Aalborg Universitet

Autonomous Routing for LEO Satellite Constellations with Minimum Use of Inter-Plane Links

Soret, Beatriz; Smith, Daniel

Published in:
ICC 2019 - 2019 IEEE International Conference on Communications (ICC)

DOI (link to publication from Publisher):
10.1109/ICC.2019.8761787

Creative Commons License
Unspecified

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Soret, B., & Smith, D. (2019). Autonomous Routing for LEO Satellite Constellations with Minimum Use of Inter-Plane Links. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* Article 8761787 IEEE (Institute of Electrical and Electronics Engineers). https://doi.org/10.1109/ICC.2019.8761787

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 04, 2025

Autonomous routing for LEO satellite constellations with minimum use of inter-plane links

Beatriz Soret*†, Daniel Smith*
*GomSpace, Aalborg, Denmark
†Aalborg University, Denmark
bsa@es.aau.dk

***Abstract*—A Low Earth Orbit (LEO) mega constellation (MC) is a satellite network of hundreds of spacecrafts organized in several orbital planes, which can be deployed at different inclinations and altitudes. Beyond the connection to the ground segment, the direct inter-satellite communication, both intra-plane and inter-plane, is also used. One of the challenges in LEO MCs is the development of efficient routing strategies that scale up with the size of the network. We propose a simple distributed routing algorithm, *GomHop*, to find the best next-hop for forwarding a packet in a connection-less network. Each satellite runs the on-board algorithm autonomously, with no need for the ground segment to calculate, maintain and distribute routing tables. *GomHop* minimizes the number of required inter-plane hops, which are typically more challenging and expensive for the network, and relies mostly on intra-plane hops. The simulation results show that the overall performance is very close to the centralized optimal solution provided by the Dijkstra algorithm, with *GomHop* having a smaller amount of inter-plane hops and low complexity, in contrast to the $\mathcal{O}(N^2)$ of Dijkstra.**

I. Introduction

Compared to Geo-stationary satellites (GEO), Low Earth Orbits (LEO), typically deployed between 300 and 2000 km, have shorter round trip delays and lower transmission power requirements. The main drawback is that many spacecrafts are needed to provide full and continuous ground coverage. The arrival of the CubeSat platform in the early 2000s has dramatically reduced the price and timescale of a fully functional spacecraft, making LEO Mega Constellations (MC) a very attractive solution for global telecommunications. Indeed, LEO MC shall be a key ingredient of 5G networks, providing a variety of services that include voice, data and IoT [1]. In [2], 3GPP has defined a number of IoT use cases supported by MCs.

A LEO MC is composed of hundreds of spacecrafts plus one or several ground stations, working all together as a communication network. Although a set of coordinated small satellites can have similar functionality as the "big" satellites, there are tight constraints of energy, weight and processing. This poses great challenges to the design, which has to be highly optimized and simple.

The constellation is organized in several orbital planes that can be deployed at different inclinations and altitudes. Within a plane, the satellites are usually evenly spaced. Orbits with a low inclination are called equatorial orbits or near equatorial orbits, and polar orbits are those passing above or nearly above both poles on each revolution (i.e., inclinations close to 90 degrees). Considering regular constellations at the same altitude, there are two classical topologies: the Walker star or polar [3], and the Walker δ or Rosette [4] [5]. This paper addresses generic (regular or irregular) constellations, with no constraints to the topology.

The satellites are connected to ground stations through the Ground-to-Satellite Link (GSL), which is used for Telemetry and Telecommand (TMTC) data and user data. The LEO satellites move at rapid speeds ($>$ 25 000 km/h) relative to the ground terminals, with the GSL available only a few minutes before handover to another satellite occurs. The satellites are connected to each other via the Inter-Satellite Links (ISL). The ISL can be intra-plane ISL, connecting with the satellite in front and the satellite behind in the same plane; and inter-plane ISL, connecting satellites from different orbital planes. Inter-satellite distances are preserved along time within a plane. Between different planes, the inter-satellite distances are time-variant: longest when satellites are over Equator, and shortest over the polar region boundaries. Besides, the orbital periods will be different if the planes are deployed at different altitudes, leading to aperiodic topologies. Overall, RF intra-plane ISL are more stable and easier than inter-plane ISL.

