofstrom, S. Euler, J. Sedin, X. Lin, O. Liberg, G. Masini, and M. Israelsson, "5G NR communication over GEO or LEO satellite systems: 3GPP RAN higher layer standardization aspects," in Proc. of the IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1–6.

[27] Y. Rao, J. Zhu, C.-a. Yuan, Z.-h. Jiang, L.-y. Fu, X. Shao, and R.-c. Wang, "Agent-based multi-service routing for polar-orbit LEO broadband satellite networks," Ad hoc networks, vol. 13, pp. 575–597, 2014.

[37] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laugh-lin, B. Maggs, and A. Singla, "Why is the internet so slow?!" in Proc. of the International Conference on Passive and Active Measurement (PAM), 2017, pp. 173–187.

[38] I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," in Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2014, pp. 103–112.

[49] Z. Lai, H. Li, Q. Zhang, Q. Wu, and J. Wu, "StarFront: Cooperatively Constructing Pervasive and Low-Latency CDNs Upon Emerging LEO Satellites and Clouds," IEEE/ACM Transactions on Networking, 2023.

[50] D. Vasisht and R. Chandra, "A distributed and hybrid ground station network for low earth orbit satellites," in Proc. of the ACM Workshop on Hot Topics in Networks (HotNets), 2020, pp. 190–196.

We hope that you will enjoy reading our new paper.

Sincerely,

Songshi Dou, Jinxian Wu, Shengyu Zhang, Xianhao Chen, and Kwan L. Yeung

Achieving Predictable and Scalable Load Balancing Performance in LEO Mega-Constellations

Songshi Dou*, Shengyu Zhang†, and Kwan L. Yeung*
*The University of Hong Kong, †Singapore University of Technology and Design

***Abstract*—With the increasing deployment scale of Low Earth Orbit (LEO) mega-constellations, more satellites are expected to become visible to users simultaneously, bringing a new opportunity to optimize the network performance by properly assigning users to satellites. In this paper, we consider LEO mega-constellations without Inter-Satellite Links (ISLs) while assuming there are enough ground relays for inter-satellite communications. To establish a path from a user terminal to the nearest ground station (which serves as a gateway to the Internet), shortest path routing is usually adopted. To focus on the problem of user-satellite assignment, as well as to make routing more scalable, we divide the routing process into two parts: assigning the user terminal to a visible satellite, and finding a path from the satellite to the nearest ground station. For simplicity, shortest path routing is assumed in the second part. Aiming at minimizing the Maximum Satellite Utilization (MSU), a Mixed Integer Linear Programming (MILP), called Optimal User-Satellite Assignment (OUSA), is formulated. Performance evaluations are conducted based on Starlink Phase I mega-constellation and AWS ground station locations. As compared with the existing solutions, we show that the average load balancing performance can be improved by up to 33.29%.**

***Index Terms*—LEO Mega-Constellations, Load Balancing, User-Satellite Assignment, Scalable Routing.**

I. Introduction

The concept of utilizing mega-constellations consisting of numerous Low Earth Orbit (LEO) satellites in space to provide global Internet services has gained significant attention from both industry and academia. These emerging LEO mega-constellations (*e.g.*, Starlink [1], OneWeb [2], and Amazon Kuiper [3]) function as Internet Service Providers (ISPs) and have the capability to offer pervasive Internet connectivity worldwide, particularly in remote areas. As the deployment of LEO mega-constellations progresses, major technology giants (*e.g.*, Amazon [4] and Microsoft [5]) are actively establishing their geo-distributed ground stations or Ground-Station-as-a-Service (GSaaS) infrastructure. This infrastructure aims to provide cost-effective, flexible, and scalable services for satellite communications, data transmission, and operational management on a global scale, eliminating the need for organizations to build and maintain their own ground station infrastructure. For instance, Starlink plans to develop more than 50 ground stations [6], and has announced a partnership with Google to install ground stations at Google data centers [7]. Similarly, Amazon's Kuiper project has strategically positioned its ground stations to ensure reliable communication with Amazon's data centers, supporting a wide range of network services (*e.g.*, Amazon Web Services) [4]. Consequently, these geo-distributed ground stations can efficiently handle users' Internet service requests across the globe.

