

# An SDN Based Testbed for Dynamic Network Slicing in Satellite-Terrestrial Networks

Fabian Mendoza

*SnT. University of Luxembourg*  
Luxembourg, Luxembourg  
fabian.mendoza@uni.lu

Mario Minardi

*SnT. University of Luxembourg*  
Luxembourg, Luxembourg  
mario.minardi@uni.lu

Symeon Chatzinotas

*SnT. University of Luxembourg*  
Luxembourg, Luxembourg  
symeon.chatzinotas@uni.lu

Lei Lei

*SnT. University of Luxembourg*  
Luxembourg, Luxembourg  
lei.lei@uni.lu

Thang X. Vu

*SnT. University of Luxembourg*  
Luxembourg, Luxembourg  
thang.vu@uni.lu

**Abstract**—6G networks are expected to meet ambitious performance parameters of coverage, data rates, latency, etc. To fulfill these objectives, the implementation of non-GEO satellite constellations is expected to improve coverage, capacity, resilience, etc. as well as the implementation of new advanced network virtualization algorithms in order to optimize network resources. However, the integration of these technologies represents new challenges, such as the execution of network slicing schemes in highly dynamic environments and network awareness requirements. In this regard, Software Defined Networking (SDN) is seen as a required 6G technology enabler in order to provide better satellite-terrestrial integration approaches and Virtual Network (VN) implementation solutions. In this paper, we present an experimental testbed for non-GEO satellite constellations integration solution and VNE algorithms implementation adapted to highly variable network conditions that builds upon SDN. A laboratory testbed has been developed and validated, consisting in SDN-based satellite-terrestrial dynamic substrate network emulated in Mininet, a Ryu SDN controller with an End-to-End (E2E) Traffic Engineering (TE) application for the VNs establishment and a Virtual Network Embedding (VNE) algorithm implemented in Matlab.

**Index Terms**—Hybrid Satellite-Terrestrial Networks, Network Slicing, Software Defined Networking.

## I. INTRODUCTION

The upcoming 6G networks foresee ambitious network performance parameters (e.g. data rates  $>Tbps$ , lower latency  $<1ms$ , reliability of 99.9999%, etc.), network capabilities (e.g. 100 times higher connection density, more intelligence for full automation, sub-centimeter geo-location accuracy, near 100% coverage, etc.) and new business models over its predecessor (e.g. Network Slicing, etc.) [1] [2]. To fulfill these objectives, is required the incorporation of new sets of technologies such as non-GEO satellite mega constellations to improve coverage, capacity, resilience, etc., and the implementation of new advanced algorithms (e.g. AI-based) for advanced network virtualization schemes in order to optimize network resources and to improve the Quality of Service provisioning in terms of data rates, latency, etc. [3].

In this sense, in recent years there is a clear trend of various disruptive initiatives that promote the use of non-

GEO satellite constellations with a large number of low-cost microsatellites to reduce cost and latency [4] [5]. Due to the high dynamicity of medium and low satellite orbits, these new satellite network configurations, however, require advanced traffic distribution algorithms able to be adapted to changing network environment conditions for an efficient use of network resources.

On the other hand, 6G networks are also expected to further boost the use of new virtualization schemes [6]. Network virtualization is defined in [7] as the ability to manage and prioritize traffic in portions of a network that might be shared among different external networks. In this sense, network slicing (NS) is a more particular case of network virtualization. A NS definition is presented in [8] as an E2E logical network/cloud running on a common underlying infrastructure, mutually isolated, with independent control and management that can be created on demand. This dynamic network resources customization, including multi-tenant capability through network slicing, can be assigned to differentiated services (e.g. internet of things, industry automation, cellular vehicle to everything, mobile broadband, etc.), given their QoS requirements (e.g. very low-latency, energy efficient systems, etc). Potential benefits of Network Slicing (NS) include decreased energy consumption, capital expenditure (CAPEX) reduction, differentiated QoS provisioning, etc. [9].

