SAST-VNE: A Flexible Framework for Network Slicing in 6G Integrated Satellite-Terrestrial Networks

Mario Minardi, *Student Member, IEEE,* Youssouf Drif, *Member, IEEE,* Thang X. Vu, *Senior Member, IEEE,* and Symeon Chatzinotas, *Fellow, IEEE*

***Abstract*—Network slicing (NS) is one of the key techniques to manage logical and functionally separated networks on a common infrastructure, in a dynamic manner. As the complexity of virtualizing a full infrastructure required unprecedented effort, the initial idea of combining satellite and terrestrial networks has not been fully implemented in 5G yet. 6G networks are expected to further bring NS to a substrate network that is more heterogeneous, due to the full integration between terrestrial and satellite networks. NS describes the process of accommodating virtual networks, typically composed of nodes and links with the respective requirements, into the main infrastructure. This is an NP-Hard problem, typically also known as Virtual Network Embedding (VNE). Existing VNE solutions are designed per use-case and lack flexibility, adaptation and traffic-awareness, especially in such dynamic satellite environment.**

In this work, we investigate the VNE implementation to integrated satellite-terrestrial networks and propose a novel flexible framework, named Slice-Aware VNE for Satellite-Terrestrial (SAST-VNE), which (1) operates based on traffic prioritization, (2) jointly optimizes the load-balancing and the migration cost when network congestion occurs, and (3) provides a near-optimal solution. We compare SAST-VNE to existing well-known near-optimal VNE algorithms such as VINEYard and CEVNE and the shortest-path SN-VNE solution for satellite networks. The simulations showed that SAST-VNE reduces the migration costs between 10% and 40% during satellite handovers while maintaining the network load under control. Furthermore, when congestion occurs, SAST-VNE proved to be flexible in matching the priority of the slice, i.e., tolerated latency, with the time complexity and optimality of the solution.

***Index Terms*—Network Slicing, Virtual Network Embedding, Integrated Satellite-Terrestrial.**

I. Introduction

THE constant growth of traffic demand in the last decade has been shown to continue to grow even for 6G networks [1]. The advent of novel technologies which aim at exploiting the network capacity and accommodating an ever-growing number of traffic has become unavoidable. Among these, the network slicing (NS) paradigm [2] has proved to play a crucial role, already in 5G networks, given its flexible assignment of infrastructure resources to logically isolated virtual networks, also known as slices. In fact, 6G networks are expected to be highly heterogeneous, due to the enormous number of different use cases, both from resource demand and from the nature of the substrate network, advertised to be a fully integrated satellite-terrestrial network.

The high hetereogenity, introduced by the multi-layer structure of the satellite networks and the difference in terms of stability of the links and on-board resources, considerably increased the complexity of managing traffic demands for such dynamic network. In this context, technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are used in combination with a full-stack orchestrator to support NS operation [3], [4]. NS is based on the concept of developing and virtualizing a layered substrate network where each slice is seen as an isolated subset of node and link resources (e.g., processing, storage and communication).

Despite the well-known broad separation of use cases provided by the International Telecommunication Union (ITU) and the Fifth Generation Public Private Partnership (5G-PPP), 6G networks are expected to define with greater granularity the differentiation of use cases, based on demands and application [5], [6]. The more specific the differentiation of the use cases, the greater the need for the infrastructure to be flexible. In fact, network virtualization became relevant because it allows the service provider to exploit the dynamic nature of traffic, increase the network usage and allocate more traffic demands. The process of accommodating a set of nodes and links (including the respective resources), known as Virtual Network Request (VNR), is named Virtual Network Embedding (VNE) [7]. The application of VNE has been investigated for more than a decade and with the advent of 6G networks, this complex optimization algorithm has become difficult to solve. As this work focuses on network slicing, throughout the contribution, we use the term "slice" to refer to the generic VNR.

This paper investigates the application of VNE as the main NS enabler for an integrated satellite-terrestrial network managed by a SDN controller with joint load-balancing and minimization of the resources to be migrated during the satellite handover or network congestions. The paper presents a flexible framework that adapts the time complexity and the optimality of the solution to be found, depending on the latency constraints.

This work is supported by Fond National de la Recherche Luxembourg (FNR) via, Industrial Partnership Block Grant (IPBG), ref. FNR/IPBG19/14016225/INSTRUCT, and the ASWELL Project, ref. FNR/C19/ IS/13718904/ASWELL/Chatzinotas

M. Minardi, Y. Drif, T. X. Vu and S. Chatzinotas are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg (email: {mario.minardi, youssouf.drif, thang.vu, symeon.chatzinotas}@uni.lu).

C. Politis is with SES Techcom, Betzdorf, Luxembourg (email: christos.politis@ses.com).

II. Related Works

The VNE is a well-known NP-Hard problem [8] in the literature with numerous contributions in the past decade. Originally, authors in [9] studied a near-optimal formulation for VNE with a static infrastructure. With a procedure of building an augmented graph with candidate nodes, they manage to provide a coordinated node and link mapping. Authors in [10], propose a similar approach with the difference that they reduce the maximum utilization of each link to 94.5 %. Although these are near-optimal solutions, some critical points can be highlighted, which need to be modified when applied to future transport networks. For example, candidate nodes are discovered only on the basis of geographical distance. However, it is relevant to take into account scenario-dependent values of the link between the candidate and virtual node, such as the Signal-to-Noise Ratio (SNR). Furthermore, the authors consider resources and networks as fixed over a period of time and do not include migrations of the mappings. Providing a framework that constantly monitors, and eventually changes the current mapping of the slices, is relevant to be investigated while studying a VNE algorithm because the dynamic nature of future transport networks brings to unavoidable handovers and a minimization of reconfiguration costs is essential for infrastructure operators. Recently developed VNE solutions for dynamic contexts are proposed in [11]–[13]. Authors in [11] investigate the embedding of Virtual Network Function (VNF) for satellite networks. However, the mapping of nodes and links of VNFs is uncoordinated which typically reduces the quality of the embedding. In [12], authors propose SN-VNE for a satellite network with low complexity, which makes it favorable to compute during handovers in a reactive way, at the expense of the load-balancing performance. In [13], authors formulated the VNE for an integrated satellite-terrestrial network with the objective of minimizing the migrations during handovers. In this case, authors simplified the problem to a single End-to-End (E2E) VNRs, instead of considering an entire slice. The previous mentioned works deal with the optimization of the problem without associating real constraints or applications to the slices. Few works in literature consider 6G use cases which are expected to coexist over the same substrate network. Authors in [14] study the VNE applied to 6G use cases but they only focus on the link mapping, taking the node mapping as already pre-defined.

