On-Demand Network Slicing using SDN/NFV-enabled Satellite Ground Segment Systems

Toufik Ahmed[1], Abdelhamid Alleg[1], Ramon Ferrus[2], Roberto Riggio[3]
[1)] CNRS-LaBRI (UMR5800), Univ. Bordeaux / Bordeaux INP, France.
[2)]Universitat Politècnica de Catalunya, Spain.
[3)] Future Networks (FuN), FBK CREATE-NET, Trento, Italy.
tad@labri.fr, aalleg@labri.fr, ferrus@tsc.upc.edu, rriggio@fbk.eu

***Abstract*—Network Slicing in 5G networks has drawn lot of research attentions recently as it allows addressing different requirements for latency, throughput, capacity, and availability over a common physical network infrastructure and thus supporting diverse services, use cases, and business models. As part of the efforts pushing for a better satellite-terrestrial integration within 5G networks, extending support for network slicing into the satellite component stands out as one important must-have feature. This paper proposes an architecture framework for the realization of on-demand satellite network slicing that is built on the introduction of Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies. In this way, service delivery with satellite networks is shifted from a network for connectivity model to a network for service model with a high degree of service customization and adaptability, including satellite bandwidth on-demand and support for cross-domain integration with terrestrial networks. Under this framework, we study the resource orchestration of satellite network services by formulating the on-demand network slicing as an optimization problem that provides flexible service chaining and provisioning taking into account diversified service requirements. The objective is to determine the optimal resource allocation for supporting a satellite network slice that minimizes resources consumption while meeting service specification requirements such as the end-to-end delay.**

Keywords: **Network slicing, flexible service chaining, Software Defined Networking (SDN), Network Function Virtualization (NFV), Federated resource management.**

I. INTRODUCTION

Key features of satellite communications such as wide-scale coverage, broadcast/multicast support and high availability, together with significant amounts of new satellite capacity coming online, anticipate new opportunities for satellite communications services as an integral part within upcoming 5G systems. To materialize these opportunities, satellite communications services have to be provisioned and operated in a more flexible, agile and cost-effective manner than done today. The combination of satellite and terrestrial components to form a hybrid network has been regarded for long as a promising approach to significantly improve the delivery of communications services [1]. In this context, it is anticipated that satellite networks shall embrace network slicing support, which is one of the foundations introduced in 5G as a network architecture evolution to support diversified services requirement such as broadband communication, mission critical communications, massive IoT, etc. Each network slice can be configured to provide specific performance.

In this context, the introduction of Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies within the satellite ground segment networks is anticipated to be a necessary step in their future evolution [2]-[4] and integration in 5G network. SDN and NFV technologies are expected to bring greater flexibility to Satellite Network Operators (SNOs), reducing both operational and capital expenses in deploying and managing SDN/NFV-compatible networking equipment as well as facilitating the integration and operation of combined satellite and terrestrial networks [5]-[7]. While SDN aims to separate the control plane from the data plane, NFV aims for abstraction of the physical network in terms of a logical network and implementing network functions in software instances that can run on a range of industry standard hardware platform. The virtualization technologies used by NFV, represent a progressive manner to design, deploy and manage network slice. Each network slice is represented by a sequence of VNFs instances, chained together to compose a Service Function that requires a particular amount of resources to provide specific performances in terms of latency, throughput, capacity, and availability. For example, deployment of mission critical services such as public safety over a network slice imposes capabilities related to always-available coverage, low-latency, and high availability/reliability one-to-many and many-to-many communications. This can be properly achieved by ensuring that network resources allocated to a slice are well provisioned and deployed with specific quality of service (QoS) policy support. Furthermore, the capacity and traffic within the slice considering the specific requirements (e.g. coverage, capacity mobility, reliability…) are correctly managed and optimized.

This paper provides the design of an innovative architecture framework for on-demand satellite network slicing built on top of SDN/NFV-enabled satellite ground segment systems. A focus will be on modeling the on-demand network slicing as an optimization problem that distributes network resources on-the-fly and on demand using flexible service chaining and provisioning while taking into account diversified service requirements. This allows improving flexibility in terms of scaling up/down network resources and reconfigurability in terms of resource control programmability and dynamic QoS policy to achieve required levels of performance. The support for cross-domain integration is featured by extending the overall framework with end-to-end network slicing spanning satellite and terrestrial domains using joined / federated satellite and terrestrial network resource management and multi-domain orchestration of network functions with SDN-based control and management across terrestrial and satellite domains.

The rest of this paper is organized as follows. Section II outlines the general reference architecture of satellite ground segment and introduces the proposed architecture framework for SDN/NFV-enabled satellite systems offering on-demand adaptive network slicing. We describe further, on-demand and dynamic bandwidth allocation for satellite network slice and propose an end-to-end network slicing spanning satellite and terrestrial domains. Section III presents the slicing problem. Section IV formulates the on-demand resource allocation model for satellite slicing (OnDReAMS) as an optimization problem for flexible placement and chaining of VNF resources with the corresponding QoS and comparing it with related work as presented in section V. On this basis, section VI evaluates the

performance of the proposed mechanism. Finally, conclusions are drawn in Section VII.

II. ARCHITECTURE FRAMEWORK FOR ON-DEMAND SATELLITE NETWORK SLICING

A. General reference architecture

The general reference architecture for the satellite ground segment is structured in different subsystems [8][9] as illustrated in Fig. 1 and summarized as follows:

- The access subsystem, commonly referred to as the satellite access network. This includes the satellite gateways (GWs) and the satellite terminals (STs), which are interconnected through the space segment consisting of radio resource in terms of frequency and power resources in one or several channels (transponders) of a communication satellite. The access network can use a variety of network topologies (star, multi-star, mesh or hybrid star/mesh). It provides a variety of types of Layer 2 (L2) and Layer 3 (L3) connectivity with bidirectional links using mechanisms for forward and return link resources sharing.
- The core subsystem, commonly referred to as the satellite core network. It comprises an aggregation network that interconnects different satellite GWs located in the same or different satellite hub/teleport facilities as well as the network nodes located in the Points of Presence (PoPs) to interconnect the satellite network with other operators, corporations and Internet Service Providers (ISPs). Typically, this core network is built around an optical backbone with L2/L3 switching and routing equipment nodes based on IP/Multi-Protocol Label Switching (MPLS) and/or Carrier Grade Ethernet technologies.

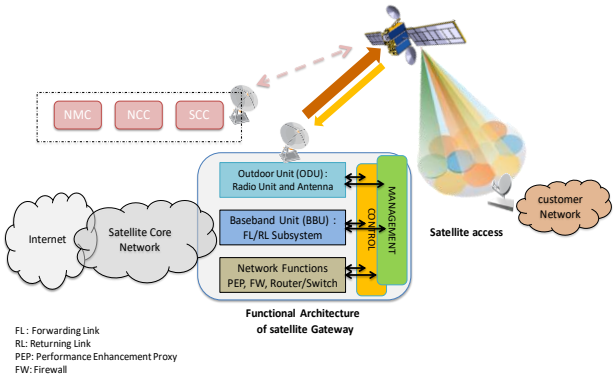

Fig. 1. Architecture for reference satellite network

- The control and management subsystems, composed of network elements such as the Network Control Centre (NCC) and the Network Management Centre (NMC). The former provides real-time control of the satellite network (e.g. connection control including the signaling necessary to set up, supervise and release connections) and the latter is in charge of the management of the system elements of the satellite network (e.g. fault, configuration, accounting, performance, and security management). In addition, the Satellite Control Center (SCC) is used to manage the satellite in-orbit platform and the satellite payload.