LEO satellite networks differ from traditional terrestrial networks in terms of their high dynamicity, as the availability of satellites for user connectivity varies over time. Routing for LEO mega-constellations involves efficiently managing the flow of data between the satellites and the ground stations. Routing algorithms need to adapt to topology changes in real-time, ensuring that data is routed through the most optimal path at any given moment. To establish a path from a user terminal to the nearest ground station (which serves as a gateway to the Internet), shortest path routing [8] is usually adopted. But shortest path routing tends to cause traffic hot-spots. Satellites are battery-powered and batteries are charged by on-board solar panels. The surge of traffic load will not only increase packet queuing delay, but also a sharp drop in battery power. The latter may force a satellite into hibernation mode for saving power. This shows that a load balanced mega-constellation is important.

Aiming at balancing network load in LEO mega-constellations, existing routing solutions face three major issues, not practical, not efficient, and not scalable. Under the assumption that Inter-Satellite Links (ISLs) are available, various ISLs-enabled routing solutions have been proposed [6], [9]. But ISLs are not yet supported by current mega-constellations (*e.g.*, Starlink Shell I of Phase I) due to the deployment cost and technical issues [10], [11], [12]. Some solutions assume that traffic demands are available. Optimization problems are then formulated to find explicit routing strategies that can optimally redistribute the traffic across the LEO satellite network [13], [14]. They are also impractical due to the huge overhead involved in measuring and collecting Traffic Matrices (TMs) among moving LEO satellites in real-time. Besides, with potentially millions of user terminals, all existing routing solutions may face the scalability issue.

As the deployment scale of LEO mega-constellations increases, more and more satellites become simultaneously visible to user terminals, *i.e.*, within the transmission range of each other. Recent study [15] show that in most terrestrial locations, the number of visible satellites at any given time can range from 15-40 if all five shells of Starlink Phase I [1] are fully deployed. We observe that the user-satellite assignment policy (alone) can significantly impact the load balancing performance of the constellation. In order to focus on the problem of user-satellite assignment, as well as to make routing more scalable, we divide the routing process into two parts: assigning the user terminal to a visible satellite, and finding a path from the satellite to the nearest ground station. Without loss of generality, shortest path routing is adopted for finding paths from all satellites to ground stations. Since the number of satellites and ground facilities (*e.g.*, ground stations and ground relays) is much smaller than the number of user terminals, routing becomes more scalable.

To solve the user-satellite assignment problem, we propose a Mixed Integer Linear Programming (MILP), called Optimal User-Satellite Assignment (OUSA). As measuring and collecting TMs among moving satellites in real-time is impractical, our goal is to balance the number of user terminals whose traffic would be handled/traversed each satellite. Assume all satellites have the same capacity. OUSA aims at minimizing the Maximum Satellite Utilization (MSU). Performance evaluations are conducted based on Starlink Phase I mega-constellation and AWS ground station locations. As compared with the existing solutions, we show that the average load balancing performance can be improved by up to 33.29%.

The main contributions of this paper are summarized as follows:

- We identify that existing user-satellite assignment solutions are not practical for today's LEO mega-constellations and also fail to achieve good load balancing performance in a scalable manner.
- We formulate the OUSA problem as a MILP and propose a solution for deciding the optimal user-satellite assignments to achieve predictable load balancing in LEO mega-constellations.
- To show the effectiveness of our proposed OUSA, we utilize real-world LEO mega-constellation parameters and ground station information to evaluate the load balancing and scalability performance of OUSA.

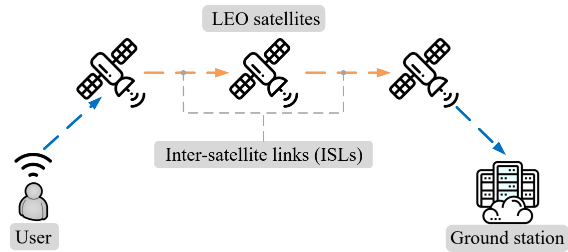
The remainder of this paper is organized as follows. In Section II, we provide background and motivation of this paper. In Section III, we present the design overview. Section IV formulates our problem as the OUSA problem, and proposes solution to this problem. The performance of our proposed solution is evaluated and analyzed in Section V. Section VI discusses the related work in this research area. Finally, in Section VII, we conclude the paper and propose future research directions.