The calculation of the optimal distribution of network resources among VNs refers to Virtual Network Embedding (VNE) problem. Given the physical (substrate) network and the virtual networks/services requirements described as a graph of virtual node and virtual links each, the aim of VNE problem is to compute the embedding of virtual node into physical node and virtual links into physical links. Due to its NP-hardness complexity [10], the VNE problem is commonly divided into two sub-problems: Node Mapping and Link Mapping. To do that, for fixed network, the algorithm inputs such as physical topology, node and link resources are usually known in advance and each time a new VN request arrives, VNE computes the embedding and the

available physical resources are updated. However, the VNE algorithm doesn't need any update of the physical topology since it is fixed. For dynamic network models (e.g. hybrid non-Geo satellite constellations and terrestrial networks), however, virtual network mappings must also consider real time network topology's changes. In this context, 6G networks require to incorporate the recently developed paradigm Software Defined Networking (SDN), whose main key principles lie on the separation of the control and user plane, centralized control and programmability. SDN can play the role of facilitator in the implementation of Virtual Network Functions (VNF) for dynamics satellite-terrestrial networks through a higher level of programmability and network awareness capability, providing flexibility, agility and dynamism for an intelligent delivery and deployment of services [9]. However, despite many research works consider indispensable the use of the SDN/NFV concepts in 5G networks and beyond [11], there are only some isolated theoretical works on the advantages obtained from these solutions in hybrid satellite-terrestrial networks.

This paper presents the development of an SDN-based testbed to validate the feasibility of SDN implementation, assessing the VNE algorithms performance in real time under a highly dynamic environment for hybrid satellite-terrestrial backhaul networks. The SDN-based testbed executes dynamically the installation and elimination of VNs, maximum rates limiters per VN, dynamic network topology readings (for any number of nodes and links), network status changes reports and real time network statistics to feed the VNE algorithm. The testbed consists of three main elements: An OpenFlow-based substrate network running in Mininet, emulating the hybrid satellite-terrestrial mobile backhaul network; an external SDN RYU controller, settling the VNs on demand and; the VNE algorithm running in a Matlab application. Different validation tests for the automatic VNs establishment and adaptability to the dynamic network conditions are presented.

The rest of the paper is organized as follows. Section 2 briefly reviews the main research studies and projects for architectural traits of the satellite-terrestrial SDN-based integration solution and VNs implementations. Section 3 describes the SDN solution for Network Slicing implementations and the implemented SDN-enabled hybrid satellite-terrestrial testbed, detailing its components and configuration settings as well as the structure and logic of the programmed E2E TE for VNs applications. On this basis, the testbed operational validation is provided in Section 4. Finally, Section 5 draws the conclusions.

## II. RELATED WORKS

Satellite networks traditionally had been combined with legacy terrestrial networks as independent systems, generating deficiencies in terms of interoperability, scalability, programmability, which is difficult to establish virtualization schemes dynamically for this type of networks (see section III). In recent years, there have been works which employ SDN as part of seamless terrestrial-satellite integration and

applicability of virtualization schemes point to a federated control capable of controlling heterogeneous network segments that involves multiple network infrastructures (e.g. 5G and next generations) [12]- [14]. Under this premise, an important advance has been carried out in recent years with regard to the analysis of the potential use cases, requirements, and definition of functional frameworks for the exploitation of SDN/NFV technologies in satellite networks [15]- [18]. [15] investigates the advantages of introducing network programmability and virtualization using SDN and/or NFV through the analysis of use cases as well as their impacts on a typical satellite system architecture. Other works have presented developments that include network architecture designs for the exploitation of SDN/NFV technologies for the seamless integration of satellite-terrestrial networks [17]. In the context of the European H2020 Virtualized Hybrid Satellite-Terrestrial Systems for Resilient and Flexible Future Networks (VITAL) research project [18], a feasibility study of different functional splits for the virtualization of a satellite gateway and developed a generic functional architecture for satellite ground segment systems embracing NFV and SDN technologies was delivered. In this respect, an overview of the current 5G initiatives and projects followed by a proposed architecture for 5G satellite networks, where the SDN/NFV approach facilitates the integration with the 5G terrestrial system is provided in [19] which also analyses a novel technique based on network coding for the joint exploitation of multiple paths in integrated satellite-terrestrial systems. Other works and projects address the topic of network slicing over integrated satellite-terrestrial 5G networks. 5G-VINNI [20], directly addresses satellite integration in 5G networks from the point of view of highly dynamic and flexible network architectures, service deployment and testing, to create new technical and commercial service deployment models, enabling virtualized functions from the network and service layer. Other research works have shown the applicability of these SDN/NFV concepts in mobile networks in a practical way, through developments of testbeds or PoCs [21]- [25], however, none of them was designed to address high dynamic networks.