B. Satellite network slicing

We consider a satellite network slice as logical, virtual and self-contained network built on top of the physical satellite network infrastructure. This network slice aggregates multiple physical network resources (core and access) and uses specific abstraction and isolation mechanisms at topology, node and link levels to achieve the required levels of performance. Multiple slices may coexist over the same physical satellite. We conceive the satellite network slice as a virtual satellite network in which most of its functions are supplied as software components running in one or several Network Functions Virtualization Infrastructure (NFVI) PoPs. Conversely, the non-virtualized functions of the slice are provided through one or several physical hardware appliances, which could be dedicated to a given slice or shared among several ones.

The satellite slice is owned, managed, and operated by a satellite virtual network operator (SVNO). Each network slice is represented by a sequence of VNFs instances, chained together to compose a Service Function Chain (SFC) which lasts for a specific period. The VNF resources can be scaled up or down and they may include a variety of network functions such as Performance Enhancement Proxy (PEP) for TCP acceleration, Firewall, Deep Packet Inspection (DPI), Virtual Private Network (VPN), Packet-based QoS, DNS cache, and so on. The placement, management, chaining, and orchestration operations of these VNFs should be carefully considered to meet the required performances for supporting diverse services

In particular, as illustrated in Fig. 2 the following entities is considered as a building block of a satellite network slice:

- One or several Satellite Network Function (SNF) VNFs, namely SNF-VNFs (PEP, TCP acceleration, etc.) and Satellite Baseband Gateway (SBG) VNFs, namely SBG-VNFs for baseband mechanisms that run over NFVI-PoP. In addition, the non-virtualized part of the SBG functions, namely SBG-PNFs such as frequency block resources, together with SNF-VNFs and SBG-VNFs constitutes data plane functions. These NFVI-PoP can be extended further by support of edge functions and mobile edge computing (MEC) that provide cloud services closer to the user for reducing latency.
- SDN-based control applications and SDN controllers (all running as VNF instances) for the realization of some control functions (e.g., QoS control, radio resource management [RRM], gateway diversity [GWD], Fading Mitigation Techniques [FMT], etc.).
- Network Management (NM) and Element Management (EM) functions, also running as VNFs, which provide a package of management functions (e.g. Fault, Configuration, Accounting, Performance and Security [FCAPS] management).
- One or several Customer Premise Equipment (CPE) VNFs namely CPE-VNFs that run over Lightweight NFVI-PoP such as service provider Whitebox. The CPE-PNFs are also part of the data plane. The NM/EM and CPE SDN-based control applications and SDN controller building blocks are provided for illustrative purpose. Furthermore, CPE-VNF may provide wired/ wireless access network (such as Ethernet and WiFi) to build the customer network.

It is worth noting that SBG-PNG controller containing functionalities for SBG-PNF slicing and sharing, Service Orchestrator (SO) agent and NFV agent is outside the Satellite Network slice. Usually is implemented by the Satellite Network Operators (SNOs) to facilitate the slicing operation. The allocation, deployment, control, management and, orchestration operations for a satellite network slice considering diverse requirements for latency, throughput, capacity, availability and reliability are crucial for supporting diverse services, use cases, and business models.

As shown in Fig. 2., the network service orchestration capabilities are logically centralized in the so-called Service Orchestrator (SO) management component, which forms part of the Operation Support Systems / Business Support Systems (OSS/BSS) of a Satellite Network Operator (SNO). Beyond this, functionalities related to the instantiation, modification and termination of the VNFs composing the satellite network slice are covered by the NFV Manager. The functionalities provided by the SO and the NFV Manager are related to the following:

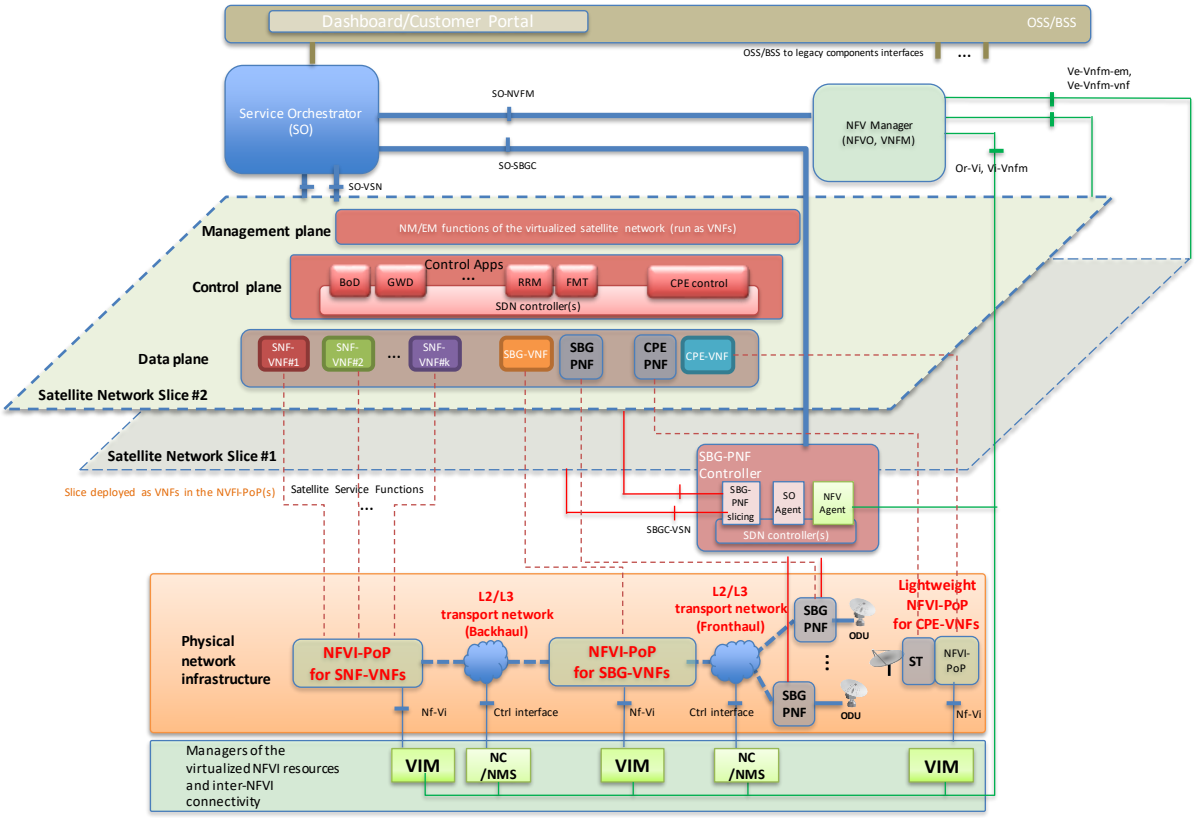

Fig. 2. An architecture for SDN/NFV-enabled satellite ground segment systems

- Lifecycle management of the slice, which can be defined as the set of functions required to manage the instantiation, maintenance (e.g. adaptive scaling up / down, QoS configuration, etc.), and termination.
- Composition of the service function chain described by a network service descriptor (NSD) that represents the part of the slice that is implemented as VNFs and executed over NFVI-PoP(s).
- Determination of the application-specific aspects of both VNFs and PNFs that form part of a slice.
- Fault, Configuration, Accounting, Performance, Security (FCAPS) management of the slice and its components, irrespective of whether they are VNFs or PNFs. These FCAPS management functionalities supported by the SO are mainly intended to guarantee the proper operation of the deployed slice in terms of performance measurement, resource usage, and alarms handling for accounting, reconfiguration and problem correction actions.
- Lifecycle management of the service chain composing the slice through interaction with the NFV Manager, which offers the Os-Ma-nfvo reference point [11] as specified in the ETSI NFV MANO architecture [16] (i.e. the SO is a consumer of reference point Os-Ma-nfvo). It's worth noting that the NFV Manager takes care of the deployment specific configuration of the VNFs that form part of the NS.
- Management of VNF packages that can be already on-boarded on the NFV Manager or can be managed/on-boarded onto it by the dashboard of the SO. In general, at the SO only the VNFDs are needed to compose the NSDs.