II. Background and Motivation

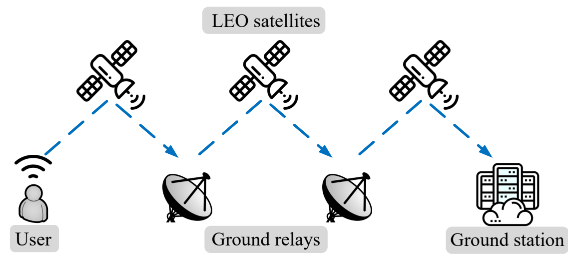
In this section, we will introduce the background, motivation, and our observations about this paper.

A. *Designs of Emerging LEO Satellite Networks*

Fig. 1(a) depicts an illustrative example of the design of an ISLs-enabled LEO satellite network. Many existing LEO satellite networks adopt this design, which utilizes ISLs to establish space routes for long-distance communications. In this design, user data packets are initially forwarded to the satellite and then transmitted through the space backbone, which is formed by satellites and ISLs. Finally, these packets are forwarded to ground stations, which serve as access



(a) ISLs-enabled LEO satellite network design.



(b) Bent Pipe LEO satellite network design.

Fig. 1: LEO satellite networks with and without inter-satellite links.

points or gateways. These ground stations are typically GSaaS infrastructures deployed by leading technology companies to facilitate satellite communications and data transmission while providing access to the Internet.

In contrast to the ISLs-enabled LEO satellite network design, current operational mega-constellations (*e.g.*, SpaceX's Starlink) adopt a Bent Pipe architecture without ISLs to integrate LEO satellites into ground facilities. Fig. 1(b) showcases an example of the Bent Pipe design used to interconnect satellites. In this architecture, user data packets are transmitted to the satellites, which then promptly return the data to the ground facilities in a manner resembling a bent pipe. Unlike the ISLs-enabled design, there are no ISLs between satellites in space. Satellites solely rely on phased array antennas for direct communications with ground facilities (*e.g.*, ground stations and ground relays) and user terminals via Radio Frequency (RF) signals. Unlike ground stations, ground relays serve as intermediaries to interconnect two satellites but do not have direct access to the Internet.

B. *User-Satellite Assignment Problem and Existing Solutions*

With the increasing deployment scale of LEO mega-constellations, many satellites can simultaneously be visible to a user located at a specific terrestrial site. Thus, there will be multiple choices to assign each user to satellites. Different assignment policies will largely affect the performance of the whole LEO satellite network [9]. Existing solutions try to tackle the user-satellite assignment problem and can be categorized into the following four types: (i) Shortest path routing solutions [10], (ii) Explicit routing solutions [13], [14], (iii) Distance-based user-satellite assignment solutions [16], [17], and (iv) ISLs-enabled routing solutions [6], [9].

Specifically, shortest path routing solutions calculate the shortest path between each user and its destination ground station, and the user-satellite assignment is decided based on the calculated shortest path. Unlike shortest path solutions, explicit routing solutions assume TMs can be collected in real-time and try to find effective routing strategies to optimize network traffic. Distance-based user-satellite assignment solutions always find the nearest satellite for each user to maintain the best quality of the RF signal. ISL-enabled routing solutions usually assume ISLs are supported in LEO mega-constellations and satellites are directly interconnected by ISLs without using ground relays. Most of the existing works follow the +Grid design to interconnect satellites [18].

C. Observations

However, we observe that existing solutions suffer from the following three major problems:

*1) **Not practical**:* ISLs-enabled routing solutions only concentrate on the LEO satellite networks with ISLs. However, ISLs are not supported in current LEO mega-constellations due to the high deployment cost and technical issues, and when ISLs will be fully supported is still unclear [10], [11], [13]. Thus, it is important to consider the practical scenario by using Bent Pipe design, which is not fully considered by existing works.

