Distributed On-Demand Routing for VLEO Constellations with 3-Terminal Inter-Satellite Links

Distributed On-Demand Routing for VLEO Constellations with 3-Terminal Inter-Satellite Links

Nicolò Benso*, Alessandro Compagnoni*, Gregory F. Stock†, Juan A. Fraire‡§, Gabriel Maiolini Capez*, Camilla Ottaviani*, Daniel Gaetano Riviello¶, Giacomo Verardo‖, Mirca Gargiulo**, Leonardo Ospizio**, Carla Fabiana Chiasserini*, Roberto Garello*

*Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy

†Saarland University, Saarland Informatics Campus, 66123 Saarbrücken, Germany

‡Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France

§CONICET – Universidad Nacional de Córdoba, Córdoba, Argentina

¶CNR-IEIIT, Consiglio Nazionale delle Ricerche, 10129 Turin, Italy

‖KTH Royal Institute of Technology, Stockholm, Sweden

**Thales Alenia Space Italia, 00131 Rome, Italy

***Abstract*—This paper investigates routing strategies for Very Low Earth Orbit (VLEO) satellite constellations constrained by a three-terminal (3T) inter-satellite architecture, which is gaining interest as currently adopted by major operators (e.g., Starlink). In the 3T configuration, two terminals enable along-track communication, while a third hemispherical terminal can dynamically connect to either the left or right cross-plane neighbor, but not both simultaneously. This contrasts with the traditional four-terminal (4T) architecture, where four directional terminals enable full inter-plane links. The 3T constraint reduces cross-plane connectivity, impacting path diversity, reachability, and latency. To address these challenges, we develop *Interleaved-Scenario MinHopCount (IS-MHC)* and *DisCo3T*, modified versions of the state-of-the-art routing algorithms *Minimum Hop Count (MinHopCount)* and *Distributed On-Demand Routing for Mega-Constellations (DisCoRoute)* for the 3T case. Our new model accurately captures the limitations of 3T connectivity and reduces hop count in this scenario. We conduct a comprehensive evaluation of *DisCo3T* in terms of latency, hop count, and computational complexity, including a comparative analysis with the original 4T implementation. Results show that the topological constraints of the 3T configuration cause an average hop count increase of 21.7% compared to the 4T baseline; however, *DisCo3T* retains the processing efficiency of the original 4T algorithm.**

***Index Terms*—Mega-Constellations, VLEO satellites, Satellite Networks, Routing, Performance Evaluation**

I. Introduction

The increasing interest in Very Low Earth Orbit (VLEO) constellations is driven by their potential to provide high-throughput, low-latency, and resilient Non-Terrestrial Network (NTN) connectivity, while also supporting high-resolution Earth observation missions [1], [2]. With altitudes ranging between 300 and 500 km, VLEO satellites offer significantly reduced propagation delays and improved signal strength. However, they also face technical challenges such as more substantial atmospheric drag, shorter satellite lifespans, higher Doppler, shorter visibility windows, and the need for denser and more dynamic constellations to maintain global coverage [3], [4].

Routing for VLEO constellations is particularly challenging due to the rapid orbital motion of satellites, which causes frequent and unpredictable topology changes that compromise reliable and timely data delivery. To address these challenges, an efficient heuristic called *DisCoRoute* [5] has recently been proposed and has shown excellent performance under highly dynamic VLEO conditions [6]. The algorithm intelligently selects the hop sequence by leveraging two key geometric characteristics of Walker-Delta topologies: the fact that all intra-plane hops have a constant and time-invariant length, and that inter-plane hops are generally shorter near the poles and longer near the Equator. *DisCoRoute* was developed for satellite architectures equipped with four terminals, a design choice that could, in principle, be adopted by many VLEO and Low Earth Orbit (LEO) constellations.

However, the number of terminals per satellite is critical in determining the network's efficiency and design complexity [7]. For this reason, an interesting architectural alternative for upcoming VLEO/LEO constellations is to consider satellites equipped with a three-terminal (3T) architecture, as currently implemented in Starlink's constellations [8]: two terminals may support along-track communication, while a third hemispherical terminal enables switching between left and right cross-plane neighbors, albeit not simultaneously [9].

One of the main challenges of the 3T configuration lies in the limited cross-plane availability; hence, traditional routing approaches, initially designed for more stable or fully connected mesh networks, may perform suboptimally or even fail under the 3T configuration.

To the best of our knowledge, the only study that addresses routing within the 3T constraint is by Chen, Yang, Zhao, *et al.* [10], which presents a detailed theoretical analysis of how this configuration affects hop count, user delay, network cost, and capacity compared to the four-terminal (4T) case.

Nevertheless, a performance evaluation of in-space routing algorithms regarding end-to-end delay and execution time under 3T constraints is still lacking.

In this work, we develop a new routing scheme by modifying *DisCoRoute* so that it operates on 3T VLEO constellations. In doing so, we introduce a novel analytical model that ensures reachability and minimizes hop count in 3T configurations.

The proposed routing schemes are evaluated under realistic VLEO constellations in terms of key performance indicators, including latency, hop count, and computational complexity.

The remainder of the paper is organized as follows. Section II introduces the system and topology models. Sections III-A and III-B detail the proposed *DisCo3T* approach. Evaluation results are presented in Section IV, and conclusions are drawn in Section V.

