Rethinking LEO Constellations Routing

Paul Grislain* Alexia Auddino*, Anna Barraqué*, François Lamothe*,
Oana Hotescu*, Jérôme Lacan*, José Radzik*, Emmanuel Lochin*

∗ Fédération ENAC ISAE-SUPAERO ONERA,
Université de Toulouse, France

Abstract

This study investigates the Unsplittable Multi-Commodity Flow (UMCF) as a routing algorithm for LEO constellations. Usually, LEO routing schemes enable the Floyd-Warshall algorithm (Shortest Path) to minimize the end-to-end latency of the flows crossing the constellation. We propose to solve the UMCF problem associated with the system as a solution for routing over LEO. We use a heuristic algorithm based on randomized rounding known in the optimization literature to efficiently solve the UMCF problem. Furthermore, we explore the impact of choosing the first/last hop before entering/exiting the constellation. Using network simulation over Telesat constellation, we show that UMCF maximizes the end-to-end links usage, providing better routing while minimizing the delay and the congestion level, which is an issue today over new megaconstellations. **Keywords:**LEO constellations, routing, Floyd-Warshall, UMCF, Hypatia, ns3 simulator

1 Introduction

Over the past couple of years, Low Earth Orbit (LEO) satellite constellations have raised a lot of interest for several "New Space" companies offering satellite-based Internet access: SpaceX with Starlink [2], Amazon with Kuiper [1] or Telesat [3]. This new approach competes with standard satellite communication using geostationary (GEO) orbit operating at $35,786\,km$ above the Earth surface, that can offer wide coverage with the cost of a high communication latency (more than $600\,ms$) [14], but also with optical fiber-based terrestrial networks [12]. LEO constellations aim at using thousands of satellites to provide connectivity everywhere on Earth, even for difficult to access areas with terrestrial networks (such as rural or mountain areas) and not in the spot of GEO satellites. Since satellites are close to Earth, low communication latency (*i.e.,* in the order of tens of milliseconds) are reachable. In addition, inter-satellite links (ISL) have the capacity to run at the speed of light in a vacuum, which is higher than in optical fiber used for terrestrial networks [12], and to provide high bandwidth thanks to the laser-based transmission technology.

LEO constellations open new communication opportunities, however, several aspects need to be investigated to take advantage of their full potential. In communication networks, the end-to-end latency and link capacity utilization are usually impacted by the routing protocols and the congestion occurring in intermediate routers. These elements can have an important impact on the performance of LEO constellations. Following the huge number of satellites and their high velocity, finding an optimal inter-satellite routing configuration to reduce latency and congestion, while maximizing the network capacity is a complex problem. This motivates the present study, which seeks to design a routing algorithm that improves satellite link usage while reducing the possibility of network congestion.

Shortest path algorithms are the commonly used routing strategy to reduce latency. However, this class of algorithm is based on a fixed path metric (length, weight, ...) and does not consider network congestion that leads to network latency. Another approach would be to prevent congestion on inter-satellite links by distributing flows on less occupied links. But how to precompute network paths that would take into account congestion? We claim that this question can be answered by a novel routing protocol based on the Unsplittable Multi-Commodity Flows (UMCF) algorithm described in [15] which is at the root of this study.

To sum up, the contributions presented in this paper are the following:

- we introduce an innovative routing protocol focusing on maximizing the amount of IP traffic crossing the constellation instead of minimizing the latency. This is considered by using the Sequential Randomized Rounding (SRR) for the Unsplittable Multi-Commodity Flow (UMCF) problem instead of the standard Floyd-Warshall Shortest Path (SP) algorithm;
- we extend Hypatia ns3-based simulator with the implementation of the UMCF algorithm, and evaluate its routing performance;

- we experimentally compare UMCF routing with the shortest path routing, and show that UMCF routing demonstrates the best performance over Telesat constellations;
- we finally investigate the impact of the choice of the first/last hop before entering/exiting the constellation. We show that choosing advantageous in and out hops, among the k-nearest satellites to source and destination ground stations, improves the performance of both routing algorithms.

This paper is organized as follows: Section 2 presents recent advances and challenges in LEO constellations. We then introduce our solving algorithm based on the Sequential Randomized Rounding (SRR) method for Unsplittable Multi-Comodity Flows (UMCF) in Section 3 and investigate the choice of the in/out hop in Section 4. Section 5 presents the Hypatia simulation environment and the simulation results obtained with the investigated routing algorithms. Finally, Section 6 concludes this study and gives leads for future work.

2 Advances and challenges in LEO constellations

This section presents recent advances and challenges in LEO constellations and new mega-constellations. We first point out a recent congestion issue observed over the Starlink mega-constellation[1] and detail routing issue and how the multi-commodity flow problem can solve it.

In the past five years, the interest in satellite Internet constellations raised due to the dropping cost of launching and increasing demand for a broader Internet access. Mega-constellations are composed of hundreds and thousands of satellites that are organized in orbits. Each orbit is characterized by a set of parameters that determine the orbit mechanics, as follows:

- orbit altitude measured from the surface of the Earth. Low Earth orbit satellites operate at an altitude of less than $2,000\,km$ above the Earth surface;
- orbit inclination, given by the angle between the orbital plane and the Equator. When the orbit inclination is low, orbits are called equatorial or near equatorial orbits. Orbits that pass above or nearly above the poles on each revolution with inclinations close to 90° are polar orbits.

