Zero-Touch Network Slicing Through Multi-Domain Transport Networks

Ricard Vilalta[1], Pol Alemany[1], Ramon Casellas[1], Ricardo Martínez[1], Carlos Parada[2], José Bonnet[2], Felipe Vicens[3], Raul Muñoz[1]

[1] *Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)*

[2]*Altice Labs*

[3]*ATOS*

e-mail: ricard.vilalta@cttc.es

ABSTRACT

The concept of network slicing is gaining a lot of interest and being under notable discussion within SDOs such as 3GPP, NGMN, ETSI and 5G Americas. A Network Slice is end-to-end service composed by a set of network functions, and the resources to run these network functions, forming a complete instantiated logical network, which meets certain network characteristics.

In this paper, we provide an overview of the design and development of a network slice manager component inside the SONATA Service Platform. The proposed component is able to align with ETSI NFV and 3GPP models.

Keywords: 5G, Network Slicing, NFV, 3GPP, ETSI, Transport Networks

1. INTRODUCTION

A Network Slice Instance (NSI) is defined in [1] as a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical end-to-end network, which meets certain characteristics in terms of available bandwidth, latency, or QoS, among others described in 5QI (5G QoS Indicator).

3GPP has proposed a data model [2], which consists of a list of Network Slice Subnetworks Instances (NSSI), which contain a set of network functions and the resources for these network functions which are arranged and configured to form a logical network. ETSI NFV and 5G Americas directly map a NSI with Network Services (NSs), saying basically that the sub-network instances are the same as NSs [3].

The NSI contains NSSIs, which in turn contain Network Functions (NFs) (e.g., belonging to Access Network (AN) and Core Network (CN)), as well as all relevant information to the interconnections between these NFs like topology of connections and individual link requirements (e.g. QoS attributes). The NSI is created by using a Network Slice Template (NST).

ETSI NFV is actually highlighting the relationship between NSs and Slices/Subnetwork Slices [3]. This is important since the SP is familiar and supports the NS (and VNF) constructions. Figure 1.a depicts how the 3GPP Network Slicing architecture might be aligned with ETSI NFV. Figure 1.b shows the suggested [3] data model alignment between 3GPP and ETSI NFV, where NSs might be implemented as Subnetwork Slices. 3GPP Network Functions might be related to VNFs or PNFs.

This paper proposes the introduction of a Network Slice Manager in SONATA Service Platform (SP) [4] and its experimental validation. Previously, SONATA SP has been presented as a MANO framework for deployment of NSs on top of multiple NFV Infrastructure Points of Presence (NFVI-PoP) interconnected across multiple network domains.

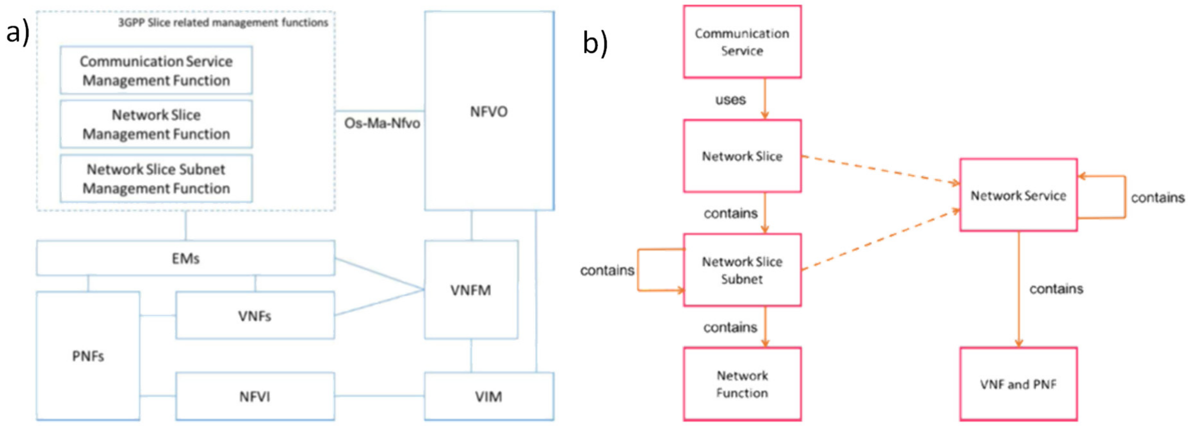

Figure 1. a) 3GPP Network Slicing and ETSI NFV alignment - Source [3]; b) 3GPP and ETSI NFV data model alignment - Source [3].