Furthermore, authors in [15] deal with the problem of E2E network slicing for NTN (Non-Terrestrial)-Terrestrial networks. However, there is no study of the response to real-time situations, such as congestion, with minimization of reconfigurations during handovers. As already proposed in works such as [14], [16], [17], the application of VNE for a satellite-terrestrial integrated network is feasible thanks to novel technologies, such as SDN, which simplify the networking and make it flexible and reactive to changes. In [16], authors exploit the idea of creating a framework for network slicing in integrated satellite-terrestrial networks, with the focus on respecting the QoS requirements, such as bandwidth and E2E delay, of different slices. Our solution intends to improve the framework proposed in [16], with the generalization of the slices to multiple links, instead of a single E2E connection, and minimization of the re-configuration cost while handovers are required. The embedding of slices is often accompanied by a SDN implementation, such as in [17], where the authors implement a SDN application that manages the embedding of slices in the satellite-5G network. However, there is no optimization involved. Our aim is to present an SDN-oriented framework that can be easily interfaced with an SDN-based platform, such as the one proposed in [14].

To the best of our knowledge, no VNE framework has been presented before to support NS for 6G use cases in an integrated satellite-terrestrial network, with a joint optimization function of load-balacing and migration cost minimization focused on real-time network events in either a reactive or proactive manner.

III. Motivations

In the previous section, we described the current literature review of VNE and discussed its application for NS in satellite-terrestrial networks. The NP-hardness of VNE has brought trade-offs between complexity and performance. The main motivations of our work are summarized as follows:

- The use of a single VNE solution cannot efficiently handle future dynamic transport networks. In fact, there is a need for a flexible framework that gives higher priority to performance, i.e., higher computing time, when there is no strict latency constraint. On the contrary, a simpler VNE solution to provide a less optimal embedding when the application is critical.
- Some proposed works in the literature showed how to coordinate the node and link mapping by building an augmented substrate graph. However, candidate nodes are typically chosen only based on a pure 2D geographical distance, which does not reflect the nature of some communications.
- Instead of randomly generating VNRs in terms of resources' demand, it is relevant to test VNE algorithms with real-traffic requirements, where each slice can reflect a proposed use-case for the next generation of transport networks.
- The majority of the near-optimal solutions, due to their complexity, do not provide any minimization of migration's cost when a slice need to be moved because of lack of available substrate resources or due to handovers. This is relevant in future networks where their dynamic exploitation can require frequent migrations.

As we listed the main motivations of our work, in the following we highlight the respective contributions as follows:

- We provide a flexible framework to support the VNE application for integrated satellite-terrestrial networks. We propose SAST-VNE, a new formulation for VNE that provides a joint objective function with load-balancing and minimization of migration cost. SAST-VNE provides a coordinated node and link mapping with a common approach of building an augmented substrate graph;
- We propose a novel way of looking for candidate nodes in the node mapping strategy. While generally this is done

via geographical location, we include additional features to improve the quality of the embedding such as the SNR. Furthermore, we run the candidate search on a 3D space instead of 2D;
- We apply SAST-VNE to the NS concept, where each VNR is seen as a real network slice with requirements coming from 6G use cases. A priority is assigned to each class of slice based on their time constraints. SAST-VNE manages the satellite handover for all classes in a proactive manner, minimizing the migrated nodes and links;
- We simulate a constant monitoring of the network with real-time analysis on whether a slice, or even only a smaller portion of it, is problematic, i.e., does not respect the QoS. Our framework minimizes the migration cost, i.e., number of migrated nodes and links, when the latency constraint of the slice allows it. For critical slices, a heuristic solution is found to minimize outage during migration.

This paper is organized as follows. Section IV formulates the problem. Section V presents the algorithm SAST-VNE and its relaxed version. Section VI describes some initial simulations and Section VII presents the performance evaluation. Finally, Section VIII concludes the paper.

IV. Network Model and Problem description

A. Substrate Network

We model the substrate network as a directed weighted graph $G_s = (N_S, E_s^t)$, where N_S is the set of substrate nodes and E_s^t the set of substrate edges at time t. Each substrate node $n_s \in N_S$ is associated with the available node-related resources (CPU, storage and memory) $R_N(n_s)$, the location $loc(n_s)$ on a three-dimensional space. Given u and v a pair of substrate nodes in N_S, the link e_{uv} is associated with the residual capacity $R_E(e_{uv})$.

B. Slice Request

As described for the substrate graph, we model each slice request as an oriented subgraph $G^v = (N^v, E^v)$. We assign the term "virtual" to the nodes and links of each slice G^v. Each virtual node n^v has a location $loc(n^v)$ and a demanded node-related resources $c_v(n^v)$. Each virtual link i is assigned with the demanded datarate b and a priority p is assigned to each slice.

C. Problem Formulation

In this section, we include the objective function and the constraints of our problem. The objective function (1) jointly optimizes the load-balancing (capacity for links and available resources for nodes) and the differences between consecutive mappings, i.e., it reduces the cost of slice migrations. It is composed of four main terms. The first one computes the sum of the percentage of used capacity for each substrate link e_{uv}. The second one, similarly, is the sum of the consumption (in percentage) of the resources for each substrate node. The last two terms compute the differences in the node and link mapping between the current solution and the newly computed one.

We formulate the objective function as:

$$\begin{aligned}\min_{f_{e_{uv}}^i, x_{mw}} \quad & \frac{1}{|E_s^t|}\sum_{e_{uv}\in E_s^t}\frac{\alpha}{R_E(e_{uv})+\epsilon}\sum_i f_{e_{uv}}^i \\ & + \frac{1}{|N_S|}\sum_{w\in N_S}\frac{\beta}{R_N(w)+\epsilon}\sum_{w\in N_{S'}\setminus N_S} x_{mw}c_v(m) \\ & + \frac{\gamma}{|E_s^t|}\sum_{e_{uv}\in E_s^t}|\mathbf{F^{t-1}}-\mathbf{F^t}| + \frac{\delta}{|N_S|}\sum_{e_{uv}\in E_s^t}|\mathbf{X^{t-1}}-\mathbf{X^t}|\end{aligned} \tag{1}$$

where $(\alpha, \beta, \gamma, \delta) \in \mathbb{R}_+^4$ are the weights for the link mapping, node mapping, difference between consecutive link and node mappings, respectively. In addition, we define $\mathbf{F^{t-1}}$, $\mathbf{F^t}$ as the link mapping matrices at time slot $t-1$ and t, respectively. Similarly, we define the node mapping matrices $\mathbf{X^{t-1}}$ and $\mathbf{X^t}$. It is important to underline that only $\mathbf{F^t}$ and $\mathbf{X^t}$ contain the variables f and x because the mapping in the previous time slot $t-1$ has already been defined. We summarize the optimization variables in Table I. In the following, we present the constraints of SAST-VNE.