One of the challenges of satellite networks is the development of routing algorithms to compute the paths connecting any pair of nodes. The early literature on routing for LEO constellations focused on connection-oriented networks (see e.g., [6]). After TCP/IP became the prefered method for computer network communication, the implementation of IP in LEO MC has attracted the attention of researchers, with the challenges identified in [7]. Examples of IP routing algorithms are found in [8] - [9]. The problem of the TPC timers is addressed in [10], by maximizing the Round Trip Times under a desired threshold in reference to the TCP timer granularity. In [11], a genetic algorithm uses delay and aging factor in the routing decisions.

Most of the research has focused on regular Iridium-like polar constellations, which greatly simplifies the problem. This is thanks to the imposed constraints to the topology and the links: two fixed intra-plane links to the satellite up and down; and two fixed inter-plane to the satellites in neighbouring planes to either side. The inter-satellite link is usually implemented only between satellites orbiting in the same direction, i.e., avoiding the cross-seam links in counter-rotating planes, which suffer from large Doppler shifts.The motivation of this paper is to

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.

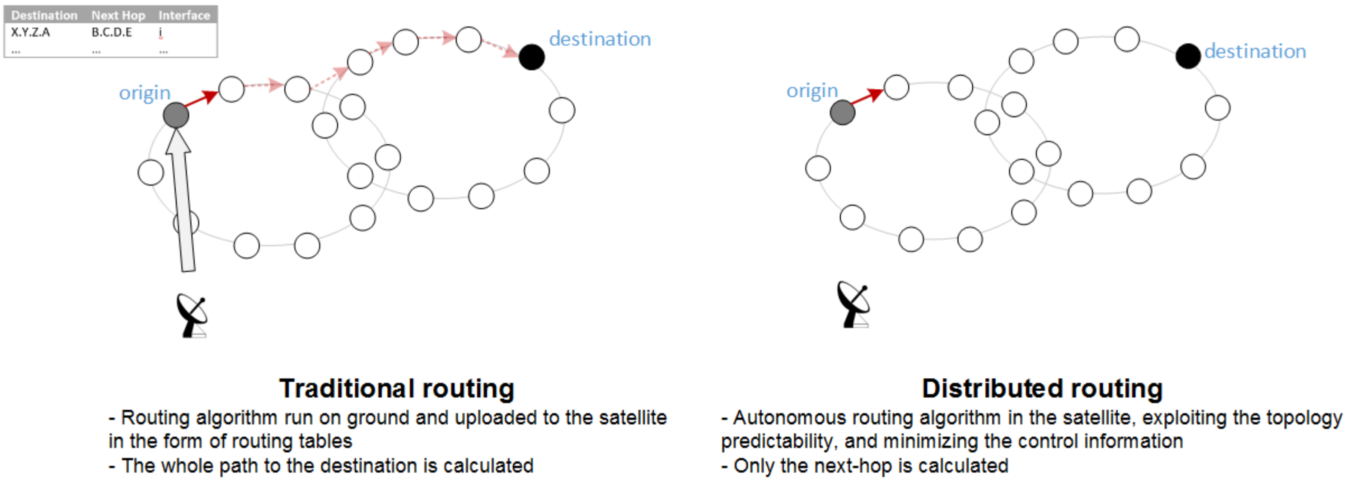

Fig. 1. Traditional centralized routing vs. distributed next-hop routing in satellite constellations.

find a simple and scalable algorithm suitable for a generic MC of small satellites, with limited processing power and ground link availability. As main drivers, the algorithm must be run independently at each spacecraft with minimum amount of control information. On the upside, the predictability and density of LEO networks can be exploited. The rest of the paper is organized as follows. In Section II the two approaches for routing are introduced. Section III details the *GomHop* algorithm. The simulation comparison with the shortest-path Dijkstra solution is presented and discussed in Section IV. Concluding remarks are given in Section V.

II. Routing for Satellite Networks

A routing algorithm is a collaborative process for deciding, in every intermediate node, the directions which must be used to reach the destination. The algorithm can be static or dynamic, depending on whether the decision varies or not with time. In a satellite network, only dynamic solutions are meaningful.