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 of two shells with orbit inclination of 98.98°, respectively 50.88° and orbit altitude of $1,015\,km$, respectively $1,325\,km$. These two shells will group a total number of $1,671$ satellites [21].

Apart from Telesat, other companies aim at bringing satellite Internet access to a broader audience. In particular, there are:

- Starlink, operated by SpaceX, composed of approximately 3000 small LEO ($550\,km$) satellites at this time, knowing more than 12000 are planned to be deployed;
- Kuiper, operated by Blue Origin, aims at sending 3236 satellites in LEO ($600\,km$);
- OneWeb, currently deployed with 462 of the 648 scheduled LEO satellites.

The performance of these mega-constellations is relatively good, as shown in [18, 13]. However, congestion has become a big issue over Starlink obliging the operator to throttle customers connection speed below a given amount of data, which is at the opposite of their initial user contract. This congestion seems obviously linked to the success of Starlink and the increase in users. As a matter of fact, from the end of the year 2021 until now, the available throughput for a Starlink user is decreasing[2].

Routing and congestion control are two interrelated processes in a resource control system managing a constellation. Both attempt to solve two distinct problems: (1) how to determine the optimal allocation for a given static condition, and (2) how to cope with dynamic changes in resources. The first problem is often called optimal routing and flow control, and is actually a special case of the multi-commodity network flow problem in operations research studies. Real optimal algorithms are often impractical for real networks due to their computational complexity and excessive information exchange. However, the algorithm proposed here is based on simplified assumptions, but yields good results that prevent congestion in constellations.

Routing over SATCOM might be considered as a low complexity task, considering the determinism that prevails in this type of system. Basically, it is possible to compute the future position of the satellites regarding the various ground relays and to determine the possible connections. Thus, a complete temporal graph of the

---

[1] See for instance: ”Congested, contested... under-regulated and unplanned”, by Stuart Eves, Surrey Satellite Technology Limited, Guildford, UK, Issue #3(29) 2021, ROOM Space Journal of Asgardia, available online `https://room.eu.com/article/congested-contested-under-regulated-and-unplanned`.

[2] https://mybroadband.co.za/news/broadband/467999-starlink-gets-fair-use-policy-will-start-throttling-in-congested-areas.html

connections can be built between each network elements. Once this tree is built, it is theoretically possible to find the route between a source and a destination that optimizes a particular function of the chosen metrics. The problem becomes an optimization problem of weight functions over a graph. For instance, the algorithm *Contact Graph Routing* (*CGR*) is based on this optimization result to make its routing decisions [10]. The main complexity relates to the necessary computing time. Actually, the graph of connections previously mentioned can become particularly large, as its size is directly linked to the number of contact opportunities and nodes present in the system. Two possibilities are then offered to improve the computation time of the solution:

- to find a search algorithm on the connection tree of less complexity;
- to simplify the connection tree without degrading the results, *i.e.,* without removing the optimal route by a clumsy truncation of the graph.

Concerning the first point, the existing algorithms used for tree analysis are already optimal. This leaves the second point, where two cases appear: whether we seek to optimize the delay or the capacity. If we consider only the transmission delay, the goal is to simplify the connection graph both in width, assuming that contacts that are very far in the future will not be useful, and in depth, assuming that the optimal route has a limited number of hops. This idea is presented and developed in H. C. Sanchez's thesis about Store and Forward routing in satellite constellations [5] and has been deeply investigated.

However, if we are interested in link usage rather than delay, there are many mathematical theories and algorithms allowing to find the routing that optimizes the amount of information that can be exchanged between the different nodes of the network. For instance, the max-flox min-cut theorem [17, 7] and the multi-commodity theory [16] which takes up the previous theorem in the case of several flows present in the network to optimize the global flow. As the latter has been only too recently considered, this motivates the present paper that aims at investigating a solving method for the Unsplittable Flow Problem (UFP), that is a widely studied variant of the Multi-Commodity Flow Problem (UMCF). Thus, we will use the Sequential Randomized Rounding (SRR) algorithm described in the paper [15]. The SRR algorithm improves the Randomized Rounding algorithm from [19] by sequentially solving the relaxed (linear) problem, stating some commodities and loop back until finished.

3 Routing inside LEO constellations with UMCF

Routing protocols based on the Shortest Path algorithm, often used in LEO constellations, aim at minimizing the distance between satellites connected to source and destination ground stations to reduce the delay required to transfer packets in the constellation. Given the network topology, the shortest path may be the same for several flows, so a subset of inter-satellite links may be more solicited than others, which can lead to an increase in the delay due to congestion. To solve this congestion problem, we propose a routing strategy based on the Unsplittable Multi-Commodity Flows (UMCF) that allows distributing flows on longer paths as well. The objective is to globally maximize the utilization of links capacity and to reduce delays by reducing congestion in the network. Next, we present this strategy and our approach to solve the optimization problem.

3.1 Unsplittable Multi-Commodity flows and Sequential Randomized Rounding

3.1.1 Problem description

The unsplittable flow problem is an extensively studied variant of the well-known maximum flow problem. In this problem, one is given a directed or undirected graph, together with capacities on its arcs. A family of commodities, each composed of an origin, a destination, and a demand, is also provided. Each commodity has to route its demand from its origin to its destination through a unique path. The routing must ensure that capacities on the arcs are not exceeded by the flow of the commodities, or at least minimize the violation of the capacities.