## III. SDN-BASED SOLUTION FOR VIRTUAL NETWORKS IN SATELLITE-TERRESTRIAL NETWORKS

Traditionally, virtualization in legacy network focused mainly on Virtual Local Area Networks (VLANs) or virtual private networks (VPN). The process consisted of establishing overlay networks, where a small set of nodes use tunnels to form their own topology on top of a legacy network. This made through labeling packets, encapsulating and sending them through the network and decapsulating them at the other end (e.g. MPLS networks). This process involves several disadvantages, such as the addition of headers to packets which reduces their efficiency, networks manually configured by administrators, and the need for additional network equipment running specific protocols which increment the costs, etc., finally hindering the integration of new technologies. In contrast, considered a technological enabler for network virtualization,

SDN through APPs running in a centralized controller, can install flows (routing tables) automatic and dynamically in the switches controlling the management of packets. Using the OF protocol, the SDN controller can add, update, and delete flow entries in flow tables in an OF switch. Each flow entry consists of match fields, counters, and a set of instructions and actions (e.g. packet forwarding actions) to apply to matching packets. Matching rules can be configured based on multiple fields such as ingress port, source/destination MAC-IP addresses, etc. and/or a combination of them. Then, through installed flows, each arrival packet can be identified with the VN it belongs to, assigning routing schemes and QoS level (e.g. queue priority, etc.) according to their respective VNs SLA. In this regard, SDN through Openflow protocol allows us to implement QoS frameworks from a different points of view; (1) One of the basic mechanisms for assigning certain QoS levels (as in legacy networks) is by assigning different priorities on switch port queues (attached to ports which the flows are forwarded to). The configuration protocols such as Open vSwitch Database Management Protocol (OVSDB) that manage Open vSwitch (OVS) implementations are used to create, configure and delete queues for QoS purposes and; (2) resource reservation mechanisms or the implementation of per-flow meters for the establishment of rate limiters. These OpenFlow capabilities together with the centralized control, allow us to play with several matching arguments combinations, as well as characteristics of the state/performance of the network in a given moment (e.g. congestion, link failures, changes in the topology, type of traffic in the network, etc.) in order to dynamically establishing the routing and QoS schemes for each VN.

### A. Experimental Testbed

A high level view of the experimental testbed is depicted in Fig. 1. The objective of the testbed is to evaluate the performance of VNE algorithms in real scenarios, as well as the feasibility of SDN for dynamic VNs implementation. The testbed comprises a mininet network emulator for the SDN-enabled hybrid satellite-terrestrial substrate network representation. The substrate network is built upon OpenFlow switches, which stand for a particular realization of the generic network elements. Between the switches there are two possible connectivity links; Satellite or Terrestrial differentiated by the assigned link latency. Any link in the substrate network can be programmed to be periodically modified (links up to down or viceversa) in order to represent the typical non-GEO satellites orbital movements from a terrestrial fixed point of view. The testbed implementation consists on a PC hosting the Mininet emulator, and another PC hosting the external Ryu SDN controller and the VNE algorithm script running in Matlab. During the emulation, the VNE algorithm randomly creates VN requests arrivals with a poisson arrival distribution, each arrival triggers the recalculation of the assigned resources (embedding) for each VN in the system. Each embedding result consists in an information matrix where each matrix row contain the embedding information of each VN. This

information are: the two end nodes (with host connection capability), the time that each VN will be active in the network, maximum rate per VN and the list of nodes belonging to each VN. Based on the embedding information, the TE application, programmed in the SDN controller, creates the path for each VN, set the rate limiters, obtains the network information statistics, reads the topology as well as receives and process any network topology modification in real time. A more detailed description of the testbed components is addressed in the following, including the configuration settings used to carry out the operational validation in Section 5.