The classical approach to space routing is to centrally compute all the paths in a terrestrial station, and then broadcast the information. Satellites forward the packets according to the on-board routing tables, which are configured based on the ground computations. This centralized approach has the potential of finding the optimal solution. However, its application in MC with a limited number of ground stations is challenging. The routing tables must be frequently updated and sent to the satellites. These updates require the GSL, but its intermittent availability can be insufficient to track the topology changes. Besides, the total bandwidth must be shared with other vital TMTC data and, depending on the mission, also user data. Furthermore, a centralized, optimal approach scales poorly in dense constellations. For instance, Dijkstra's shortest path algorithm, which searches in all directions until it finds the target, has a computational complexity $\mathcal{O}(N^2)$.

Distributed routing refers to having no control center, but all nodes are peers (each node decides its paths autonomously). However, these solutions typically rely on exchanged control information (or metadata) between nodes to refine the decision with e.g., packet acknowledgement. A simpler approach for MC LEO is a hop-by-hop *autonomous* routing, consisting of choosing the next-hop in each satellite from the list of available links, and with the aim of routing the packet in the best direction towards the destination. The end-to-end routing from origin to destination is the result of the network collective behavior. There is a two-fold motivation for this approach in MC LEO: on the one hand, increasing the resilience of the routing, with satellites able to run the routing algorithm independently of the rest of the network; on the other hand, exploiting the predictability of a LEO topology, where nodes do not move randomly, but according to well-known physical laws. The predictability allows to skip the "Hello" messages utilized to discover adjacent nodes in terrestrial ad-hoc networks. This minimize the control information and overheads. Taking local decisions at each satellite facilitates also the management of congestion and failures in the network. A distributed algorithm can be run alone, or used as a backup solution of more traditional routing table-based solutions.

The two approaches are illustrated in Figure 1.

III. GomHop Algorithm

A. Overview

GomHop is a connectionless routing method for LEO satellite constellations which does not require routing tables. It is well-suited for dense MC, where there are usually several paths to connect any pair of satellites. The routing decisions are met on a per-packet basis, i.e., each satellite processes every incoming packet independently. Even though the constellation is moving, the movement is much slower than the time scale of the network layer. We take the typical approach in related literature: fixed length time intervals in which the topology can be regarded as static. For simplicity, the GSL is not included in the description, but it is naturally part of the routing process. Thus, the final destination d is a spacecraft in the formulation of the paper, but in practice it can be both a satellite of the space segment or a ground

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.

station/user terminal. For the latter, there shall be a final leg to download the packet to ground, and a mapping process to get the satellite(s) with ground coverage, i.e., currently covering the area where the ground terminal is located [1].

The algorithm assumes geometry awareness in each satellite, i.e., knowledge of the topology, geometry and dynamics of the network, hence the set of available links at each time instant. This is not a strong assumption: even in small satellites with limited processing capacity, other subsystems in the spacecraft, such as the Attitude and Determination and Control (ADCS) subsystem, shall rely on geometry awareness to e.g., point to neighbour satellites or ground stations at a given time. Therefore, there is no need to provide this information from the ground station. When some links (or full nodes) are not working, the network (started by the ground station) is assumed to broadcast the information to the rest of nodes in the constellation, such that non-functional links are avoided.

B. Algorithm

The constellation consists of N satellites, whose relative or absolute coordinates are known, in P orbital planes. For simplicity, we set the number of satellites per plane to be the same for all planes, $M = \frac{N}{P}$, but it can be easily generalized. There is a surjective correspondence between satellites and orbital planes, with a large number of satellites per plane, i.e., $N >> P$. The orbital plane of a given satellite i is obtained from the geometry information as

$$p_i = \left\lfloor \frac{i-1}{M} \right\rfloor + 1 \qquad (1)$$

The distance between any pair of nodes is computed using relative or absolute coordinates. For example, it can be given in a duplet (lon, lat) indicating the longitude and latitude, respectively, with positive value for longitude east and latitude north, and negative value for longitude west and latitude south. Another option is to consider not only distance but also spacecraft and antenna orientation, introducing the angle between nodes.