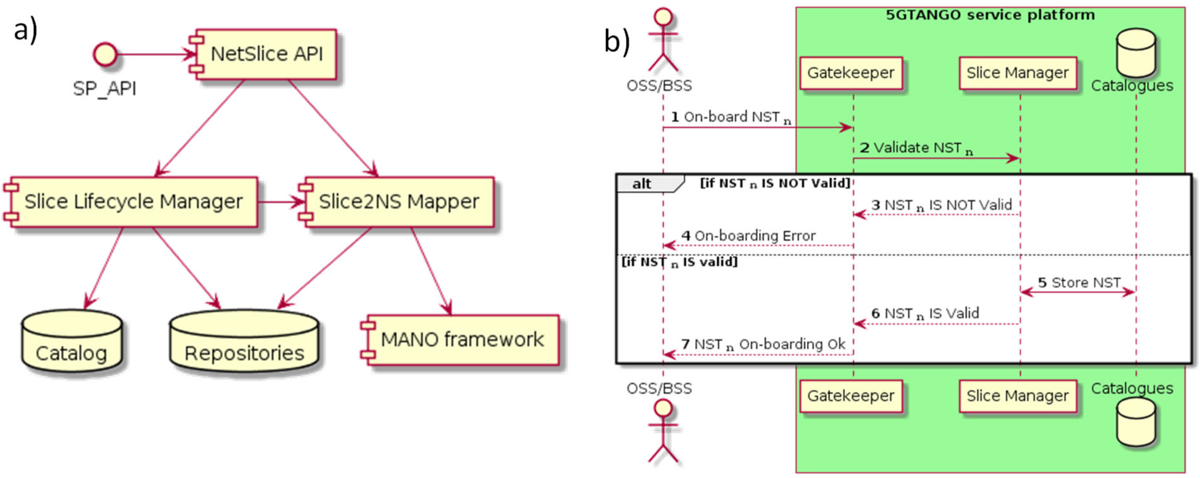

Figure 2 a) SONATA Slice Manager's internal architecture, b) Network Slice Template Creation

2. SLICE MANAGER ARCHITECTURE

The proposed Network Slice Manager internal architecture is shown in Figure 2.a. The Network Slice Manager is a SONATA Service Platform (SP) functional block that interacts with OSS (Operations Support Systems) and is responsible to access the SP MANO APIs for NS management.

The Network Slice Manager consumes the REST APIs exposed by the MANO Framework (reference point Os-Ma-nfvo [5]). Within a Network Slice Manager, one can identify two main functions:

- The first function is responsible for managing the lifecycle of these slices, assigning NSs and VNFs and applications to network slices. We will call it the Slice Lifecycle Manager.

- The second function is responsible for mapping network slices to NSs. We will call it the Slice2NS Mapper.

2.1 Descriptors

The NST and NSI Record (NSIR) proposed YAML Schemas are based on ISO/IEC DIS 19086-2. They are detailed in [6]. A Network Slice Template (NST) refers to the descriptor that details the Network Slice descriptor, which incorporates the requested NSs to be deployed for a network slice instance, and the integration among them. A NSIR refers to the descriptor that includes the information of a running NSI, which will use a network slice template reference for its deployment. The NSIR will include references to the deployed NS Records (NSR).

2.2 APIs

In this section, we present the proposed NBI for the Network Slice Manager Component.

2.1.1 Network Slice Template APIs

Table 1 shows the API primitives to manage NSTs.

Action	HTTP Method	Endpoint
CREATE NST	POST	/api/nst/v1/descriptors
GET ALL NST	GET	/api/nst/v1/descriptors
GET SPECIFIC NST	GET	/api/nst/v1/descriptors/{uuid}
DELETE NST	DELETE	/api/nst/v1/descriptors/{uuid}

Table 1 Network Slice Template APIs

2.1.2 Network Slice Instance APIs

Table 2 shows the API primitives to manage NSIs. Note that the instantiation of a Network Slice is achieved with the first endpoint of this API, /api/nsilcm/v1/nsi.

Action	HTTP Method	Endpoint
Create instance NSI	POST	/api/nsilcm/v1/nsi
Terminate NSI	POST	/api/nsilcm/v1/nsi/{nsiId}/terminate
GET all NSIs	GET	/api/nsilcm/v1/nsi
GET NSI	GET	/api/nsilcm/v1/nsi/{nsiId}

Table 2 Network Slice Instance APIs

3. DESIGN AND IMPLEMENTATION

The first release of the Slice Manager has been programmed using Python2. Several libraries are used, such as Flask, flask-restful, python-dateutil, python-uuid. Flask library is extensively used for the Slice Manager North Bound Interface (NBI) API programming, as each API call results in a flask route. Moreover, connectors to Repositories and MANO Framework have been introduced for the interaction with other SP components, to store NSIRs and manage the NSs lifecycle, respectively.