II. System Model

This section introduces the adopted orbital configuration, based on the Walker-Delta constellation model, and outlines the decentralized routing algorithms that form the basis of our *DisCo3T* design.

A. Walker-Delta Constellations

Walker-Delta constellations [11] – also known as *Ballard-Rosette* configurations [12] – are characterized by circular orbits arranged in a flower-like pattern, evenly spaced along the Equator. These constellations are described using the notation $\alpha : T/P/F$, where:

- α denotes the inclination of each orbital plane relative to the Earth's equatorial plane,
- T is the total number of satellites,
- P is the number of orbital planes, and
- F is the phasing factor, which defines the relative angular displacement of satellites in adjacent planes.

All orbits in a Walker-Delta configuration have the same altitude and inclination and are circular. Each plane contains $Q = T/P$ satellites, uniformly distributed along the orbit. The phasing factor F controls the angular offset between satellites in adjacent planes. If $F = 0$, satellites in different planes are aligned; for $F \neq 0$, satellites in successive planes progressively shift forward or backward along the orbit.

Each satellite can be uniquely identified by a pair of indices (o, i), where $o \in \{0, \ldots, P-1\}$ is the index of the orbital plane, and $i \in \{0, \ldots, Q-1\}$ is the satellite's position within that plane. Alternatively, satellites can also be referred to using a global ID in the range 1 to T.

In terms of connectivity, satellite (o, i) typically maintains four inter-satellite links (ISLs):

- Two *intra-plane* links, connecting to the previous and next satellites on the same orbit: $(o, (i-1) \bmod Q)$ and $(o, (i+1) \bmod Q)$, respectively.
- Two *inter-plane* links, connecting to satellites in adjacent planes. The left neighbor is $(o-1, i)$ if $o > 0$, or $(P-1, (i-F) \bmod Q)$ if $o = 0$. The right neighbor is $(o+1, i)$ if $o < P-1$, or $(0, (i+F) \bmod Q)$ if $o = P-1$.

In the literature, links to satellites in the same orbital plane are referred to as *vertical hops* or *intra-plane links*. In contrast, links to satellites in adjacent planes are referred to as *horizontal hops* or *inter-plane links*.

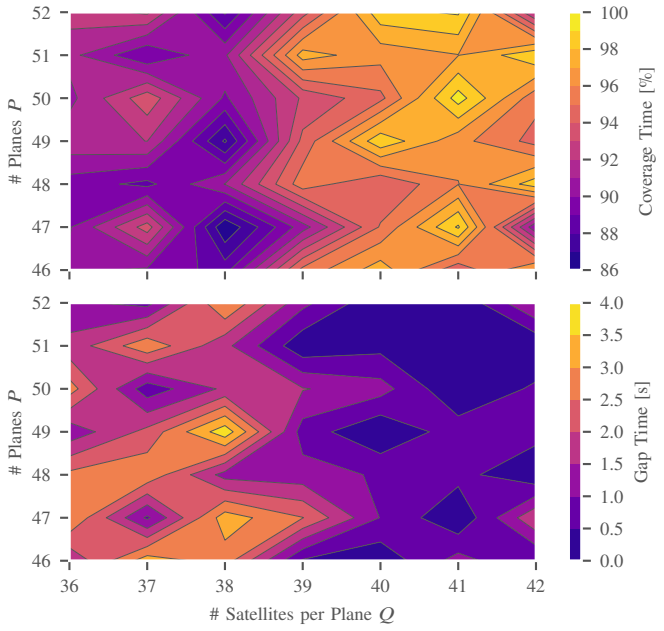

Fig. 1. Coverage time percentage and gap time vs. number of orbital planes and satellites per plane in the constellation.

B. Constellation Design Methodology

Thales Alenia Space Italia designed the reference Walker-Delta constellation in this work [13] to ensure continuous coverage and uninterrupted service over the target area, while optimizing orbital parameters based on both coverage constraints and the satellite's onboard architecture. The final configuration adopted is a Walker-Delta 65° : 2080/52/10. The satellites operate in circular orbits at a nominal altitude of approx. 300 km, typical for VLEO constellations. A key design constraint was the total antenna Field of View (FoV), which played a central role in the coverage analysis and the definition of orbital density required to guarantee service continuity.

The orbital configuration was optimized through a two-step parametric study, with results evaluated using two primary performance metrics: the *coverage time percentage* (i.e., the fraction of time a target point within the service area is covered by at least one satellite) and the *gap time* (i.e., the maximum duration during which a target point remains uncovered). In the first phase, these metrics were analyzed as functions of the number of satellites and orbital planes, allowing the identification of configurations with near-optimal performance.

a) Phase 1: Results are shown in Fig. 1. In the coverage contour plot, the bright yellow region represents configurations with a coverage time percentage exceeding 99% over the service area. Conversely, in the gap contour plot, the dark blue region highlights configurations where the maximum service gap duration is below 0.5 seconds.

b) Phase 2: Subsequently, the inter-plane phase increment was optimized to maximize coverage uniformity, thereby enabling continuous service area coverage in full compliance with the mission requirements. Fig. 2 shows the result of this

