



**HAL**  
open science

# The Nearest Is Not The Fastest : On The Importance Of Selecting In/Out Routing Hops Over A Satellite LEO Constellation

Alexia Auddino, Anna Barraqué, Oana Hotescu, Jérôme Lacan, Jose Radzik,  
Emmanuel Lochin

## ► To cite this version:

Alexia Auddino, Anna Barraqué, Oana Hotescu, Jérôme Lacan, Jose Radzik, et al.. The Nearest Is Not The Fastest : On The Importance Of Selecting In/Out Routing Hops Over A Satellite LEO Constellation. IEEE Vehicular Technology Conference - VTC2022-Fall, Sep 2022, Londres, United Kingdom. pp.1-5, 10.1109/VTC2022-Fall57202.2022.10012796 . hal-03766403

**HAL Id: hal-03766403**

**<https://hal.science/hal-03766403>**

Submitted on 1 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The Nearest Is Not The Fastest : On The Importance Of Selecting In/Out Routing Hops Over A Satellite LEO Constellation

Alexia Auddino\*, Anna Barraqué\*, Oana Hotescu\*, Jérôme Lacan\*, José Radzik\*, Emmanuel Lochin†

\*ISAE-SUPAERO, University of Toulouse, Toulouse, France

†ENAC, University of Toulouse, Toulouse, France

{alexia.auddino, anna.barraque}@student.isae-supaeero.fr

{jerome.lacan, oana.hotescu, jose.radzik}@isae-supaeero.fr

emmanuel.lochin@enac.fr

**Abstract**—This study investigates the importance of choosing the first (respectively last) hop to access (respectively to exit) a Low Earth Orbit (LEO) satellite constellation, which is of utmost importance for the LEO routing performance. Usually, basic routing strategies connect a ground station to its nearest satellite, and this strategy does not always lead to the optimal routing path. We propose to select this first/last satellites within a subset of  $k$ -nearest satellites. After performing routing simulations over one of the next-generation satellite constellations, preliminary results show that this in/out hop selection strategy leads to a better link capacity usage and a lower data loss rate, allowing a faster TCP bulk data transfer.

**Index Terms**—Mega constellation, LEO routing, Hypatia, ns-3 simulator.

## I. INTRODUCTION

Nowadays, satellite constellations are a key topic. Several private companies are deploying multiple satellites on Low Earth Orbits (LEO) : SpaceX with Starlink [1], Amazon with Kuiper [2] or Telesat [3]. The main objective of these massive constellations of satellites is to provide a complete coverage of the Earth and thus Internet connection to everyone, regardless of their location. As a matter of fact, this increase in connectivity allows reducing the digital fracture. Even now, some regions are indeed poorly equipped with antennas, cables, or fiber optics and have a slower or non-existent Internet connection.

Currently, most of satellite constellations do not use their full potential. The signal from a ground station is sent to a satellite that sends back the signal to a ground station and so on, before it reaches the final destination. Instead, satellites should communicate within the constellation and directly transfer the incoming signal from the source ground station to the destination. However, due to the movement of the satellites, inter satellite links (ISL) cannot be established between any couple of satellites and in most cases, a satellite is only connected to its closest satellite moving in the same direction. This implies that the graph of the ISL is very sparse and regular. The route in this graph, connecting two satellites, must be then carefully designed. For example, two satellites travelling in opposite directions (one to the north and the other

one to the south) can be very close geographically, but very far considering the ISLs graph. The same issue appears with the ground stations, which can be relatively close, but their respective nearest satellite can be very far inside the ISL graph.

The algorithm proposed in this paper, named FROG, is based on the idea that the ISL pattern in a constellation is fixed because of connectivity complexity. Satellites are always connected to the same satellites and routes from a source satellite to a destination satellite are somehow fixed. Therefore, the choice of these source and destination satellites is of utmost importance as it will establish the route to carry IP traffic. The aim is to find a source and destination satellite that are in the range of visibility of the ground station, and that are the closest in the constellation's network graph. The best-case scenario is when the source and destination satellite are the same : two ground stations have the same satellite in their visibility range. This observation leads us to create an algorithm that compares the satellites, in the visibility range of our source and destination ground stations to prioritize : connections to the same satellite; satellites that are directly linked by an ISL (neighbors satellites); or satellites offering a shorter path. This selecting algorithm does not replace LEO routing, but improves the performance of already existing algorithms through the additional calculation on the in and out hops.