The space segment topology is modeled like a dynamic weighted graph $\mathcal{G}(\mathcal{S}, \mathcal{L})$ where the vertices are the reachable satellites $\mathcal{S} = [s_1, ..., s_n]$ and the edges are the available ISL links $\mathcal{L} = [l_1, ..., l_n]$, which can be intra- or inter-plane ISL. Each satellite keeps updated $\mathcal{S}$ and $\mathcal{L}$. When satellite s (in plane p_s) receives a packet with destination satellite d (in plane p_d), the next hop is to be calculated. Instead of treating each satellite individually, the algorithm looks for the best orbital plane, i.e., the one that gets closer to the destination, knowing that in a dense constellation there are several opportunities for jumping between pairs of orbital planes. The goal of reducing the number of required inter-plane ISL hops is achieved by jumping to the target plane as soon as possible, and from that moment the algorithm relies uniquely on intra-plane ISL.

[1]To provide full coverage, the MC is typically designed with coverage overlapping between neighbouring satellites, hence the plural. In case of several satellite covering a given location, the strongest signal is selected.

Algorithm 1 GomHop

Input: satellite ID s
Longer time scale :
Update $\mathcal{G}(\mathcal{S}, \mathcal{L})$
Shorter time scale :
while new packet **do**
  Get the new packet from the queue, with destination d
  if $d == s$ **then**
    {The packet has arrived to the destination}
  else
    {Choose the next hop $l^* \in \mathcal{L}$}
    Calculate the best direction dir to get as close as possible to d in the orbital plane (forward or backward)
    $l^* =$ get_next_hop$(\mathcal{G}, \mathcal{S}, \mathcal{L}, s, d, dir)$
  end if
end while

The *GomHop* routing is sketched in Algorithm 1. It requires updating the available ISL links on the time scale of the constellation movement (e.g., 1-2 minutes), and the mapping to reachable satellites and planes. On a different (shorter) time scale (e.g., 20 ms or the ISL frame duration), the next packet in the queue is picked, with destination d. If the packet has not arrived to the destination, then the next hop is obtained from the set of available links using the following sorted rules (see also Figure 2):

1) If the destination is in the current orbital plane: forward the packet to the next satellite in the plane, following the best direction dir
2) If the target plane is reachable: forward the packet to the available satellite in the target plane. If there are several available satellites in the target plane, choose the one whose distance to the destination is smallest
3) If the current plane and all the reachable planes are different from the target plane: forward the packet to the satellite whose distance to the destination is smallest. This action requires calculating the distances from each reachable satellite to the destination.

Notice that the complexity of *GomHop* is not proportional to N, like Dijkstra's shortest path algorithm. Instead, it depends on the maximum number of reachable satellites, which is much lower than N, and related to the geometry, link budget and RF parameters.

C. Number of ISL hops

The total number of hops, intimately related to the end-to-end latency, is a combination of intra- and inter-plane hops. If $p_s = p_d$, then no inter-plane hops are needed. The total number of hops n_{total} equals the number of intra-plane hops. Said number is expressed as the orbital distance between s and d, denoted by $r_{s,d}$, divided by the number of satellites per plane (assuming evenly distributed satellites within the orbital plane), i.e.,

$$n_{total} = n_{s,d}^{intra} = \frac{r_{s,d}}{N/P} \qquad (2)$$

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.