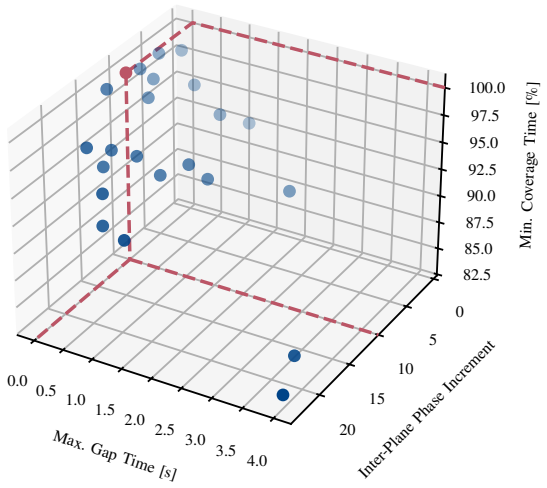

Fig. 2. Inter-plane phase increment optimization.

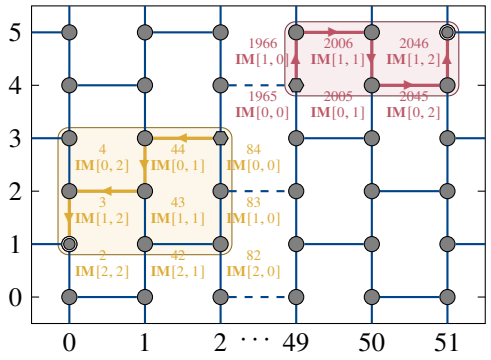

Fig. 3. Extract of a 3T ISL topology for 65°: 2080/52/10 with interleaved inter-plane links. The yellow path is an example of the case $H_v \geq H_h$, where the hop counts for 4T and 3T coincide. In contrast, the red path shows that two additional hops are needed in 3T. The two source nodes are marked as ⬣ while the destination nodes are shown as ◉. In addition, the annotations show the global node IDs and how nodes are assigned to the **IM** matrix as described in Section III-A.

optimization process, highlighting the phasing factor ($F = 10$) that led to the optimal constellation configuration, achieving 100% coverage time with no service gaps.

The analysis was carried out using the System Tool Kit (STK) software. The final selected configuration guarantees a regular distribution of satellites along the orbital planes and a high spatial density, both essential for achieving uninterrupted coverage across the entire area of interest. The resulting orbital mesh ensures that each satellite maintains available connections both within its own orbital plane and with an adjacent plane at any given time, following the interleaved topology imposed by the 3T architecture (see Section II-D).

Despite the architectural constraint of having only one cross-plane link per satellite, the constellation's spatial organization and phasing ensure full reachability among nodes. This enables the implementation of efficient intra-plane and inter-plane routing strategies even under conditions of partial connectivity.

C. Routing Algorithms Based on Minimum Hop Count

The routing algorithms we analyze utilize the output of the *Minimum Hop Count (MinHopCount)* [14], which mathematically determines the minimum number of ISL hops required to connect any two satellites in the constellation. In addition, it identifies the direction (Northeast (↗), Northwest (↖), Southeast (↘), or Southwest (↙)) along which this minimum-hop path can be found. *MinHopCount*-based routing algorithms take this result as input and restrict their search space to a spherical, rectangular grid, with the source and destination positioned at opposite corners. As a result, any path confined within this grid has the minimum number of hops. Although the resulting route may not yield the shortest distance or propagation delay, it ensures the lowest hop count—often a preferred criterion, as each hop introduces additional delay due to onboard processing and queuing.

In this paper, we focus on and adapt the *DisCoRoute* algorithm [5], [6] that employs a heuristic to intelligently select the hop sequence based on two geometric characteristics of Walker-Delta constellations: (i) all intra-plane hops are equal in length and remain constant over time, and (ii) inter-plane hops are shortest near the poles and longest around the Equator. As a result, *DisCoRoute* aims to position all horizontal hops as close to the poles as possible – that is, at high latitudes in terms of absolute value.

D. ISL Terminal Configuration

In order to provide a fully reachable ISL topology with the 3T constraint, we connect satellite (o, i) to either the right or left neighbor based on its position in the constellation. A satellite with an even position $i = 2m$ is connected either to its neighbor on the right if it lies on an even-numbered orbital plane $o = 2k$, or to its neighbor on the left if it lies on an odd-numbered plane $o = 2k + 1$ where $k, m \in \mathbb{N}$. The opposite applies for satellites with an odd position $i = 2m + 1$, where "right" and "left" are exchanged. We call this approach an *interleaved topology*. A representative section of the final topology is shown in Fig. 3.

III. Proposed Solution

The *DisCoRoute* algorithm has been specifically designed for a 4T configuration. As demonstrated in [5], [6], it achieves over 98% reduction in computational time compared to Dijkstra-based algorithms while maintaining almost the same end-to-end latency (−2% in the worst case). These results highlight *DisCoRoute*'s suitability for onboard execution in dynamic LEO and VLEO constellations, where routing efficiency and scalability are critical.

