Virtual Network Embedding for NGSO Systems: Algorithmic Solution and SDN-Testbed Validation

Mario Minardi, *Student Member, IEEE,* Thang X. Vu, *Senior Member, IEEE,* Lei Lei, *Member, IEEE,* Christos Politis, and Symeon Chatzinotas, *Fellow, IEEE*

***Abstract*—Non-Geostationary Orbit satellite (NGSO) is an essential element in 5G Non-Terrestrial Networks (NTNs), which can operate either independently or as complementary parts to terrestrial systems to boost the network capacity, coverage and resilience. Due to the highly dynamic topologies, one of the challenges in NGSO is how to harmonize the network virtualized resources to satisfy diverse quality of service requirements in an efficient manner. In this paper, we investigate Virtual Network Embedding (VNE) for integrated NGSO-terrestrial systems while considering dynamic topologies. We propose a Mixed Binary Linear Programming (MBLP) formulation for a Dynamic Topology-Aware VNE (DTA-VNE) algorithm. Given priori information about the network's evolution over time, DTA-VNE plans the embedding for each Virtual Network Request (VNR) over its lifetime. In a highly dynamic environment, the VNE decision can be varying for different VNRs at the expense of a considerable cost of migrating traffic and reconfiguring resources. DTA-VNE aims at minimizing this migration cost to avoid unnecessary re-mappings. To tackle the exponential complexity of the MBLP, we propose an efficient algorithm based on relaxation approaches (DTA-R) to solve large-scale problems. In numerical results, the effectiveness of the proposed DTA-R is demonstrated with much lower migration cost than the conventional implementations. The trade-off between the computation time and migration cost of DTA-R is studied. Finally, we test DTA-R and the baselines in our developed MultI-layer awaRe SDN-based testbed for SAtellite-Terrestrial networks (MIRSAT) to precisely quantify the packet lost for each migration. DTA-R proved to reduce the packet lost by $\sim 2.5 - 5\%$ compared to baselines.**

***Index Terms*—Non-Geostationary Orbit (NGSO), Virtual Network Embedding (VNE), Virtual Network Request (VNR), Network Slicing, Software Defined Networking (SDN).**

I. Introduction

THE current telecommunication networks are being overloaded during the recent years and the trend seems to keep increasing for the upcoming years. The 5G standard has brought in the telecom sector a very large variety of applications, which differ for demanded resources and for Quality of Service (QoS) requirements. The advent of network virtualization has proved to be efficient to guarantee different QoS over the common infrastructure (substrate network).

One of the main topics in network virtualization is defined as Network Slicing [1]–[3]. Due to the heterogeneous nature of novel virtual networks, network slicing algorithms aim to provide intelligent mappings, with the intent to satisfy and guarantee to each of them a pre-defined QoS agreement. Network operators, both terrestrial and satellite, have invested effort and economic resources, due to the potential and benefits that network slicing could have brought into the market. The embedding of different demands onto the same substrate, or physical network, gives a very relevant added-value to the telecom industry-academic sector and has opened the way for the development of those very attracting use-cases of 5G, such as enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC). Currently, network slicing has drawn a considerable attention of research activities, due to its strong impact and benefits [4]. The large field of network virtualization aims at efficiently embedding Virtual Network Requests (VNRs), onto the physical network, which is defined as Virtual Network Embedding (VNE).

Furthermore, Non-Geostationary Orbit (NGSO) satellite networks have recently gained interest due to their capability of cooperating with terrestrial networks to satisfy the increasing network demand. NGSO networks, comprising the Low Earth Orbit (LEO), High Elliptical Orbit (HEO) and Medium Earth Orbit (MEO) constellations, can satisfy relevant common QoS requirements from current networks such as global high-speed coverage with a reduced latency, if compared to GSO networks [5], [6]. NGSO satellite networks, due to their dynamic nature, especially LEO constellations, make the development of an efficient VNE algorithm challenging to deal with frequent network-topology variations. In general, the lower the altitude of the constellations, the more often the topology changes over a defined period of time. This is due to the fact that the Line-of-Sight (LoS) visibility between satellite and gateway has a limited temporal duration. The time-dependent LoS visibility between a gateway and a satellite is in general priori known and, as such, does not require any statistical prediction. This is not the case for link failure/degradation, for example, which usually are simulated and predicted through a statistical description.

In this context, this paper proposes an efficient VNE algo-

This work is supported by Fond National de la Recherche Luxembourg (FNR) programme, Industrial Partnership Block Grant (IPBG), ref. 14016225, and the FNR project, Autonomous Network Slicing for Integrated Satellite-Terrestrial Transport Networks (ASWELL), ref. 13718904.

Mario Minardi, Thang X. Vu and Symeon Chatzinotas are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg (email: mario.minardi@uni.lu; thang.vu@uni.lu; symeon.chatzinotas@uni.lu).

Lei Lei is with the School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China (email: lei.lei@xjtu.edu.cn)

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

rithm for dynamic substrate networks with priori knowledge of temporal variations. The main idea is to subdivide the time horizon into time slots and to exploit the known information of the time dependent network connections in order to embed VNRs, not only for a single time slot, but also for multiple future time slots, minimizing, for each VNR, the migration cost between consecutive time slots. The "*migration cost*" is defined as the ratio between the number of consecutive time slots where the traffic has experienced a migration over the total planned time slots. Later, we provide a more detailed description for this metric.

This paper is organized as follows. We initially provide the state-of-the-art for VNE in Section II. Then, an insight of the motivations and contributions is presented in Section III. Section IV introduces the formulation of the problem and the network model, Section V presents the algorithm's steps and complexity and in Section VI the relaxation procedure is described. Sections VII and VIII present the performance evaluation from simulations and practical implementation, respectively. Finally, Section IX concludes the paper.

II. Related works

In this subsection, we firstly consider the main trends in the literature for VNE, and, secondly, we discuss the satcom-based VNE solutions.

A. Main VNE trends in literature

The relevant literature comprises a large variety of different VNE solutions. In fact, the general NP-hardness of VNE problems [7] has inspired different objectives and solving strategies. Some detailed and recent surveys can be found in [8]–[10]. Online solutions for VNE are one of the main common trends among the proposed implementations [11]–[14], where node and link mapping are computed either independently, which most of the times provides an heuristic approach [15]–[17], or in a joint optimization [18], which usually is an higher quality solution. In Section III-A, we explain why online solutions are not the most efficient solution for satellite networks.

In the literature, some *dynamic* VNE solutions have been proposed. [19] considers a dynamic scenario but the dynamicity is provided by the time varying VNRs over a static substrate network. Indeed, the algorithm applies online re-mapping with the main objective of maximizing the acceptance ratio, given the actual demand, without giving priority to the migration cost. Authors in [20] investigate a time-dependent VNE algorithm where the VNR's migration cost is taken into consideration. However, in this case the migration process is driven by a reward system with the objective of improving the average network utilization and maximizing the acceptance ratio, without considering any migrations caused by a dynamic substrate network. Few other implementations are considering a dynamic substrate network. Authors in [21] and [22] propose resource mapping algorithms for satellite networks. However, these implementations are mostly relying on node caching optimization systems to maximize the network flow and acceptance ratio, without considering any migration cost.

B. Satcom-based VNE

Some VNE solutions specifically target satellite networks. Few satcom-based VNE implementations with a dynamic substrate graph can be found in the literature, in addition to the ones presented above. [16] and [23] address this problem. In both cases, the topology of the network changes over time. However, the algorithms are proposed as online algorithms, which means that, at each instant, the algorithm has to verify the entire network topology. Those VNRs, affected by the link change, will be re-mapped. This approach, as the previously mentioned online solutions, might produce a considerable high number of migrations over time per each VNR. While for terrestrial infrastructure, migrations might not be a relevant issue, for SatCom nodes, an excessive amount of migrations might significantly affect the network performance. In fact, migrations over satellite networks are not often smooth because of some temporal loss of service or dropped sessions. In addition, migrations require considerable signalling which creates traffic overhead and might be prone to errors.

Last, but not least, the majority of the proposed VNE works lack of testing the algorithm in a built-in testbed. Clearly, this is the most interesting and challenging aspect of the implementation, where the algorithm faces the technical effects of a software/hardware emulated system.

Given these premises, from theoretical perspective, a relevant attention should be given to the VNR migration cost, when the substrate network is highly dynamic, such as satellite networks. From the practical side, a testbed should be provided to validate the presented algorithm and to help understanding the pros and cons of its integration in a real network. To the best of our knowledge, a VNE algorithm which exploits the priori time-variation of NGSO constellation, in order to minimize the migrations of VNRs over their lifetime, has not been proposed before.