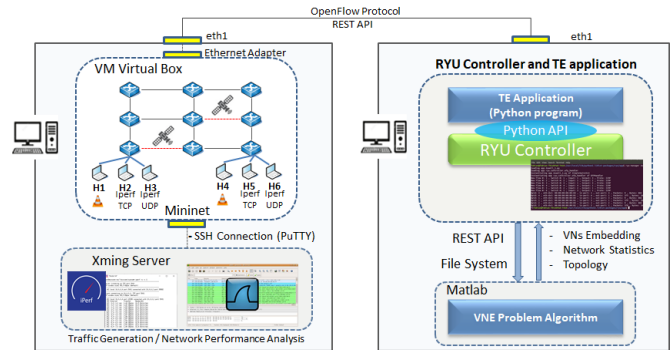


Fig. 1. Illustrative view of an SDN-based testbed.

### B. Virtual Network Embedding Algorithm (VNE)

For the embedding process, each VN is considered as a simple end-to-end service, composed of 2 nodes and 1 link. The node mapping of the two end-points is defined a priori, thus, only the link mapping remains to be computed. We use an Integer Linear Programming (ILP) formulation for the link mapping with the well-known Load-Balancing objective function. Load-Balancing is a well-investigated objective function for VNE and it aims to reduce as much as possible the sum of the bandwidth utilization across the physical network. On top of that, we compute the link mapping in a parallel way among the entire set of VNs. In other words, the Load-Balancing function is not computed for each VN in a sequential way, but only once, considering the entire set of VNs together. Furthermore, we consider a not-splittable path embedding. This means that, given a VN, its link will be embedded in one and only one physical path between the two end-points. The selected path is then a combination of multiple physical links, given the fact that we do not assume virtual nodes embedded in adjacent physical nodes. The parallel computation will further decrease the average physical link utilization. The Virtual Network Embedding problem is executed in Matlab with the help of the GNU Linear Programming Kit (GLPK). GLPK is one of the most common mathematical solvers for VNE problem which provide a solution, in a reduced amount of time using branch-and-cut method. The Matlab script also instantiates the respective node mapping, reads in real-time the physical topology, runs the link mapping computation and sends the embedding results to the SDN Controller.

### C. SDN-Based TE application for VNs

The OF switches operation are controlled by a Ryu SDN controller, which is a component-based software fully written in Python. The Ryu controller exposes application programming interfaces (API) for deploying network management and control applications as Python scripts. Such APIs are a collection programming libraries that give access to the previously mentioned set of mechanisms supported by OpenFlow protocol. The OF switches information as network topology, network state (e.g. switch status, port status, traffic load per port/flow, etc.) and flow table information (e.g. flows information) are visible at application level. The exposed API capabilities, allows us to program and demonstrate the operation of an SDN-based TE application able to set dynamically the embedded VNs in the substrate network and to deliver the input information to VNE algorithm module. More specifically, the implemented SDN-based TE application is able to (1) learn the network topology, (2) real time monitor the network/port status, (3) create monitoring network statistics, (4) identify the VNs new traffic based on a set of user information (e.g. origin IP address) at input/output ports, (5) set the forwarding path based on a VN embedding information by populating the flow tables across the OF switches, and (6) enforce the maximum rate per VN by establishing meter\_ids linked to each match condition. An illustration of the implemented SDN-based TE application is given in Fig. 2, showing its main internal organization and the exploited APIs.

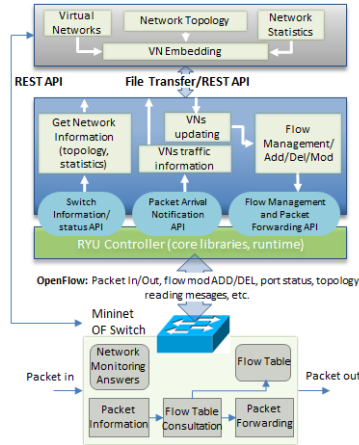


Fig. 2. Illustration of the implemented SDN-based TE.