Table I: Variables in eq. (1) - (12)

Variable	Explanation
$f_{e_{uv}}^i \in \mathbb{R}_+$	flow variable to indicate if the virtual link i is embedded in $e_{uv} \in E_s^t$
$x_{e_{uv}} \in \{0,1\}$	node mapping variable

To facilitate reading, we highlighted the variables in bold.

$$R_N(w) \geq \mathbf{x_{mw}}c_v(m), \forall m \in N_{S'} \setminus N_S, \forall w \in N_S, \tag{2}$$

$$\sum_i (\mathbf{f_{e_{uv}}^i} + \mathbf{f_{e_{vu}}^i}) \leq R_E(e_{uv})\mathbf{x_{e_{uv}}}, \ \ \forall i, \forall (e_{uv}) \in E_s^t, \tag{3}$$

$$\sum_{w\in N_{S'}} \mathbf{f_{uw}^i} - \sum_{w\in N_S} \mathbf{f_{wu}^i} = 0, \ \ \forall i, \forall u \in N_{S'} \setminus \{s_i, d_i\}, \tag{4}$$

$$\sum_{w\in N_{S'}} \mathbf{f_{s_i w}^i} - \sum_{w\in N_S} \mathbf{f_{w s_i}^i} = b(e_i), \ \ \forall i, \tag{5}$$

$$\sum_{w\in N_{S'}} \mathbf{f_{d_i w}^i} - \sum_{w\in N_S} \mathbf{f_{w d_i}^i} = -b(e_i), \ \ \forall i, \tag{6}$$

$$\sum_{w\in \Omega(m)} \mathbf{x_{mw}} = 1, \ \ \forall m \in N_{S'} \setminus N_S, \tag{7}$$

$$\sum_{m\in N_{S'}\setminus N_S} \mathbf{x_{mw}} \leq 1, \ \ \forall w \in N_S, \tag{8}$$

$$\mathbf{x_{e_{uv}}} = \mathbf{x_{e_{vu}}}, \ \ \forall (e_{uv}) \in E_s^t, \tag{9}$$

and then the variable constraints:

$$\mathbf{f^i_{e_{uv}}} \geq 0, \ \ \forall i, \ \forall u, v \in N_{S'}, \tag{10}$$

$$\mathbf{x_{e_{uv}}} \in \{0, 1\}, \ \ \forall i, \ \forall u, v \in N_{S'}, \tag{11}$$

Constraint (2) makes sure that among the candidate substrate nodes, only the ones with enough available resources are selected. Equation (3), for every virtual link, selects the substrate link with enough capacity. Then, for each virtual link i, the flow's conservation law is enforced by constraints (4)-(6). Constraint (7) sets the maximum number of selected substrate nodes, among the candidates, equal to 1 while constraint (8) ensures that each substrate node is not selected by more than one virtual node. In constraint (9), for each substrate link, the x variable is set to 1 as long as there is a flow in any direction on that link. Finally, constraint (10) sets the positivity of the flow variable and constraint (11) forces the integrity of the node variable x.

D. Linear Programming relaxation and rounding technique

As the problem formulated in the previous section is a Mixed Integer Programming (MIP) which is known to be computationally complex, we provide a relaxed version of the problem. Accordingly, we relax the constraint (11) so that it becomes as follows:

$$0 \leq \mathbf{x_{e_{uv}}} \leq 1, \ \ \forall i, \ \forall u, v \in N_{S'}. \tag{12}$$

V. SAST-VNE Framework

The results of the optimization problem (1) in Section IV provide the optimal coordinated node and link mappings to meet the demands. The logic behind the coordinated node and link mapping is as follows. For each virtual node, the cluster is defined with the candidate substrate nodes to host it. For each candidate node, a fractional value of $x_{e_{uv}}$ is obtained. For each metalink (from candidate to virtual node), the flow variable is computed, normalized to be in $[0, 1]$ and multiplied by $x_{e_{uv}}$. Finally, we select the substrate node based on the highest value of the obtained products, while the other candidates are discarded. This process is repeated for each virtual node. Once the node mapping is defined, the link mapping is computed sequentially by solving the Multi-Commodity Flow (MCF) algorithm for each link. In this section, we present the structure of our proposed SAST-VNE framework, see Algorithm 1, and we analyze the complexity of (1).

A. Description

The pseudo-code of algorithm 1 is described in the following. There is an initial step where the algorithm requires to set parameters, such as the substrate network features (capacity for the links, time description of network behavior, node-related resources). Then the slice generation mechanism is initiated. Once the substrate network and slices are randomly generated, the simulation with a time-slotted strategy runs. For each time slot, some operations are performed. Initially, the KPIs of any mapped slice(s) are checked. If one or more KPIs are not respected or the slice will undergo satellite handover in the following time slot, the slice is considered *problematic* and will be inserted into a queue to be analyzed. In case of any new arrived slice(s), the embedding according to D-ViNE is computed. Lastly, for any problematic slice, our VNE algorithm runs. In case it succeeds, part or the full slice will be moved toward the new embedding. For this last cycle, to be more precise, depending on the priority of the slice and on its situation (handover or KPI failure), different versions of the VNE are run. In fact, if a high priority slice is impacted by congestion, each affected link is re-mapped with shortest-path algorithm, given the source and destination of the affected link(s), to speed up the embedding process and provide connectivity in low time. On the contrary, lower priority slices are mapped using our proposed algorithm, which minimizes the migration cost.