III. Motivations and Contributions

In this section, we firstly provide a comprehensive analysis about the relevance of an ad-hoc topology-aware VNE solution for satellite networks and the added-value of a built-in testbed to validate the results. Secondly, we detail our contributions.

A. Motivations

The literature review has highlighted how online VNE solutions are important and preferred for their simpler and efficient nature.

Figure 1 depicts a traditional online VNE approach for dynamic substrate networks [16]. The time domain is divided into time slots such that the substrate network can be considered static within each time slot. A random VNR arrives at the time slot $t = t_3$, with a lifetime L. This means that the VNE algorithm is asked to guarantee the embedding, at least, until the time slot $t = t_3 + L$. At the time slot t_3, the VNE decisions have been made based on the current status of the substrate network. However, at time slot t_i, the substrate network has changed, and it might happen that the VNR is affected by this change. In this case, the VNE has to be computed again, providing a different embedding and, consequently, triggering the

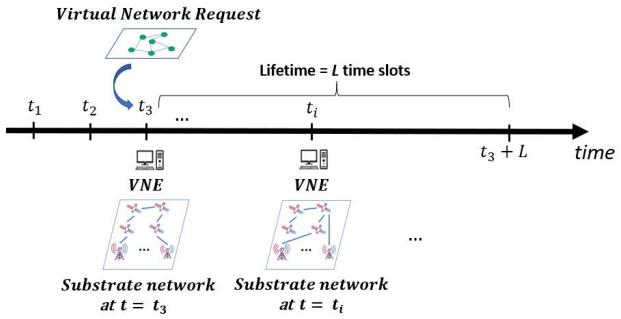

Figure 1. Traditional online VNE solution for dynamic substrate network

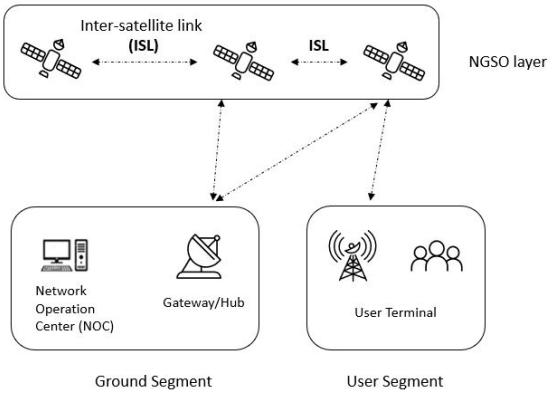

Figure 2. Migrations over terrestrial/NGSO network

traffic migration to the new path. This process is repeated until the VNR expires. Clearly, this approach lacks a global view in VNE optimization and may result in a typical suboptimal solution. A suboptimal solution might cause higher number of traffic migrations which corresponds, not only to economic expenses, but also to network performance degradation such as temporal outage, signalling, satellite handovers.

More precisely, from a network provider point of view, computational and/or technical costs might be involved for each VNR's migration. Clearly, the migration cost might be dependent on several factors such as the used technologies, the quantity of traffic to be migrated and, quite relevant, the considered layer of International Organization for Standardization/Open Systems Interconnection (ISO/OSI) protocol stack. For example, at the physical layer, migrations correspond to changes of hardware settings (antenna direction, etc.). At the layer 2, the time needed to re-configure one or several path might correspond to packet losses, frequent handovers, data plane signalling, which most likely has to be faced with some ad-hoc algorithms.

At the networking layer, the change of routing paths corresponds to modification of routing tables while, at the application layer, migrations are more related to the management of necessary computability resources to solve the complex VNE problem. In this paper, we refer to the *migration cost* as the number of times, in percentage, that each VNR has to be migrated from one path to another one, due to topology variation, along its lifetime. It is worth underlining that this metric has a relevant impact on the computation time. In fact, on one side, the more planned time slots are considered, the lower the migration cost, in probability, due to a longer analyzed period of time. On the other side, the more the time slots, the higher the complexity of the problem, i.e. higher computation time. In this paper we analyze this trade-off, driven by the choice of the number of time slots to be planned for.

In the VNE research field, there are widely used common metrics to evaluate the quality of VNE solutions such as *load balancing* [18], [19], *energy-saving* [24], [25], *acceptance ratio* [11], [13], [21], [22] and *computation time* [12], [26], [27]. However, it is evident that the *migration cost* should be considered as a relevant metric to evaluate the VNE's performance and it should be given sufficient attention, especially in high dynamic scenarios such as the one considered in this paper. Figure 2 depicts a global view of the considered scenario. A VNR, which is embedded over a satellite network, might be required to be migrated or forwarded toward a different satellite. This clearly does not happen often for GSO networks, known to be very stable and static, but quite regularly for NGSO.

The last relevant motivation for this work has a more practical meaning. The VNE literature lacks overall of practical validation of the VNE solutions. Usually, VNE algorithms are simulated and validated in Matlab. While this approach is time effective in comparing multiple algorithms and flexible in analyzing several performance, it considers ideal settings of the scenario. However, this approach might not highlight hidden details, coming from the real world. For high dynamic scenarios, for example, the traffic is likely to be migrated over time, and the effect of a migration on the experienced quality of service (e.g. packet lost) is difficult to be simulated in Matlab. This is the main reason why we developed and used a built-in testbed.

B. Contributions

Considering the highlighted trends of VNE's literature in Section II and the motivations for our work in Section III-A, our contributions are summarized as the following:

- We propose a Dynamic Topology-Aware VNE (DTA-VNE) algorithm that minimizes the average migration cost for dynamic network requests over planned time slots. We employ a linearization process, to transform the original non-linear objective into a linear form;
- Since the DTA-VNE algorithm is based on a MBLP formulation, its complexity increases exponentially with the network size and the planned time slots. To tackle this, we propose DTA-R, a relaxed version of DTA-VNE, which mitigates binary constraints and allows to study performance-complexity trade-off;
- We study the impact of the number of planned time slots on the trade-off between the migration cost and the computation time, comparing the results to the online shortest-path and load balancing algorithms. The proposed algorithms have proved to avoid unnecessary

migrations already with few time slots considered, while keeping the computation time a reasonable value;
- As the literature lacks of VNE practical validations, we test the feasibility of our developed VNE algorithms via a MultI-layer awaRe SDN-based testbed for SAtellite-Terrestrial networks, MIRSAT, which employs the realistic topology of a MEO satellite system [28]. The implementation results show that, while analyzing the traffic of a VNR, each migration foresees a considerable outage. MIRSAT intents to further motivate the prioritization of the migration cost in the optimization problem.

IV. Network Model and Problem description

A. Network Model

We model the integrated satellite-terrestrial substrate network as a directed weighted graph $G_s = (N_s, E_s)$, where N_s is the set of substrate nodes and E_s is the set of substrate edges. Given u and v a pair of substrate nodes in N_s, we define $c(u, v)$ as the available capacity of the link (u, v) and $p(u, v)$ as the propagation delay. VNRs are randomly generated as end-to-end connection between random pairs of substrate nodes in the set N_s, given m the source node and d the destination node. In other words, the node mapping is already defined. While this might seem a strong assumption, in the following we provide two reasons to motivate this choice. Given the intent to validate the VNE algorithm in the built-in testbed, we visualize the VNR as client-server requests. This gives already defined nodes in the substrate network (source or client and destination or server). In addition, in order to have a validation consistent with the simulation results, we avoid to consider node related constraints (e.g. CPU) which is not trivial to be simulated in the testbed. Each VNR demands a minimum data rate b, a maximum tolerated latency l, it has a lifetime L time slots and it arrives at time slot t_a. Therefore, each VNR's mapping is computed for the interval $[t_a, t_a + L]$ time slots. It is relevant to clarify that the unit of measure for the lifetime and simulation time is the "time slot". However, the formulation and results do not depend on the real duration of the time slots. Nevertheless, in Section VII, a real duration value is computed from the use-case and provided to give consistency and truthfulness to the emulated scenario. The VNRs follow a poisson distribution with average arrival rate λ VNRs per time slot. We assume that the topology of VNRs does not vary over time. In order to make the notations easier and more comprehensible, we do not assign the VNR's index to each VNR related parameter introduced before.

We consider the time-slotted system and assume the system operates for t_{sim} time slots. Despite the substrate network can be considered static within each time slot, the modelling variables should be considered as function of time. Indeed, we introduce the time dependency in the substrate graph notations. Accordingly, the substrate graph is modelled as a time expanded graph, following the approach proposed in [29], [30], $G_s^t = (N_s, E_s^t)$, with $c(u, v, t)$ as the available capacity of the substrate link (u, v) at the time slot t. It is worth noting that, while E_s^t and, consequently, the link resources, $c(u, v, t)$, are dependent on time, due to both the temporal topology variation and the consumption of resources over time assigned to the incoming VNRs, the set N_s, on the opposite, is not dependent on time because we assume that the available substrate nodes do not change over time. The same applies to the propagation delay.