This problem is NP-hard which makes it very difficult to find optimal solutions for instances of the problem with several hundred nodes and commodities such as the one we are considering. However, some heuristics and approximation algorithms can find solutions of excellent quality even on such large instances. In this study, we will consider the Sequential Randomized Rounding (SRR) algorithm described in [15] which is an extension of the Randomized Rounding algorithm [19].

3.1.2 Mathematical formulation for unsplittable flows

To formally introduce the problem through a mathematical formulation, we use the following notations:

- $G = (V, E)$ is a directed or undirected graph, with V the set of nodes and E the set of arcs;
- $L = (o_k, d_k, D_k)_{k \in K}$ is a set of commodities defined by their origin, destination, and demand;

- $(c_e)_{e \in E}$ are capacities on the arcs.

We also use the Kronecker notation, δ_x^y equals 1 if $x = y$ and 0 otherwise. The sets of arcs incoming and outgoing of node v will be noted $E^-(v)$ and $E^+(v)$ respectively. With these notations, the unsplittable flow problem can be more formally described with the following mixed integer linear program :

$$\min_{f_{ek}, \Delta_e} \sum_{e \in E} \Delta_e \tag{1a}$$

such that

$$\sum_{e \in E^+(v)} f_{ek} - \sum_{e \in E^-(v)} f_{ek} = \delta_v^{o_k} - \delta_v^{d_k} \qquad \forall k \in K,\ \forall v \in V, \tag{1b}$$

$$\sum_{k \in K} f_{ek} D_k \leq c_e + \Delta_e \qquad \forall e \in E, \tag{1c}$$

$$f_{ek} \in \{0, 1\},\ \Delta_e \in \mathbb{R}^+ \qquad \forall k \in K,\ \forall e \in E. \tag{1d}$$

In this formulation, the variable f_{ek} indicates whether commodity k pushes flow on arc e and the variable Δ_e represents the overflow on arc e. Equation (1b) corresponds to the flow conservation constraints. It ensures that, for each commodity and every node except the origin and the destination of the commodity, the same amount of flow of the commodity goes in and out of the node. Equation (1c) corresponds to the capacity constraints. It ensures that the capacity of an arc is respected or that the overflow is recorded in Δ_e. Finally, the fact that $f_{ek} \in \{0, 1\}$ ensures that each commodity is allowed to use only one path to route its flow.

3.1.3 Description of the heuristic

Before diving into the heuristic, we introduce a central concept for randomized rounding algorithms, which is the linear relaxation of a mixed integer linear program. To obtain the linear relaxation of a mixed integer linear program, one just removes the integrality constraint on the variables that to have one. In our case, this means replacing the constraint $f_{ek} \in \{0, 1\}$ with a constraint $f_{ek} \in [0, 1]$. This relaxation has an impact on the modeled problem, and in our case the linear relaxation of the unsplittable flow problem is the multi-commodity flow problem. This relaxed problem is identical to the previous one, except that each commodity is allowed to split its flow on several paths.

Let us now describe the heuristic. The SRR algorithm, presented in Algorithm 1 alternates between two different steps:

- solving the linear relaxation where each commodity can send its flow on several paths;
- fixing some commodities to a unique path among those proposed by the linear relaxation using randomized rounding.

Algorithm 1 The SRR heuristic

```
Require: $G = (V, E, c)$ a capacitated graph, $L = (o_k, d_k, D_k)_{k \in K}$ a list of commodities
Sort the commodities by decreasing demand
Set $K_{fixed} = \varnothing$        ▷ $K_{fixed}$ is the set of indices of commodities fixed to a single path
for each commodity $k^*$ in decreasing demand order do
    if an actualization is needed then
        $((x_{pk})_{p \in P_k})_{k \in K} = Solve\_Linear\_Relaxation\left(G, L, K_{fixed}, (p_k)_{k \in K_{fixed}}\right)$
    end if
    Draw a path $p^*$ from $P_{k^*}$ with probability $x_{p^* k^*}$
    Add index $k^*$ to $K_{fixed}$.
    $p_{k^*} = p^*$
end for
return $(p_k)_{k \in K}$
```

These steps are iterated upon until all the commodities have been assigned to a unique path. Solving the linear relaxation is easier than solving the original problem because polynomial time algorithms are known to efficiently solve the multi-commodity flow problem. This can be done, for example, by the above formulation to a linear programming commercial solver, but more specific algorithms have been designed in the multi-commodity flow literature [9, 6]. In the experiments, we use the commercial solver Gurobi [11] on the above formulation with an aggregation technique presented in [15] to reduce the computing time. More details on when to actualize the linear relaxation between randomized rounding steps are also given in [15].

As for the fixing of the path of a commodity, a randomized rounding step proceeds as follows. In the solution of the linear relaxation, each commodity may send its flow on several paths. Let us note x_{pk} the proportion of the flow of a commodity k that is sent on path p. A path will be chosen randomly among the one used by commodity k in the linear relaxation. The probability of choosing path p is x_{pk}. The commodity is then forced to use only this path in the final solution and in the subsequent resolution to the linear relaxation.

An important detail of the SRR algorithm is the order in which the commodities are fixed to a unique path. In the SRR heuristic, paths are assigned to the commodities in decreasing order of commodity's demand, *i.e.,* the commodities with larger demands have their paths chosen first. The rationale behind this ordering is to allocate commodities with a large demand first, while a large amount of capacity is left in the arcs. Commodities with smaller demands are then used to fill the remaining gaps. It has been shown that this order has a large impact on the quality of the solution returned by the heuristic.