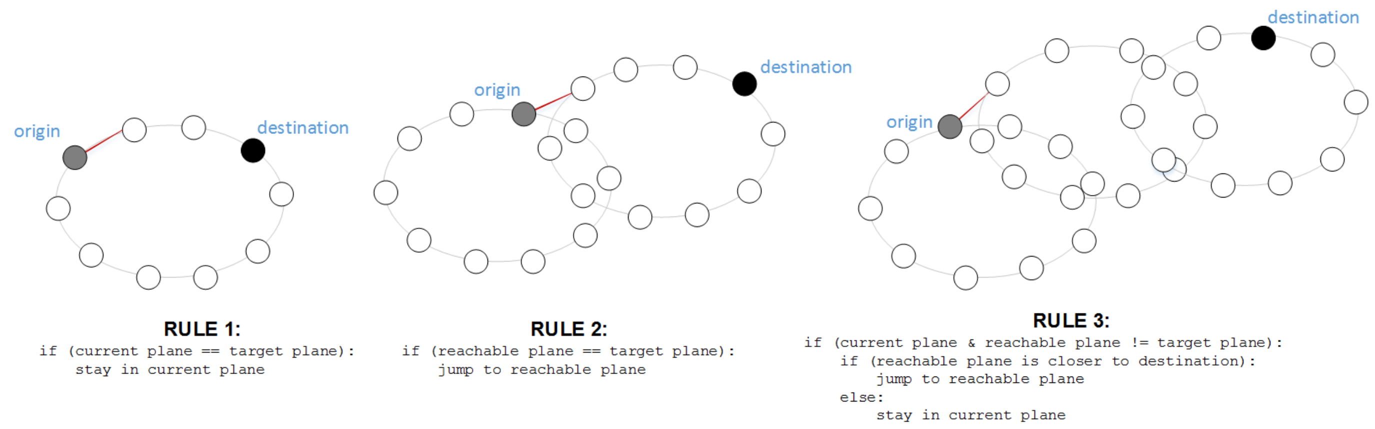

Fig. 2. *GomHop* next-hop calculation. In red, the best next hop for the origin node

When only one inter-plane is used, requiring the destination orbital plane to be reachable for the source orbital plane, then n_{total} is written

$$n_{total} = \underbrace{n_{s,i}^{intra} + n_{j,d}^{intra}}_{\text{intra-plane}} + \underbrace{1}_{\text{inter-plane}} = \frac{r_{s,i} + r_{j,d}}{N/P} + 1 \quad (3)$$

where the closest inter-plane link as seen from s is between satellites i and j.

Finally, if p_d and p_s are not reachable to each other, a number of inter-plane hops and corresponding intra-plane hops are needed, leading to

$$\begin{aligned} n_{total} &= n_{s,i}^{intra} + \sum_{i,j \in \mathcal{P}} \left(1 + n_{i,j}^{intra}\right) + n_{j,d}^{intra} \\ &= \frac{r_{s,i} + r_{j,d}}{N/P} + \sum_{i,j \in \mathcal{P}} \left(1 + \frac{r_{i,j}}{N/P}\right) \end{aligned} \quad (4)$$

where we have used the notation (i, j) to indicate the intermediate planes that are not neither the origin nor the destination.

D. Discussion

We have presented the most basic version of the routing algorithm, where scalability and minimal control information are the main drivers. As it will be shown in the simulations, the performance results are highly satisfactory despite the simplicity of our formulation, and the algorithm does not lead to routing loops. Nevertheless, there are several enhancements to *GomHop* that arise when considering extra complexity and extra metadata. We can split into three tracks:

1) Instead of purely physical distance, the availability of a neighbouring link can be defined in terms of a path cost, which can capture link budget, antenna orientation, capacity, bandwidth, as well as combinations of them.
2) More complex decisions to prioritize preferred routes can be introduced. A performance metric could capture measured parameters such as the throughput, the delay or the bit error rate. Such metric can be compared with the QoS requirements of an individual packet to discard those links that cannot provide the requisites. The rules in Figure 2 can be used by the algorithm as a direction priority, with the aim of using inter-plane as less as possible. The final decision shall then combine the performance metric and the direction priorities.
3) Several hops can be taken into account instead of a single one, at the cost of increased state and feedback information to be exchanged between near nodes. As a main advantage, knowing the available links of neighbouring satellites allows not only for a more founded decision, but also to implement better load balancing strategies in the network.

IV. Simulation Results

The *GomHop* algorithm is simulated and compared to Dijkstra's algorithm. Dijkstra finds the shortest paths between nodes in a graph. Two constellations are evaluated, one irregular and the other one regular. The irregular constellation is equatorial, consisting of 250 satellites organized in 5 orbital planes deployed at altitudes between 700 and 750 km, and covering latitudes between -15° and $+15^\circ$. The regular constellation is the Iridium polar constellation comprising 66 satellites into 6 orbital planes at an altitude of 780 km, and with global coverage. The dynamic topology of the two constellations is obtained with ns-2, for a total time of five Earth rotations. This is used as the input to a dedicated-purpose MATLAB simulator. The simulator calculates the link opportunities among satellites and implements the routing algorithms. When using Dijkstra, the routing tables are previously calculated, which is very time- and memory- consuming task.

The queues in each satellite are of infinite length and use FIFO. The traffic is Poisson, with an arrival rate $\lambda = 100$ and a packet size of 6 kB generated every 20 ms. Each spacecraft is equipped with five modems: two for intra-plane ISL (one forward, one backward), two for inter-plane ISL and one for GSL, the latter not simulated (GSL is ommited). All the modems are full duplex, with separated frequency bands for the uplink and downlink direction. The intra-plane ISL is always available with the two neighour satellites. The

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.