The operation of the SO and NFV Manager relies on a set of descriptors that are needed for the characterization of a slice and its components. In general terms, a Satellite Network Slice Descriptor (SNSD) is the input provided to the SO that describes the characteristics of the slice as requested by the customer/tenant. Based on the SNSD, the SO composes the NSD, which describes the virtualized part of the slice, and the slice application-specific descriptors, which contain the configuration of both VNFs and PNFs within the satellite network slice. However, the details of the above-mentioned components, extensively studied and developed in [2] are beyond the scope of this paper and introduced here for sake of clarity.

C. *Multi-domain orchestration for end-to-end network slicing*

End-to-end networking slicing spanning multiple administrative domains including both satellite and terrestrial network aims to deploy end-to-end network slice with dedicated resources. This can be achieved in multiple approaches such as:

- Broker-based: both satellite and terrestrial domains expose SDN/NFV services to a third party player, namely a broker. The broker will be able to order cross-domain network services.
- Peering: using direct negotiation to allow both satellite and terrestrial domains to deploy a portion of the end-to-end slice.

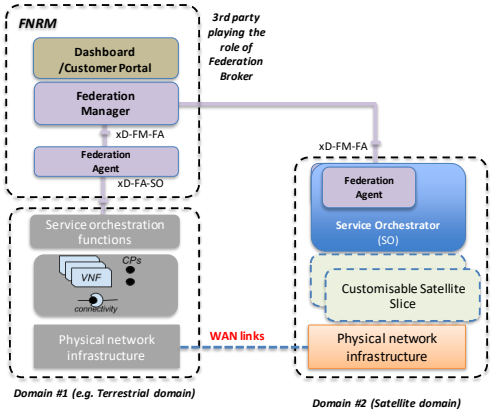

Fig. 3. Illustration of a possible architecture for the multi-domain federation

As part of the proposed framework, a solution for cross-/multi-domain orchestration has been developed. This entails the introduction of a new management component called Federation Network Resource Manager (FNRM), as illustrated in Fig. 6. This component consists of two separate functions: a Federation Manager (FM) and a Federation Agent (FA). The FM supports the logic to federate different domains and orchestrating Multi-Domain Network Services (MD-NSs), while the FA handles the heterogeneity of service orchestrators used in each of the federated domains, interfacing them with the FM. Centralized and peer-to-peer federation models can be supported. As an example, the scenario depicted in Fig.3 shows a SDN/NFV-enabled satellite ground segment infrastructure, owned and operated by a SNO, and a terrestrial network infrastructure, such as a mobile or fixed communication network, owned and operated by a Terrestrial Network Operator (TNO). The support of federation capabilities may even lead to new business cases for third party companies that could play the role of a Federation Broker [13] and offer added value services through resources allocated across multiple domains, as illustrated in Fig. 3. Further details on the capabilities of the proposed FNRM can be found in [14].

III. SATELLITE NETWORK SLICING MODEL

In this section, we formulate the proposed satellite network slicing model as an optimization problem and we define different notations, parameters and terminologies relative to network slicing topic.

A. Network Slicing

The satellite network slice is considered as virtual, self-contained and isolated network built on top of aggregated distributed physical resources at core, edge, access and user levels. It is a network of capabilities rather than a network of entities aiming to provide specialized functions deployed at different points to support diverse services requirements. It is represented by a sequence of VNFs instances and PNFs resources, chained together to compose a Service Function Chain (SFC). For simplicity, we ignore the PNF resources at both Satellite Baseband Gateway (SBG) and Customer Premise Equipment (CPE) as an SBG-VNF is always attached to an SBG-PNF and a CPE-PNF is always attached to a CPE-VNF. However, the proposed model provides appropriate requirements to support normal satellite gateway operation when the functional splitting between VNF and PNF parts is performed. The link between SBG-VNF and SBG-PNF called fronthaul link is carefully examined as the introduction of latency over the fronthaul link can affect the normal network operation such as synchronization, handover decision, guard time, etc. An SFC defines an ordered (*resp.* partially ordered) set of virtual network functions VNFs [13] that require tailored resources to guarantee predictable network performances defined by a traffic class associated to a slice.

We define a Slice Request (SR) by the following parameters:

- SFC description (ρ)
- Slice lifetime (τ)
- Tenant identifier (∇)
- Traffic Class (λ)

We note R_s the set of slice requests that needs to be instantiated on the satellite network infrastructure. Each slice request $\delta \in R_s$ is modeled as a quadruple $\delta (\rho, \tau, \nabla, \lambda)$ where ρ is the SFC that corresponds to the deployed service, τ corresponds to slice lifetime, ∇ is the tenant identifier and λ is the class of traffic to which belongs the slice request δ. The on-demand network slicing is managed as a new request for scaling up / down the SFC network resource. Each SFC ρ is modeled as a subgraph $G_v^\rho (N_v^\rho, E_v^\rho)$ where $N_v^\rho \subseteq N_v$ is a set of VNFs and $E_v^\rho \subseteq E_v$ is a set of directed edges called virtual links connecting these VNFs. In addition, each VNF instance $n' \in N_v^\rho$ has its own requested amount of resource denoted $\theta_\delta^{n'}$. Also, each virtual link $(k,l) \in E_v^\rho$ connecting two VNFs $k, l \in N_v^\rho$ is characterized by key performance metrics (capacity, performance, delay, etc.) denoted $\psi_\delta^{(k,l)}$.

In this work, we concentrate on the slice end-to-end delay (or end-to-end latency) as the key performance indicator of a specific traffic class and we define D_{th}^λ, the end-to-end delay threshold associated to each traffic class $\lambda \in R_c$. The value of D_{th}^λ is expected to meet specific requirements for diverse services that will be running on a slice (see Table I.). We define the end-to-end delay provided by a deployed SFC as the sum of processing delay $D_{Proc}^{n'}$ of its component VNFs instances and the time needed to forward the flow between these VNFs.

We depict in Fig. 4, the message chart for instantiation of satellite network slice, where the satellite network infrastructure is assumed to belong to Satellite Network Operator (SNO) that deploys our SDN/NFV-enabled satellite ground segment infrastructure. The slice instantiation can be done online using a customer portal with self-service features. The main steps involved are:

- Slice request: the customer such as a Mobile Network Operator (MNO) that needs to establish a satellite network slice for mobile backhauling services uses the online self-service portal to select and configure the characteristics of the slice (SFC description, lifetime, traffic class, etc.)
- Resource information: based on the current network status, an admission control mechanism for a slice is performed by the SO to determine how the slice is deployed in both the virtualized and the non-virtualized infrastructure of the SNO (including the CPE).
- Slice allocation resource: after different steps, SO completes the operational activation of the new slice. This process might involve a set of NM/EM systems within the SO to activate/turn-on the slice components from a management perspective and bring them into operational state. Monitoring processes are also started at this point for the SO to supervise the operational status of the slice.