To evaluate the impact of this selecting strategy, we perform simulations over a simulator called Hypatia [4], created by a team of researchers from ETH Zurich [5]. To the best of our knowledge, Hypatia is the only existing satellite constellations simulator able to consider a wide range of relevant variables for network evaluation. Using public data from Starlink, Kuiper and Telesat constellations, Hypatia computes them to generate a network graph. This network graph is then passed to the satellite version of the well-known ns-3 simulator [6]. Performing over ns-3, Hypatia finally provides the results of the traffic and paths of the data packets through graphs and interactive maps generated by Cesium [7]. This simulation framework is used to illustrate the interest, no matter the LEO routing algorithm is, that it is more advantageous to consider

the  $k$ -nearest satellites to a ground station, rather than only the nearest one.

In this paper, we proceed in three steps to measure the impact of first/last hop choice on the network performance. First, in Section II, we introduce the strategy selecting the nearest in/out routing hops and illustrates its poor performance on an example of source and destination ground stations. Then, we propose a new selection strategy that can be adjusted to any routing algorithm in Section III. In Section IV, we evaluate our approach by simulation on the Telesat constellation and show the importance of the selection strategy for a shortest path-based routing algorithm. In Section V, we conclude the paper and give leads for future work aiming at extending these preliminary results to several LEO constellations and routing algorithms.

## II. PROBLEM STATEMENT

This section provides a background on the LEO constellations and routing, and then presents the problem tackled.

### A. LEO constellations background

Mega-constellations are composed of hundreds and thousands of satellites that are organized in orbits. For each orbit, the mechanics is determined by the *orbit altitude* measured from the surface of the Earth (*e.g.* LEO satellites operate at an altitude of less than 2,000 km above the Earth surface) and the *orbit inclination* given by the angle between the orbital plane and the Equator.

A set of orbits with the same altitude and inclination crossing the Equator at uniform separation from each other is called an orbital shell. Recently proposed LEO constellations may have several shells. For instance, Telesat constellation (on which we base our experiments in this paper) consists in two shells with orbit inclination of  $98.98^\circ$ , respectively  $50.88^\circ$  and orbit altitude of 1,015 km, respectively 1,325 km. These two shells will group a total number of 1,671 satellites [8].

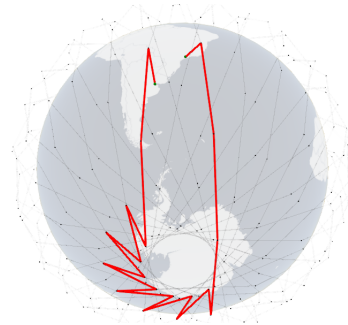
Each satellite can have two types of full duplex communication links: Inter-Satellite links (ISL) that ensure communication between satellites and Ground-to-Satellite links (GSL) for up/down communication with Ground Stations. ISL can be intra-plane connecting neighbor satellites on the same orbit and inter-plane connecting satellites on neighbor orbits. Thus, a satellite in the constellation can communicate to 4 other satellites via ISL, leading to a *+grid* pattern on the network topology.

### B. GS-to-GS: constellation routing and access

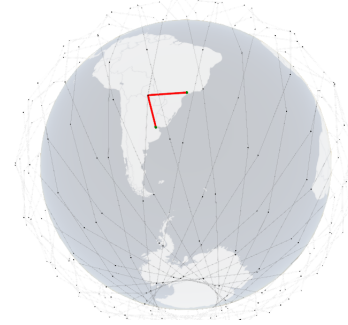
In order to establish communication between ground stations, a path on which IP packets are relayed has to be established. Inside the constellation, a routing protocol is implemented to specify for each satellite the next hop to reach the destination satellite. Given the fact that satellites change position in time and inter-satellite distances between different planes can vary in time: longest when satellites are positioned over the Equator and shortest when satellites are over the poles, the routing protocol needs to adapt as well in time and

update routes to the new satellites positions at different time steps.