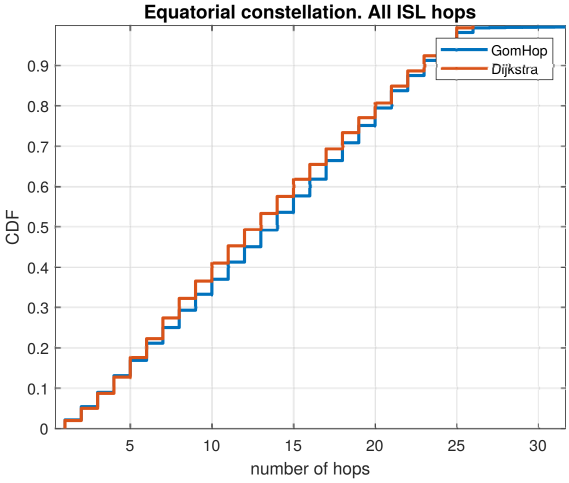

Fig. 3. Simulation results for an equatorial constellation. All ISL hops.

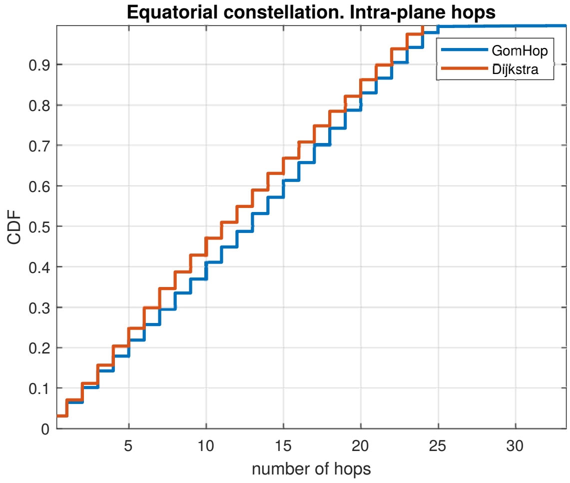

Fig. 5. Simulation results for an equatorial constellation. Intra-plane hops.

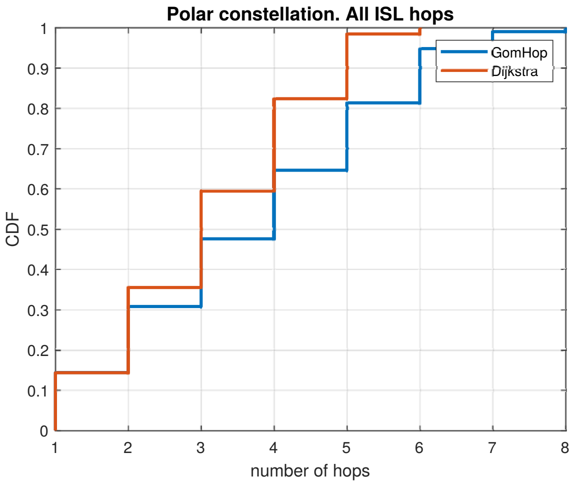

Fig. 4. Simulation results for a polar constellation. All ISL hops.

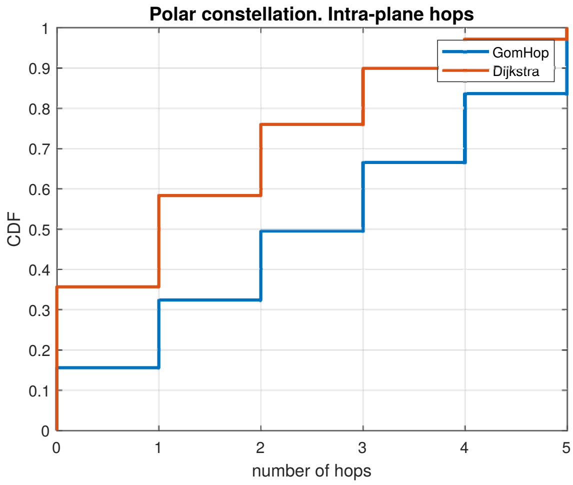

Fig. 6. Simulation results for a polar constellation. Intra-plane hops.

inter-plane ISL communication is possible when the distance between two satellites in different orbital planes is lower than the intra-plane ISL distance, which equals $\sim$ 900 km in the equatorial constellation and $\sim$ 4000 km in the polar constellation. QPSK modulation and constant bit rate are used in the available ISL links.