In addition, explicit routing solutions propose to do explicit routing for realizing satisfactory load balancing performance based on the network traffic status (*e.g.*, TMs). However, real-time TMs are hard to collect, and also introduce extremely high processing overhead. Amassing sufficient data from direct measurements to populate a traffic matrix is even typically prohibitively expensive in today's terrestrial networks [19]. Due to the special characteristics of LEO mega-constellations (*e.g.*, highly dynamical and non-terrestrial), collecting TMs among these moving LEO satellites in real-time is even more impractical and not worth the expense of measurements.

*2) **Not efficient**:* Existing shortest path routing solutions and distance-based user-satellite assignment solutions fail to efficiently utilize the network resource without considering load balancing performance. Load balancing is an essential to help distribute incoming network traffic, which improves network performance, adds redundancy to the whole network, and prevents traffic bottlenecks at any one satellite and to ensure that the satellites are utilized efficiently [20]. The imbalanced traffic load on each satellite will lead to inefficient network resource management [13].

*3) **Not scalable**:* Due to the time-varying topology with high dynamicity, the shortest path algorithm has to be run periodically to update all flow entries in each satellite's forwarding table. Calculating shortest path between sources (*i.e.*, user terminals) and destinations (*i.e.*, ground stations) can be time-consuming and not scalable in this case. With potentially millions of user terminals in LEO satellite networks in future, the scale of the LEO satellite network topology will continuously grow, and the forwarding table size of each satellite can be increased correspondingly. This will result in larger storage requirements, longer execution time of the shortest path algorithm and slower table lookup.

III. Design Overview

In this section, we introduce how our proposed solution outperforms state-of-the-art solutions, and further present the processing logic.

A. How Does Our Solution Overcome These Limitations?

*1) **Real-world LEO mega-constellation design**:* Given that current operational LEO mega-constellations do not support ISLs due to technical issues and deployment cost [12], our proposed solution follows the practical Bent Pipe design by using ground relays interconnecting satellites to build the LEO satellite backbone. Furthermore, our solution proposes to balance the number of users instead of real-time traffic volume among all satellites. Compared with traditional terrestrial networks, the data speeds provided by LEO satellite networks are relatively limited and stable (*e.g.*, from 50 Mbps to 150 Mbps) [15]. Thus, LEO satellite network users usually have similar traffic demands, and the number of users can essentially represent the real-time traffic status, given that the number of users is way easier to collect than TMs.

*2) **Load balancing-aware user-satellite assignment**:* Satellites rely on batteries that are charged by on-board solar panels. Unbalanced satellite traffic load may increase packet queuing delay and cause a sharp drop in battery power. To improve the network performance, avoid traffic congestion, and ensure the network resources are fully utilized, our solution proposes to maintain predictable user load balancing upon LEO mega-constellations since satellites can easily become traffic hotspots under Bent Pipe design [13]. It first calculates all shortest paths between each user terminal and its visible satellites periodically. Then, based on the calculated shortest paths, the proposed OUSA decides the proper user-satellite assignment to balance user load among satellites. The optimal user-satellite assignments are decided by solving our formulated OUSA problem, which will be detailed in the following Section IV.

*3) **Two-step routing architecture**:* To make routing more scalable, we divide the routing process into two parts: assigning the user terminal to a visible satellite, and finding a path from the satellite to the nearest ground station. Specifically, each user will be assigned to the proper visible satellite based on the predictable load balancing performance. Then, the routing between the satellite and its destination ground stations will follow the pre-calculated shortest paths. Given that the number of satellites and ground facilities (*e.g.*, ground stations and ground relays) is much smaller than the number of users, the higher scalability of our solution would benefit from a much smaller scale of network topology without involving the end users.

B. Processing Logic

Fig. 2 shows the processing logic of our proposed solution, which has five main steps. First, the network status (*e.g.*, the visibility between users and satellites) will be collected periodically at each user terminal. Then, our solution pre-calculates all the shortest paths between each user terminal and its visible satellites. Furthermore, our solution decides

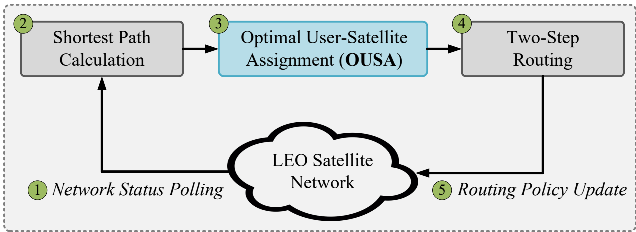


Fig. 2: Processing logic of our proposed solution.

the optimal user-satellite assignment strategies by solving our formulated optimization problem OUSA, which will be introduced in Section IV, and then follows pre-calculated shortest paths when forwarding data packets. Finally, the routing policy of this two-step routing is updated to the LEO satellite network.