Communication between the ground station and the constellation is usually based on the nearest satellite [9]. Among the satellites visible and accessible at a given moment by a ground station, the nearest one (in terms of distance) is chosen directly for data sending. When reaching the destination ground station, the packets are transmitted to the nearest satellite to the destination. Routing algorithms therefore uses the same first hop for a single source to all destinations, and the same last hop for all flows arriving at a destination ground station.



(a) Nearest satellite to source and destination ground stations.



(b) One of the  $k$ -nearest satellites to source and destination ground stations.

Fig. 1: Path through Telesat constellation - Buenos Aires to São Paulo.

### C. Selection strategy problem illustration

Communication between ground stations and the satellite constellation can be a determining factor for the overall network performance. To illustrate the access problem on up/down GSL, we propose to highlight this problem using Hypatia simulations. First, there is an important constraint to consider: the topology built by inter-satellite links in a constellation. Satellites are linked together inside a constellation by the same pattern, in *+grid*, allowing the satellites to remain connected to the same four satellites and to have a constant graph. This results in simplifying the routing. However, if satellites in the *+grid* are travelling in opposite directions (one to the north and the other one to the south), although they are neighbors in the network graph, direct communication

cannot be established. This implies that two cities very close geographically can cross a very long satellite path.

Consider the following example: São Paulo to Buenos Aires. Both cities are relatively geographically close over the map in Figure 1. However, Figure 1a shows the path used when only the nearest satellite to the ground station is considered; while figure 1b shows the path obtained when considering the 3-nearest satellites of the destination ground station. The routing path is obviously longer when the nearest satellite of a ground station is only considered. As a matter of fact, the first/last hop choice using the  $k$ -nearest satellites must be considered as it directly impact on the LEO routing performance. This observation is at the root of the FROG algorithm proposal detailed thereafter.

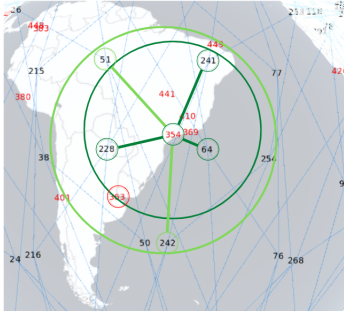


Fig. 2: 5-nearest satellites from the ground station 354 (São Paulo).

### III. CHOOSING AMONG THE $k$ -NEAREST SATELLITES

FROG algorithm improves the performance of already existing routing algorithms through an additional calculation on the in and out hops allowing to enter and exit the constellation. This section explains its basic principle.

The common format of a forwarding table is *source, destination, nexthop* where, in our context, sources are satellites or ground stations. Once again, FROG proposal is not to design a novel LEO routing algorithm, as it only focuses on selecting the entering and exiting hops of the constellation which are outside the routing decision. Basically, FROG algorithm only focuses on the lines from the forwarding table where the source is a ground station or when the source satellite is the last hop of the path (i.e., when the destination and the last hop are the same). For all other cases where the source is a satellite and that the next hop is different from the destination, the forwarding table is managed by the existing routing algorithm considering the nearest satellite. FROG algorithm keeps in memory the specified  $k$ -nearest satellites to all the ground stations at each time step. From this, and for all the considered pairs of ground stations, it compares the found  $k$ -nearest satellites to the source and destination ground stations, offering  $k^2$  possible connections. FROG runs through these  $k^2$  possibilities and writes in the forwarding table the optimal in/out hops. There are 3 different cases of optimization:

- 1) the same satellite exists in the  $k$ -nearest satellites of the source and destination ground stations, meaning that one satellite is enough to connect the two;
- 2) one of the  $k$ -nearest satellites of the source ground station has a direct ISL to one of the  $k$ -nearest satellites of the destination ground station, meaning that the satellites are direct neighbors in the constellation graph;
- 3) linking one of the  $k$ -nearest satellites of the source ground station to one of the  $k$ -nearest satellites of the destination ground station offers a shorter path than only considering the nearest satellite.

FROG algorithm prioritizes, in the above order, these cases and writes in the forwarding table the optimal combination.