## IV. OPERATIONAL VALIDATION

The testbed operation is validated through the execution of three illustrative examples. The first one, referred to as the VNs installation validation, shows how the implemented SDN-based TE application is able to enforce a desired routing scheme over the satellite-terrestrial network each time a new VN is embedded. The second one, validates the dynamic VN reconfiguration each time the VNE is recalculated after a network topology change occurs due to a typical non-GEO satellite constellation movement. Finally, the third example

validates the dynamic configuration for a full network with several established VNs after an unexpected link failure occurs and how is handled by the different testbed modules (Mininet substrate network, SDN controller and VNE module). For demonstration purposes, we consider the substrate network represented by Fig. 3. The physical network is composed by 3 types of OF switches representations: the switches 13 to 16 represents MEO satellites, the switches 4, 5 and 9 represents the backhaul NEs with satellite link availability and the rest represents the access NEs with hosts connectivity. The available bandwidth on terrestrial links are settled to 800 kbits/s and the available bandwidth on satellite link as 400 Kbits/s with latency settled at 125ms. Virtual Networks, each one with only one link and 2 hosts.

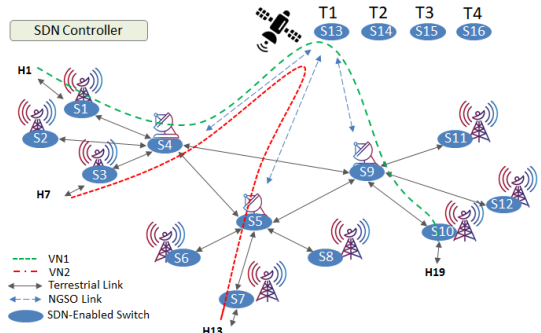


Fig. 3. Network Emulation Scenario.

### A. VNs Implementation

The first example considers the embedded VNs depicted in Fig. 3. An established flow basically consist on a list of match conditions and an linked action, so for the same match conditions cant be assigned different actions. Fig. 4, presents the 4 created flows in switch 13 for VN1 and VN2. From the point of view of switch 13, for the VN 1 and 2, there are two flows arriving at the same input port (s13-eth1) but have different assigned actions (output port=s13-eth3 and s13-eth2). Therefore, to differentiate among VNs, the match conditions also considers the destination address. Then, the TE APP recognizes all packets generated by each Hosts in the network (by its origin IP address), so each packet can be linked to its VN, assigning the right flow action (output port) at each OF switch in order to create the VN path. Then, the same match conditions are linked to a meter\_id in order to stablish the maximum rates per VN.

```
File Edit View Search Terminal Help
mininet@mininet-ubuntu:~$ sudo ovs-ofctl dump-flows s13
cookie=0x0, duration=1.394s, table=0, n_packets=36, n_bytes=2160, priority=65535,d1_dst=01:80:c2:00:00:0e,d1_
type=8XB8C actions=CONTROLLER:65535
cookie=0x0, duration=0.3037s, table=0, n_packets=4, n_bytes=392, priority=1,in_port="s13-eth1",d1_dst=5a:fs:b3
:af:2c:0a actions=output:"s13-eth3"
cookie=0x0, duration=0.5023s, table=0, n_packets=4, n_bytes=392, priority=1,in_port="s13-eth3",d1_dst=e6:09:13
:31:a5:1e actions=output:"s13-eth1"
cookie=0x0, duration=0.5475s, table=0, n_packets=3, n_bytes=294, priority=1,in_port="s13-eth1",d1_dst=c2:84:51
:93:30:1e actions=output:"s13-eth2"
cookie=0x0, duration=0.488s, table=0, n_packets=3, n_bytes=294, priority=1,in_port="s13-eth2",d1_dst=02:30:e6
:9d:37:7f actions=output:"s13-eth1"
cookie=0x0, duration=1.252s, table=0, n_packets=4, n_bytes=392, priority=0 actions=CONTROLLER:65535
```

Fig. 4. Created flows in Node 13 for 2 conexions (VN1 and VN2).

## B. Dynamic Virtual Network Embedding

### a) Dynamic Non-Geo Satellite-Terrestrial Topology:

For the demonstration of the dynamic VN computation/establishment procedure, the testbed is started with the configuration depicted in Figure 3 (T1). The two VNs are embedded through the satellite link. Every 30 seconds the current satellite connection is deactivated (up to down) and the next satellite link is activated (down to up) in order to represent a temporal MEO satellite connections. This process is repeated through switches 13 to 16 consecutively (T1 to T4). After being embedded, the VN provides the basic IP connectivity service between the hosts 1 to 19 (VN1) and between hosts 7 to 13 (VN2). Then, a VLC application is launched so UDP traffic begins to flow between hosts belonging to each VN.