B. Complexity Analysis

To check the complexity of SAST-VNE, it is enough to analyze the solution of the complex algorithms, i.e., equation (1), VINEYard and shortest-path. As the equation 1 is a more complex version of VINEYard and the shortest-path, due to the introduction of two main elements in the objective function, it is the element that dominates the time complexity. The time complexity of (1) is the sum of the complexity of the first two main elements, which can be described as $O((|E^{S'}|(1 + |E_V|))^{3.5})$, plus the complexity of the third element which is similar to the objective proposed in [13], restricted to the first 2 time slots, i.e., $O(g^{2\cdot(h-1)})$, with g the maximum degree of the substrate graph and h the path length, plus the complexity of the last element which is comparable to the number of permutations of $|N_V|$ elements among the set of $|N_S|$ elements, i.e., $O(\frac{|N_S|}{|N_S|-|N_V|})$.

VI. System Model

In this section, we initially present the simulation setup and, then, we provide several performance evaluations to initially set some problem parameters such as the criteria to define and compute the augmented substrate graph (i.e., the expanded substrate graph with the candidate nodes) and the weights involved in the objective function. Therefore, some trade-offs are discussed.

A. Simulation setup

The substrate network is simulated via STK [18] and is an integrated satellite-terrestrial network, as in [14], composed of 32 LEO satellites from Iridium constellation and one GEO satellite (SES-6) (see Figure 1). We assign a location to each substrate node according to the geographical position on a 3D space of dimensions 300x100x300. For the sake of simplicity, we consider the LEO satellites, when in Line-of-Sight (LoS), in a fixed location inside the 3D space. Specifically, we locate the LEO satellite on the plane xyz with z = 100, while we locate the GEO satellite at z = 300. One could argue that these values do not reflect the reality as the LEO satellites orbit at around 800 km while the GEO at 36000 km. However,

Algorithm 1: SAST-VNE

```
Initialization
Substrate network G_s, E_s^t, c(e_uv) and d(e_uv)
 ∀e_uv ∈ E_s^t;
Slices' generation process: arrival, graph G_V and
 node/link requirements);
for t ∈ [t_0, t_sim] do
    save current status of substrate network;
    for any mapped slice(s) do
        for any virtual link do
            check KPIs;
        end
        save current utilization;
        if KPI(s) not respected then
            add to problematic slice
        end
        if handover in the next slot then
            add to problematic slice
        end
    end
    for any arrived slice(s) do
        sort based on priority;
        solve VINEYard;
        if successful then
            update current network resources;
        else
            add to failed slice(s)
        end
    end
    for any failed/problematic slice(s) do
        sort based on priority;
        if high priority then
            solve shortest-path for affected links;
        else
            solve (1) s.t. (2)-(10), (12);
        end
        if successful then
            update current network resources;
        end
    end
end
```

as in our work we build the augmented graph based on the geographical distance, see equation (13), which is controlled in value by the parameter S, with the smallest difference in altitudes between terrestrial, LEO and GEO nodes, we avoid to consider very low values of S. The terrestrial components are the GEO gateways [19], LEO gateways and users and GEO users, which are clearly located at z = 0. As this work is intended to be an extended version of [14], we consider the same type of traffic differentiation as described in Table II. However, unlike the previous work, we consider an entire slice, i.e., a set of interconnected nodes and links, instead of a single E2E connection. Each substrate node is initiated with available resources $\in [25, 45]$ while each link with available capacity = 400 for terrestrial links, 1000 for LEO links and 10000 for GEO. We intentionally assign less capacity to the terrestrial links to make sure the system chooses the satellite links with more likelihood so that our algorithm can be tested in a more dynamic environment.

Table II: Slices description

Slice priority	Application
High - 1	emergency services, collision avoidance scenarios
Medium - 2	infotainment, IFC, earth observation
Low - 3	autonomous boat driving, device manufacturer updates

Each slice is randomly generated as a graph N^V, with the number of nodes as a random number in the interval [2, 5]. Then, we set the probability of every pair of nodes to be connected, equal to 0.5. If the randomization process produces a graph with nodes that are not connected, the slice is discarded as we intentionally aim at connected graphs. Each node in E^V is located in a random position in the xyz plane (with z randomly $\in [0, 20]$ to simulate traffic demands from terrestrial users or plane), and it demands for random resources $\in [2, 6]$. Each link in E^V demands a datarate $\in [50, 150]$.

B. Creation of augmented graph

As already proposed in [9], when embedding a slice, an augmented graph (Figure 2) is suggested. It is built starting from the initial substrate graph which is expanded with *metanodes*, i.e., the nodes to be embedded and *metalinks* as the link between *metanodes* and the candidate subtrate nodes to host the *metanodes*.

The selection of the candidate nodes plays a key role in the complexity of the problem because the larger the selection of the candidates, the higher the number of links and, consecutively, the higher the complexity. As shown in [9], and reported here in Section V-B, the complexity of the relaxed version is linearly dependent on the number of links in the augmented substrate graph. Authors in [9] select the candidate nodes based on the geographical location on a 2D space. As anticipated in section III, our work brings several novelties in the definition of the candidate nodes. Firstly, we build a 3D structure, as the nature of integrated terrestrial-satellite networks foresees. Furthermore, we assign to each link between a generic metanode A and a substrate node B, a signal-to-noise ratio coefficient $S \in [0, 1]$ that multiplies the euclidean distance $d(A, B)$ as explained in (13).

$$d(A,B) = S \cdot \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}. \tag{13}$$

The parameter S is used to modify the distance between the candidate node and the virtual one based on the current SNR. This is used to make the scenario more realistic. For instance, a terrestrial node which is physically closer to the virtual node on ground, eventually can have a worse link quality, e.g., due to terrestrial obstacles, compared to the satellite link. We randomized the value of $S \in [1, 4]$ for the terrestrial nodes, while we select a more constant interval for the LEO links at $S \in [0.5, 1]$. We fix $S = 0.2$ for the GEO link as we assume it to be stable. With these premises, this subsection aims to