---

#### Algorithm 1 Heuristic for $k$ -nearest satellites - FROG

---

**Input:** time step, constellation graph, set of GS, set of visible satellites per GS,  $k$ , previous forwarding state

**Output:** forwarding state for the given time step

Get  $k$ -nearest satellites from each ground station

**for** each pair of considered ground stations **do**

compare their  $k$ -nearest satellites and find :

**if** they have a satellite in common **then**

Write directly the optimal source and destination satellites corresponding to the pair of ground stations in a list

**else if** they have satellites which are directly connected by an ISL (direct neighbors) **then**

Write the optimal source and destination satellites corresponding to the pair of ground stations in a list

**else if** a pair of source and destination satellites offering a shorter path **then**

Write the optimal source and destination satellites corresponding to the pair of ground stations in a list

**end if**

Compute the forwarding state calculation algorithm with the list of optimal pairs

**end for**

Write the forwarding tables accordingly

---

#### A. FROG in a nutshell

The goal of this algorithm is to consider multiple source and destination satellites to create a shorter path for a set of pairs of cities. During the experiments, we noticed that the path, sometimes, went around the poles to connect two cities of the same country or continent. It means that some paths could have been set only by an ISL link or only GSL links (always considering inter-satellite links in plus-grid pattern, that is to say, connected to the satellites in front of it and behind it on its orbit, and to its sides on adjacent orbits).

Thus, the objective is to set up a heuristic to obtain shorter paths in terms of number of satellites crossed, and to reduce at the same time the unnecessary traffic in the constellation leading to potential congestion.

From the different possibilities of connections of a ground station to the satellites that are visible to it, FROG establishes a

list of the  $k$ -nearest satellites as shown in Figure 2 for  $k = 5$ . In our experiments, we did tests with  $k = 3$  and then  $k = 5$  closer or less depending on the visibility. From this list, FROG builds another list which evaluates, for each of the possible pairs of source and destination ground stations (and in priority the 100 that we evaluate in our study), the optimal pair of source and destination satellites.

### B. Implementation in Hypatia

We implemented FROG in Hypatia simulator to evaluate its performance conjointly with a LEO routing algorithm. Basically, everything is driven at the routing table calculation level (i.e., forwarding state). The forwarding state calculation function is called at each simulation time steps. FROG adds further conditions to select the first satellite, but also the last, if possible. It is important to consider that not all ground stations can always see  $k$  satellites, some are indeed not visible. FROG then seeks among the  $k$ -nearest or among those visible in case there would be less than  $k$ . All the remaining operations are unchanged.

## IV. EXPERIMENTAL VALUATION AND ANALYSIS

The measurements presented provide : 1) graphs allowing to observe the progression of data transmission with TCP for different versions of the algorithm (when considering the closest satellite, the 3 closest satellites, and finally the 5 closest satellites); 2) the evolution of the number of satellites used on average; 3) the average throughput over all pairs of source-destination ground station; 4) the distribution of the average elevation angles for each constellation which allow the stations to observe  $k = 1$ ,  $k = 3$  or  $k = 5$  satellites.

### A. Simulation setup

Hypatia is a complex but complete simulator based on a modified version of *ns-3-sat-sim*. At this stage, Hypatia only supports TCP New Reno and TCP Vegas for testing. In this paper, simulations are done with TCP New Reno. Hypatia is divided in three main parts :

- *satgen*, which 1) creates the graph of the constellation of satellites; 2) set of ground stations passed in parameters and 3) generates the corresponding routing tables for each time step specified;
- *satviz*, to visualize with Cesium the constellation ISL pattern or the path established between a specific pair of ground stations;
- and finally several scripts to specify simulation parameters and drive the final analysis.

We have therefore added in the *satgen* section FROG algorithm is therefore added to *satgen* and particularly to provide the calculation of the in and out hops of the constellation<sup>1</sup>. The worst-case scenario of our algorithm, that is to say when no advantageous pairs of satellites have been found between the two ground stations, is the usual path of the previously

<sup>1</sup>For the sake of reproducibility of these results, FROG algorithm is available on github project <https://github.com/abx13/Hypatia>

existing algorithm connecting the ground station to the nearest satellite.

We consider the first shell of the current Telesat constellation. Telesat has 351 satellites distributed in 27 orbits of 13 satellites, at an altitude of 1015 km and an inclination of 98.98°. The simulation runs for 120 seconds and for an ISL capacity of 10 Mbit/s. By default, the GSL capacity is also set to 10 Mbit/s.