Table I. SERVICE REQUIREMENTS

Type of Service / traffic class	Requirements
Satellite Multimedia Broadcast Multicast Services (MBMS)	High capacity, large packet size, low loss rate, caching at the edge, bulk data, one and many-to-many communications,
Satellite IoT and Massive Machine Type Communications (MTC)	Large converge, one-to-one and one-to-many communication
Satellite Mission Critical Communications (MCC)	Low-latency, high reliability, real-time, jitter sensitive and high interaction
Satellite Mobile Backhauling (MB)	High bandwidth, low latency
Satellite Mobile Direct Access (MDA)	Mobility support, low latency, high reliability, large converge,

For each time window noted T_{win} and according on admission control decisions the SO allocates the required network resources to the accepted slices during their respective lifetimes τ while taking into account the traffic classes λ.

B. NFVI Model

The infrastructure layer hosts the physical and virtual resources needed to create the satellite network slices. These include both virtualization software and hardware comprised of memory, compute, storage, and networking resources. We adopt a Network Function Virtualization architecture composed of NFVI (physical network), set of VNFs and NFV management and orchestration (MANO) platform, in accordance with the terminology presented in [15]. We model the network the NFVI that consists of hardware resources as a graph $G_i(N_i, E_i)$, where N_i is the set of Point of Presence (PoPs) that compose network and E_i is the set of bidirectional links (PLs). Each PoP $n \in N_i$ represents a possible location that can host a single or multiple VNFs instances depending on their resource capacities. PoPs are connected via Physical Links (PLs) that forward traffic between VNFs composing a SFC.

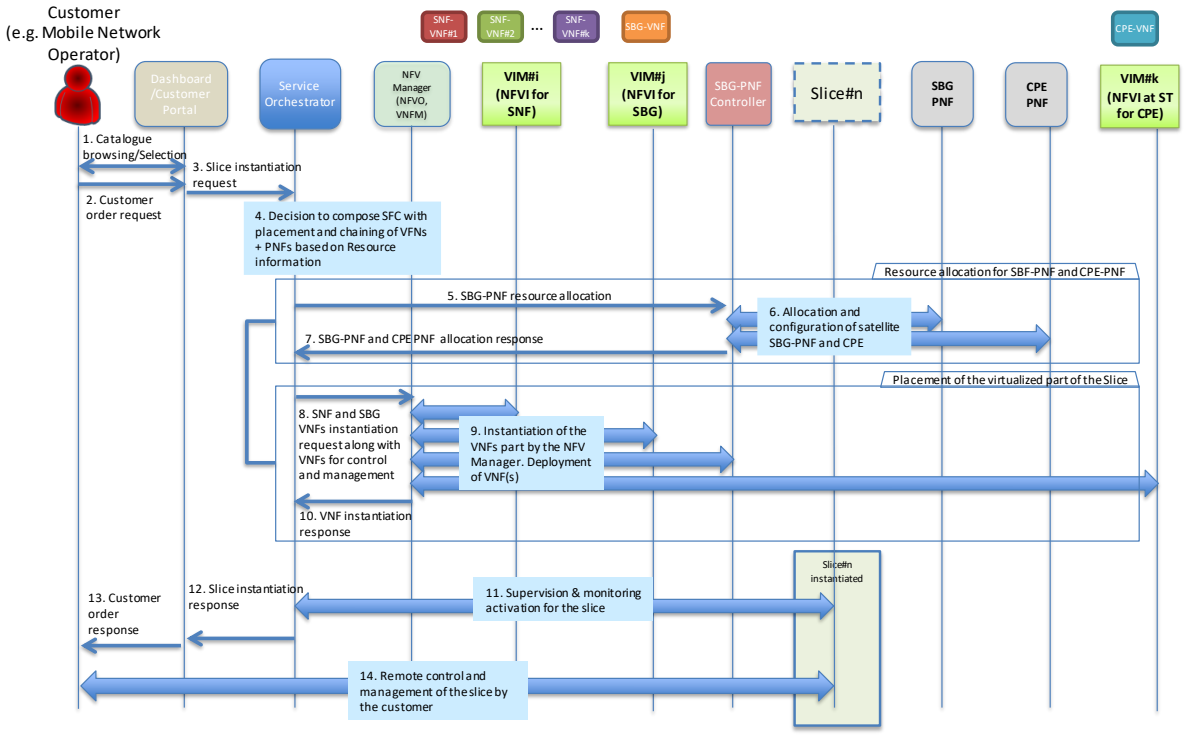

Fig. 4. Slice instatiation process

Each PoP $n \in N_p$ represents the quantity of available resources in terms of Computing, Memory and Storage denoted Θ_n^λ and reserved to SR using traffic class λ . Similarly, each PL $(n,m) \in E_p$ connecting two PoPs $n,m \in N_p$ has its capacity (Bandwidth, Bitrate, etc.) denoted $\Psi_{(n,m)}^\lambda$ used exclusively by SR with traffic class λ.

Usually, placement and chaining involves two main steps:

- Placement, which consists in assigning a set of VNFs to a set of PoPs (physical locations) in the NFVI.
- Chaining, which builds paths that interconnect the VNFs previously assigned to different PoPs (during placement step) in order to constitute SFCs that corresponds to a service supported by a slice.

Table II summarizes the NFVI and SR notation and parameters used in our model.

Table II. NFVI AND SR NOTATION

PAR.	DESCRIPTION
NFVI	
G_i	NFVI graph
N_i	Set of PoPs in G_i
E_i	Set of physical links between PoPs
Θ_n^λ	Available resource at PoP $n \in N_i$ resereved for traffic class λ
$\Psi_{(n,m)}^\lambda$	Available capacity of physical link $(n,m) \in E_i$ resereved for traffic class λ
$D_{Tran}^{(n,m)}$	Transmission delay of the physical link in terms of latency $(n,m) \in E_i$
Slice Request "SR"	
R_s	Set of slice requests
R_c	Set of traffic classes
ρ	SFC that corresponds to the deployed service
τ	Slice lifetime
$\overline{V}$	Tenant identifier
λ	Class of traffic to which belongs the slice request δ .
G_v	SFCs graph
N_v	Set of VNFs in G_v
E_v	Set of virtual links between VNFs in G_v
N_v^ρ	Set of VNFs composing the request r where $N_v^\rho \subseteq N_v$
E_v^ρ	Set of links between VNFs $\in N_v^\rho$ such as $E_v^\rho \subseteq E_f$
$\psi_\delta^{(k,l)}$	Required capacity of virtual link $(k,l) \in E_v^\rho$
$\theta_\delta^{n'}$	Requested resources of VNF $n' \in N_v^\rho$
$D_{Proc}^{n'}$	Processing delay generated by VNF $n' \in N_v^\rho$ using exaclty the required amount of resources $\theta_\delta^{n'}$
D_{th}^λ	End-to-end delay threshold associated to $\rho \subseteq R_{sfc}$

IV. ON DEMAND RESOURCE ALLOCATION MODEL FOR SATELLITE SLICING (ONDREAMS)

Our proposal solution is based on a mathematical program combined with an online algorithm. First, we model the slicing problem using a Mixed Integer Linear Program (MILP). The objective of this formulation is to deploy efficiently the services carried on each slice request. The MILP proposes an optimal placement and chaining of the sequence of VNFs that compose each SFCs.