3.1.4 On the constellation dynamic

Because they orbit the earth at low altitude, constellations of satellites are highly dynamic systems. To take this characteristic into account, the constellation is thus considered at different time steps. Moreover, due to the ever-changing structure of the network in these systems, users need to change the path used to transmit their traffic over time. However, path changes disrupt the connection of the user and tend to lower the quality of service. To have a high service quality, the routing decisions must try to minimize the number of path changes over time. Thus, when making routing decisions at a time step, we seek to penalize routing decisions that do not use the same path as in the previous time step. This can be done by slightly modifying the UMCF problem and thus the mixed integer linear program that models it. More precisely, if we note x_k the variable deciding whether user k uses the path p_k from the previous time step and P the penalization endured in case of a path change for any user, then the modified formulation for the UMCF problem is the following:

$$\min_{f_{ek},\Delta_e} \sum_{e\in E} \Delta_e + P\sum_{k\in K}(1-x_k) \tag{2a}$$

such that

$$\sum_{e\in E^+(v)} f_{ek} - \sum_{e\in E^-(v)} f_{ek} = \delta_v^{o_k} - \delta_v^{d_k} \qquad \forall k\in K,\ \forall v\in V, \tag{2b}$$

$$\sum_{k\in K} f_{ek}D_k \le c_e + \Delta_e \qquad \forall e\in E, \tag{2c}$$

$$x_k \le f_{ek} \qquad \forall e\in p_k, \forall k\in K \tag{2d}$$

$$f_{ek}\in\{0,1\},\ \Delta_e\in\mathbb{R}^+ \qquad \forall k\in K,\ \forall e\in E. \tag{2e}$$

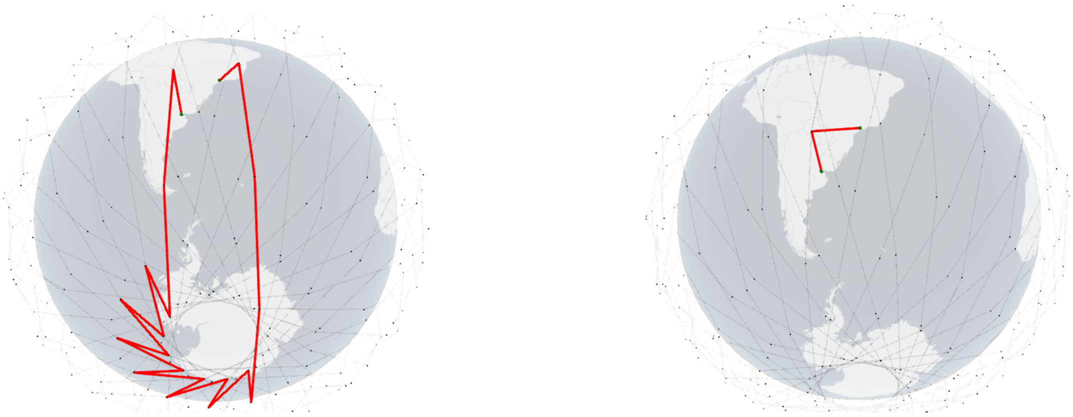
The new term $P\sum_{k\in K}(1-x_k)$ in the objective function counts the number of penalization induces by the solution, and the new constraint $x_k \le f_{ek}$ links the variable x_k with the variables f_{ek} that decide what path user k uses.

4 From the Ground Station (GS) to the constellation: on selecting advantageous in and out hops among the k-nearest satellites

Communication between ground stations and the satellite constellation can be a determining factor for the overall network performance and is usually based on the nearest satellite[8]. 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.

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



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

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

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

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.

Choosing among the k-nearest satellites 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.

5 Performance evaluation by simulation

In this section, we evaluate the performance of the proposed routing algorithms by simulation with Hypatia [20]. First measurements investigate the impact of routing strategies on the link capacity usage, followed by end-to-end delay measurements.

5.1 Simulation environment

The Hypatia simulator is built on a set of tools allowing the analysis of satellite networks. The main tools are illustrated in Figure 2.

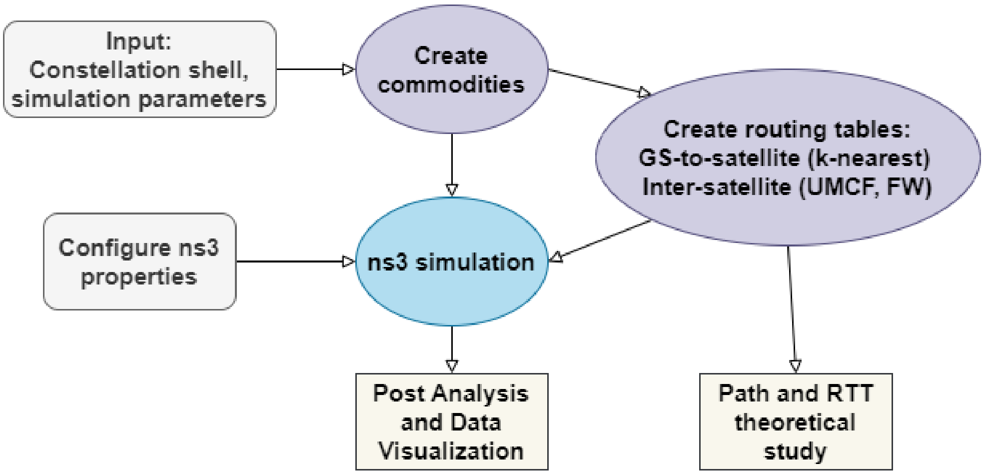


Figure 2: Flow diagram of Hypatia scripts

Hereafter, we enumerate the functions of these tools:

- a generator of satellite constellation, which gathers information from a single constellation shell;
- a grid generator, which establishes connections between satellites in the constellation. Based on these connections, a graph is generated corresponding to the following situations:
  - inter-satellite communication: one satellite is connected to four other satellites, which are always the same. Thus, the satellite graph is constant;

  - communication between ground stations and satellites: on this satellite graph, ground stations are connected by links. These links allow a ground station to connect to the k-nearest satellite.