The dynamic process is triggered each time a satellite link status changes (up to down). When this happens, the two involved OF switches send a port status change notification to the controller (OFPT\_PORT\_STATUS message) with the port\_down notification. The detailed message information description is captured with a protocol analyzer is presented in Figure 5. In that particular case, the port s13-eth2 is highlighted, and we can see that the flags OFPPC\_PORT\_DOWN and OFPPS\_LINK\_DOWN are True, which means that actually that port has been shut down. Figure 6 presents the generated traffic by VN 1. Every 30 seconds the throughput goes down to 0 kbit/s or close. Then, the system triggers the new topology reading, and VNE recalculations in order to restore the VN path. The graph also presents the restored traffic after few seconds. It is clear that for this example, link drops or capacity drops are predictable and these abrupt changes (traffic drops) can be avoided by triggering the recalculations before link drops. However, for demonstration purposes we have defined the cut first and then a new calculation. All this information is collected with the help by inspecting traffic on the SDN controller, and with a network protocol analyzer tool.

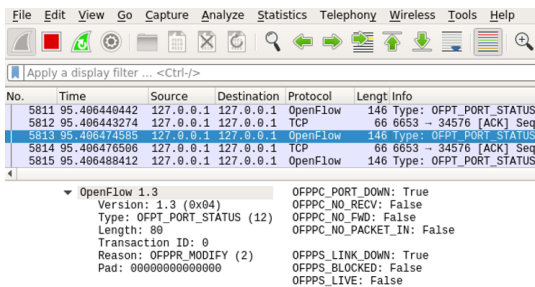


Fig. 5. Port status change notification messages.

b) *Dynamic VNE Re-calculations:* For this example, the scenario consists of 6 VNs embedded, and each VNR requires a bandwidth of 200 kbits/s. This scenario reproduces the same dynamic satellite link change process each 30 seconds of the previous example, representing the temporal MEO satellite-terrestrial backhaul connections. However, after a given period of time, we also reproduce a terrestrial backhaul link failure between the node 4 and 9. Then we observe the system

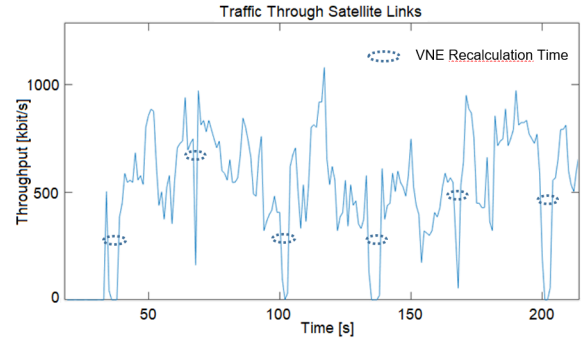


Fig. 6. Traffic load over satellite links.

TABLE I  
VNS CONFIGURATION

VNs Configuration before/after the link failure									
VN	Before Failure				After Failure				
1	1/ H1	4	9	10/ H9	1/ H1	4	5	9	10/ H19
2	2/ H4	4	9	11/ H22	2/ H4	4	5	9	11/ H22
3	2/ H5	4	9	12/ H25	2/ H5	4	5	9	12/ H25
4	3/ H7	4	5	6/ H10	3/ H7	4	5	6/ H10	–
5	3/ H8	4	9	10/ H20	3/ H8	4	13	9	10/ H20
6	6/ H10	5	9	10/ H21	6/ H10	5	9	10/ H21	–

reaction for both satellite and terrestrial links changes. We observe that for the periodic satellite link changes, the VN embedding keeps the same distribution, only changing the consecutive satellite connection. However, after reproducing the terrestrial backhaul link failure there are many VN embedding configuration changes. The embedding configuration is presented in Table I. After the terrestrial link failure, we observe three different possible embedding configurations represented by VN1 (s1-s10), that changes from a terrestrial backhaul path for a different terrestrial backhaul path, the VN4 (s3-s6) keeps the same terrestrial path and VN5 (s3-s10) that changes from a terrestrial backhaul path to a satellite backhaul path. The paths of these three VNs before and after a terrestrial failure are depicted in Figure 7 and 8 respectively. This configuration can also be validated by the measured latency changes on each VN presented in Figure 9. The figure presents the round trip time for each VN before and after the terrestrial link failure (after 70 seconds). Each time there is a change in the topology, the flows in the network are deleted to establish the new VN paths, then the peaks of time each satellite link change or terrestrial link failure occurs represents the time took by the system to create the flows in all OF switches involved in the new VN path.