This paper adapts *DisCoRoute* to operate within a 3T architecture, where two terminals are dedicated to along-track communication, while a third hemispherical terminal can dynamically switch to either the left or right cross-plane neighbor, though not both at once. In the 4T configuration scenario, the algorithm assumes full horizontal connectivity, which is no longer guaranteed in the new setup due to the reduced number of terminals. Furthermore, the original *MinHopCount* model

becomes unreliable in 3T settings, particularly when the number of required horizontal hops exceeds the available vertical hops, leading to infeasible or suboptimal paths.

To address these limitations, we propose the *Interleaved-Scenario MinHopCount (IS-MHC)* (Section III-A) and *DisCo3T* (Section III-B) algorithms, which are adaptations of *MinHopCount* and *DisCoRoute* for the 3T scenario, respectively.

A. Interleaved-Scenario MinHopCount

Assuming no initial link failures, the interleaved topology inherently constrains minimum-hop paths to typically follow a staircase-like pattern.

Given the constant availability of vertical hops, it is evident that the horizontal hops are the determining factor. Let H_h and H_v denote the minimum number of horizontal and vertical hops, respectively, required to connect the source and destination satellites in the 4T scenario [5], following a given direction (i.e., one of $\{\nearrow, \nwarrow, \searrow, \swarrow\}$). Similarly, let H_h^* and H_v^* denote the corresponding values defined for the 3T scenario.

Trivially, $H_h^* = H_h$ always holds in the interleaved scenario. We observe that when $H_v \geq H_h$, then $H_v^* = H_v$, as it is always possible to construct a "staircase" path connecting the source and destination planes between the satellites' latitudes (see, for example, the yellow path in Fig. 3). In all other cases, the number of hops in 3T increases. Specifically, unlike in the 4T scenario, each missing horizontal hop between the source and destination in 3T adds two additional vertical hops to the original value of H_v (e.g., the red path in Fig. 3).

To properly compute the number of hops in 3T, we consider the search space as the rectangular grid determined by the *MinHopCount* model, with the source and destination satellites at opposite corners of this space.

Specifically, we can extract two matrices from this search space: (i) the *index map matrix* $\mathbf{IM} \in \{1, \ldots, T\}^{m \times (n+1)}$ collecting all (global) satellite IDs in the search space, and (ii) the *inter-plane link adjacency matrix* $\mathbf{H} \in \{0,1\}^{m \times n}$, where $m \triangleq H_v + 1$ and $n \triangleq H_h$.

The $\mathbf{IM}$ matrix is defined such that $\mathbf{IM}[0,0]$ corresponds to the source satellite, while a generic satellite in the search space is indicated by $\mathbf{IM}[i^*, o^*]$, where o^* denotes the offset in orbital planes relative to the source's orbital plane, and i^* represents the offset in position within the plane relative to the source's position, with $i^* \in \{0, \ldots, m-1\}$ and $o^* \in \{0, \ldots, n-1\}$.

The inter-plane links in the search space are represented by the adjacency matrix $\mathbf{H}$ as follows: $\mathbf{H}[i^*, o^*] = 1$ indicates the presence of a horizontal link between satellites $\mathbf{IM}[i^*, o^*]$ and $\mathbf{IM}[i^*, o^* + 1]$, while $\mathbf{H}[i^*, o^*] = 0$ denotes its absence.

For example, the index map $\mathbf{IM}$ and the adjacency matrix $\mathbf{H}$ for the red search space with source $(o, i) = (49, 4)$, with ID = 1965, and destination $(o, i) = (51, 5)$, with ID = 2046, in Fig. 3 are given by $\mathbf{IM} = \left(\begin{smallmatrix} 1965 & 2005 & 2045 \\ 1966 & 2006 & 2046 \end{smallmatrix}\right)$ and $\mathbf{H} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, where the top-left entry of $\mathbf{H}$ is 0 since there is no inter-plane link from $(49, 4)$ to $(50, 4)$, or, equivalently, between satellites 1965 and 2005.

Starting from this representation, it is possible to develop a model that estimates the number of vertical hops H_v^* necessary to complete the route from source to destination.

Given the $m \times n$ matrix $\mathbf{H}$, we distinguish two cases:

- $m > n$: $H_v^* = H_v$
- $m \leq n$:

$$H_v^* = \begin{cases} H_v + 2(k+1) & \text{if } n - m = 2k + 1 \\ H_v + 2k & \text{if } n - m = 2k \wedge tr(\mathbf{M}) = m \\ H_v + 2(k+1) & \text{if } n - m = 2k \wedge tr(\mathbf{M}) \neq m \end{cases}$$

  where $k \in \mathbb{N}$, $\mathbf{M}$ is an $m \times m$ sub-matrix of $\mathbf{H}$ defined starting from the source position, and $tr(\mathbf{M})$ denotes the trace of the matrix $\mathbf{M}$.

We can now apply the proposed formulation to the solutions found by *MinHopCount* in 4T ($H_v^{\nwarrow}, H_v^{\swarrow}, H_v^{\nearrow}, H_v^{\searrow}$) to compute the corresponding directional intra-plane hop counts in 3T, while the inter-plane hop counts are directly given as $H_h^{*\leftarrow} = H_h^{\leftarrow}$ and $H_h^{*\rightarrow} = H_h^{\rightarrow}$. The (overall) minimum hop count is determined by identifying the smallest value among all possible combinations:

$$\min \left\{ H_h^{*\leftarrow} + H_v^{*\nwarrow},\ H_h^{*\leftarrow} + H_v^{*\swarrow},\ H_h^{*\rightarrow} + H_v^{*\nearrow},\ H_h^{*\rightarrow} + H_v^{*\searrow} \right\}$$

It is important to note that *IS-MHC* also includes indicators of the direction of travel needed to achieve the minimum number of hops. In the next section, we exploit this model to define the *DisCo3T* algorithm.