- a simple commodity generator, which creates flows between ground stations in different cities;
- a route generator, which computes routes at regular time steps. The satellites rotate around Earth, and the route generator updates positions and distances between nodes before computing an optimal path between ground stations. It outputs routing tables.
- simulation parameters: choose commodities (from city, to city, data rate or data size), configure links capacities, transport protocol, data to record (mainly link usage and latency) and other specific parameters;
- the ns3 simulator uses these parameters, the constellation and the pre-computed routing tables to simulate the data flows in the network;
- post-processing tools are used. Measures can be compared to theoretical analysis based on the routing tables.

Different configurations of satellites and ground stations interfaces can be used. We mainly used the configuration where a ground station can only connect to one satellite, but a satellite can be connected to many ground stations via the same ground interface embedded in the satellite. Hypatia brings simple laser inter-satellite connection, and radio connection to the ground.

5.2 Experimental setup

In our experiments with Hypatia, we consider the first shell of the Telesat constellation with 351 satellites at an altitude of $1,015\,km$, distributed over 27 orbits, at an inclination of 98.98° [21]. The model of communications between ground stations and satellites is elementary: each ground station has one interface to communicate with the satellite and each satellite has one ground interface allowing communication with several ground stations. The ground interface of the satellite is limited by a maximum capacity, *e.g.,* the same as the Inter-satellite links.

In our experiments, we consider one ground station in each 100 largest cities of the world. Each ground station is connected to one of the k-nearest satellites. Communication flows are generated by the commodity generator as follows. First, it creates 50 pairs between the 100 most populated cities in the world by a random combination without repetition. Then, for each of these 50 couples, communication flows are exchanged between connected cities at a random data rate value ranging from 0.7 to $1.3\,Mbps$. As we are interested in traffic flow and routing performance, all flows are handled with the UDP protocol, allowing links utilization at their maximal capacity.

We aim at studying how the routing protocols perform when congestion occurs, so we introduce this phenomenon by varying the capacity of Inter-satellite links (ISL) and Ground to Satellite links (GSL). In our experiments, we thus consider the links capacity ranging from $2\,Mbps$ (highly congested links) to $10\,Mbps$ (no congestion). Note these capacities are chosen to quickly converge to congestion and are not representative of real ISL capacities indeed (usually, ranging from $20\,Gbps$ to $50\,Gbps$).

In our experiments, we consider scenarios of $120\,s$ duration, where routing tables are updated every $2\,s$. The distance computations between nodes are updated for each packet transmission using the current satellite position.

A particular aspect of LEO constellations is their dynamic. Satellites move on the orbit with a high velocity (around $27,000\,km/h$ [4]) leading to changes in the network graph. We consider these changes in the simulations and update the routing tables frequently. To validate the consistency of the frequency on which routing tables are updated, we check if there is an impact of the time step on the network performance. In Figure 3, we investigate the impact of time step considered between consecutive computations of the routing tables. We consider time steps varying between $500\,ms$ to $10\,s$ and evaluate the performance of the SP and UMCF routing algorithms combined with the choice of the 1-nearest and 3-nearest first/last hop satellite. We observe no notable difference in the amount of data received with all the algorithms in the less favorable scenario, considering $2\,Mbps$ ISL capacity. We observe that routing algorithms are stable in time and do not require frequent updates of the routing tables.

5.3 Simulation results

Next, we present the experimental results obtained with routing tables generated with the Shortest Path algorithm and the UMCF algorithm.