The simulator has been validated by matching the simulated routes with the expressions for the total number of hops in equations (2)-(4), for each pair of source/destination and the two topologies. The empirical Cumulative Distributed Function (CDF) of the number of ISL hops in the simulation is illustrated in Figures 3-8 for the equatorial and polar topology. Naturally, the number of ISL hops is in general larger in the equatorial example, with 250 satellites, in contrast to the 66 hops of the polar constellation. The maximum number of intra-plane ISL hops with Dijkstra is half the size of the ring, i.e., 25 in the equatorial MC and 5 in the polar constellation. This corresponds to the case when the destination is at the other side of the globe. Looking at the total amount of ISL hops, it is observed that GomHop gets very close to the optimal solution provided by Dijsktra, with a slightly longer tail. The same behaviour is confirmed in the intra-plane ISLs. As desired, *GomHop* minimizes the number of inter-plane hops, which are less used than with Dijkstra.

Finally, the carried load per inter-plane ISL modem and per satellite is plotted in Figure 9, for both constellations. Again, we observe that the use of inter-plane is lower when using *GomHop*, with a median carried load of 240 kbps versus the 335 kbps of Dijkstra in the equatorial constellation, and 275 kbps versus 395 kbps in the polar constellation.

V. Conclusions

The main advantage of the proposed *GomHop* algorithm is its simplicity, which makes it suitable for implementation in small satellites deployed in dense constellations. There is no need for the ground segment to keep and distribute updated routing tables, nor for sending periodic control information among spacecrafts. The principle is to autonomously calculate the best next-hop from the list of available links, with the aim of getting closer to the final destination and minimizing the

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.

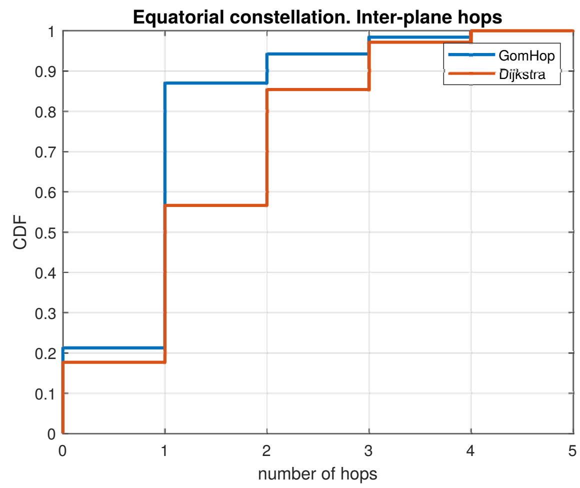

Fig. 7. Simulation results for an equatorial constellation. Inter-plane hops.

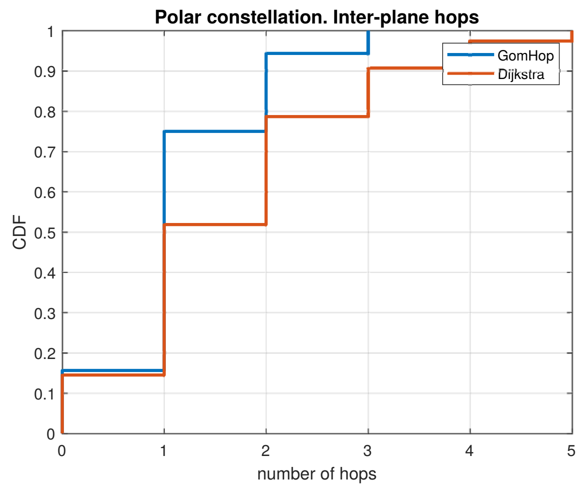

Fig. 8. Simulation results for a polar constellation. Inter-plane hops.

number of inter-plane hops. Despite its simplicity, the simulation results have shown that the performance of *GomHop* is very close to the optimal solution that computes Dijkstra to get the shortest path for all source-destination combinations. Particularly, the global use of inter-satellite links is very close, whereas the inter-plane hops are less used with *GomHop*. This is very interesting for practical implementations, since RF inter-plane hops are much more challenging than intra-plane hops, due to the relative motion between planes and the misalignment.