## V. CONCLUSIONS

This paper has presented an experimental testbed validation of an integration solution based on SDN technologies



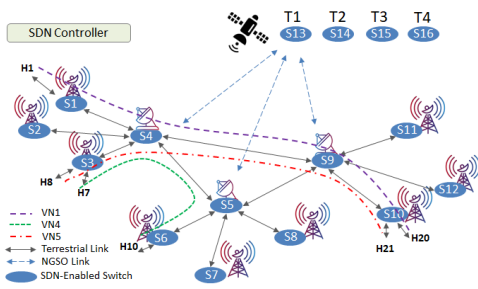


Fig. 7. VNE before terrestrial link failure.

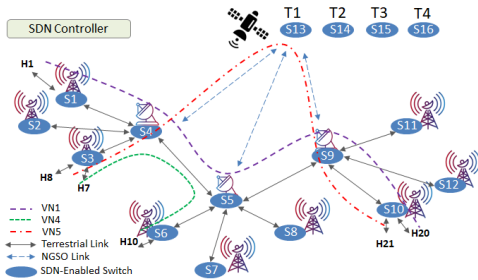


Fig. 8. VNE after terrestrial link failure.

for the realization of VNE implementation to tackle highly dynamic hybrid satellite-terrestrial backhaul networks. The implemented testbed has allowed us to assess the feasibility of the proposed SDN-based integration solution under a practical laboratory. Moreover, we have also validated the versatility of using a high-level language like Python and the existing OF libraries in the Ryu controller for programming the TE application in order to enforce VNE implementations dynamically. The results achieved in this testbed demonstrate the potential and feasibility of applying SDN technologies for improved satellite-terrestrial integration and VN implementations.

## REFERENCES

[1] Shahraki A., Abbasi M., Piran J., Chen M. and Cui S., Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts. *Sci China Inf Sci*, 2021, 64(1): 110301. <https://doi.org/10.1007/s11432-020-2955-6>.  
 [2] Bariah L., et al., A prospective look: Key enabling technologies, applications and open research topics in 6G networks. *IEEE Access PP(99):1-1*. August 2020. DOI:10.1109/ACCESS.2020.3019590.

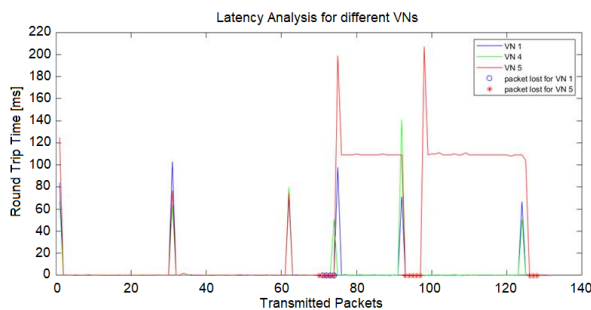


Fig. 9. Latency measurements over VNs.