The optimization objective of our MILP is to minimize the amount of allocated resource to VNFs (Equation 1). Since we are in context of slicing, this objective has the most significant impact on network management. Furthermore, this objective could be easily adapted to aim other purposes such as number of active PoPs or cost utilization, etc. The optimization objective and the constraints of the MILP are presented below.

$$Min\left(\sum_{\delta\in R_s}\sum_{n\in N_i}\sum_{n'\in N_v^\rho}\theta_\delta^{n'}.C_\lambda^\delta.B_n^{n'}\right) \qquad \forall\,\lambda\in R_c \quad (1)$$

Subject to:

$$\sum_{\delta\in R_s}\sum_{n'\in N_v^\rho}\left(\theta_\delta^{n'}.C_\lambda^\delta.B_n^{n'}\right)\le\Theta_n^\lambda \quad \forall\, n\in N_i \quad \forall\,\lambda\in R_c \qquad (2)$$

$$\sum_{\delta\in R_s}\sum_{(k,l)\in E_v^\rho}\left(\psi_\delta^{(k,l)}.C_\lambda^\delta.B_{(n,m)}^{(k,l)}\right)\le\Psi_{(n,m)}^\lambda\,\forall\,(n,m)\in E_i \quad \forall\,\lambda\in R_c \qquad (3)$$

$$\sum_{n'\in N_v^\rho}B_n^{n'}=1 \qquad \forall\, n\in N_i \qquad (4)$$

$$\sum_{m\in N_i}B_{(m,n)}^{(k,l)}-\sum_{m\in N_i}B_{(m,n)}^{(k,l)}=B_n^k-B_n^l \quad \forall\, n\in N_i,\forall\,(k,l)\in E_v^\rho \qquad (5)$$

$$\sum_{n\in N_i}\sum_{n'\in N_v^\rho}\left(D_{proc}^{n'}.B_n^{n'}\right)+\sum_{(n,m)\in E_i}\sum_{(k,l)\in E_v^\rho}\left(D_{Trans}^{(k,l)}.B_{(n,m)}^{(k,l)}\right)\le\ D_{th}^\lambda$$

$$\forall\,\lambda\in R_c \quad (6)$$

$$\sum_{n'\in N_v^{\rho*}}B_n^{n'}=1 \qquad \forall\, n\in\ N_i^* \qquad (7)$$

$B_n^{n'}$ (*resp.* $B_{(n,m)}^{(k,l)}$) is a binary variable indicating whether VNF instance n' (*resp.* virtual link $(k,l) \in E_v$) is mapped into a particular PoP n (*resp.* into the physical link $(n,m) \in E_i$). Also, we note C_λ^δ a binary variable indicating whether λ corresponds to the traffic class of the slice request δ.

Constraint (2) ensures that the sum of allocated computing resources required by VNF n' mapped into PoP n does not exceed the amount of available resources in its class of traffic. Similarly, constraint (3) ensures that each link has enough available capacity to support the virtual links mapped over it. Constraint (4) states that each VNF has to be mapped only once into the physical infrastructure. In other words, the whole amount of resource (Computer, Memory and Storage) allocated to a given VNF must be provided by *exactly* one physical node to avoid dispatching a VNF over multiple POPs.

Constraint (5) consists in building the virtual paths between the required endpoints. This chaining constraint is used to enforce the condition that for each virtual link there must exist a continuous path allocated between the pair of physical nodes in which VNFs have been mapped.

Constraint (6) ensures that each deployed SFC will not exceed the end-to-end delay threshold that is specific to each traffic class. The first part of the equation is a sum of the delay incurred by packet processing on VNFs, while the second part defines the delay incurred by transmitting packets between these VNFs.

Last, constraint (7) allows to place a specific type of VNF ($n' \in N_v^{\rho+}$) into a particular physical placement ($n \in N_i^*$). For example, the SBG-VNF must be placed in the PoPs near the satellite Hubs that implements the physical part of the gateway in terms of SBG-PNFs. The fronthaul link between SBG-VNF and SBG-PNF imposes strict requirements in terms of performance such as latency. This allows the satellite gateway to work properly. In fact, some control functions of the gateway need a control loop interaction between VNF and PNF parts such as MODCOD selection based on feeder link condition (SNIR level). Thus, the objective of this constraint is to model such need.

Algorithm 1: pseudo-code for ONDREAMS

Input: R_s, R_c, G_i, T_{win}
Output: S

1 // Round 0
2 $T_{current} \leftarrow 0$
3 **Update** (available$_{res}$)
4 //Sorting Slice request
5 $SR_{buffer}(\lambda) \leftarrow \boldsymbol{Sort}(R_s, \tau, decreasing)$
6 **for** λ **in** R_c
7 | $S(\lambda) \leftarrow \boldsymbol{Solve}$ $(MILP\ Model)$
8 **endfor**
9 **Slice**$_{\boldsymbol{Allocation}}(S)$
10 **while** $(T_{current} \leq T_{win})$
11 | **Read** R_s
12 | **Update** (available$_{res}$)
13 | **for** δ **in** S
14 | | $\tau(\delta) \leftarrow \tau(\delta) - 1$
15 | | **if** $\tau(\delta) = 0$ **then**
16 | | | **Free**(resources)
17 | | **elseif** $\rho(\delta).old < \delta.new$ **then**
18 | | | $UpScaling^\delta \leftarrow \boldsymbol{SCheck}(\delta.new)$
19 | | **elseif** $\rho(\delta).old < \delta.new$ **then**
20 | | | $DownScaling^\delta \leftarrow True$
21 | | **Endif**
22 | **Endfor**
23 | $T_{current} \leftarrow T_{current} + 1$
24 **Endwhile**
25 **for** δ **in** S
26 | **If** $UpScaling^\delta = True$ **then**
27 | | $\boldsymbol{DoUpScaling}(\delta.new)$
28 | **Elseif** $DownScaling = True$ **then**
29 | | **Free**(resources)
30 | **Endif**
31 **Endfor**

The MILP defined above is used to process a set of slices by providing an optimal placement and chaining of their corresponding SFCs while meeting the traffic class requirements (end-to-end delay). In a second step, and in order to add a dynamic on demand aspect to our solution we developed an online algorithm that calls the previous MILP at periodically at a fixed time window T_{win} as presented in algorithm 1.

At the begging of each time window, the online algorithm updates available resources (line 3). For this solution, we propose to favor Slice Request (SR) with a long lifetime (line 5), other variants may be proposed such as short lifetime slice request first. Therefore, the algorithm calls the original MILP for each class of traffic using as inputs an ordered set of SRs according to their lifetime τ. The MILP returns the set of accepted SRs and the placement of their corresponding SFCs (line 7). During a time window, the algorithm cannot accept any new SR until the next round. However, it keeps track of the possible upscaling of SRs that have been already instantiated (line 17 and 19). Also, when a given SR reaches its lifetime its allocated resources will be immediately released (line 15) and available for resource pool ready to be allocated. At the end of the time window, the SR with possible upscaling is handled by *DoUpscaling* function that reallocates the new required resources by the SR δ (Line 27). Depending on the available resources and the current placement of the SFC, the upscaling of SR can be realized by satisfying locally the new required resources (e.g. a VNF may obtain new CPUs from the same PoP in which it is already mapped. Similarly, a virtual link capacity can be enhanced over the same physical link without need to be mapped over another link) or by applying a global optimization and calling the original MILP. It is worth noting that upscaling the resources over the satellite link will use the mechanism for Bandwidth on Demand presented such as presented in [12]. Another upscaling scenario that should be carefully considered could be envisioned with the migration of VNFs to other POPs. Furthermore, when a resources downscaling are released, they will be available in the pool for the next time window (line 29). In addition, several scenarios can be handled in this procedure.