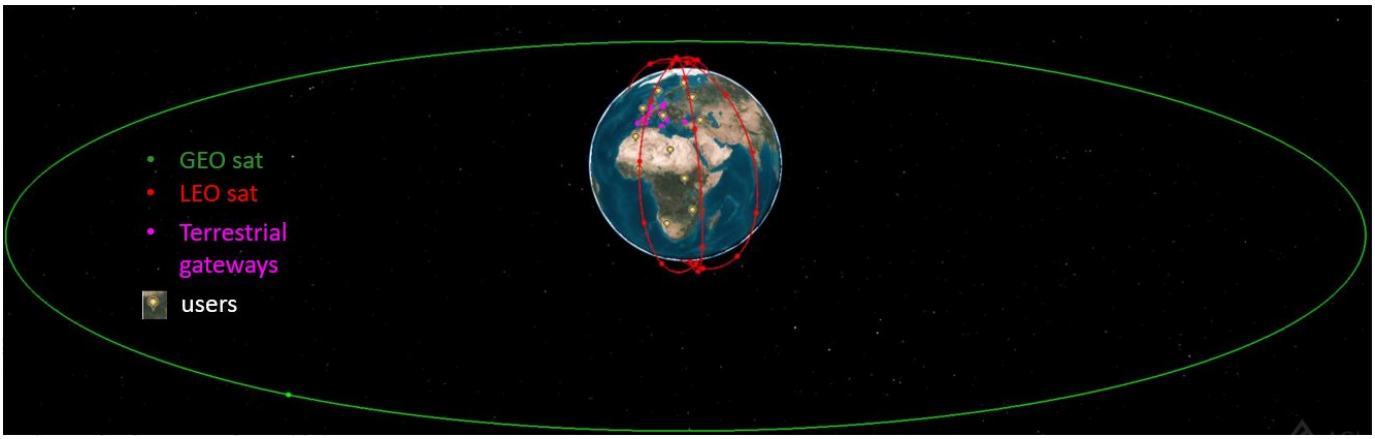


Figure 1: Combined GEO-LEO network

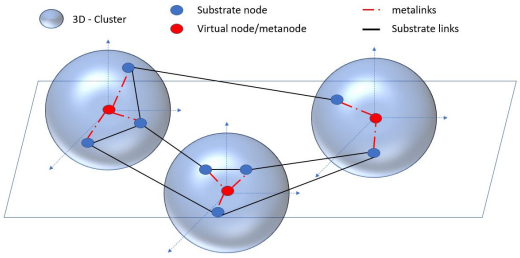


Figure 2: Definition of the augmented substrate graph

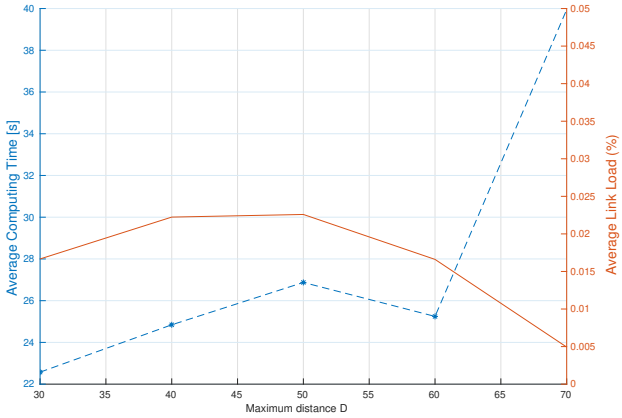


Figure 3: Time complexity and Load-Balancing vs minimum distance

fix a priori a reasonable value of the maximum distance D between the metanode and the substrate nodes at which the substrate node can be considered a candidate one. It is worth noting that this selection is driven by the trade-off between the time complexity and performance, as shown in Figure 3, and it is strictly linked to the chosen substrate topology.

In fact, as shown in Figure 3, the higher the maximum distance D, the higher the average time complexity. In this case, when D $>$ 60, the complexity grows exponentially. On the other hand, the higher D, the more the candidate nodes,

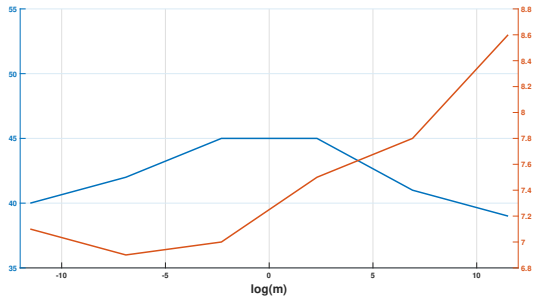


Figure 4: Average node migration and substrate node load vs cost factor

and thus, the easier the load-balancing. In fact, the average link load decreases as D increases. This preliminary analysis sets the maximum distance to be considered for candidate substrate nodes to 50. We also note that, when the substrate network is dense, the reduction of D does not automatically correspond to a reasonable decrease of complexity. As explained in Section V-B, because the complexity is exponentially linked to the number of links of the augmented substrate graph, in a dense network, even with small D, the number of connected nodes can be very high. For this reason, we introduce a feature in SAST-VNE that limits the maximum number of candidate nodes when building the augmented substrate graph.

C. Weight parameters definition in objective function

In this subsection, we study the impact of the weight parameters introduced in (1) to properly set their initial values. Even in this case, a trade-off is highlighted. The weights α and β are assigned to the link and the node load balance, respectively. While γ and δ are for the link and node mapping differences between previous and new mappings, respectively. To properly select the values for the four weights, we initially check the order of magnitude of each of the four terms of (1). Intuitevely, all the terms are percentages as they are divided by the number of nodes or links. In this simulation, we fix the parameters as follows:

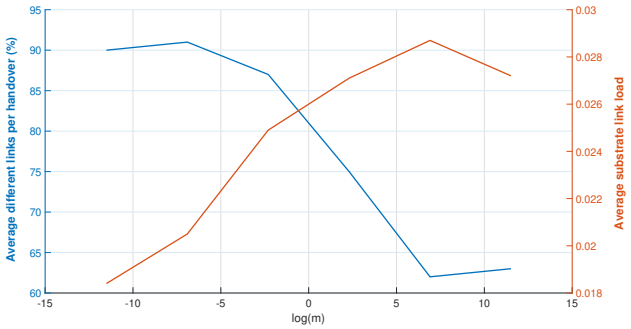


Figure 5: Average link migration and average substrate link vs cost factor

$$\alpha = \beta = 1 \tag{14}$$

$$\gamma = \delta = m. \tag{15}$$

We test the SAST-VNE with different orders of magnitude for m. The aim is to find the suitable value to have a fair trade-off between load-balancing and low migration cost. We consider separately the cost of node migration and link migration. The average node migration is computed as the number of nodes that are migrated everytime there is a change in a slice. Then it is averaged over all slices. Similarly, the average link migration is computed. In Figure 4 we compare the average node migrations and the average network load vs. the cost factor m. Some comments can be highlighted. First, as expected, the overall trend of the performance measuring differences after a congestion, on average, decreases when the cost factor is increased, i.e., the weight in the objective function (1). The opposite happens for the average load of the substrate nodes, computed as the average among the nodes of their CPU capacity utilization during the simulation. It can also be noticed that the average substrate node load remains more or less constant. This is probably due to the fact that the congestion is created only in the links, and therefore, the chance that the nodes remain unchanged but different links are chosen is higher.