B. DisCo3T Routing Algorithm

In this section, we present the proposed *DisCo3T* algorithm and evaluate its performance against the original 4T *DisCoRoute*.

The development of the *DisCo3T* algorithm is strongly based on what was done for *DisCoRoute* in 4T. However, due to the 3T topology and the constraints it places on the construction of paths between two satellites, in the version we propose, it is no longer necessary to distinguish between the two cases of ascending-to-ascending (or descending-to-descending) and ascending-to-descending (or descending-to-ascending), where ascending or descending refers to the propagation direction of a satellite along its plane of belonging, and two different algorithms were used depending on the case, as described in [5]. The basic idea of *DisCoRoute* of using horizontal hops as close to the poles as possible to minimize the distance between satellites of adjacent planes is still maintained, as it is still advantageous in terms of end-to-end delay. In particular, the route construction is done in parallel both from the source ($route_s$) and from the destination ($route_t$), as long as the number of hops used is less than the number of hops needed to connect the two satellites calculated by *IS-MHC*, and giving precedence to horizontal hops, being a scarce resource. The two routes will develop towards each other, in a "staircase" fashion, until the number of horizontal hops is exhausted, and, if the construction is done correctly, the two routes can finally be joined using the remaining vertical hops, as we will see later. This method allows to keep the route construction as

Algorithm 1 DisCo3T

1: **procedure** DisCo3T(*Const.*, *src*, *dst*, *adjM*) ▷ *adjM = adjacency matrix*
2: $H_h^*, H_v^*, H_h, H_v, SearchDir \leftarrow$ IS-MHC(*Const.*, *src*, *dst*, *adjM*)
3: $\mathbf{IM}, j, l, p, q \leftarrow$ ESS(*Const.*, *src*, *dst*, *adjM*, H_h, H_v, *SearchDir*)
4: **while** $H_h^* > 0$ **do**
5: Get $\mathbf{IM}[j, l]$'s horizontal successor $\mathbf{IM}[j, l+1]$
6: Get $\mathbf{IM}[p, q]$'s horizontal successor $\mathbf{IM}[p, q-1]$
7: $Cond_s = adjM[\mathbf{IM}[j, l], \mathbf{IM}[j, l+1]]$ ▷ *1 iff. horiz. hop available*
8: $Cond_t = adjM[\mathbf{IM}[p, q], \mathbf{IM}[p, q-1]]$
9: **if** $Cond_s \wedge Cond_t$ **then** ▷ *true if both horizontal hops available*
10: Compute $reward_s$, $reward_t$ (based on latitude)
11: Favor direction with higher latitude (closer to poles)
12: Update corresponding route and index
13: $H_h^* \leftarrow H_h^* - 1$
14: **else if** $Cond_s \vee Cond_t$ **then** ▷ *true if one horizontal hop available*
15: Take the available horizontal hop
16: Update route and index
17: $H_h^* \leftarrow H_h^* - 1$
18: **else** ▷ *Enter this case when no horizontal hops are available*
19: Get $\mathbf{IM}[j, l]$'s vertical successor $\mathbf{IM}[j+1, l]$
20: Get $\mathbf{IM}[p, q]$'s vertical successor $\mathbf{IM}[p-1, q]$
21: Compute vertical hop rewards for both directions
22: Take hop with higher latitude reward
23: Update route and index
24: $H_v^* \leftarrow H_v^* - 1$
25: **if** $H_v^* > 1$ **then**
26: Append remaining vertical path to $route_s$
27: **return** $route_s + route_t$

Algorithm 2 ExpandSearchSpace

1: **procedure** ESS(*Const.*, *src*, *dst*, *adjM*, H_h, H_v, *SearchDir*)
2: Compute m, n based on H_v, H_h
3: **if** $m \leq n$ **then**
4: **if** $(n-m)$ is odd **then**
5: $expValue \leftarrow \frac{n-m-1}{2} + 2$
6: Retrieve satellite *srcExp* in position $(o, i \pm expValue)$
7: **else** ▷ $n-m$ *is even*
8: $expValue \leftarrow \frac{n-m}{2} + 1$
9: Retrieve satellite *srcExp* in position $(o, i \pm expValue)$
10: $H_v \leftarrow H_v + 2 \cdot expValue$
11: $\mathbf{IM} \leftarrow$ ConstructIM(*Const.*, *srcExp*, *adjM*, H_h, H_v, *SearchDir*)
12: $m^* = size(\mathbf{IM}, rows)$ ▷ *Retrieve the number of rows*
13: $n^* = size(\mathbf{IM}, cols)$ ▷ *Retrieve the number of columns*
14: Initialize $route_s = [src]$, $route_t = [dst]$
15: Set $j = 0 + expValue$, $l = 0 + expValue$
16: Set $p = m^* - 1 - expValue$, $q = n^* - 1 - expValue$
17: **else** ▷ *No expansion needed*
18: $\mathbf{IM} \leftarrow$ ConstructIM(*Const.*, *src*, *adjM*, H_h, H_v, *SearchDir*)
19: Initialize $route_s = [src]$, $route_t = [dst]$
20: Set $j = 0$, $l = 0$, $p = m-1$, $n-1$
21: **return** $\mathbf{IM}$, j, l, p, q

close to the poles as possible, with the consequent advantages mentioned above.