B. Problem Formulation

In this subsection, we formulate the VNE embedding algorithm running for one VNR at a time. Considering that each VNR is simulated as an end-to-end request between two randomly selected substrate nodes, the node mapping is already defined. Thus, in the following, the VNE formulation is restricted to only the link mapping. Our objective is to minimize the migration cost, taking into account the known temporal description of the substrate network topology. Before presenting the formulation of the objective function, we initially introduce the following flow variable:

$$z_{uv}^t = \begin{cases} 1, & \text{if the VNR is mapped in link } (u, v) \text{ at time slot } t, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The algorithm focuses on the difference of link mappings between consecutive time slots, for each substrate link (u, v). In other words, the aim is to keep, as much as possible, a consistent link mapping over multiple consecutive time slots in order to minimize the migration of the VNR's traffic due to topology variations. Thus, given that the algorithm works with differences, the absolute value is required to avoid negative values, while comparing the link mapping between consecutive time slots (2).

$$|z_{uv}^{t+1} - z_{uv}^t|, \ \ \forall (u, v) \in E_s^t, \tag{2}$$

At this stage, we introduce the following assumption. We do not give importance to how many links have changed between consecutive time slots, but, instead, to the presence or absence of any change. In other words, while comparing the mappings between consecutive time slots, if one link changes or all links are different, it is considered equally as a migration, i.e. binary migration variable, introduced in the following, equal to 1. It is worth underlining that this does not force the objective function to be always null. In fact, the objective function keeps minimizing the percentage of different consecutive mappings and the results show that, almost always, migrations occur if VNRs between at least one non-terrestrial node are involved. Accordingly, the algorithm is based on the following function:

$$f(t) = \begin{cases} 1, & \text{if } \sum_{(u,v) \in E_s^t} |z_{uv}^{t+1} - z_{uv}^t| > 0, \ \ \forall t \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$f(t)$ is a binary indicator function, which has the value "1" if there is at least one difference between the link mapping at time slot t and the one at time slot $t + 1$. Based on the previous definition, equation (4) shows the initial formulation of the objective.

$$\min_{z_{uv}^t} \left(\frac{\sum_{t \in [t_a, t_a + T]} f(t)}{T} \right), \tag{4}$$

The objective function (4) minimizes the migration cost, previously defined as the percentage of migrations experienced by the VNR, between consecutive time slots, over T planned time slots. It can be noticed that the function is not linear. In fact, in Section V, we provide an explanation of the linearization process and the obtained final linear objective function.

In the following, the constraints of the problem are presented.

$$z_{uv}^t \cdot b \leq c(u,v,t), \;\; \forall (u,v) \in E_s^t, \forall t \in [t_a, t_a + T], \tag{5}$$

$$\sum_{w \in N_s} z_{mw}^t - \sum_{w \in N_s} z_{wm}^t = 1, \;\; \forall t \in [t_a, t_a + T], \tag{6}$$

$$\sum_{w \in N_s} z_{dw}^t - \sum_{w \in N_s} z_{wd}^t = -1, \;\; \forall t \in [t_a, t_a + T], \tag{7}$$

$$\sum_{v \in N_s} z_{vu}^t - \sum_{v \in N_s} z_{uv}^t = 0, \;\; \forall t \in [t_a, t_a + T] \quad \forall u \in N_s \setminus \{m, d\}, \tag{8}$$

$$z_{uv}^t \in \{0,1\}, \;\; \forall (u,v) \in E_s^t, \tag{9}$$

$$\sum_{u \in E_s^t} z_{uw}^t + \sum_{u \in E_s^t} z_{wu}^t \leq 2, \;\; \forall t \in [t_a, t_a + T], \quad \forall w \in N_s, \tag{10}$$

$$\sum_{(u,v) \in E_s^t} z_{uv}^t \cdot p(u,v) \leq l, \;\; \forall t \in [t_a, t_a + T], \tag{11}$$

$$\sum_{(u,v) \in E_s^t} z_{uv}^t \leq h, \;\; \forall t \in [t_a, t_a + T], \tag{12}$$

Constraint (5) states that, for each time slot and for each substrate link (u, v), the assigned data rate cannot exceed the actual available capacity. It is relevant to clarify the reason why the minimum data rate is being upper bounded by the link capacity. Considering that the proposed algorithm is formulated for an SDN-enabled scenario, the SDN controller (responsible for executing the VNE algorithm) initially assigns the minimum data rate to be guaranteed for each VNR (which cannot clearly be embedded in links with insufficient capacity). Given the ability of the SDN controller to constantly check for the traffic usage of any mapped VNR, a higher data rate can be assigned real-time if any VNR requires that (higher data rate might happen e.g. unexpected peak of traffic), only in case of available capacity. Constraints (6), (7) and (8) refer to the flow's conservation law. Indeed for (6) and (7), given the source and destination substrate nodes of the VNR, the sum of outgoing flows from the source node m has to fully flow over the network because the demanded data rate should be guaranteed. The same applies for the incoming traffic in the destination node d, with a negative sign. (8) guarantees the flow's conservation law for the intermediate nodes, where the sum of incoming flows has to always match the sum of outgoing flows.

Equation (9) defines the integrity constraint for the flow variable. From a physical point of view, the integer flow variable means that splitting-path is not accepted in this configuration, so either the flow is passing through a substrate link, or it is not.

The constraint (10) guarantees the absence of loops. This is relevant because the objective function (4) minimizes the link mapping differences between consecutive time slots, but it does not guarantee that loops are avoided. In addition, constraints (6)-(8) guarantee the flow's variable conservation, but they also do not guarantee that loops are avoided because loops would still keep, the flow's conservation law, valid. This is the reason why it is important to set the constraint (10) where, for each substrate node, the sum of incoming flows and the outgoing flows, cannot exceed "2". This value is given by the fact that each substrate node can, at most, manage two flows per VNR (if it is an intermediate node), one incoming and one outgoing, giving "2" as total sum. If the sum of flows is greater than "2", it means that there is a loop. Constraint (11) keeps the end-to-end propagation delay, computed as the sum of the delay of each link from source to destination, lower or equal to the tolerated latency. It is important to compare the propagation delay to the maximum latency allowed by the VNR, especially in satcom, where delays may not be negligible, to avoid that low-latency communications are affected by large delays, which makes the scenario not realistic anymore. Finally, constraint (12) reduces the solution's space, limiting the search space to the paths of length h. In Section V, we highlight the impact of the value h on the complexity of the problem. Intuitively, the value h drives the trade-off between the quality of the solution and the computation time. Very low values of h will limit the solution's quality while keeping the computation time low. On the opposite, higher values of h will improve the solution at the expense of the computation time. The value h is very much dependent on the number of links and possible paths that are available over the network.

V. The Dynamic Topology-aware VNE algorithm

The formulation presented in the previous section is not linear. In this section, we provide a comprehensive description of the linearization process, the overall procedure of the proposed Dynamic Topology-aware VNE (DTA-VNE) algorithm and the problem's complexity.

A. Linearization process

The functions (3)-(4) are not linear due to the presence of the absolute value. In order to linearize them, we use the approach presented in [31]. For the sake of simplicity, we describe the process for one generic link (u, v) at the generic instant t. Thus, we can consider the initial objective function, without summations, as:

$$\min(|z_{uv}^{t+1} - z_{uv}^t|). \tag{13}$$

We define a second variable x_{uv}^t and add the two following constraints:

$$(z_{uv}^{t+1} - z_{uv}^t) \leq x_{uv}^t, \; \forall (u,v) \in E_s^t, \tag{14}$$

Table I
FORMULATION VARIABLES

z_{uv}^t	binary flow variable to indicate if the VNR is embedded in (u, v) at the time slot t
x_{uv}^t	absolute value $\lvert z_{uv}^{t+1} - z_{uv}^t \rvert$ defined for each substrate link (u, v)
y^t	binary variable to indicate if there is a migration from time slot t to $t+1$

$$-(z_{uv}^{t+1} - z_{uv}^t) \leq x_{uv}^t, \ \forall (u,v) \in E_s^t, \tag{15}$$

Finally, the linear objective function is written with the new variable as:

$$\min(x_{uv}^t). \tag{16}$$

The variable x_{uv}^t is introduced to substitute the absolute value in (13). However, as previously mentioned in Section IV-B, we are interested to use a migration variable in the objective function which denotes the presence or absence of migrations between consecutive time slots. Since the variable x_{uv}^t counts the differences between the flow variables between consecutive time slots, for each substrate link, separately, we introduce, according to (17), the migration variable y^t, which does not count how many links differ between consecutive time slots, but, rather, only denotes the differences. This is the reason why it is an integer variable:

$$y^t = \begin{cases} 1, & if \sum_{(u,v)\in E_s^t} x_{uv}^t > 0, \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Finally, the original problem is reformulated as follows:

$$\begin{aligned} &\min_{y^t, z_{uv}^t} \left(\frac{\sum_{t\in[t_a, t_a+T]} y^t}{T}\right), \\ &\text{s.t. } (5)-(12), (14), (15) \end{aligned} \tag{18}$$

Table I summarizes the optimization variables.