IV. Problem Formulation and Solution

In this section, we present the formulation of the OUSA problem, and introduce a solution to solve the problem.

A. System Description

Assume time is slotted, and we can divide time into successive operation periods. We use synodic period of the constellation to denote repeated cycles for the satellites as observed from the earth's surface. Specifically, to describe the dynamic characteristics of LEO satellites, we divide the synodic period into T time intervals, and the set of time intervals is denoted as $\mathcal{T} = \{1, 2, ..., t, ..., T\}$. We assume that the network topology remains unchanged between adjacent time intervals. Assume N satellites are evenly spaced in their orbits. The set of satellites is denoted as $\mathcal{S} = \{s_1, s_2, ..., s_i, ..., s_N\}$. M active users are at geo-distributed locations, and the set of these users is $\mathcal{U} = \{u_1, u_2, ..., u_j, ..., u_M\}$. We denote the set of ground stations as $\mathcal{G} = \{g_1, g_2, ..., g_l, ..., g_L\}$. x_{ij}^t denotes the assignment between users and satellites at time interval t. If user u_j is assigned to satellite s_i at time interval t, we have $x_{ij}^t = 1$; Otherwise, $x_{ij}^t = 0$.

B. Problem Constraints

1) User-Satellite Assignment Constraint: Each user must be assigned to one satellite at each time interval. Thus, the user-satellite assignment constraint can be formulated as follows:

$$\sum_{i=1}^{N} x_{ij}^t = 1, \forall j \in [1, M], \forall t \in [1, T]. \tag{1}$$

2) Connection Visibility Constraint: When assigning users to satellites, we have to consider the connection visibility between them. A user can be assigned to a satellite only when this satellite is visible to this user at the time interval. We use β_{ij}^t to denote the connection visibility between users and satellites at each time interval. If satellite s_i is visible to user u_j at time interval t, we have $\beta_{ij}^t = 1$; Otherwise, $\beta_{ij}^t = 0$. Thus, our connection visibility constraint can be expressed as follows:

$$x_{ij}^t \le \beta_{ij}^t, \forall i \in [1, N], \forall j \in [1, M], \forall t \in [1, T]. \tag{2}$$

3) Satellite Load Constraint: Typically, each user's data will be transmitted to the assigned satellite. Then, the transmission of data will follow the Bent Pipe design (*e.g.*, traversing several satellites and ground relays) and eventually to a designated ground station. Assume each user will be handled by one specific ground station. We use ζ_{jl}^t to denote the relationship between users and ground stations. If user u_j is eventually handled by ground station g_l at time interval t, we have $\zeta_{jl}^t = 1$; Otherwise, $\zeta_{jl}^t = 0$. However, each satellite can only handle a limited number of users. We use Cap_i to denote the capacity of satellite s_i. During each time interval, we run the shortest path algorithm (*e.g.*, Dijkstra algorithm) to calculate the shortest path between each satellite and each ground station. We use $\alpha_{il}^{i't}$ to denote which satellite the shortest path will traverse between satellite s_i and ground station g_l at time interval t. ω^t is used to denote the MSU at each time interval t. Thus, the satellite load constraint can be formulated as follows:

$$\sum_{l=1}^{L} \sum_{j=1}^{M} \sum_{i=1}^{N} (x_{ij}^t * \zeta_{jl}^t * \alpha_{il}^{i't}) \le Cap_{i'} * \omega^t, \forall i' \in [1, N], \forall t \in [1, T]. \tag{3}$$