Similar considerations can be highlighted when comparing the average migration and load of the link, as shown in Figure 5. In this case, the pattern of both curves is closer to the expected one (cross shape) because we simulate the congestion over the links. Precisely, we randomly select 30 % of the links where the current mapped slices are accommodated, and set their capacity to 85 % of usage to almost always force the mapping over different substrate links.

To avoid having a too unbalanced solution, from these simulations, it looks clear that values of the cost factor close to unity can be taken. Clearly, a further trade-off may also be analyzed in case more priority should be given to node migrations rather than link migrations. In other words, this would mean considering $\gamma \neq \delta$. As this would increase the computational cost, but, nevertheless, without bringing any relevant additional information, we can skip it in this work.

VII. Simulation results

In the previous section we introduced the simulation setup and described some preliminary evaluations to start our SAST-VNE algorithm. In this section, we aim to compare the performance of SAST-VNE to the following well-known VNE baselines:

- CEVNE [10];
- VINEYard [9];
- SN-VNE [12].

The first two baselines run a joint optimization of the node and link mapping, as summarized in Table III, with the construction of the augmented substrate graph. SN-VNE, instead, is a satellite-based VNE algorithm, which deals with the dynamic nature of the substrate network in an heuristic manner, using a shortest-path based approach.

A. Acceptance probability - Network usage

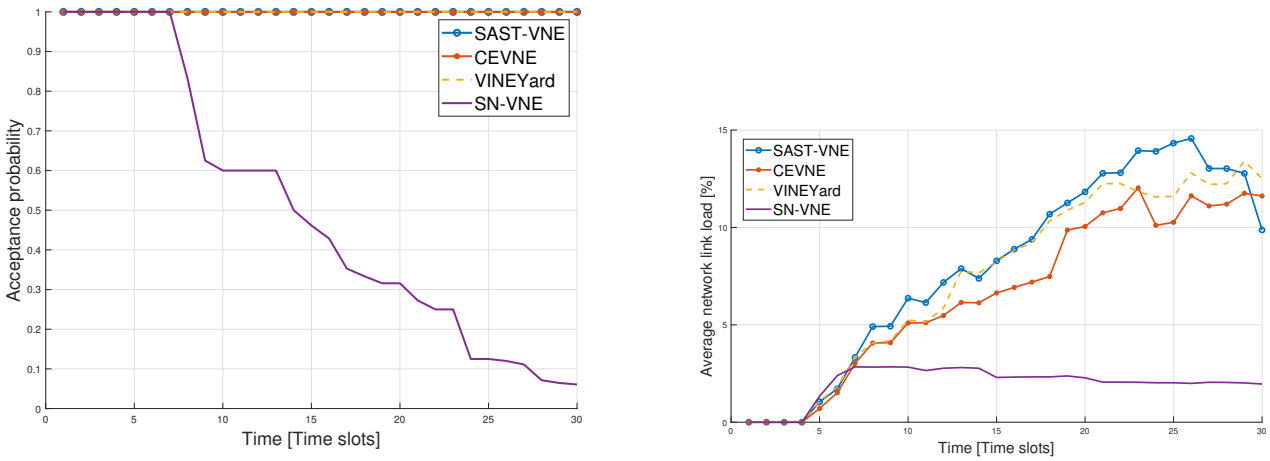
In this section, our objective is to compare (1) the average acceptance probability per slice over the duration of the simulation and (2) the average link load. The acceptance probability is computed every time slot as the number of mapped slices over the arrived ones. The average load on the link is computed as the average capacity consumption of each substrate link, i.e., defined in percentage. For this performance, we do not differentiate the class of the slices. Intuitively, the better the load-balancing, the higher the probability of a slice being accepted. It is worth mentioning that this is the most tricky and challenging performance to evaluate SAST-VNE with, because it is not implemented to further reduce the load-balancing provided by the baselines and CEVNE, which is already near optimal (due to relaxation), but to minimize the average node and link migrations of each slice to be migrated.

We run two simulations with different level of traffic, i.e., in case of low and high load. Figures 6a and 6b depict the results for low load. The aim is to show that when the load is low, our algorithm does not worsen the near-optimal solution provided by CEVNE and VINEYard. Figure 6a shows the acceptance probability vs. time. SAST-VNE matches VINEYard and CEVNE while SN-VNE, as a heuristic solution, decreases the acceptance probability very quickly. As the network load is low, the near-optimal solutions manage to map all the incoming slices, so the acceptance probability is fixed to 1. On the other hand, the more time passes, the higher the load on the network as the number of mapped slices, which consume resources, increases. Even in this case, Figure 6b shows that SAST-VNE does not deviate too much from the near-optimal solutions.

Similarly, we analyze the comparison under high load conditions (see Figure 7). Even in this case, we see that SAST-VNE behaves similarly to CEVNE and in both figures. SN-VNE has a “wave” behavior in Figure 7a because with the higher number of arrived slices, the probability that there is one which can be embedded is higher with respect to the low load scenario. This is also reflected in the average link load (Figure 7b). It is worth to underline that we do not assume that slices expire, hence, network resources are consuming with the time to avoid the comparison to be influenced by that.

Table III: Baselines' features comparison

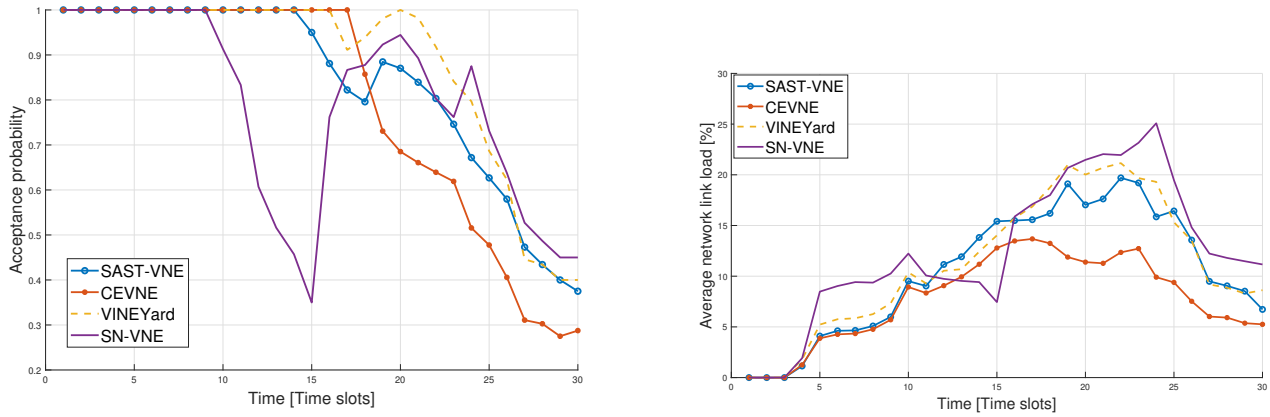
Algorithm	Substrate network	Node/Link mapping	QoS Policy	Optimization
CEVNE [10]	static graph	coordinated	best-effort	load-balancing and congestion-aware
VINEYard [9]	static graph	coordinated	best-effort	load-balancing
SN-VNE [12]	satellite network	uncoordinated sequential	best-effort	shortest path
SAST-VNE	integrated terrestrial-satellite	coordinated	priority-based	joint load-balancing and migration minimization