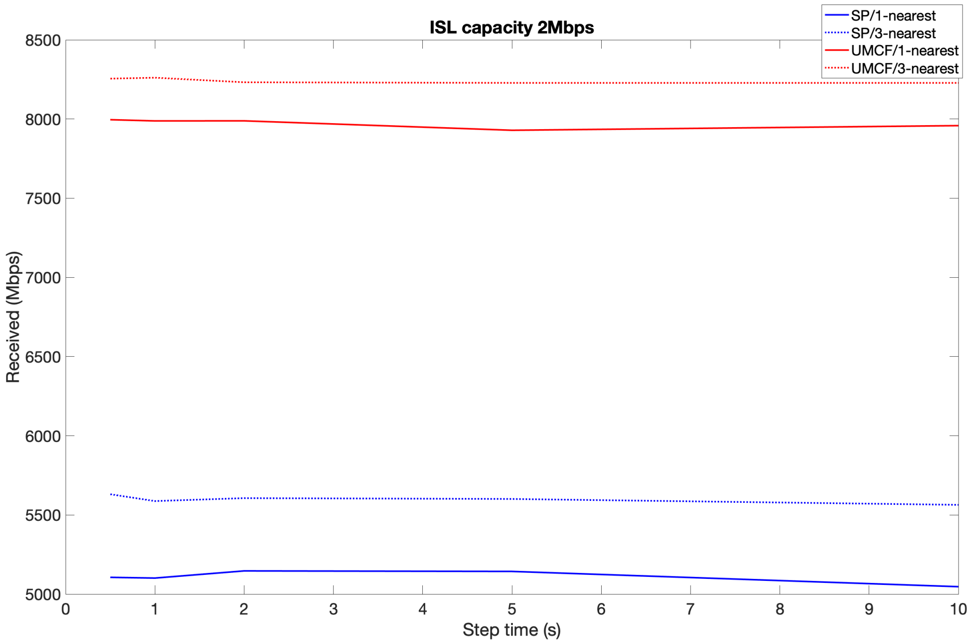


Figure 3: Impact of the time step between consecutive routing tables generations on the performance of proposed routing algorithms (11919.67 Mbps of data are sent during this experiment).

5.3.1 Link usage evaluation

In Figure 4, we evaluate the link usage of the routing algorithms for different ISL capacities. This link usage is a simple ratio between the total amount of data received over data sent. When the ISL link capacity is high, near $10\,Mbps$, we observe no congestion and both algorithms SP and UMCF get similar results. But as the ISL capacity decreases, there is more congestion, and we observe that distributing flows on different routes reduces losses. Concerning the choice of the first and last satellite, we see that choosing between the 3-nearest satellites offers a better performance than the 1-nearest satellite for both SP and UMCF algorithm.

In Figure 5, we evaluate the differences between cities. We get the commodities link usage of every simulation. By aggregating them by the source city, we get around 15 measurements per city. For each city, we take the median value of these measurements for different ISL rates. The commodities differ a lot by their destination and their flow demand. The standard deviation between results is large, but taking the median value gives us an idea of a representative travel, with a good compromise between the data path length and the flow limitations.

When the ISL rate is high, there are few differences between cities with both SP and UMCF algorithms, combined with 1-nearest or 3-nearest satellite at first/last hop. This can be checked by seeing that success ratio is near to 1 for all scenarios involving an increasing number of cities on the X-axis of Figure 5.

When the ISL rate decreases, we observe that this median value varies much more. Using the same commodities, results show that depending on the routing algorithm, the commodities are not served fairly. With the Shortest Path routing algorithm, we see that a few cities get good results at the expense of all the other commodities. In this case, 3-nearest algorithm offers slightly better results than the 1-nearest algorithm.

On the contrary, the UMCF routing algorithm helps to flatten this plot, the measured packet ratio becoming closer to 1 for low ISL data rates with both 1-nearest and 3-nearest algorithms. In the congested case considering $2\,Mbps$ ISL capacity, we can note that the packet ratio is higher than the one obtained with the Shortest Path algorithm.

5.3.2 Latency evaluation

In Figure 6, we measure the median round trip time of commodities generated by each city. We can observe that for high ISL capacities, the performance of the UMCF algorithm is similar to the SP algorithm one, in terms of latency (both less than $0.3\,s$). We observe in Figure 6 that for $2\,Mbps$ ISL capacity, the severe congestion prevents the transmission of some commodities when the SP algorithm is used. On the contrary, the UMCF algorithm better handles congestion for low ISL capacity and allows transmission of a higher number of commodities, at the cost of a higher latency. So, for the congested case, we can note that the best performance is obtained with the UMCF algorithm combined with the 3-nearest satellite first/last hop.

6 Conclusion and future work

We investigated the performance of an Unsplittable Multi-Commodity Flow (UMCF) algorithm to generate routing tables on the Telesat LEO mega-constellation. Results show that for high ISL data rates, when there is no congestion in the network, *i.e.*, the network is oversized, the shortest path computed with Floyd-Warshall algorithm and the UMCF algorithm have similar performance. But when congestion occurs, results show that the proposed UMCF routing algorithm establishes better routing tables than the shortest path algorithm. In addition, we study the impact of choosing the first/last hop among several visible satellites before entering/exiting the constellation. We show that selecting the nearest satellite does not always lead to the best performance. However, considering instead the k-nearest satellites to the ground stations allows optimizing the paths taken by selecting advantageous in and out hops with both shortest path and UMCF algorithms. As future work, we aim at investigating the impact of the routing algorithm on the Quality of Service of applications using different constellations variant such as Kuiper and Starlink.

Acknowledgement

The authors wish to thank Nicolas Pelissier for the help in setting the initial simulation experiments during his Master research project.

References

[1] 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.

[2] Spacex's approach to space sustainability and safety. `https://www.spacex.com/updates/`. 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] Debopam Bhattacherjee and Ankit Singla. Network topology design at 27,000 km/hour. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 341–354, 2019.

[5] Hugo Cruz Sanchez. *Routage pour une constellation de satellites Store and Forward dans des conditions difficiles*. PhD thesis, Paris, ENST, 2008.

[6] Weibin Dai, Jun Zhang, and Xiaoqian Sun. On solving multi-commodity flow problems: An experimental evaluation. *Chinese Journal of Aeronautics*, 30(4):1481–1492, 2017.

[7] George Bernard Dantzig and Delbert Ray Fulkerson. On the max flow min cut theorem of networks. Technical report, Rand Corp Santa Monica, Ca, 1955.

[8] Chaofan Duan, Jing Feng, Haotian Chang, Bin Song, and Zhikang 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)*, pages 845–851. IEEE, 2018.