As shown in Algorithm 1, we obtain the number of horizontal and vertical hops for both 3T and 4T scenarios, and the travel direction required to achieve the minimum number of hops using the *IS-MHC*. Then we assign the search space dimensions m and n, and construct the index map $\mathbf{IM}$ (line 3) using ExpandSearchSpace (Algorithm 2), whose procedure can be split into two cases as follows:

1) $m \leq n$: the $\mathbf{IM}$ matrix is expanded according to Algorithm 2, to allow the algorithm to search for more convenient paths also outside the rectangle defined by *MinHopCount*. To avoid overloading the memory by over-expanding the matrix, i.e., adding more rows than necessary, we calculated the maximum value of vertical hops that a path can make before violating the *IS-MHC*. Following what was done in the *IS-MHC* formulation, this depends on the difference between m and n, and whether it is even or odd. Depending on the case, we calculate the maximum value for which it is needed to expand the index map (lines 5 and 8 of Algorithm 2).

Since one cannot know a priori the most advantageous direction of expansion, one has to expand the matrix both above the source and below the destination by *expValue* rows, resulting in a matrix with a number of rows equal to $2 \cdot expValue + H_v + 1$. To expand the matrix above the source, we move *expValue* satellites vertically from the source out of the search space. By doing so, we get the *srcExp*, the satellite that should be positioned at the top left corner of the index map, and from which the matrix is constructed.

Note that (i) *srcExp* and *src* do not coincide, and (ii) the expansion is applied only in the vertical direction – that is, by adding $2 \cdot expValue$ rows. This choice is due to the fact that, in 3T, the additional hops compared to the 4T case occur exclusively in the vertical direction. Expanding the search space horizontally would violate the *IS-MHC*, since the minimum number of horizontal hops required to complete a route between two satellites remains unchanged between 3T and 4T, as discussed in Section III-A. We then initialize the matrix indexes and the routes from the source node $(j, l, route_s)$ and the destination node $(p, q, route_t)$, taking into account the offsets given by the *expValue*.

2) $m > n$: We construct the two matrices without any expansion and initialize the indexes and the routes from the source to the destination. Indeed, in this case, expanding the search space is unnecessary, as the most advantageous path is already included therein.

Subsequently, we construct the routes following the procedure described next. For starters, we obtain the horizontal successors of the last node visited from the source moving toward the destination, $\mathbf{IM}[j, l+1]$, and of the last node visited from the destination moving toward the source, $\mathbf{IM}[p, q-1]$. Then, we compute two boolean conditions, $Cond_s$ and $Cond_t$. These take on the values true (1) or false (0) depending on the presence or absence of a horizontal hop between satellites $\mathbf{IM}[j, l]$ and $\mathbf{IM}[j, l+1]$, and satellites $\mathbf{IM}[p, q]$ and $\mathbf{IM}[p, q-1]$. The values of the conditions are assigned from the adjacency matrix *adjM*, a square matrix indicating whether pairs of satellites in the whole constellation are adjacent. If both cross-links are available, i.e., $Cond_s = 1$ and $Cond_t = 1$, we apply the algorithm shown from line 10 to line 12 in Algorithm 1. The procedure computes two rewards, one for each direction, based on the sum of absolute value of the latitudes of the satellites $\mathbf{IM}[j, l]$ and $\mathbf{IM}[j, l+1]$, and of the satellites $\mathbf{IM}[p, q]$ and $\mathbf{IM}[p, q-1]$. If $reward_t$ exceeds $reward_s$, this indicates that the nodes along the route from the destination are located at higher latitudes compared to those from the source. Consequently, they are situated closer to the poles and therefore represent the more advantageous path between the two. The successor of (p, q) is then appended to $route_t$. The same approach is

applied in the opposite case, when $reward_s$ is greater than or equal to $reward_t$, appending the successor of satellite $\mathbf{IM}[i,l]$ to $route_s$.

Otherwise, the sole horizontal hop available at the source or the destination is appended. If no horizontal links are available, it advances with a vertical link, implementing a procedure identical to the horizontal hops case described above, choosing the path with the highest sum in absolute value of latitudes. Every time a new satellite is appended to $route_s$ or $route_t$, the number of horizontal hops H_h^* or the number of vertical hops H_v^* is decreased by one.

This process is repeated for H_h^* times. The route from the source node is then completed by adding the remaining vertical hops required to connect it to $route_t$'s last visited node. Finally, the route from the destination node is appended to the route originating from the source node, and the total path is returned.