B. DTA-VNE algorithm

DTA-VNE solves the link mapping for each VNR, separately and independently, in a sequential way. As highlighted in Section III, the core idea is to plan, in advance, the VNR embedding for multiple time slots. Thus, we firstly focus the attention to the single VNR mapping (Algorithm 1). Secondly, we present the main environment (Algorithm 2) of the simulation, which manages the Input and Output of every instance of Algorithm 1, the VNRs generation and the resources update over time.

Algorithm 1 takes, as input, the current status of the substrate network (topology and resources), the demanded data rate and the end nodes of the VNR, and the time slots to be planned. Thus, the embedding is computed for the current time slot and for T following time slots (results will be shown for different values of the parameter T). Lines 6-7 are solved and, if a feasible solution exists, the embedding for T time slots is provided as output.

In the following, the pseudo-code for Algorithm 2 is presented and discussed. The simulation runs for t_{sim} time slots (main *for* cycle) where each iteration lasts one time slot. For each iteration, the priority is given to the VNRs, embedded in the previous time slot, which have to be re-embedded, solved by Algorithm 1 via the well-established LP solvers, in case the link mapping they have been assigned with, is not available anymore due to network topology variation. After this step, once the queue of VNRs to be re-mapped is empty, in case some have expired, the system releases the corresponding substrate resources. Afterwards, the new arrived VNR(s), in case any, will be mapped via Algorithm 1. After each VNR mapping, the substrate available capacities are updated.

Algorithm 1: Single VNR mapping for the interval $[t_a, t_a + T]$

```
1  Input
2  current substrate network status G_s^t, ∀t ∈ [t_a, t_a + T];
3  demanded data rate b, tolerated latency l;
4  m and d, source and destination substrate nodes;
5  T time slots to be planned;
6  Solve (18)
7      s.t. (5)-(12), (14), (15)
8  if successful then
9      Output
10     link mapping for T time slots;
11 end
```

Algorithm 2: Dynamic Topology-aware VNE

```
1  Initialization
2  Substrate network definition G_s^t, ∀t ∈ [t_0, t_sim];
3  VNR parameters (demanded physical resources, arrival
     distribution, end-to-end nodes);
4  for t ∈ [t_0, t_sim] do
5      for VNR(s) to be migrated do
6          execute Algorithm 1;
7          if successful then
8              update residual capacities of substrate links;
9          end
10     end
11     for expired VNR(s) do
12         release assigned substrate resources;
13     end
14     for new VNR(s) arrived do
15         execute Algorithm 1;
16         if successful then
17             update residual capacities of substrate links;
18         end
19     end
20 end
```

C. Complexity's Analysis of DTA-VNE Algorithm

To analyze the complexity of DTA-VNE, it is enough to consider Algorithm 1, given that Algorithm 2 is only managing different instances of the first one. Algorithm 1 (lines 6-7) is solved through an exhaustive search which will be interrupted by the first solution that makes the objective function null,

i.e. no migrations over T time slots. Even though there is an exhaustive search, the number of feasible paths between two given nodes (source and destination of the VNR) is strongly dependent on the topology of the substrate network. Given that the considered substrate graph is significantly simpler than a complete graph, this considerably reduces the complexity of the problem. We assume g the maximum degree of the substrate graph, such that $g < |N_s| - 1$, with $|N_s|$ the number of the substrate nodes. Thus, the complexity of finding all feasible paths of length h, given the two end nodes, is upper bounded by $O(g \cdot (g-1)^{h-2})$. This formula is obtained by the fact that, starting from the source node, there are at most g possible adjacent nodes. Then, from the second until the penultimate node, there are at most $g-1$ possible adjacent nodes that can be chosen as next hop, given that the flow does not go back from the link it has arrived. Considering h the path length, there are $h+1$ nodes in total over the path, among which 2 nodes are source and destination. The second node of the path is considered in the first selection which has g possibilities. Finally, there will be $h-2$ nodes to be selected in the path among at most $g-1$ nodes every time. It is worth noting that the provided complexity is an upper bound because the value g is the maximum degree of the substrate graph. In addition, given that the objective function considers the combination of the single time slot embedding over T time slots, the total complexity is upper bounded by $O((g \cdot (g-1)^{(h-2)})^T) \sim O(g^{T \cdot (h-1)})$ if $g >> 1$. This highlights the impact of T, h and the density of the substrate graph (g), on the complexity.

VI. LP RELAXATION

While the non-linearity of (13) is solved by the above-mentioned linearization process, the binary nature of the flow variable z_{uv}^t exponentially increases the computation time with the increase of the planned time slots and/or the network size. As the computation time is crucially relevant for online VNE algorithms and considering the size and dynamicity of the targeted substrate networks, i.e. MEO and LEO constellation, we propose a relaxed version of DTA-VNE, named DTA-R. To obtain DTA-R, we relax the binary flow variable, in constraint (9), to continuous values. We underline that the exponential complexity of DTA-VNE is given by the binary constraint on z_{uv}^t, because the migration variable y^t grows only linearly with T, which is not large in practice. For this reason, the relaxation process is applied only to z_{uv}^t.

In addition, we introduce the following penalty function

$$P(\bar{\boldsymbol{z}}) = \sum_{(u,v) \in E_s^t} z_{uv}^t - {z_{uv}^t}^2 \tag{19}$$

to penalize the values of the flow variable, far from the integer extremes 0 and 1. We run an iterative process which stops when $|q^{i-1} - q^i| < \varepsilon$, with q^i be the solution at the i-th iteration and ε a very small number. Since (19) is not convex, we introduce the first order approximation which results as

$$P(\bar{\boldsymbol{z}}) = \sum_{(u,v) \in E_s^t} z_{uv}^t + {\bar{z}_{uv}^t}^2 - 2 z \bar{z}_{uv}^t, \tag{20}$$

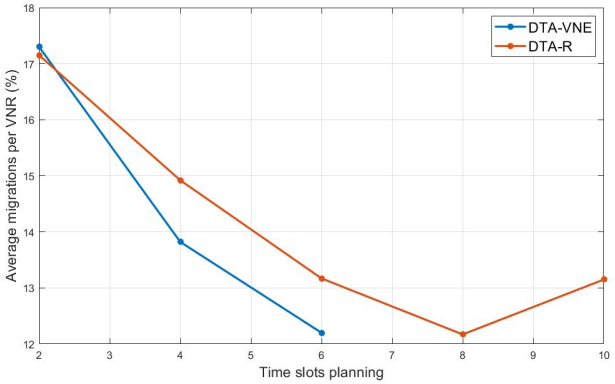

Figure 3. Comparison of average migration cost vs time slot planning between integer formulation (DTA-VNE) and relaxed one (DTA-R)

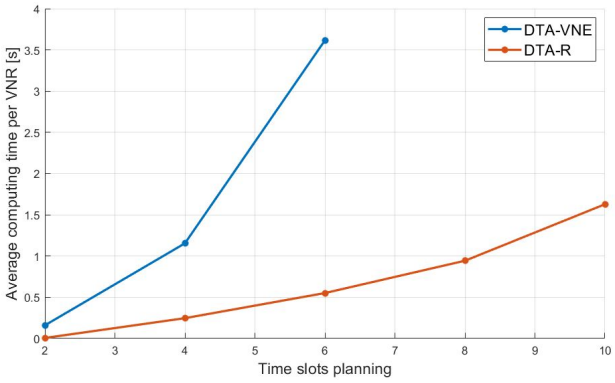

Figure 4. Comparison of average computing time vs time slot planning between integer formulation (DTA-VNE) and relaxed one (DTA-R)

with $\bar{z}_{uv}^t$ the solution at the previous iteration. With the addition of $P(\bar{\boldsymbol{z}})$ to the objective function (18), the final objective function of DTA-R is expressed as

$$\begin{aligned} &\min_{y^t, z_{uv}^t} \left(\frac{\sum_{t \in [t_a, t_a+T]} y^t}{T} + \gamma P(\bar{\boldsymbol{z}}) \right), \\ &\text{s.t. } (5)-(8), (10)-(12), (14), (15) \end{aligned} \tag{21}$$

with $\gamma > 0$ as the penalty weight parameter.