[9] Lisa K Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13(4):505–520, 2000.

[10] Juan A Fraire, Olivier De Jonckere, and Scott C Burleigh. Routing in the space internet: a contact graph routing tutorial. *Journal of Network and Computer Applications*, 174:102884, 2021.

[11] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020.

[12] Mark Handley. Delay is not an option: Low latency routing in space. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pages 85–91, 2018.

[13] Mohamed M Kassem, Aravindh Raman, Diego Perino, and Nishanth Sastry. A browser-side view of starlink connectivity. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 151–158, 2022.

[14] Oltjon Kodheli, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, Jesus Fabian Mendoza Montoya, Juan Carlos Merlano Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff, et al. Satellite communications in the new space era: A survey and future challenges. *IEEE Communications Surveys & Tutorials*, 23(1):70–109, 2020.

[15] François Lamothe, Emmanuel Rachelson, Alain Haït, Cedric Baudoin, and Jean-Baptiste Dupé. Randomized rounding algorithms for large scale unsplittable flow problems. *Journal of Heuristics*, 27(6):1081–1110, 2021.

[16] Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. Technical report, Massachusetts Inst Of Tech Cambridge Microsystems Research Center, 1989.

[17] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[18] François Michel, Martino Trevisan, Danilo Giordano, and Olivier Bonaventure. A first look at starlink performance. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 130–136, 2022.

[19] Prabhakar Raghavan and Clark D Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[20] snkas. Github of hypatia. Accessed: 2022-03-10.

[21] Telesat. Application for modification of market access authorization, 2020. Accessed: 2022-03-10.

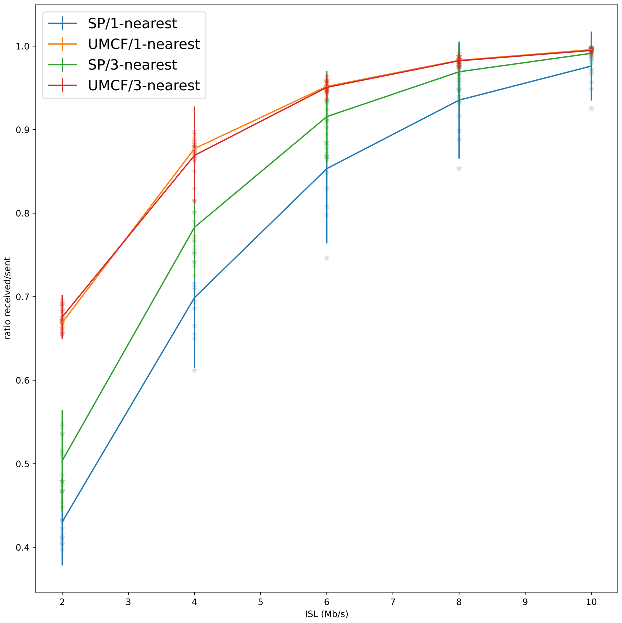
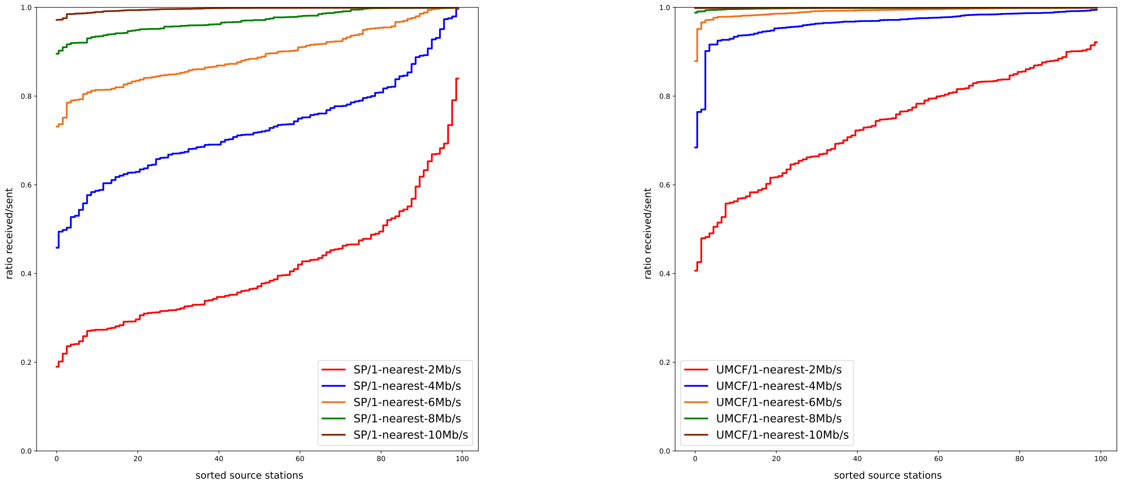
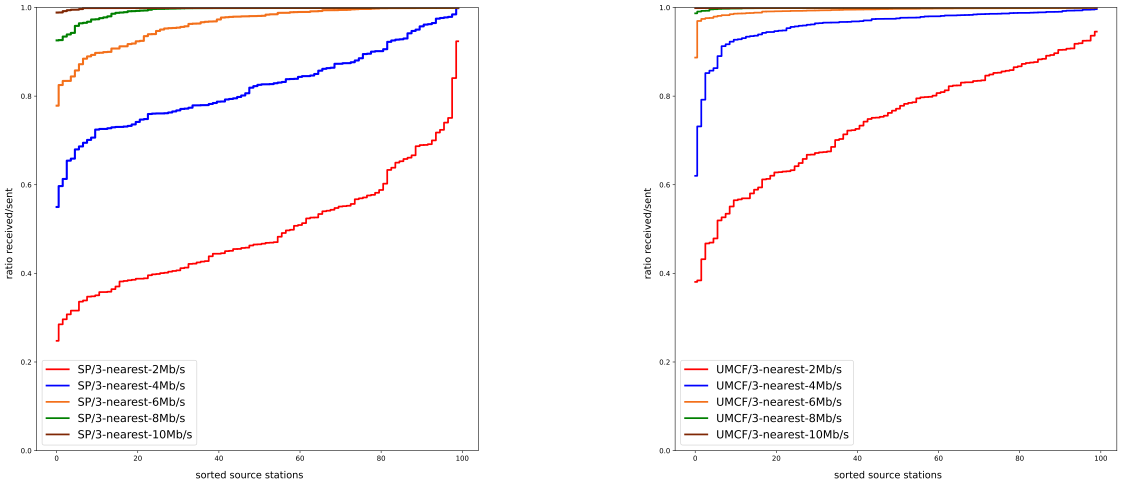