C. Objective Function

In our formulated problem, we aim to achieve balanced user load among all satellites. The objective of our proposed OUSA problem is to minimize the MSU at each time interval. Thus, the objective function can be formulated as follows:

$$obj = \omega^t.$$

D. Problem Formulation

Based on all the above-mentioned problem constraints and objective function, the problem can be formulate as follows:

$$\begin{aligned} \min_{\omega, x} \quad & \omega^t \\ \text{s.t.} \quad & (1)(2)(3), \\ & x_{ij}^t \in \{0, 1\}, \forall i, i' \in [1, N], \\ & \forall j \in [1, M], \forall l \in [1, L], \forall t \in [1, T], \end{aligned} \tag{P}$$

where $\{x_{ij}^t\}$ are binary design variables, and ω is a continuous design variable. $\{Cap_i\}$ are given continuous constants, and $\{\alpha_{il}^{i't}\}$, $\{\beta_{ij}^t\}$, and $\{\zeta_{jl}^t\}$ are given binary constants. Given that the objective function is linear, this problem is a MILP.

E. Solution

The typical solution to the above OUSA problem is to get its optimal result with IP optimization solvers (*e.g.*, GUROBI [21] and CPLEX [22]), and our proposed solution also follows this way. In our future works, we will also develop efficient heuristic algorithms for solving the OUSA problem to cope with more complicated scenarios (*e.g.*, with more satellites visible to users) and achieve the trade-off between the performance and time complexity.

TABLE I: Parameters of the LEO mega-constellation used in our simulations.

Primary parameters	Starlink (Shell I of Phase I)
Inclination	53°
Altitude	550 km
Number of orbits	72
Number of satellites	1584
Synodic period	5,731s

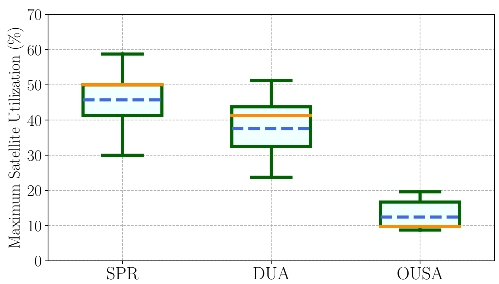


Fig. 3: Performance of the MSU. The lower, the better. Orange lines show the medians of each scheme, and blue dash lines show the mean values of each scheme.

V. Performance Evaluation

A. Evaluation Setup

We adopt the mega-constellations performance simulator StarPerf [23] with modifications to conduct our simulations and use a realistic and time-dynamic satellite system simulation tool Ansys STK [24] to calculate inter-visibility and time-varying geo-locations of each node. Our simulations are based on SpaceX's Starlink (Shell I of Phase I) constellations [25], and primary parameter settings used in our simulations are presented in Table I. To ensure our simulations' validity, we select 40 representative cities from Starlink's availability map [26] and generate 7,500 active users from these 40 selected cities simultaneously. Given that each Starlink LEO satellite can support a maximum of 2,000 simultaneously active user terminals with limited bandwidth for each user terminal [27], we set the capability of each satellite to 2,000 users during our simulations. We select 11 ground stations worldwide based on Amazon's AWS ground station locations to transmit and receive data for user terminals [28]. Each user terminal will be handled by its nearest ground station, calculated using the Haversine formula [29]. Ground relays are evenly distributed and placed between every two satellites within visible connectivity [23]. The time interval is set as 5s in our simulations, which can perfectly show the dynamic characteristics of LEO mega-constellations without introducing much topology updating overhead and is also widely used in previous works [6]. The simulation lasts for 5,731s, which is equal to the synodic period of the Starlink LEO mega-constellation.

B. Comparison Algorithm

1) Shortest Path Routing (**SPR**) [10]: This scheme calculates the shortest path between each user terminal and its destination ground station. The user-satellite assignments between user terminals and satellites are based on their shortest paths.
2) Distance-based User-Satellite Assignment (**DUA**) [16], [17]: This scheme first finds the nearest satellite for each user terminal and then assigns each user terminal to its nearest satellite.
3) Optimal User-Satellite Assignment (**OUSA**): This scheme pre-calculates the shortest paths between users' visible satellites and their destination ground stations in each time interval by using Dijkstra algorithm [30] and get the user-satellite assignment strategy to achieve optimal load balancing performance by solving the proposed problem (P). We solve the problem using the optimization solver GUROBI [21].