Once the iterative process terminates, the rounding technique follows a deterministic approach, where feasible, as proposed in D-ViNE [18]. To understand the performance losses, we provide the trade-off analysis between DTA-VNE and DTA-R, in terms of average migrations and computing time for few planned time slots.

Figure 3 shows the loss in terms of average migrations, for multiple values of planned time slots. It can be noticed that when T = 2, there is almost no difference and, anyway, the algorithm does not perform well in both cases because there is almost no planning. When T increases, a difference can be noted. Despite that, the relaxed version allows to go further with the planning, such as T = 8. This evaluation aims to show that the relaxed version experiences a reasonable loss in performance and it makes sense to use it for performance evaluation, even for MEO constellations.

The other side of the trade-off is depicted in Figure 4, where the gain in computing time is shown. DTA-R shows that even

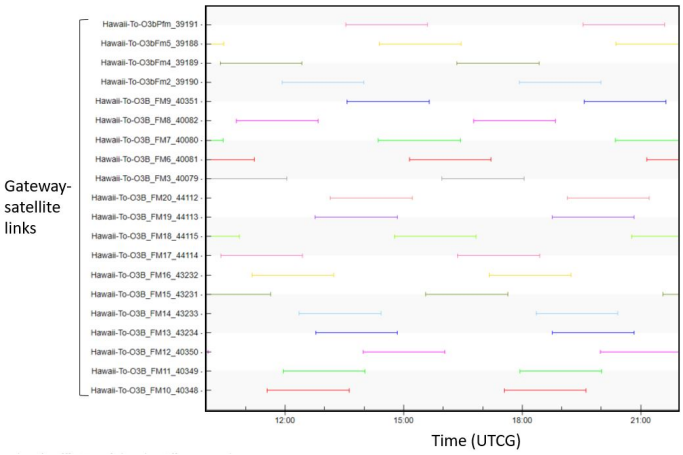

Figure 5. Access time scheme of Hawaii gateway to the 20 O3b satellites over 12 hours

T = 10 has still a reasonable complexity to be computed.

VII. Performance Evaluation with Simulations

In this section, we firstly describe the simulation setup and, secondly, we provide several performance evaluation analysis, while comparing DTA-R to online load balancing and shortest-path baseline algorithms.

A. Simulation setup

The substrate network is an integrated satellite-terrestrial network. We considered the SES O3b-MEO constellation with real location data for the MEO gateways [28] across the globe. The constellation is composed of 20 MEO satellites while the ground segment is composed of 9 terrestrial gateways. We additionally introduce Inter-Satellite Links (ISLs) among the MEO satellites, as a circular shaped topology. Thanks to the Systems Tool Kit (STK) [32] software, we have obtained the time series of the satellite-gateway link for each satellite and for each gateway over the entire duration of the simulation. The orbit of MEO satellites has an altitude of 8063 km, and considering that, for the purpose of this algorithm, we have to subdivide the simulation into time slots such that connections are static within each of them, the duration of a single time slot is $\sim$ 15 minutes, when applied to the real time scale. Later on in the section, we show in details how this value is obtained.

In STK tool, once the location of the gateway and the constellation parameters are set, we can obtain, for each terrestrial gateway, the description of the access over a defined amount of time, for all satellites. In this case, Figure 5 shows the time-dependent connectivity of the terrestrial gateway, located in Hawaii, with the entire constellation.

The graph is taken from 12 hours of observation. From Figure 5, considering the time on the x-axis, it can be noticed that for approximately 15 minutes, the network remains static. Therefore, we assume the duration of each time slot equal to 15 minutes. The obtained value is the result of a trade-off between the number of time slots and the precision of their duration with respect to the real network behavior. In fact, from 5, it can be deducted that, in reality, the network is not static for exactly 15 minutes. A rounding to the closest multiple of 15 has been applied. For example, if the connection stops at 15.55, it will be counted as 16.00. While if it stops at 15.50, it will be counted as 15.45. While this might seem introducing an error $\in [0, 5 - 7]$ minutes, it should also be noticed the following. Each gw-sat connection lasts for 6 time slots, which correspond to 2 hours, 7200 seconds. In the reality, each sat-gw connection lasts for $\in [7023, 7029]$ seconds, therefore a maximum error of 177 seconds is introduced. This is due to the fact that whenever there is a rounding in one time slots, this is compensated from the rounding of the following slot. In other words, if the connection is between 15.55-16.10, this is reported as 16.00-16.15, which still keeps the 15 minutes duration.

In total, the simulation lasts t_{sim} = 72 time slots, which correspond to 18 hours of real time. As also highlighted above, the duration of the time slot is extremely dependent on the substrate network. However, the proposed VNE algorithms are not dependent on the real duration of the time slot and, in addition, a time slotting strategy with variable time slot duration would still be applicable. Figure 5 depicts the time-dependent connection for only one terrestrial gateway and it can be obtained for every terrestrial gateway. It is relevant to underline that, if we project the simulated time slots to a real case, intuitively, as we have just shown, the duration in minutes is dependent on the considered satellite network. This real duration of the time slots will be relevant later on in this paper for the simulation results and the final discussion due to the following reason. The proposed algorithm is thought to be applied to a variety of dynamic networks such as LEO and MEO, which differ especially for the amount of time the status of the satellite network can be considered static. In fact, if we consider a MEO constellation, the link connectivity between a terrestrial gateway and a MEO satellite can remain up to 2 hours. When the orbit of the constellation is considerably lower (LEO), the duration of the time slot decreases significantly, even up to few minutes, depending on the SatCom related parameters such as the orbit's altitude.

From the terrestrial connections point of view, we consider that the gateways are also connected via terrestrial links (no variations over time) among themselves as a mesh fully-connected network. The available capacity of both terrestrial and satellite links is initially set equal to 300 Mbps. The source and destination nodes of each VNR are randomly selected among the substrate nodes to consider the worst possible scenario, i.e. when a priori prediction is not available.

The two following different cases for the source-destination random generation are being considered:

- **Case 1**: the source is randomly selected among the satellites and the destination is randomly selected among the terrestrial gateways;
- **Case 2**: source and destination are randomly selected among the entire set of substrate nodes.

The two use cases are selected to analyze the efficiency of the proposed algorithm, varying the diversity among the type of VNRs. In particular, use case 1 selects randomly the source and destination nodes which belong to two different segments,

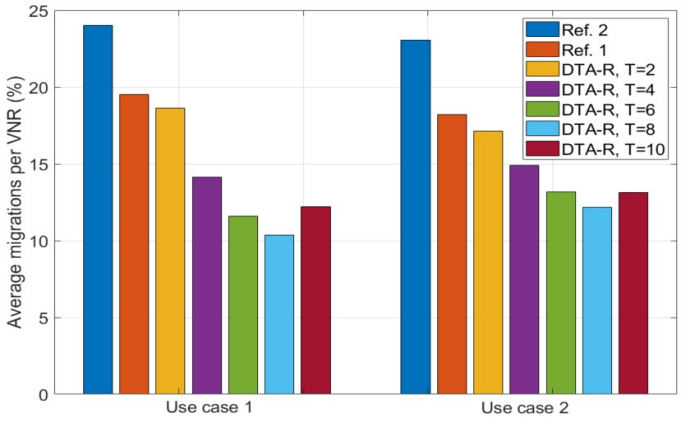

Figure 6. Average migration cost per VNR for use cases 1 and 2

one satellite and one terrestrial. In this scenario, migrations are more likely to happen. In use case 2, instead, nodes belonging to the same layer might also be selected, thus, migrations might not be involved by default and the efficiency of this approach would reduce with respect to a greedy approach. For the subsections VII-B and VII-C, a comparison between these use cases is presented while, for the results in subsection VII-D, only use case 1 is considered since it was proved to be the most interesting scenario for applying DTA-R algorithm. The demanded data rate by each VNR is randomly generated in the interval [40,100] Mbps while two different values of latency requirement are randomly selected among [30, 1000] ms. The propagation delay $p(u, v)$ is set to 27 ms for the ground-sat links, 1 ms for ground-ground links and 3 ms for sat-sat links. The parameter h is fixed to 10. We compare DTA-R to the following online baseline algorithms:

- **Ref. 1: Shortest path** [23];
- **Ref. 2: Load Balancing** [16].

Both algorithms are also considering a node mapping optimization which is not taken into consideration in this paper, thus they have been adopted to only link mapping optimization.

B. Average migration cost per VNR

In this first simulation result, the metric is the average percentage of migrations that each VNR experiences, during its lifetime. The results are shown for both cases, mentioned in the previous subsection. From Figure 6, several considerations can be done.