IV. Numerical Results

We implemented the considered routing algorithms in MATLAB R2024b. The analysis is performed using the default orbit propagator provided by the Aerospace Toolbox for Satellite Mission Analysis and the Satellite Communication Toolbox. We applied a proximity-based approach within the Walker-Delta constellation pattern to determine which ISLs to establish with nearby satellites. We ran the experiments on an HP Z2 Tower G9 Workstation, with an Intel Core i9-13900 processor and 32 GB of RAM.

We collect the performance statistics of *DisCo3T* and *DisCoRoute* over 100 000 randomly selected pairs of source-destination satellites on the 65°: 2080/52/10 constellation designed by Thales Alenia Space Italia [13].

Latency. Regarding latency, the results show a significant increase when using the 3T topology: on average, there is a 34.14% increase compared to the 4T configuration. This provides an intuition on the performance cost associated with transitioning from a 4T configuration to one with 3T, which is due to the reduced path diversity and the 3T, hence routing, constraints. It therefore underscores the importance of trading off between architectural simplicity and performance in constellation design. The results are visualized in Fig. 4.

Number of Hop Count. The impact of the 3T topology is again very noticeable in this context. We observe an average increase of 21.7% in the number of hops with respect to *DisCoRoute* in the 4T case. Additional experiments highlighted that the interleaved topology ensures that the path with the shortest total length is the one with the minimum number of hops. The results confirm the structural impact of the 3T configuration on path length, and reflect the constrained path diversity resulting from the reduced number of ISLs per satellite, which forces routes to follow longer, stair-like paths across the interleaved mesh. In contrast, the 4T case exhibits a visibly tighter distribution, centered on lower hop counts, with an average hop count of 23 hops, against the average of 28 hops displayed by the 3T case, which showcases the benefit of additional ISL degrees of freedom. Again, these findings underscore a key trade-off in constellation design:

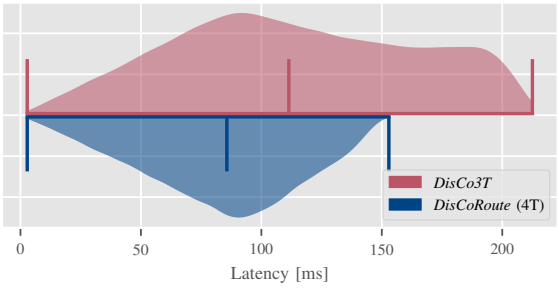

Fig. 4. Latency distribution: *DisCo3T* vs. *DisCoRoute* (4T).

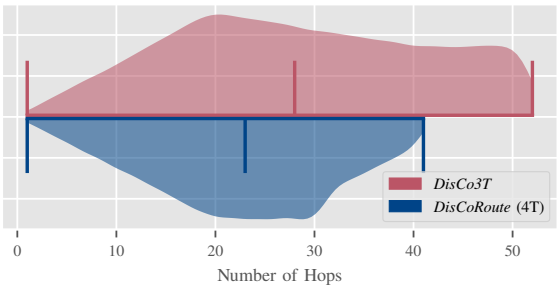

Fig. 5. Hop Count distribution: *DisCo3T* vs. *DisCoRoute* (4T).

reducing network cost and onboard complexity (by using fewer terminals) comes at the expense of higher average hop counts, with implications for end-to-end delay, energy consumption, and routing table size. Hop count results are reported in Fig. 5.

Computational Time. *DisCo3T*, compared to the version in the 4T scenario of *DisCoRoute*, incurs higher computational costs, as the execution time has more than doubled, due to the higher complexity of the *IS-MHC* model, compared to that of the original *MinHopCount*. However, the processing time is still very competitive, confirming the strong point of this approach even in the 3T scenario. Results on computational time are reported in Fig. 6.

Takeaways. The difference in performance observed in the 3T configuration – specifically, a 34.14% increase in latency and a 21.7% increase in hop count – is a direct consequence of the reduced cross-plane connectivity imposed by the topology, not a limitation of the routing algorithm itself. While optimal paths

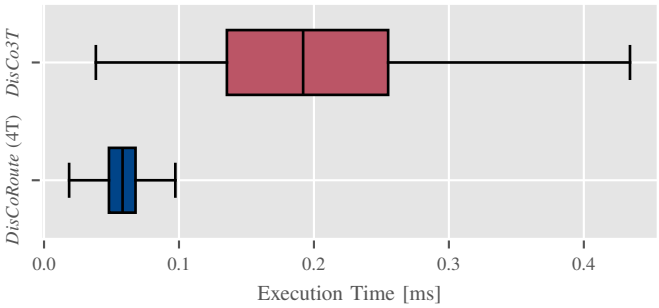

Fig. 6. Computational time bar plot: *DisCo3T* vs. *DisCoRoute* (4T).

in 4T scenarios are naturally shorter due to full inter-plane availability, our proposed *DisCo3T* successfully adapts the *DisCoRoute* heuristic to operate under the structural constraints of 3T constellations.

These challenges – such as discontinuities in wrap-around connectivity and longer staircase paths – are mitigated through a static interleaved topology and the *IS-MHC* model, ensuring full network reachability. Importantly, *DisCo3T* achieves acceptable performance under the 3T constraints, while maintaining substantially lower computational complexity. We also remark that our work is, to the best of our knowledge, the first practical and efficient routing solution tailored for 3T architectures, offering a lightweight alternative suitable for onboard execution.