[3] Amin Shahraki, et al., A Comprehensive Survey on 6G Networks: Applications, Core Services, Enabling Technologies, and Future Challenges. *Networking and Internet Architecture*. 2021-01-29, DOI: arxiv-2101.12475.  
 [4] S programme. ESA announces dedicated support for the development of megaconstellations. <https://artes.esa.int/news/esa-announced-dedicated-support-development-megaconstellations>. Last updated at 02 February 2016.  
 [5] Bhat J., and AlQahtani S., 6G Ecosystem: Current Status and Future Perspective *IEEE access* 2017.  
 [6] Kitindi E., Fu S., Jia Y., Kabir A. and Wang Y., *Wireless Network Virtualization with SDN and C-RAN for 5G Networks: Requirements, Opportunities, and Challenges* Access 2017.  
 [7] B. Prakash and R. Tholeti, *Handbook of Fiber Optic Data Communication: Chapter 16. Hypervisors, Virtualization, and Networking*. 2013, academic Press; 4th edition.  
 [8] J. Ordonez-Lucena, et al., Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges, in *IEEE Communications Magazine*, vol. 55, no. 5, 2017, pp. 8087.  
 [9] Boero L., Bruschi R., et al., Satellite Networking Integration in the 5G Ecosystem: Research Trends and Open Challenges. *IEEE Network*. Volume: 32, Issue: 5, pp. 9-15, September 2018. DOI: 10.1109/MNET.2018.1800052.  
 [10] Amaldi E., Coniglio S., Koster A. and Tieves M., On the computational complexity of the virtualnetwork embedding problem, *Electronic Notes in Discrete Mathematics* Volume 52, June 2016, pp 213-220. <https://doi.org/10.1016/j.endm.2016.03.028>  
 [11] Xu S., Wang X. and Huang M. *Software-Defined Next-Generation Satellite Networks: Architecture, Challenges, and Solutions*. *IEEE Access*, Vol: 6, pp. 4027-4041. January 2018. DOI: 10.1109/ACCESS.2018.2793237.  
 [12] Giambene G, Kota S, Pillai P. *Satellite-5G integration: a network perspective*. *IEEE Netw*. 2018;32(5):25-31.  
 [13] Xu S, Wang X, Huang M. *Software-defined next-generation satellite networks: architecture, challenges, and solutions*. *IEEE Access*. 2018;6:4027-4041  
 [14] Li T, Zhou H, Luo H, Yu S. *SERVICE: a software defined framework for integrated space-terrestrial satellite communication*. *IEEE Trans Mobile Comput*. 2018;17(3):703-716.  
 [15] Bertaux L, Medjiah S, Berthou P, et al. *Software defined networking and virtualization for broadband satellite networks*. *IEEE Commun Mag*. 2015;53:54-60.  
 [16] Toufik A, Emmanuel D, Jean-Baptiste D, et al. *Software-defined satellite cloud RAN*. *Int J Satellite Commun Networking*. 2017;36(1):108-133.  
 [17] Ferrs R, Koumaras H, Sallent O, et al. *SDN/NFV-enabled satellite communications networks: opportunities, scenarios and challenges*. *Elsevier, Phys Commun*. 2016;18(Part 2):95-112.  
 [18] VITAL Project. *Virtualized hybrid satellite-terrestrial systems for resilient and flexible future networks*. Project website at <http://www.ictvital.eu/>. Start date 1 February 2015.  
 [19] Giambene G, Kota S, Pillai P. *Satellite-5G integration: a network perspective*. *IEEE Netw*. 2018;32(5):25-31.  
 [20] 5G-VINNI: 5G Verticals Innovation Infrastructure, <https://www.5g-vinni.eu/>.  
 [21] Vergtz A., Prates N., et al., *An Architecture for the Performance Management of Smart Healthcare Applications, Sensors (Basel)*. 2020 Sep 28;20(19):5566. doi: 10.3390/s20195566.  
 [22] Esmaeily A., et al., *A Cloud-based SDN/NFV Testbed for End-to-End Network Slicing in 4G/5G*. 2020 6th IEEE Conference on Network Softwarization (NetSoft). July 2020. DOI: 10.1109/NetSoft48620.2020.9165419.  
 [23] Carella G.A., Pauls M., et al., *Prototyping NFV-based Multi-access Edge Computing in 5G ready Networks with Open Baton*. 2017 IEEE Conference on Network Softwarization (NetSoft). July 2017. DOI: 10.1109/NETSOFT.2017.8004237.  
 [24] Costa-Requena J., Poutanen A., et al., *SDN-based UPF for Mobile Backhaul Network Slicing*. 2018 European Conference on Networks and Communications (EuCNC). June 2018. DOI: 10.1109/Eu-CNC.2018.8442795.  
 [25] Mekikis P., Ramantas K., et al., *NFV-enabled Experimental Platform for 5G Tactile Internet Support in Industrial Environments*. *IEEE Transactions on Industrial Informatics*. Vol: 16, Issue: 3, pp. 1895-1903. March 2020. DOI: 10.1109/TII.2019.2917914.