(a) Acceptance probability vs time

(b) Average link load over time

Figure 6: Acceptance probability and average load vs time in low load conditions



(a) Acceptance probability vs time

(b) Average link load over time

Figure 7: Acceptance probability and average load vs time in high load conditions

B. Handover procedures

This section focuses on the analysis of the number of migrations during handovers, due to loss of LoS within a satellite communication, which require parts or the full slice to be migrated towards different areas of the substrate network. It is worth mentioning that handovers are procedures usually known a priori because the trajectories of satellites, and consequently their link connectivity, are pre-established. This is the reason why a relatively high time complexity should not be immediately categorized as a drawback, considering that the computation can be done in a proactive manner. In addition, handovers are unavoidable for Non-Geostationary (NGSO) links but the migration cost should be considered, especially in those cases where only part of the slice is affected.

In fact, in this section, we compare SAST-VNE with the baseline algorithms. We compare the average number of differences in terms of node and link mapping between consecutive time slots when a handover occurred. It is worth underlining that this metric plays a key role because, for any handover (link or node), there is a cost. To simplify the analysis, we do not consider the specific cost, so we focus only on the pure percentage of differences, which is directly proportional to the cost. To measure the efficiency of an algorithm in minimizing node and link handovers, we look at the percentage of nodes and links that are different between consecutive handovers.

Table IV: Average Node migrations during handovers

Slice's class	SAST-VNE	VINEYard	CEVNE	SN-VNE
1	30.2	68.75	65.4	100
2	29.72	72.50	55.36	100
3	15.56	76.1	55.83	99.6

Table V: Average Link migrations during handovers

Slice's class	SAST-VNE	VINEYard	CEVNE	SN-VNE
1	75	100	94.8	100
2	76.67	90.83	95.24	100
3	89.17	92.86	96.67	100

Then we average over the mapped slices. Clearly, we do not count the slices that are not subjected to handover procedures, as they will affect the average. For this comparison, we subdivide the results per class of slices. We run this simulation with low load to avoid that eventual failures would reduce the number of slices, i.e., samples, from any algorithm and, consequently, would negatively impact the final comparison.

Table IV compares the average migration of the nodes. It can be seen that the percentage of node variations in SAST-VNE is considerably lower with respect to the other algorithms. SN-VNE is a shortest-path based algorithm, so the likelihood that all nodes are different is considerably high. CEVNE and VINEYard perform roughly similarly. This is due to the fact that their only difference is that CEVNE only uses 95% of the available capacity at each link. However, considering that the traffic load is low for this simulation, the probability that a link cannot accommodate the slice anymore is very low. Thus, the overall difference of performance between CEVNE and VINEYard is very limited. Following the same approach, we analyze and compare the average link differences, as shown in Table V. Clearly, the trend matches the previous discussion related to nodes, as they are strictly related. However, for the link comparison, higher values can be observed since, on average, the number of links impacted during handovers is higher than the number of nodes.

C. Congestion intensity

In this section, we simulate a congestion over some links of the substrate network. To be more precise, for the sake of simplicity, we only simulate sudden congestion on some substrate links. Eventually, congestion can also occur on substrate nodes, such as over-utilization of computatibility or storage resources, but this is outside the scope of this article. A random algorithm selects a set of links and it reduces their available capacity for a period of time. To avoid simulating congestion in links that are not accommodating links, we select a random number of slices and simulate congestion to a randomly selected number θ of links which accommodate them. Congestion is simulated as 85% of their capacity being used. As previously mentioned, a relatively high number is selected to make sure that the congestion triggers re-computation of the embedding for that slice. We use $\theta = 30\%$ because higher values are likely to create unavailable paths, especially for the mapped slices over satellite links. First, it is

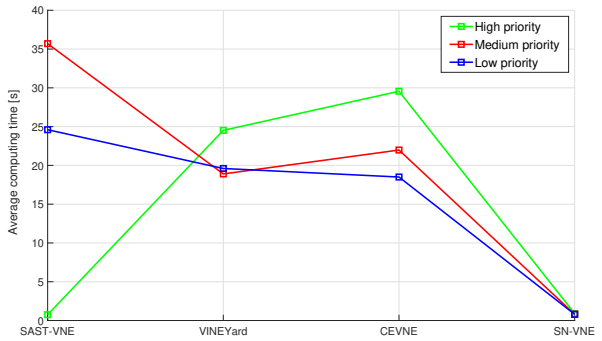


Figure 8: Average computing time for different priority

worth mentioning that the more congested links, the higher the probability that more slices are impacted. At the same time, among the impacted slices, some high priority slices may not afford outage time. In general, as we assume that congestions are not easily predictable, a reactive approach is required. Thus, the time complexity of finding a new embedding and the delay or service outage play a relevant role.