- Simple request / response: there is no negotiation about the on-demand resources allocation that can be granted to the slice. The slice request can be accepted or refused based on the current available resources.
- Complex request / response with negotiation: the procedure of granting on-demand resources can use several round with multiple alternatives if possible, on which the tenant can react. This can be coupled with QoS parameters in which the tenant can select (accept/refuse) or make new request. This process can still until an agreement can be reach or a fail.
- Modification of resources request parameters such modification in the parameter of the bandwidth profiles coupled with the parameter of the QoS class.

To understand our model for QoS constraint, Fig. 5 illustrates a creation of slice by deploying an SFC composed of 5 VNFs over a simple satellite segment topology composed of 6 PoPs. Three QoS classes are implemented in different PoPs and links providing differentiated performances. For example, the link between PoP1 and PoP2 has a latency of 5ms for C1 (QoS Class 1) and 7ms for C2 (QoS class 2) and 15ms for C3. Furthermore, each PoP may provide a set of differentiated resources (CPU, disk, memory) corresponding to a specific QoS class, which impacts the processing delay of a given VNF and generates different performances depending on the attributed resource (for example high performance vCPU vs. low performance vCPU). In this case, the slice QoS ensure the aggregation of individual QoS offered to the SFC.

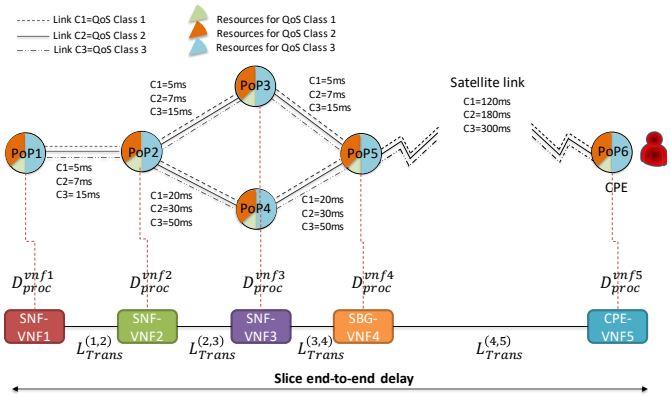

Fig. 5. Example of 5 VNFs SFC and 6 POPs toplogy with two QoS classes

As we will concentrate on the end-to-end delay, our model guarantees minimum resources allocation while meeting the constraint of slice end-to-end delay threshold.

The deployment of the SFC requires an optimal placement of the different VNFs over the PoPs topology. Let us consider a slice request composed of 5 VNFs: {SNF-VNF1, SNF-VNF2, SNF-VNF3, SBG-VNF4, CPE-VNF5} with traffic class for a slice that requires 180ms end-to-end delay threshold. For simplicity, we suppose that the processing delay of each VNF is set to be $D_{proc}^{n'} = 2ms$. The result our optimal placement and chaining of this SFC will be over {PoP1, PoP2, PoP3, PoP5, PoP6} with QoS class 1. This slice will provides an end-to-end delay of {2+5+2+5+2+5+2+120+2}=145ms which is less than the end-to-end delay threshold (180ms). The other placement {PoP1, PoP2, PoP4, PoP5, PoP6} will not be selected as it cannot guarantee the delay threshold.

V. Related Work

Numerous research papers have been published on the placement and chaining of VNF (PC-VNF) problem [20] with the goal of optimizing the placement of VNFs subject to different optimization objectives (number of VNF instances, resource utilization, provising cost, etc.). Works in this field can be classified according to their proposed formulations (model and objective) proposed by their solutions.

Works [21] - [26] proposed solutions for VNF service chain placement and chaining to optimize both network link and compute resource utilization. However, most of these works ignored the network class of traffic to construct the service chain and especially the QoS parameters such as the end-to-end delay have not been integrated in their models. In particular, Taleb *et al.* [25] considered two competing objectives for VNF placement in mobile core network and proposed three solutions. First, ensuring an acceptable QoE by a near-user placement of data anchor gateway. Second, avoiding the mobility anchor gateway relocation by placing VNFs far enough from users. The third solution is a trade-off between the two previous solutions modeled using game theory. The scope is however limited to only two particular mobile core network functions and VNF resource requirements have not been considered. In a similar context, Baumgartner *et al.* [26], investigated the placement of virtual mobile core network functions excluding VNFs on the radio access network. They aimed to minimize resource provising cost while meeting VNF requirements in terms of bandwidth, processing and storage. However, they do not address QoS constraint such as end-to-end delay. Riggio et *al.* [18] examined the VNF placement problem in the radio access network (RAN) domain including functions such as load-balancing, firewall, and virtual radio nodes. An ILP model and a heuristic are proposed. Their objective is to minimize the cost of mapping virtual functions to substrate network (nodes and links) while satisfying VNF requirements in terms of CPU, memory, storage, radio, and bandwidth resources.

For comparison purposes, we summarize the related work in a modified version of our model OnDReAMS that does not consider the QoS parameters in terms of end-to-end delay when instantiating the slice and placing its SFC. This second model is called E2E QoS Agnostic Model (QoSAM) and it is obtained by retrieving QoS class differentiation and retrieving the constraint (6 and 7) from the original model.

VI. Performance Evaluation

In this section, we evaluate the slice creation in terms of placement performances of both OnDReAMS and QoSAM models using different type of SFC. Both models were implemented using AIMMS Modeling Optimization version 4.3 [17] and experiments were conducted on Windows 8 server with Intel Core i7-3740QM processor with 16GB of memory. All evaluations are repeated 20 times. We first describe the simulation environment and then we discuss the performances evaluation metrics.

A. Simulation environment

We use the topology described in Fig. 5 as baseline topology. In our simulation, the available PoPs resources (resp. requested resources) and the available capacities of physical links (PLs) (resp. requested virtual links capacities) are configured to have fixed values as presented in Table III. In addition, we define three type of generic slice requirement using three traffic classes (QoS1, QoS2, and QoS3) to cater different categories of service with different QoS levels. Each QoS level imposes performance parameters mainly the end-to-end delay threshold D_{th}^{λ} as expressed in Table III.

All formalized models are evaluated using four structural variants of SFC. The first component **L1** is a linear chain composed of a sequence of VNFs connecting two endpoints "S" and "D". The second component **B1** consists of a bifurcated chain using different VNFs in each path connecting two endpoints "S" and "D". The third and fourth components (**L2** and **B2**) use the same structure of the ones described previously but with varying the number of VNFs (see Table III.).