First of all, it is important to remind that the metric *average migration cost* is expressed in percentage because, as stated previously, we refer to the migration cost as the ratio between the number of migrations, experienced by each VNR, over the lifetime. For example, if we analyze one single VNR and we assume its lifetime L time slots, it means that there are L transitions over time where the traffic might have to be migrated due to topology variation. If we assume that n migrations have occurred, the migration cost would be n/L. Then, at the end of the simulation, we take the average migration cost among all VNRs.

The proposed algorithm is compared to the baselines presented in VII-A. As mentioned above, the DTA-R is based on a planning of a defined number of time slots T. Accordingly, in the results, different values of T are analyzed, 2, 4, 6, 8 and 10. Figure 6 shows, in both use cases, that the average migration cost per VNR obtained with DTA-R is similar or considerably lower than the baselines' one, for all values of T. In addition, the more the value of T, the more DTA-R is efficient because the path with less migrations, over a longer period of time, is selected.

Furthermore, a comparison between the two use cases can be highlighted. Figure 6 depicts, on the left side, the use case 1 where migrations are more likely to happen while, on the right side, use case 2, where VNRs can be generated even among terrestrial nodes. It can be noticed that the benefit of DTA-R for use case 1 is slightly higher, with respect to the baselines, than in use case 2, with $T > 2$. For example, when $T = 4$, for use case 2, DTA-R gains $\sim$ 36 % with respect to baseline 2, while, on the opposite, in use case 1, DTA-R gains $\sim$ 41.2 %. However, even in the latter case, an advantage of DTA-R can still be appreciated, but the benefit is slightly lower than use case 1, due to the less dynamicity of the scenario. It has also been noticed that for values higher than T = 8, the rounding procedures is more likely to fail, falling into a shortest path approach, which worsens the performance on average. While this might seem a limitation, it is worth underlining the following. Despite the type of algorithm or substrate network, when dealing with dynamic scenarios, such as NGSO, there should be an upper-bound in the number of priori planned time slots. In fact, it makes sense to plan in advance for some time, exploiting the priori knowledge, but it does not make sense to plan too much in advance, because the more distant the future, the more difficult to predict the status of the network. For example, in a MEO constellation, considering 6 or 8 planned time slots, it corresponds to a planning of already $\sim$ 2 hours.

C. Computation time

The second performance evaluation considers the computation time, which is a very critical analysis in this field. As depicted in Section V-C, the complexity grows exponentially with the number of planned time slots T, therefore, the computation time is expected to be very sensitive to the increase of T. As shown in Figure 4, the DTA-R considerably reduces the computing time, and it allows to consider values of $T = 6$ or 8, without introducing long delays. Despite the type of DTA utilized, we should not expect lower values than baselines' ones, but, rather, the purpose of this comparison is to give an idea on which could be the ideal value of T, while keeping the solution time reasonable.

Table II shows that, in both use cases, the computation time grows exponentially with T, as expected. For use case 2, the reason why there are overall lower values, is related to the fact that the computation time is an average over the simulation time. For the VNRs generated among terrestrial nodes, the computation time is almost none, since it is easier to find paths with no migrations, which reduces the overall average. This situation does not happen in use case 1.

Table II
AVERAGE COMPUTATION TIME PER SINGLE VNR EMBEDDING

Algorithm	Case 1	Case 2
Ref. 1	0.0056 s.	0.0013 s.
Ref. 2	0.0201 s.	0.0148 s.
DTA-R, T=2	0.0026 s.	0.0069 s.
DTA-R, T=4	0.26 s.	0.24 s.
DTA-R, T=6	0.55 s.	0.44 s.
DTA-R, T=8	0.94 s.	0.71 s.
DTA-R, T=10	1.74 s.	1.62 s.

The evaluation on which is the right value of T, significantly depends on the scenario. For example, for delay-tolerant applications, such as multimedia content download, even 2 seconds per computation might still be a reasonable value. On the opposite, for those less delay-tolerant applications, this delay might not be reasonable anymore. Thus, the selection of the right value of T is very much use-case dependent.

It is worth underlining that the computation time metric has a relevant importance for another reason. If we consider that the algorithm takes, on average, the computation time depicted in the Table II, this means that if there is a queue of tens of VNRs, waiting to be embedded, the last VNR(s), might experience a long delay because the algorithm is embedding in a sequential way. Obviously, VNRs might be re-arranged in the buffer following criteria such as priority, maximum allowed latency, etc. but it is still very important to control the computation time, considering the number of VNRs to be embedded. For the following performance evaluations, $T = 8$ is considered since, overall, it seems to be the most reasonable value.

D. Average network load over time

In this section, we evaluate the average substrate link load over time. This is a common metric, also known as load balancing, used to evaluate how the traffic is spread in the network. Indeed, the average substrate link load metric takes the capacity utilization of every substrate link (percentage) and computes the average over all links. For the comparison, the load balancing algorithm is considered as benchmark because it provides a very good solution.

It is important to underline that this metric is quite relevant in current transport networks. A low average value of utilization is equivalent to say that the traffic is fairly spread which also means that unexpected peaks of traffic might be more easily managed in the network, with lower probability of bottlenecks. In this context, the performance evaluation of this subsection aims to prove that the proposed approach provides reasonable results even in terms of load balancing. For the comparison, we fix some parameters for DTA-R such as the planned time slots $T = 8$.

Obviously, considering the average substrate link utilization as the metric, worse results of the DTA-R algorithm, compared to load balancing, are expected for any level of planned time slots. However, the promising result, which can be noticed in Figure 7, is that the DTA-R algorithm provides values that follow the same trend of the load balancing ones, keeping the distance, with the load balancing graph, always a reasonable

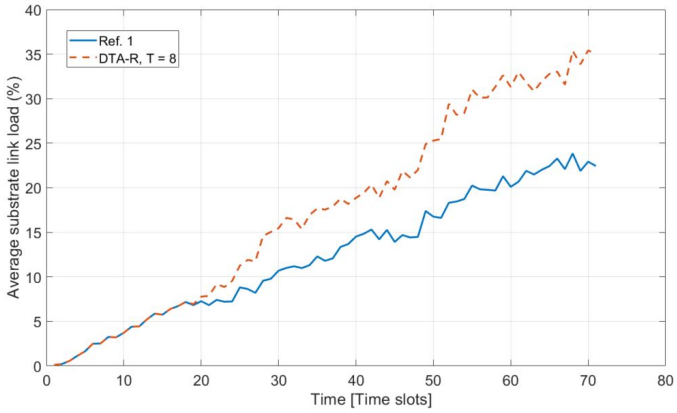

Figure 7. Average substrate load vs time

Table III
AVERAGE PATH LENGTH (HOPS)

Algorithm	Case 1	Case 2
Ref. 1	2.76	3.09
Ref. 2	2.80	3.12
DTA-R, T=2	2.79	3.12
DTA-R, T=4	2.83	3.19
DTA-R, T=6	2.93	3.33
DTA-R, T=8	3.49	3.98
DTA-R, T=10	3.59	3.86

value. The more the time passes, the more the accommodated VNRs, and the higher the probability that DTA-R deviates from the better solution (Load Balancing). This is expected because indeed, the objective of DTA-R is not to reduce the average substrate load. In addition to that, the VNR is an end-to-end connection, which gives less degree of freedom to the embedding algorithm, if compared to cases where VNRs are composed by several links and nodes. If we consider for example, larger size networks, such as a LEO constellation instead of MEO, the story might be different and the difference might be larger. In this context, a joint (load balancing and migrations minimization) objective function could be considered, where complementary weights are given to the two objectives and different combinations of the weights can be studied in order to keep also the load, over the network, under control.

E. Average path length per VNR over lifetime

In this subsection, we analyze the average path length for the baselines and the DTA-R. As for the analysis presented in VII-D, this performance evaluation does not foresee to improve the results obtained by the Ref. 1 algorithm, which is a pure shortest path approach, but rather to show that the solutions of DTA-R do not explode, not only in average load, but also in terms of path length.

In fact, Table III shows that the average path length for every considered T value, is always not very far from the lowest one, i.e. Ref. 1. Despite this, a trend can be highlighted. The more the planned time slots, the higher the average path length. The reason for this result is linked to the ability of the algorithm to find static links over T planned time slots. For intra-segment VNRs (from satellite to terrestrial node), it

is more likely to have longer paths in order to minimize the difference between consecutive time slots rather than shortest paths but more dynamic over time. This is an additional reason why it is important to do not increase too much the number of planned time slots.