### B. Simulation results

1) *Coverage evaluation*: In the histogram of Figure 3, we show how the choice of  $k$ -nearest satellites is related to the elevation angle the first hop satellite is making with the ground station. For the Telesat constellation, it can be noticed that the higher the elevation angle, the fewer satellites are visible. For a better performance of the routing algorithm, choosing elevation angles lower than 54°, allowing more than  $k = 3$  visible satellites, should be a good compromise.

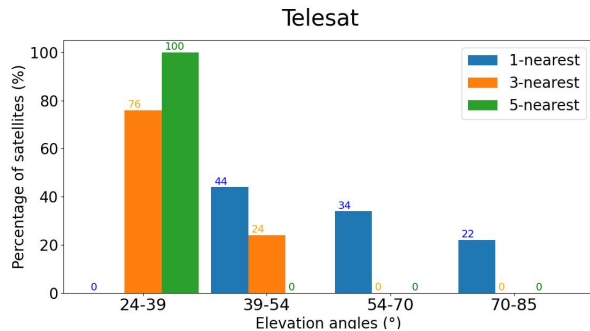


Fig. 3: Visible satellites depending on their elevation angle.

2) *Path length evaluation*: The length of the paths is evaluated in terms of number of satellite hops between the source and destination ground station in Table I. Paths that are not better have nevertheless the same number of satellites as in the  $k = 1$  case.

TABLE I: Hop reduction rate in the path length.

$k$ -nearest satellites	Number of shorter paths	Average hop reduction
3	14 out of 100	10.57%
5	24 out of 100	19.05%

The average reduction Table I is computed as follows:

$$\frac{1\text{-nearest length} - k\text{-nearest length}}{1\text{-nearest length}} \cdot 100$$

In both cases ( $k = 3$  and  $k = 5$ ), the path used is shorter by about 10% and almost 20%, which is significant and also means we do not use certain satellites for the same pairs of ground stations source and destination considered. Thus, as the constellation usage decreases, the congestion level *de facto* decreases.

3) *Throughput evaluation*: Concerning the throughput, as shown in Table II, there is a clear average increase in throughput when adjusting the choice of the first satellite. For the Telesat constellation, with our FROG algorithm, we obtained

an increase of around 7.3% for  $k = 3$  and around 18.2% for  $k = 5$ .

TABLE II: Average throughput.

$k$ -nearest satellites	Average throughput (Mbit/s)
1	2.47
3	2.65
5	2.92

4) *TCP performance evaluation*: Another important criterion of this evaluation is to determine the impact of FROG selection algorithm on the transport protocol behavior. In Figure 4, we show experimental measurements for TCP New Reno using standard metrics such as TCP congestion window size, round-trip time (RTT) and data transmission rate.

TCP data transmission performs better if 3-nearest or 5-nearest algorithm is used, as shown on the bottom graph of all three sub-figures from Figure 4 (for different pairs of cities). For example, for data transmission between Paris and Kabul, with the use of 5-nearest, TCP transmits much faster the data (in 70-80 seconds) than with the use of 1-nearest (not finished yet in 120 seconds of simulation). We can clearly see a lower data rate loss and a faster transmission of data throw the constellation.

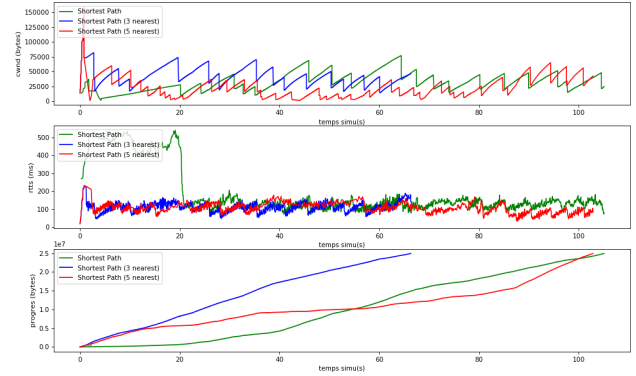
## V. CONCLUSION

Considering the  $k$ -nearest satellites to the ground stations allows optimizing the paths taken by choosing advantageous in and out hops. The preliminary results showed in this paper have been computed on Telesat and using the Floyd-Warshall shortest path algorithm as a base, but the FROG algorithm can be based on any routing algorithm and implemented on any constellation.