Table III. Simulation parameters

Parameters	**Value range**		
Number of VNFs per service for linear (L) and bifurcated (B) SFC	L1,B1		[1, 3]
	L2, B2		[4, 6]
Delay threshold D_{th}^{λ} for a slice	Traffic class « QoS 1 »		150 ms
	Traffic class « QoS 2 »		300ms
	Traffic class « QoS 3 »		600ms
Available resources at PoPs	set at 100%		
Requested resource	set at 1%		
Available capacity of PL	set at 100%		
Required capacity of virtual link	set at 1%		
Processing delay $D_{proc}^{n'}$	2ms		
	Scenario	1	2
PoP and link ressource distribution per QoS class	QoS1	20%	50%
	QoS2	30%	25%
	QoS3	50%	25%

In order to evaluate the two formulations (OnDReAMS and QoSAM), we selected some result metrics adopted in several works [18][19]. Thus, for each model, we measured the average end-to-end delay, the average number of QoS violation and the average of accepted slice requests, using two type of the SFCs (linear and bifurcated) with different resource distribution per QoS class according to 2 scenarios for resources distribution as described in Table III.

B. Simulation Results

First, we analyze the slice end-to-end delay provided with both OnDReAMS and QoSAM. Fig. 6 depicts the average delay measured between endpoints when increasing the number of slice requests for scenario 1 (same results for scenario 2). The end-to-end delay is computed as a sum of VNFs processing

delays and transmission delays along the SFC path (as illustrated in topology of Fig. 5). The results shows that OnDReAMS provides the adequate end-to-end delay compared to QoSAM regardless the number of processed slice requests. When a slice requests is accepted by OnDReAMS, its end-to-end delay is guaranteed. This difference in performance is due to the delay constraint that guides the solver to place VNFs of a given SFC in a manner to ensure not exceeding the required delay threshold specific to each QoS class (result for QoS1, QoS2, and QoS3). In addition, in the case of OnDReAMS, the fact of partitioning resources between different QoS classes guarantees a more efficient placement by allocating the appropriate resource to reach the needed performance in terms of delay. While, QoSAM is unable to differentiate between QoS classes, still less meeting their delay requirements, which may increase the resulted end-to-end delay and generate QoS violation cases for the slice.

When increasing the number of slice requests, OnDReAMS provides a better end-to-end delay, especially with QoS1 and QoS2. This is due to its ability to place the VNFs in a manner to meet its end-to-end delay by allocating exclusively the dedicated resources according to QoS level. However, we observe that allocation of slice with QoS3 experiences less delay with QoSAM compared to OnDReAMS. Such result is mainly due to the ability of QoSAM to use resources without distinction that leads to a convergence of end-to-end delay to an average delay over the path.

Furthermore, by conception, QoSAM is not supposed to respect the distribution of resources per QoS class. This, allows QoSAM to minimize resources consumption by performing a free-class placement of VNFs that uses any available resources without considering delay requirements.

In order to investigate the possible hidden problems behind QoSAM, we measured the number of QoS violation defined as the percentage of slice request exceeding the end-to-end delay threshold among total number of request. Fig. 7 depicts the evolution of QoS violation percentage observed by QoSAM for different portion of QoS1 slice when increasing the number of requests. In the case of OnDReAMS there is no QoS violation because of the strict delay threshold constraint that obliges the solver to reject a request when its QoS class requirement (mainly end-to-end delay) cannot be honored.

We notice that QoSAM starts generating QoS violation cases since 5 requests and their number depends on the portion of QoS1 slice requests. Indeed, such stringent QoS class requirements are more likely to be violated since QoSAM has no delay constraint to respect. Furthermore, the QoS violation reaches it maximum when 50% of slices requests are of type QoS1 class while with 10% QoSAM provides a low QoS violation level. In other words, when using a model that ignores completely delay constraints, the number of QoS violation depends on the number of requests with strict QoS requirements.

To better understand the behavior of OnDReAMS and QoSAM in terms of requests acceptance, Fig. 8 shows the average rate of accepted slice requests for both models in different scenarios using different SFC. As expected, QoSAM solution achieves a better rate of globally accepted slice requests in overall scenarios whereas OnDReAMS tends to reject requests when exceeding a specific number of slices. QoSAM continue to accept slice requests until overloading the network resources but without guaranteeing a convenient QoS performance. Additionally, QoSAM may provide unnecessarily high QoS performances to satisfy slice requests of class QoS2 or QoS3, which leads to a possible QoS violation of QoS1 slice request.

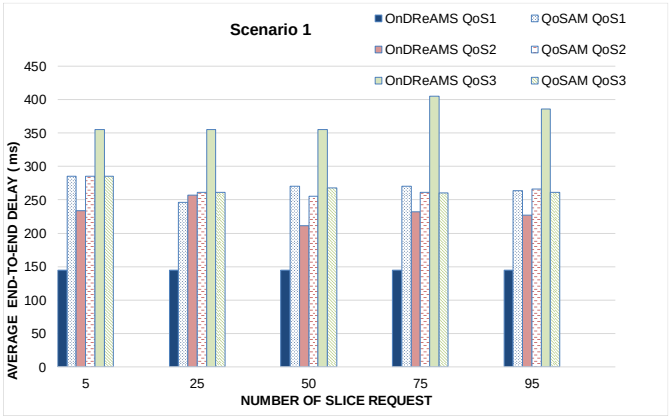

Fig. 6. Effect of number of slice requests on the end-to-end delay for scenario 1

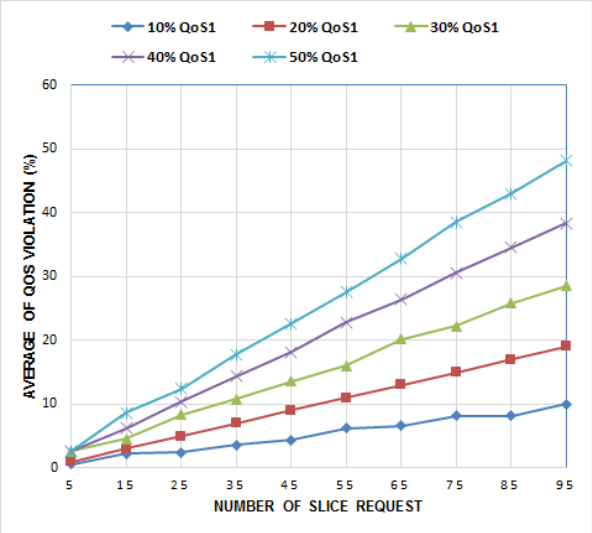

Fig. 7. Evolution of QoS violation cases for QoSAM according to QoS class request.

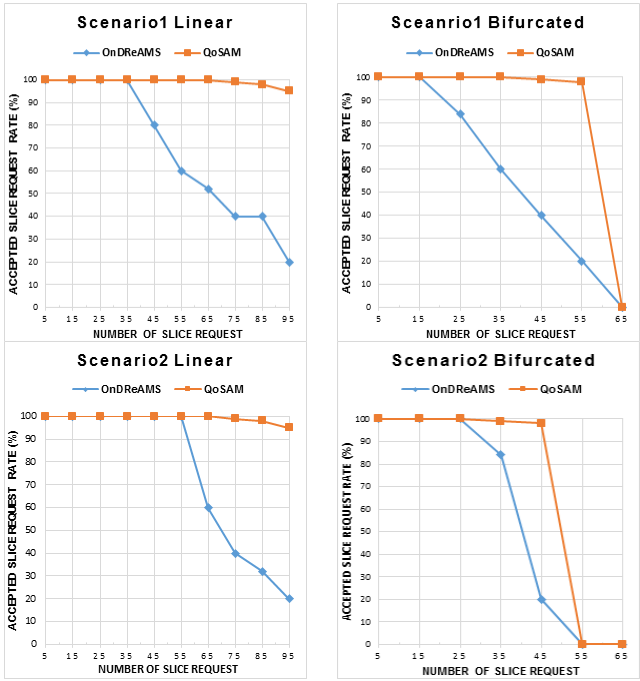

Fig. 8. SFC acceptance comparison in different scenarios

In the other hand, the delay constraint adopted by OnDReAMS avoids QoS violation but provokes an early slice requests rejection (for example in scenario 2 “linear” SFC, rejection is noticeable from 55 slice requests). OnDReAMS, as it is designed, begins to reject requests of a given QoS class when the resource reserved to this class is overloaded, even though unused resources reserved to the other class are available. Such problem can be addressed by integrating an auction-based mechanism to deal with idle non-used resources or by allowing sharing / borrowing resources among slices [27]. In addition, introducing QoS negotiation mechanism to allow one slice to move from one QoS level to another (upgrading or downgrading) can be a solution to this problem.