TABLE II: Average maximum satellite utilization. The lower, the better.

Schemes	Percentage
SPR	45.73%
DUA	37.53%
OUSA	**12.44%**

C. Evaluation Results

1) Load balancing performance. We use the MSU performance as a metric to evaluate the load balancing performance. MSU denotes the maximum satellite utilization among all satellites in each time interval. Fig. 3 shows the boxplot of all MSU results during the whole simulation (*e.g.*, 5,731s). Compared with SRP and DUA, OUSA exhibits the best load balancing performance with a much lower median and mean value of the MSU. The satisfactory load balancing performance of OUSA gives the credit to OUSA takes the user load balancing performance among all satellites into consideration. As depicted in Table II, OUSA can improve the average load balancing performance by up to 33.29% compared with SPR.

2) Scalability performance. To show the scalability performance of all three solutions, we use the LEO satellite network topology scale as an evaluation metric. The network topology scale greatly impacts the scalability performance (*e.g.*, the computation efficiency when calculating shortest paths). Table III exhibits the scale of the LEO satellite network topology. Compared with SPR, OUSA benefits from a smaller scale of the network topology without involving all end users since the number of end users is much more than satellites and ground facilities. With the increasing number of users willing to pay for the services offered by the emerging ISPs (*e.g.*, Starlink and Kuiper) in the future, the scalability difference is expected to widen.

VI. Related Work

Emerging LEO Mega-Constellations. Bhattacherjee *et al.* [18] explore the problem of designing the inter-satellite network for low-latency and high-capacity Internet service using LEO satellite constellations. They propose a novel method based on repetitive patterns, called motifs, to optimize the

TABLE III: The scale of the LEO satellite network topology. The lower, the better.

Schemes	Scale of the topology
SPR	Users + Satellites + Ground facilities
DUA	Satellites + Ground facilities
OUSA	**Satellites + Ground facilities**

network topology and avoid frequent link changes due to the high-velocity nature of LEO satellites. Hauri *et al.* [11] compare the performance and benefits of LEO satellite networks with and without ISLs, which enable direct communication between satellites using lasers. They discusses some scenarios where ISLs are not enough or can be complemented by ground and fiber connectivity to optimize network design. Lai *et al.* [15] study a new computation paradigm called Space Edge Computing (SEC), which integrates LEO satellites with terrestrial networks to provide pervasive, low-latency and high-throughput network services on-demand for futuristic 6G communications. They further analyze the feasibility of SEC, propose a novel satellite-cloud architecture, and demonstrate the potential benefits of SEC for several applications.

User-Satellite Assignment in LEO Satellite Networks. Handley [10] explores the use of ground-based relays as a substitute for ISLs to provide low-latency wide-area networking in large LEO satellite constellations. He argues that ground relays can still be useful in hybrid networks that employ both ISLs and ground relays, as they can reduce latency, supplement capacity, and increase route diversity. Zhang *et al.* [6] propose to integrate LEO satellite networks with terrestrial networks to provide global Internet services. They propose a Coordinated Satellite-Ground Interconnecting (CSGI) algorithm, which coordinates the establishment of ground-satellite links among distributed ground stations. AEROPATH [9], a space-ground integrated data delivery architecture, is proposed to leverage emerging LEO satellites and ground station networks. The main idea of AEROPATH is to use distributed ground-station-driven routing to achieve high-throughput and low-latency data transmission. However, most of these works [6], [9] only focus on the LEO Satellite Network with ISLs, which is far from practical in real-world deployment. Some existing works [10] consider the Bent Pipe design without ISLs, but they fail to achieve good load balancing performance, compared with our proposed OUSA.

VII. Conclusion and Future Work

In this paper, we propose to achieve predictable load balancing performance upon LEO mega-constellations in a scalable manner. Evaluation results demonstrate that our proposed OUSA can efficiently balance the user load among all satellites. In future works, we will focus on developing an efficient heuristic algorithm for solving the proposed OUSA problem to achieve the trade-off between the performance and time complexity for futuristic LEO mega-constellation deployment.

Acknowledgment

We thank the anonymous reviewers for their valuable comments. The corresponding author of this paper is Songshi Dou.