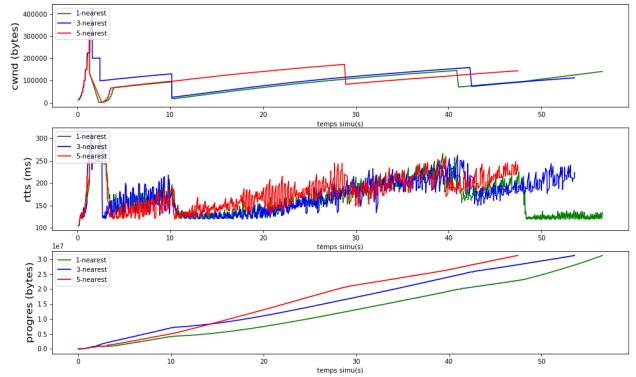
For future work, we expect to evaluate the FROG algorithm on Kuiper and Starlink. For this, we have to note that the altitude of the shell chosen is critical: it will influence the number of satellites visible for a ground station and therefore impact the performance of the FROG algorithm. We will also take other routing algorithms as a base (i.e. Multi Commodity Network Flow) to also optimize their routing.

## REFERENCES

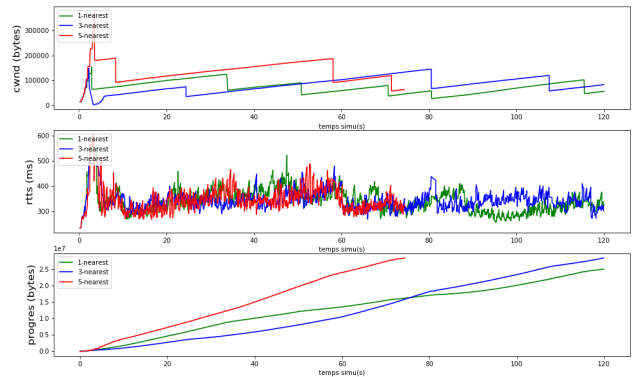
- [1] "SpaceX's approach to space sustainability and safety," <https://www.spacex.com/updates/>, accessed: 2022-07-30.
- [2] "Amazon receives FCC approval for Project Kuiper satellite constellation," <https://www.aboutamazon.com/news/company-news/amazon-receives-fcc-approval-for-project-kuiper-satellite-constellation>, accessed: 2022-07-30.
- [3] "Telesat lightspeed technology infographic," <https://www.telesat.com/wp-content/uploads/2021/04/Telesat-Lightspeed-Technology-Infographic.pdf>, accessed: 2022-07-30.
- [4] snkas, "Github of hypatia," accessed: 2022-07-30. [Online]. Available: <https://github.com/snkas/hypatia>
- [5] S. Kassing, D. Bhattacharjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the" internet from space" with hypatia," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 214–229.
- [6] P. M. Silva, "ns3-satellite," 2016. [Online]. Available: <https://gitlab.inesctec.pt/pmms/ns3-satellite/-/tree/master/satellite>
- [7] Cesium, "The platform for 3d geospatial," accessed: 2022-07-30. [Online]. Available: <https://cesium.com/>



(a) Qingdao (432) to Shantou (446)



(b) Madrid (405) to Bogotá (383)



(c) Paris (375) to Kabul (431)

Fig. 4: TCP behavior in Telesat constellation with SP algorithm using the nearest satellite ( $k = 1$  in green), one of the 3-nearest ( $k = 3$  in blue), one of the 5-nearest ( $k = 5$  in red). For each figure, TCP congestion window in bytes on top, RTT in ms in the middle, and the cumulative number of bytes transferred at the bottom.

- [8] Telesat, "Application for modification of market access authorization," 2020, accessed: 2022-07-30. [Online]. Available: <https://fcc.report/IBFS/SAT-MPL-20200526-00053/2378318.pdf>
- [9] C. Duan, J. Feng, H. Chang, B. Song, and Z. Xu, "A novel handover control strategy combined with multi-hop routing in LEO satellite networks," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 845–851.