V. Conclusion

To reduce system complexity, future LEO/VLEO satellite constellations may adopt three-terminal (3T) architectures instead of the conventional four-terminal (4T) design. This paper analyzed the routing implications of this architectural shift and proposed *DisCo3T*, an adaptation of the *DisCoRoute* algorithm for 3T scenarios. We also introduced a modified hop-count model (*IS-MHC*) to restore reachability in topologies with limited cross-plane connectivity.

Our results confirm that the increased latency and hop count in 3T configurations stem from structural constraints, not algorithmic limitations. Despite these challenges, *DisCo3T* delivers acceptable routing performance compared to the 4T version, offering similarly low computational complexity.

Unlike state-dependent schemes, *DisCo3T* relies solely on static orbital parameters, enabling fully decentralized routing with minimal control-plane overhead. Its analytical structure makes it lightweight, scalable, and well-suited for onboard deployment and large-scale simulations.

Future work will extend *DisCo3T* to handle dynamic congestion, node failures, and other runtime uncertainties. We also plan a broader performance comparison with alternative routing strategies and explore their applicability to other constellation designs.

Acknowledgments

This work was supported by the European Space Agency under the ARTES-AT activity "Handover, Data Routing and Radio Resource Management for Very Low Earth Orbit (VLEO) Broadband Constellations" [https://connectivity.esa.int/projects/handingover]. The authors thank Alberto Mengali and Veronica Spirito of ESA for their valuable discussions and suggestions. This work was also in part supported by the European Union – Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), through Mission 4, Component 2, Investment 1.3, CUP E13C22001870001, partnership on "Telecommunications of the Future" (PE00000001 – program "RESTART"), and through the project "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics", under Grant Prot. IR0000013 – Avviso n. 3264 del 28/12/2021.

References

[1] N. Crisp, P. Roberts, F. Romano, *et al.*, "System Modelling of Very Low Earth Orbit Satellites for Earth Observation," *Acta Astronautica*, vol. 187, pp. 475–491, 2021. DOI: 10.1016/j.actaastro.2021.07.004.

[2] H. Luo, X. Shi, Y. Chen, *et al.*, "Very-Low-Earth-Orbit Satellite Networks for 6G," 2022. [Online]. Available: https://www.huawei.com/en/huaweitech/future-technologies/very-low-earth-orbit-satellite-networks-6g.

[3] N. H. Crisp, P. C. E. Roberts, S. Livadiotti, *et al.*, "The Benefits of Very Low Earth Orbit for Earth Observation Missions," *Progress in Aerospace Sciences*, vol. 117, p. 100 619, 2020. DOI: 10.1016/j.paerosci.2020.100619.

[4] V. Ray, T. E. Berger, Z. C. Waldron, *et al.*, "The Impact of Space Weather on Very Low Earth Orbit (VLEO) Satellites," in *AMOS*, 2022.

[5] G. Stock, J. A. Fraire, and H. Hermanns, "Distributed On-Demand Routing for LEO Mega-Constellations: A Starlink Case Study," in *ASMS/SPSC 2022*, 2022, pp. 1–8. DOI: 10.1109/ASMS/SPSC55670.2022.9914716.

[6] C. Ottaviani, A. Compagnoni, J. A. Fraire, *et al.*, "Advanced Routing Strategies for LEO and VLEO Constellations: Ensuring Polar Coverage," in *2025 12th ASMS/SPSC*, 2025, pp. 1–8. DOI: 10.1109/ASMS/SPSC64465.2025.10946062.

[7] W. Wang, Y. Zhao, Y. Zhang, X. He, Y. Liu, and J. Zhang, "Intersatellite laser link planning for reliable topology design in optical satellite networks: A networking perspective," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2612–2624, 2022. DOI: 10.1109/TNSM.2022.3168148.

[8] "Starlink: Satellite technology." (2025), [Online]. Available: https://www.starlink.com/technology (visited on 06/12/2025).

[9] Y. Li, J. Wu, G. Kang, L. Chen, Y. Qiu, and W. Zhou, "A flexible topology control strategy for mega-constellations via inter-satellite links based on dynamic link optimization," *Aerospace*, vol. 11, no. 7, 2024, ISSN: 2226-4310. DOI: 10.3390/aerospace11070510.

[10] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, and X. Chen, "3-isl topology: Routing properties and performance in leo megaconstellation networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 2, pp. 4961–4972, 2025. DOI: 10.1109/TAES.2024.3512535.

[11] J. G. Walker, "Satellite Constellations," *Journal of the British Interplanetary Society*, vol. 37, no. 12, pp. 559–572, 1984.

[12] A. H. Ballard, "Rosette Constellations of Earth Satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-16, no. 5, pp. 656–673, 1980. DOI: 10.1109/TAES.1980.308932.

[13] Thales Alenia Space Italy and Politecnico di Torino, "TN2100: Quantified Comparison and Trade Off Analysis Report," in *ESA ARTES project "Handover, Data Routing and Radio Resource Management for Very Low Earth Orbit Broadband Constellations (HANDING-OVER)"*, 2025.

[14] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen, "Analysis of Inter-Satellite Link Paths for LEO Mega-Constellation Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2743–2755, 2021. DOI: 10.1109/TVT.2021.3058126.