References

[1] "Starlink," https://www.starlink.com/, accessed on August 17, 2023.
[2] "OneWeb," https://oneweb.net/, accessed on August 17, 2023.
[3] "Project Kuiper," https://www.aboutamazon.com/what-we-do/devices-services/project-kuiper, accessed on August 17, 2023.
[4] "AWS Ground Station," https://aws.amazon.com/ground-station/, accessed on August 17, 2023.
[5] "New Azure Orbital, ground station as a service, now in preview," https://azure.microsoft.com/en-us/updates/new-azure-orbital-ground-station-as-a-service-now-in-preview/, accessed on August 17, 2023.
[6] Y. Zhang *et al.*, "Enabling low-latency-capable satellite-ground topology for emerging leo satellite networks," in *IEEE INFOCOM'22*.
[7] "Google wins cloud deal from SpaceX for Starlink internet service," https://finance.yahoo.com/news/google-wins-cloud-deal-spacex-145258186.html, accessed on August 17, 2023.
[8] Z. Wang and J. Crowcroft, "Analysis of shortest-path routing algorithms in a dynamic network environment," *ACM SIGCOMM CCR*, vol. 22, no. 2, pp. 63–71, 1992.
[9] W. Liu *et al.*, "Enabling ubiquitous and efficient data delivery by leo satellites and ground station networks," in *IEEE GLOBECOM'22*.
[10] M. Handley, "Using ground relays for low-latency wide-area routing in megaconstellations," in *ACM HotNets'19*.
[11] Y. Hauri *et al.*, ""Internet from Space" without Inter-satellite Links," in *ACM HotNets'20*.
[12] P. Yue *et al.*, "Low earth orbit satellite security and reliability: Issues, solutions, and the road ahead," *IEEE COMST*, 2023.
[13] S. Zhang *et al.*, "Segment routing for traffic engineering and effective recovery in low-earth orbit satellite constellations," *Digital Communications and Networks*, 2022.
[14] Z. Lai *et al.*, "Achieving Resilient and Performance-Guaranteed Routing in Space-Terrestrial Integrated Networks," in *IEEE INFOCOM'23*.
[15] ——, "Futuristic 6g pervasive on-demand services: Integrating space edge computing with terrestrial networks," *IEEE Vehicular Technology Magazine*, vol. 18, no. 1, pp. 80–90, 2022.
[16] W. Jiang and P. Zong, "An improved connection-oriented routing in LEO satellite networks," in *WASE ICIE'10*.
[17] E. Papapetrou and F.-N. Pavlidou, "QoS handover management in LEO/MEO satellite systems," *Wireless Personal Communications*, vol. 24, pp. 189–204, 2003.
[18] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *ACM CoNEXT'19*.
[19] A. Medina *et al.*, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM CCR*, vol. 32, no. 4, pp. 161–174, 2002.
[20] N. Wang *et al.*, "An overview of routing optimization for internet traffic engineering," *IEEE COMST*, vol. 10, no. 1, pp. 36–56, 2008.
[21] "Gurobi Optimization Solver," http://www.gurobi.com, accessed on August 17, 2023.
[22] "IBM ILOG CPLEX Optimizer," https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer, accessed on August 17, 2023.
[23] Z. Lai *et al.*, "Starperf: Characterizing network performance for emerging mega-constellations," in *IEEE ICNP'20*.
[24] "Ansys STK — Digital Mission Engineering Software," https://www.ansys.com/products/missions/ansys-stk, accessed on August 17, 2023.
[25] "Application for Fixed Satellite Service by Space Exploration Holdings, LLC," https://fcc.report/IBFS/SAT-MOD-20200417-00037, accessed on August 17, 2023.
[26] "Starling Availability Map," https://www.starlink.com/map, accessed on August 17, 2023.
[27] L. Liu *et al.*, "Geographic Low-Earth-Orbit Networking without QoS Bottlenecks from Infrastructure Mobility," in *IEEE/ACM IWQoS'22*.
[28] "AWS Ground Station Locations," https://aws.amazon.com/ground-station/locations/, accessed on August 17, 2023.
[29] C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.
[30] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022, pp. 287–290.