VIII. Experimental results in the built-in SDN-based testbed for VNE algorithm

In this section, we test DTA-R and the baselines algorithm into our testbed MIRSAT. The aim is to use MIRSAT to analyze the impact of the migrations on the real traffic, which cannot be easily simulated in Matlab. Initially we run Matlab simulations in an ideal case and we focus on the optimization aspect, showing the benefit of priori planning. However, simulations lack of precise measure of the traffic lost per each migration (which is not necessarily always the same). In fact, MIRSAT aims to give a precise description of the traffic lost per each migration. MIRSAT is an SDN-enabled testbed, with a centralized entity, the SDN controller, and a dynamic substrate network, i.e. the satellite network. SDN is a useful technology to test mapping algorithms, such as VNE, because it enables an easy translation of mapping decisions into forwarding rules, also known as flow rules. Deploying the routing control externally in the SDN controller, brings to a simplification of the network nodes, which forward the traffic according to the pre-computed path, instead of internally running routing algorithms. This relevant feature matches very well with the satellite context, where the majority of the network nodes are satellites, i.e. nodes with limited resources. Consecutively, satellite nodes are seen as Layer-2 enabled switches.

A. Experimental testbed setup

We developed MIRSAT [33] as an enhanced version of the testbed that we proposed in [34]. In [34], the testbed has three main software components. The substrate network, emulated in Mininet [35], where the nodes are OpenFlow enabled switches, the RYU Software Defined Networking (SDN) controller [36] which has full control of the Mininet switches and the VNE algorithm in Matlab, directly connected to the SDN controller. MIRSAT is a more complete testbed, where we additionally propose the integration of the STK toolkit, connected to Mininet (Figure 8). This brings flexibility to the testbed, allowing to emulate, in Mininet, any satellite scenario that is simulated in STK. The testbed is fully running in a virtual machine with 10 GB of Random Access Memory (RAM), 1 Central Processing Unit (CPU) and 50 GB of storage capacity. In fact, once the scenario is defined in STK, the description of the substrate network topology and its temporal variations are translated into Mininet code.

The VNE algorithm retrieves the network topology at any time through the Representational State Transfer (REST) Application Programming Interface (API) of the SDN controller, and uses it as substrate network to embed any incoming VNR. Once the VNE problem is computed, the output is the link mapping of the arrived VNR, which is translated, by the SDN controller, into flow rules, and sent, through the OpenFlow

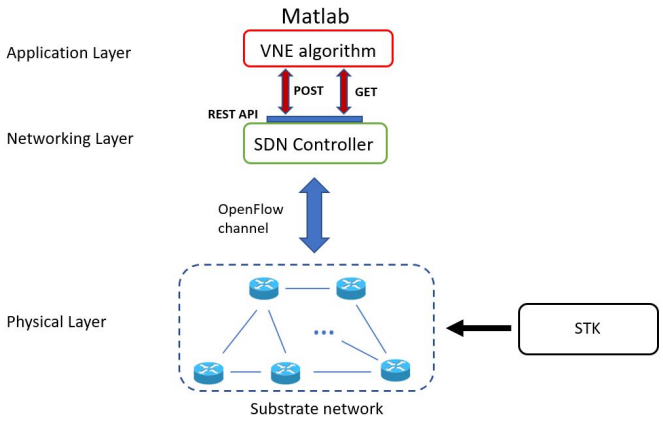

Figure 8. Architecture of MIRSAT

channel, to the switches involved in the link mapping (source, destination and intermediate nodes). MIRSAT emulates the scenario described so far. This means that Mininet emulates the integrated MEO/terrestrial network, with properly emulated latency for MEO links. RYU controls each node (terrestrial and satellite) and, in Matlab, we run the three different algorithms, to test the performance. In order to make the emulation faster, we consider that each time slot lasts for 60 seconds, instead of 15 minutes (real value).

B. Migration cost

In this subsection, we analyze the traffic of a single VNR, over 8 time slots, which correspond to 7200 s. in the reality. In order to simplify the presentation of the results from the testbed, we reduce the duration of each time slot to only 60 s., instead of 15 minutes. This helps to have a good understanding of the transition time and the traffic lost when entering in a new time slot. In fact, if we kept the real duration, few seconds out of 15 minutes, would have not been clearly noted. In addition, the difference between DTA-R and baselines can be appreciated more easily. We use UDP traffic via *iperf* to test the connection between source and destination nodes of a randomly selected VNR, among the arrived ones (Figure 9). As for theoretical simulations, the DTA-R with $T = 8$ is compared to Ref. 1 and Ref. 2 algorithms. Figure 9 shows the throughput behavior experienced by the VNR during the considered time, for the three different algorithms. To ease the comprehension, we have drawn a continuous line, external to the graph, with the description of the time slots. The drops of throughput to values close or equal to 0, correspond to an ongoing migration of traffic. In this scenario, it is evident that a period of outage between consecutive time slot, is experienced. This is due to the SDN controller, which takes few tens/hundreds of second to realize the new links. Once RYU can again see all the links in the topology, the traffic will flow towards the new paths. The outage period varies, depending on the amount of new links that the SDN controller has to discover. RYU can take up to $\sim$ 4-5 seconds to discover all new connections. The testbed results are consistent with the simulated ones. In fact, the average migration cost is considerably higher

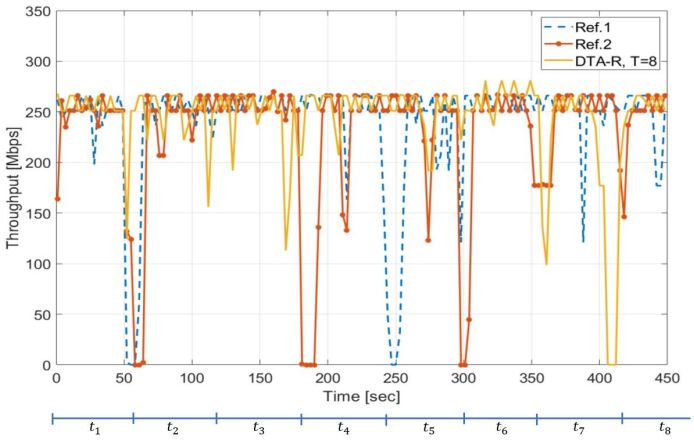

Figure 9. Comparison of throughput obtained via iperf session from one VNR

Table IV
TRANSMITTED VS RECEIVED PACKETS

	Transmitted	Received	loss (%)
Ref. 1	451	430	4.6
Ref. 2	451	420	6.9
DTA-R, T=8	451	442	2.0

while considering the baseline algorithms. DTA-R proved to significantly reduce the migration cost.

In addition, we have run ping sessions between VNR's end nodes to retrieve the precise amount of lost packets and get a final percentage of outage. Table IV provides the outputs from the ping sessions, where one ICMP packet is sent every second, thus, overall $\sim$ 450 packets are sent. The outage percentage achieved while using DTA-R has been reduced by $\sim 2.5 - 5\%$, compared to the baseline results.

IX. CONCLUSION

In this paper, we proposed a formulation for a dynamic and topology-aware VNE algorithm with dynamic substrate network. The outcome of this work can be served as a guideline for performance evaluation of VNE on integrated satellite/terrestrial networks, which is useful to both satellite and terrestrial service providers. In fact, we have considered an integrated terrestrial/MEO satellite substrate network, with also MEO-MEO links. We simulated the real time series of O3b MEO constellation, via STK toolkit. Given a subdivision of the simulation into time slots, where the substrate network could be considered static within each time slot, the DTA-VNE algorithm was able to plan, in advance, the embedding of each VNR for a certain amount of time slots, with the objective of minimizing the migration cost over the planned time slots. Due to the exponential complexity of DTA-VNE with the increase of planned time slots and network size, we have proposed DTA-R, a relaxed version of DTA-VNE. Simulation results have shown that even DTA-R outperforms baseline solutions, such as shortest-path and load balancing algorithms, while considering the migration cost per VNR. We proved that the more planned time slots, the higher the gain of DTA-R. Simulation results have also shown the trade-off between the computation time and the number of planned time slots. This is fundamental to determine the maximum allowed time slots, given the scenario (QoS requirements) and the size of the substrate network. In order to deeper investigate the application and impact of the proposed algorithm, we developed a proof-of-concept SDN-based testbed, MIRSAT, with a proper time-dependent emulation of the substrate network in Mininet. The testbed has shown the importance of avoiding migrations. In fact, we analyzed the traffic of one randomly selected VNR and we proved the outage of service, quantified as lost packets, experienced for each migration.

REFERENCES

[1] M. Chahbar, G. Diaz, A. Dandoush *et al.*, "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.

[2] H. Zhang, N. Liu, X. Chu *et al.*, "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] I. Afolabi, T. Taleb, K. Samdanis *et al.*, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[4] L. Lei, Y. Yuan, T. X. Vu *et al.*, "Dynamic-Adaptive AI Solutions for Network Slicing Management in Satellite-Integrated B5G Systems," *IEEE Network Magazine*, 2021.