Another aspect of OnDReAMS is related to its capacity to use Algorithm 1 to provide PoPs resources and network resources on demand. By the introduction of SDN in the satellite network, the procedure for BoD can be centrally computed and re-arranged dynamically at slice-level granularity in front of some events (for example, need for more satellite capacity for mobile backhauling slice when terrestrial backhaul fails). As a result, the end-to-end slice across the satellite and terrestrial components is updated to provide the appropriate network performance.

VII. CONCLUSION

In this paper, we presented an approach that pushes SDN/NFV technology enabler into the satellite domain to provide enhanced satellite communications service delivery and achieve a better integration of the satellite segment within the 5G ecosystem. In particular, we designed an SDN/NFV-based architecture framework for on-demand satellite network slicing along with its extension to end-to-end slicing using broker-based approach. This architecture provides flexible service chaining and provisioning taking into account diversified service requirement while meeting performance expectations from service level perspective. In addition, we presented procedure for flexible Bandwidth on Demand (BoD) that aims to provide dynamic allocation and sharing of resources between different tenants operating the satellite network slice. Performance measurements of the slicing procedure have been conducted and results show that the proposed OnDReAMS model provides better QoS level in terms of end-to-end delay to meet service requirement.

ACKNOWLEDGMENT

Research leading to these results has received funding from the European Union’s H2020 Research and Innovation Programme (H2020-ICT-2014-1) under the Grant Agreement H2020-ICT-644843.

REFERENCES

[1] ETSI TR 103 124 V1.1.1 (2013-07), “Satellite Earth Stations and Systems (SES); Combined Satelliteand Terrestrial Networks scenarios”

[2] H2020 VITAL research project website at http://www.ict-vital.eu/

[3] Bertaux L., Medjiah S., Berthou P., Abdellatif S., Hakiri A., Gelard P., “Software Defined Networking and Virtualization for Broadband Satellite Networks”. IEEE Communications Magazine, March 2015

[4] Roberto Ferrus, H. Koumaras, O. Sallent, G. Agapiou, T. Rasheed, M.-A. Kourtis, C. Boustie, P. Gelard, Toufik Ahmed, SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges, Physical Communication, November 2015

[5] Sacchi, C.; Bhasin, K.; Kadowaki, N.; Vong, F., "Toward the "space 2.0" Era [Guest Editorial]," Communications Magazine, IEEE , vol.53, no.3, pp.16,17, March 2015

[6] NetWorld2020’s – SatCom WG The role of satellites in 5G, Version 5 – 31th July 2014

[7] 3GPP TR 22.891 V1.1.0, “Feasibility Study on New Services and Markets Technology Enablers;Stage 1 (Release 14)”, November 2015

[8] Gerard Maral, Michel Bousquet, Zhili Sun (Contributing Editor), “Satellite Communications Systems: Systems, Techniques and Technology, 5th Edition, ISBN: 978-0-470-71458-4, December 2009

[9] ETSI TR 103 272 V1.1.1, Satellite Earth Stations and Systems (SES); “Hybrid FSS satellite/terrestrial network architecture for high speed broadband access”, March 2015

[10] Toufik Ahmed, Emmanuel Dubois, Jean-Baptiste Dupé, Ramon Ferrús, Patrick Gélard and Nicolas Kuhn, “Software Defined Satellite Cloud RAN”. International Journal of Satellite Communications and Networking, 2017.

[11] H2020 VITAL Deliverable D2.3, http://www.ict-vital.eu/documents/deliverables

[12] Toufik Ahmed, Ramon Ferrus, R., Fedrizzi, R., and Sallent, O. (2017, May). Towards SDN/NFV-enabled satellite ground segment systems: Bandwidth on Demand use case. In Communications Workshops (ICC Workshops), IEEE International Conference on (pp. 894-899). IEEE. May 2017.

[13] NetWorld2020, “Public Private Partnership in Horizon 2020: Creating a Smart Ubiquitous Network for the Future Internet”, November 2013.

[14] A. Francescon, G. Baggio, R. Fedrizzi, R. Ferrús, I. G. Ben Yahia, R. Riggio, “X–MANO: Cross–domain Management and Orchestration of Network Services”, 3rd IEEE Conference on Network Softwarization (NetSoft 2017), Bologna (Italy), 3-7 July 2017

[15] P. Quinn and T. Nadeau, “Problem Statement for Service Function Chaining” IETF RFC 7498, April 2014.

[16] ETSI ETSI GS NFV-MAN 001 V1.1.1 (2014-12), “Network Function Virtualisation (NFV); Management and Orchestration”.

[17] J.Bisschop,. AIMMS optimization modeling. Lulu. com, 2006.

[18] Roberto Riggio, Abbas Bradai, Tinku Rasheed, J. Schulz-Zander, S. Kuklinski, and Toufik Ahmed, “VNFs Orchestration in Wireless Networks,” In Proc. of IEEE CNSM, 2015.

[19] M. Barshan, H. Moens, S. Latre, and F. De Turck, “Algorithms for efficient data management of component-based applications in cloud environments”, in Proc. of IEEE NOMS, 2014

[20] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network Function Virtualization: State-of-the-Art and research challenges”, In IEEE Communications Surveys & Tutorials, 2015.

[21] B. Addis, D. Belabed, M. Bouet, S. Secci, “VNFs Placement and Routing Optimization”. In Proceedings of IEEE Cloud Networking Conference (CLOUDNET), 2015.

[22] Luizelli MC, Bays LR, Buriol L, Barcellos MP, Gaspary LP. "Piecing Together the NFV Provisioning Puzzle: Efficient Placement and Chaining of VNFs”. In IFIP/IEEE Integrated Network Management Symposium, 2015.

[23] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and Placing Chains of VNFs”, In Proc. IEEE 3rd Int. Conf. CloudNet, 2014.

[24] Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. “On Orchestrating VNFs.” Network and Service Management (CNSM), 2015 11th International Conference on. IEEE, 2015.

[25] T. Taleb, M. Bagaa, and A. Ksentini. "User mobility-aware virtual network function placement for virtual 5G network infrastructure." IEEE International Communications Conference, 2015.

[26] A. Baumgartner, V. S. Reddy, and T. Bauschert, “Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization,” in 1st IEEE Conference on Network Softwarization (NETSOFT). IEEE, 2015.

[27] M. Jiang, M. Condoluci, and T. Mahmoodi, “Network slicing in 5G: an auction-based model”. In IEEE International Conference on Communications. 2017.