Figure 4: Link usage (ratio between traffic received/sent) as a function of the ISL capacity



(a) SP 1-nearest

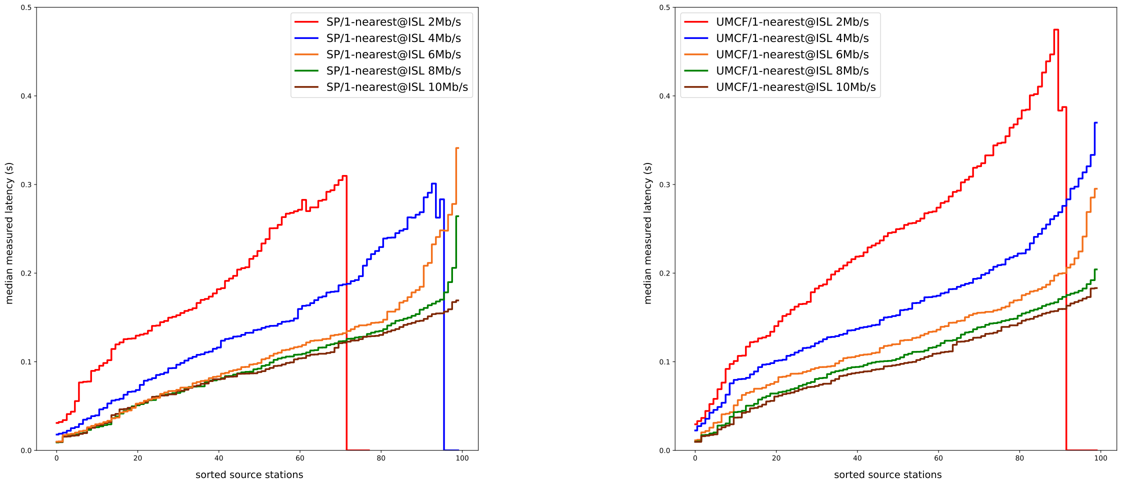
(b) UMCF 1-nearest



(c) SP 3-nearest

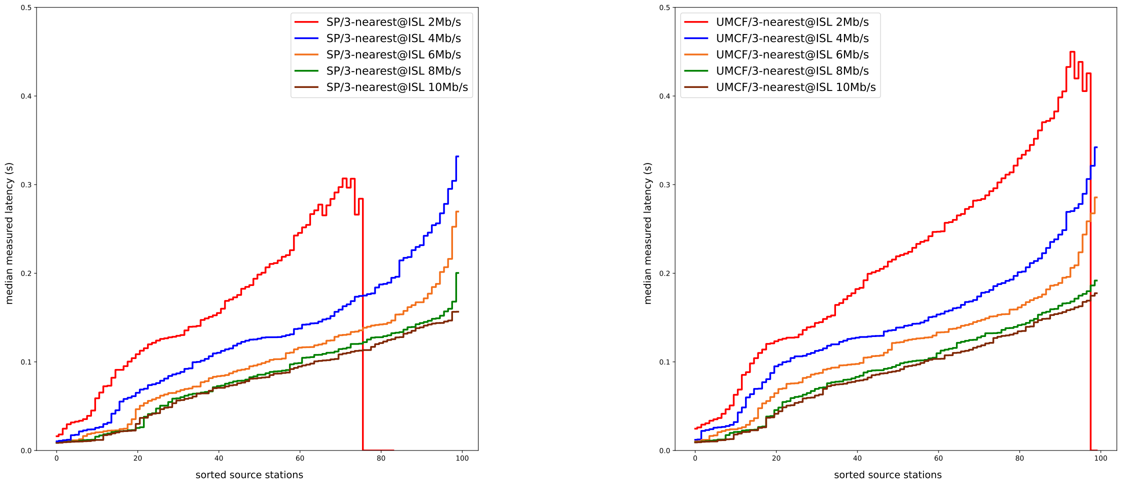
(d) UMCF 3-nearest

Figure 5: Median success rate of commodities gathered by departure city



(a) SP 1-nearest

(b) UMCF 1-nearest



(c) SP 3-nearest

(d) UMCF 3-nearest

Figure 6: Median latency of commodities gathered by departure city.