[5] O. Kodheli, E. Lagunas *et al.*, "Satellite Communications in the New Space Era: A Survey and Future Challenges," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[6] H. Al-Hraishawi, S. Chatzinotas, and B. Ottersten, "Broadband Non-Geostationary Satellite Communication Systems: Research Challenges and Key Opportunities," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[7] 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.

[8] H. Cao, S. Wu, Y. Hu *et al.*, "A survey of embedding algorithm for virtual network embedding," *China Communications*, vol. 16, no. 12, pp. 1–33, 2019.

[9] 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.

[10] A. Hashmi and C. Gupta, "A Detailed survey on Virtual Network Embedding," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, 2019, pp. 730–737.

[11] A. Blenk, P. Kalmbach, P. van der Smagt, and W. Kellerer, "Boost online virtual network embedding: Using neural networks for admission control," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 10–18.

[12] K. T. Nguyen and C. Huang, "An Intelligent Parallel Algorithm for Online Virtual Network Embedding," in *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2019, pp. 1–5.

[13] C. Aguilar-Fuster, M. Zangiabady, J. Zapata-Lara, and J. Rubio-Loyola, "Online Virtual Network Embedding Based on Virtual Links' Rate Requirements," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1630–1644, 2018.

[14] M. Zangiabady, C. Aguilar-Fuster, and J. Rubio-Loyola, "A virtual network migration approach and analysis for enhanced online virtual network embedding," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 324–329.

[15] P. Zhang, H. Yao, and Y. Liu, "Virtual Network Embedding Based on Computing, Network, and Storage Resource Constraints," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3298–3304, 2018.

[16] 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.

[17] H. Kuang and Y. Fu, "VNE-TS: A novel virtual network mapping algorithm in SDN," in *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, 2016, pp. 657–661.

[18] 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.

[19] C. K. Dehury and P. K. Sahoo, "DYVINE: Fitness-Based Dynamic Virtual Network Embedding in Cloud Computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1029–1045, 2019.

[20] Q. Lin, Y. Huang, and Y. Li, "A New Algorithm for Virtual Networks Reconfiguration with Adaptive Interval," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 2567–2571.

[21] P. Wang, X. Zhang, S. Zhang *et al.*, "Time-Expanded Graph-Based Resource Allocation Over the Satellite Networks," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 360–363, 2019.

[22] T. Zhang, H. Li, S. Zhang *et al.*, "STAG-Based QoS Support Routing Strategy for Multiple Missions Over the Satellite Networks," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 6912–6924, 2019.

[23] D. Yang, J. Liu, R. Zhang, and T. Huang, "Multi-Constraint Virtual Network Embedding Algorithm For Satellite Networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[24] 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.

[25] M. Minardi, S. K. Sharma, S. Chatzinotas, and T. X. Vu, "A Parallel Link Mapping for Virtual Network Embedding with Joint Load-Balancing and Energy-Saving," in *IEEE International Mediterranean Conference on Communications and Networking (MEDITCOM) 2021*, 2021, pp. 1–6.

[26] Q. Lu and C. Huang, "Distributed Parallel VN Embedding Based on Genetic Algorithm," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[27] A. Song, W.-N. Chen, T. Gu *et al.*, "Distributed Virtual Network Embedding System With Historical Archives and Set-Based Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 927–942, 2021.

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

[29] J. Fraire, P. Madoery, and J. Finochietto, "Traffic-aware contact plan design for disruption-tolerant space sensor networks," *Ad Hoc Networks*, vol. 47, 05 2016.

[30] J. Alonso and K. Fall, "A linear programming formulation of flows over time with piecewise constant capacity and transit times," Tech. Rep., 2003.

[31] [Online]. Available: http://lpsolve.sourceforge.net/5.1/absolute.htm

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

[33] [Online]. Available: https://wwwen.uni.lu/layout/set/print/snt/research/sigcom/sdn_lab

[34] F. Mendoza, M. Minardi, S. Chatzinotas *et al.*, "An SDN Based Testbed for Dynamic Network Slicing in Satellite-Terrestrial Networks," in *IEEE International Mediterranean Conference on Communications and Networking (MEDITCOM) 2021*, Athens, Greece, 2021.

[35] [Online]. Available: http://mininet.org/

[36] [Online]. Available: https://ryu.readthedocs.io/en/latest/getting_started.html

Mario Minardi is currently a Doctoral Researcher at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. He received the B.S. in Electronic and Telecommunications Engineering and the M.Sc. in Telecommunications Engineering, from the University of Bologna, Italy, in 2017 and 2020, respectively. His research interests are in the field of software engineering for multi-layer satellite-terrestrial integrated networks, with particular interests in optimization of traffic engineering algorithms for software-defined networking (SDN) and network function virtualization (NFV).

Thang X. Vu (M'15–SM'22) received the B.S. and the M.Sc., both in Electronics and Telecommunications Engineering, from the University of Engineering and Technology, Vietnam National University, Hanoi in 2007 and 2009, respectively, and the Ph.D. in Electrical Engineering from the University Paris-Sud, France, in 2014. In 2010, he received the Allocation de Recherche fellowship to study Ph.D. in France. From July 2014 to January 2016, he was a postdoctoral researcher with the Singapore University of Technology and Design (SUTD), Singapore. Currently, he is a research scientist at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. His research interests are in the field of wireless communications, with particular interests of the applications of Machine Learning and Optimization Theory in design, analyze and optimize multilayer 6G Networks. He was a recipient of the SigTelCom 2019 best paper award and has served as an Associate Editor of the IEEE Communications Letters since 2022.

Lei Lei (S'12-M'17) is currently associate professor at Xi'an Jiaotong University, School of Information and Communications Engineering. He received the B.Eng. and M.Eng. degrees from Northwestern Polytechnic University, Xi'an, China, in 2008 and 2011, respectively. He obtained his Ph.D. degree in 2016 at the Department of Science and Technology, Linköping University, Sweden. He was with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg as a research associate and research scientist from 2016 to 2021. He was a research assistant at Institute for Infocomm Research (I2R), A*STAR, Singapore, in 2013. His current research interests include resource allocation and optimization in terrestrial-satellite networks, energy-efficient communications, and deep learning in wireless communications. He received the IEEE Sweden Vehicular Technology-Communications-Information Theory (VT-COM-IT) joint chapter best student journal paper award in 2014. He was a co-recipient of the IEEE SigTelCom 2019 Best Paper Award.

Christos Politis is Senior Communications Systems Engineer within SES Technology. Prior to joining SES in January 2018, he was PhD Researcher at Interdisciplinary Centre for Security, Reliability and Trust (SnT) within University of Luxembourg in 2014-2017, and Satellite Communications Engineer at Centre National d' Etudes Spatiales (CNES), France in 2012. He has technical experience in national and international RD and Innovation projects on ICT with major focus on Satellite Communications and he has authored several journals, conference and book chapter publications. In particular, he has been contributing to the 5G related projects H2020 Work Program 2018-2020 "iNGENIOUS", H2020 5G PPP Phase III "5G-VINNI", H2020 5G PPP Phase II "SaT5G", ESA ARTES "5G-EMERGE", ESA ARTES "SATis5" and ESA ARTES "EdgeSAT", as well as to the PPDR related national funded study "ENGINE". He obtained his PhD degree in Spectrum Monitoring Algorithms for Wireless and Satellite Communications from the University of Luxembourg, Luxembourg in 2018, his MSc. degree in Communications and Signal Processing from the University of Newcastle, UK in 2014, his MSc. degree in Space Communication Systems from ISAE-SUPAERO, France in 2012, and the Dipl.-Eng. degree in Electrical and Computer Engineering from the University of Patras, Greece in 2011.

Symeon Chatzinotas [S'06, M'09, SM'13, F'23] is Full Professor and Head of the SIGCOM Research Group at SnT, University of Luxembourg. He is coordinating the research activities on communications and networking across a group of 80 researchers, acting as a PI for more than 40 projects and main representative for 3GPP, ETSI, DVB. He is currently serving in the editorial board of the IEEE Transactions on Communications, IEEE Open Journal of Vehicular Technology and the International Journal of Satellite Communications and Networking. In the past, he has been a Visiting Professor at the University of Parma, Italy and was involved in numerous RD projects for NCSR Demokritos, CERTH Hellas and CCSR, University of Surrey. He was the co-recipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award and Best Paper Awards at WCNC, 5GWF, EURASIP JWCN, CROWNCOM, ICSSC. He has (co-)authored more than 700 technical papers in refereed international journals, conferences and scientific books.