With these premises, we introduce the most important novelty developed in our framework. SAST-VNE is flexible in managing congestion depending on the priority of the impacted slice. In fact, if the slice has the highest priority, we assume that the outage must be as smallest as possible, thus, an heuristic VNE algorithm is implemented to minimize as more as possible the embedding time, at the expense of the load-balancing performance. On the other hand, if the impacted slice does not have strict latency requirements, SAST-VNE solves the joint optimization in (1), s.t. (2)-(10), (12). In this simulation, we compare the average embedding time and migration cost when a congestion occurs. Even in this analysis we assume low load of traffic to avoid that failures, due to lack of capacity, will impact the comparison between SAST-VNE and the baselines. In Figure 8, we compare the average embedding time for the 4 algorithms. Each line represents the average embedding time among the slices with the same priority. For the high priority slices, the difference of SAST-VNE, compared to CEVNE and VINEYard, can be appreciated. In fact, SAST-VNE is more comparable, for high priority slices, to the SN-VNE, as they are both shortest-path based approaches. Differently works for the medium and low priority slices, where we assume low latency constraints are not applicable. In this case, our framework SAST-VNE prioritizes the reduction of cost migration over time complexity. For this reason, SAST-VNE has the highest time complexity compared to the other three baselines. The time complexity of SAST-VNE is always higher than CEVNE and VINEYard due to the added element of the objective function that seeks the minimum migration cost that, as explained in Section V-B, is not a negligible factor. On average, it showed an additional $5 - 10$ s.

In Figure 9, on the other hand, we compare the average link difference for each class of slices and for each algorithm. Since we showed in section VII-B that there is a correlation

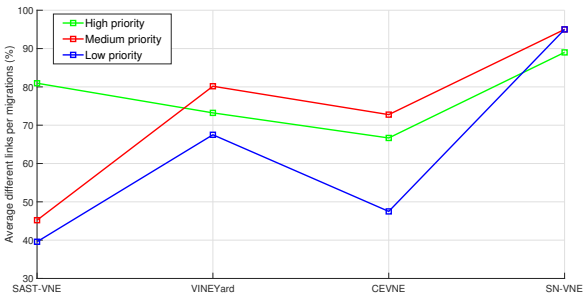


Figure 9: Average link migrations for different priority

between the average node and link differences, for the sake of simplicity, in this section we only consider the average link differences. It can be noted that, even in this case, for the high priority slices, SAST-VNE behaves similarly to the heuristic SN-VNE. In fact, both provide $\sim 10 - 15\%$ higher migrated links than the other two. Similarly to the previous discussion, for the medium and low priority slices, SAST-VNE improves the three baselines between 8 and 50%.

VIII. Conclusion

In this paper, we proposed a novel flexible framework, named SAST-VNE, to implement VNE for network slicing in a dynamic scenario, such as the Beyond 5G integrated satellite-terrestrial network, with a flexible approach. Unlike traditional VNE approaches, which use a fixed mapping scheme for any VNR or slice, we provide a full framework that runs different types of VNE algorithms depending on the latency requirements of the slice being mapped. A joint objective function is provided to maintain load balance while minimizing migration cost. First, as 6G networks will also include NGSO links, SAST-VNE is programmed to handle procedures such as handovers, due to the satellite movement, in a proactive way, as the satellite movements are known. According to some 6G use cases, we simulate slices with three different priority classes. We show that SAST-VNE, compared to baselines, reduces the migration cost, both for links and nodes, during satellite handovers while minimizing the difference between the previous and current embedding. In addition, we compared the acceptance probability during both low and high load conditions and SAST-VNE proved to match the near-optimal solutions provided by VINEYard and CEVNE. Furthermore, we tested SAST-VNE while randomly generating congestion over some network links. SAST-VNE proved to improve the migration cost for the non-critical slices compared to the baselines. On the contrary, a low-complexity solution is provided for the high priority slice which drastically reduces the computing time and provides an almost immediate solution.

References

[1] M. Asad, A. Basit, S. Qaisar, and M. Ali, "Beyond 5G: Hybrid End-to-End Quality of Service Provisioning in Heterogeneous IoT Networks," *IEEE Access*, vol. 8, pp. 192 320–192 338, 2020. DOI: 10.1109/ACCESS.2020.3032704.

[2] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017. DOI: 10.1109/MCOM.2017.1600951.

[4] O. Foundation, "Applying SDN Architecture to 5G Slicing," 2016.

[5] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, "A Comprehensive Survey on the E2E 5G Network Slicing Model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 49–62, 2021. DOI: 10.1109/TNSM.2020.3044626.

[6] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021. DOI: 10.1109/COMST.2021.3067807.

[7] A. Fischer, J. F. Botero, M. T. Beck, *et al.*, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[8] M. Rost and S. Schmid, "On the Hardness and Inapproximability of Virtual Network Embeddings," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 791–803, 2020. DOI: 10.1109/TNET.2020.2975646.

[9] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012. DOI: 10.1109/TNET.2011.2159308.

[10] M. Pham, D. B. Hoang, and Z. Chaczko, "Congestion-Aware and Energy-Aware Virtual Network Embedding," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 210–223, 2020. DOI: 10.1109/TNET.2019.2958367.

[11] I. Maity, T. X. Vu, S. Chatzinotas, and M. Minardi, "D-ViNE: Dynamic Virtual Network Embedding in Non-Terrestrial Networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 166–171. DOI: 10.1109/WCNC51071.2022.9771560.

[12] J. Liu, X. He, T. Chen, *et al.*, "SN-VNE: A Virtual Network Embedding Algorithm for Satellite Networks," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, 2019, pp. 1–6. DOI: 10.1109/ICCChinaW.2019.8849950.

[13] M. Minardi, T. X. Vu, L. Lei, C. Politis, and S. Chatzinotas, "Virtual Network Embedding for NGSO Systems: Algorithmic Solution and SDN-Testbed Validation," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022. DOI: 10.1109/TNSM.2022.3225748.

[14] M. Minardi, Y. Drif, T. Vu, I. Maity, C. Politis, and S. Chatzinotas, "SDN-based Testbed for Emerging Use Cases in Beyond 5G NTN-Terrestrial Networks," in *2nd International Workshop on Autonomous Network Management in 5G and Beyond Systems*, 2023.

[15] Y. Drif, E. Lavinal, E. Chaput, *et al.*, "Slice Aware Non Terrestrial Networks," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 24–31. DOI: 10.1109/LCN52139.2021.9524938.

[16] T. Ahmed, A. Alleg, R. Ferrus, and R. Riggio, "On-Demand Network Slicing using SDN/NFV-enabled Satellite Ground Segment Systems," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 242–246. DOI: 10.1109/NETSOFT.2018.8460139.

[17] S. Hendaoui and C. N. Zangarz, "Leveraging SDN slicing isolation for improved adaptive satellite-5G downlink scheduler," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, 2021, pp. 1–5. DOI: 10.1109/ISNCC52172.2021.9615755.

[18] [Online]. Available: https://www.agi.com/products/stk.

[19] [Online]. Available: https://www.ses.com/our-coverage/teleport-map.