The algorithm can be broaden to include QoS constraints and heterogeneous traffic. Another interesting generalization is to consider several hops in the decision instead of a single one. The tradeoff among number of hops in the decision, complexity and performance is left for further study. Both of them come at the cost of more control information. These extension could then be compared to state-of-the-art QoS-aware solutions.

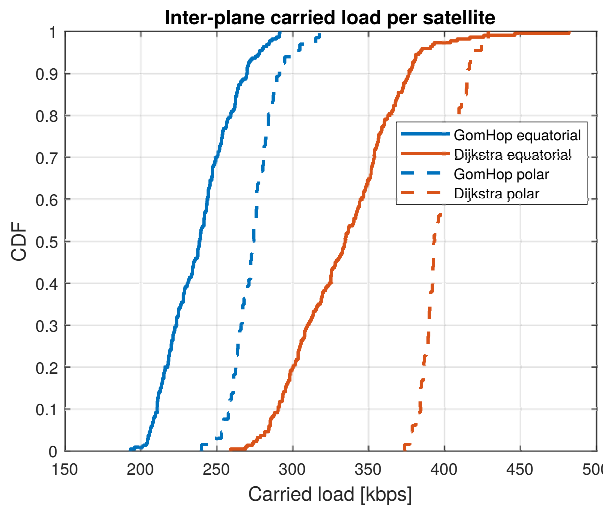

Fig. 9. CDF of the inter-plane carried load per satellite, for equatorial and polar constellations. The packet size is 6 kB and the time slot is 20 ms.

Acknowledgment

This work has been in part supported by the *Danmarks Innovationsfonden* under the Megaman project and the European Research Council (Horizon 2020 ERC Consolidator Grant Nr. 648382 WILLOW). The authors would like to thank Kim Hansen from GomSpace for his valuable input to this work.

References

[1] 3GPP, "Study on Scenarios and Requirements for Next Generation Access Technologies," 3rd Generation Partnership Project (3GPP), Technical Recommendation (TR) 38.913, June 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/38_series/38.913/

[2] ——, "Study on using Satellite Access in 5G; Stage 1," 3rd Generation Partnership Project (3GPP), Technical Recommendation (TR) 22.822, Feb. 2018, version 0.2.0.

[3] J. G. Walker, "Satellite constellations," *Journal of the British Interplanetary Society*, vol. 37, p. 559:571, Nov 1984.

[4] ——, "Circular orbit patterns providing whole earth coverage," *Journal of the British Interplanetary Society*, vol. 24, pp. 369–384, 07 1971.

[5] A. H. Ballard, "Rosette constellations of earth satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-16, no. 5, pp. 656–673, Sept 1980.

[6] M. Werner, C. Delucchi, H. . Vogel, G. Maral, and J. D. Ridder, "Atm-based routing in leo/meo satellite networks with intersatellite links," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 1, pp. 69–82, Jan 1997.

[7] L. Wood, A. Clerget, I. Andrikopoulos, G. Pavlou, and W. Dabbous, "Ip routing issues in satellite constellation networks," *International Journal of Satellite Communications and Networking*, vol. 19, no. 1, pp. 69–92, Feb 2001.

[8] Y. Hu and V. Li, "Logical topology-based routing in leo constellations," in *Proceedings IEEE International Conference on Communications ICC*, vol. 10, 2001, pp. 3172–3176.

[9] J. Wenjuan and Z. Peng, "An improved connection-oriented routing in leo satellite networks," in *2010 WASE International Conference on Information Engineering*, vol. 1, Aug 2010, pp. 296–299.

[10] S. A. M. Makki, N. Pissinou, and P. Daroux, "Leo satellite communication networks: a routing approach," *Wireless Communications and Mobile Computing*, no. 3, pp. 385–395, May 2003.

[11] D. Yiltas and A. H. Zaim, "Routing in low earth orbit satellite systems based on the optimization," in *2008 IEEE Globecom Workshops*, Nov 2008, pp. 1–5.

Authorized licensed use limited to: Aalborg Universitetsbibliotek. Downloaded on August 12,2020 at 11:09:38 UTC from IEEE Xplore. Restrictions apply.