3.1 Slice Lifecycle Manager

The Slice Lifecycle Manager (SLM) is responsible for the entire lifecycle management of the created NSI, until it is terminated. Subsequent lifecycle events are likely to have an impact on the lifecycle of the underlying NSs, but not systematically.

The SLM function is responsible for the definition and update of NSTs. If a customer facing service needs to be assigned to an existing NST, it can request a NST update. If not, a novel NST, might be created.

NSTs are then converted by the Slice2NS mapper into a list of NSDs and flavours. Once the NSDs have been on-boarded, a slice instantiation request can be triggered, resulting in multiple NS instantiation requests with the appropriate flavour identifier being sent to the MANO Framework. The SLM is responsible for interacting with SP Repositories and Catalogues.

3.2 Slice2NS Mapper

The Slice2NS Mapper has to maintain an association between NST and NSDs identifiers, as well as an association between NSI identifiers and NS instance identifiers. In next releases, it might also deal with the required combination/integration/concatenation of NSs.

4. EXPERIMENTAL VALIDATION

This section focuses on demonstrating the proposed workflow diagrams involving the interaction of the Slice Manager and the SONATA Service Platform.

4.1 Creation of a NST

The Slice Manager is the service platform new component, responsible to manage the life cycle of Network Slices. The on-boarded NSTs are stored in the Catalogue after previous validation. The NST on-boarding workflow is depicted in Figure 2.b. Depending on the validation level to be performed, the Catalogue could also be involved, to verify whether the comprising NFV building blocks (NSs) are available.

4.2 Instantiation of a NSI

Once a NST is on-boarded, multiple NSI can be created using this template. The Network Slice instantiation triggers the instantiation of the list of underlying NSs defined in the NST, which are used as building blocks to compose the NSI. The multiple NSs may need to be combined to form a meaningful network/service.

OSSs are the entities that usually trigger the Network Slice instantiation, although it may also be triggered automatically (autonomous behaviour). The NSI creation workflow is depicted in Figure 3.a.

Note: For the sake of simplicity, the workflow does not show the situation where any NS cannot be deployed (e.g. lack of resources). In this case, the NSI creation is aborted and all NSs already deployed are terminated.

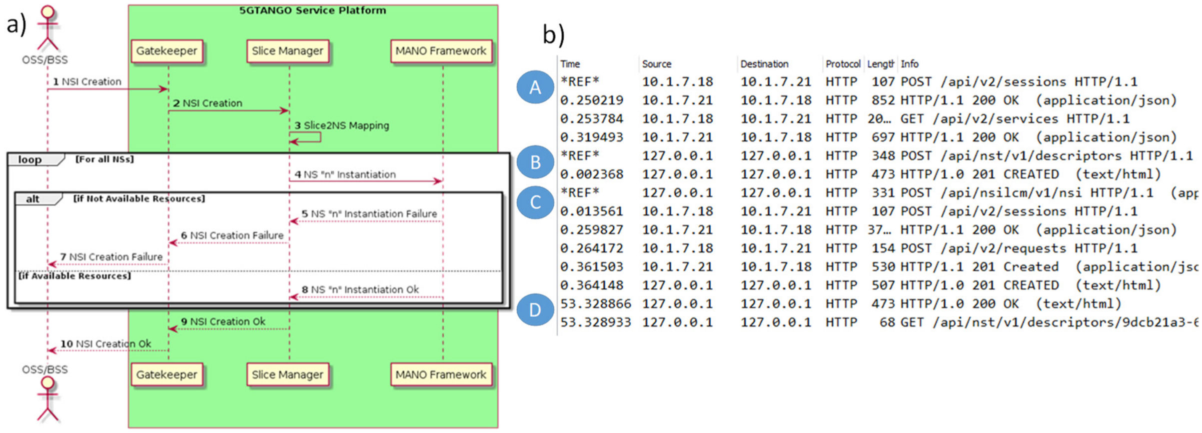

Figure 3 a) Network Slice instantiation, b) Wireshark capture between Slice Manager and MANO

4.3 Evaluation of NST on-boarding and NSI instantiation on SONATA Service Platform

Finally, Figure 3.b shows the Wireshark capture between the OSSs, Slice Manager and MANO Framework. Access and information retrieval of network services is shown in Step A. On Step B, the OSSs request the on-boarding of an NST, which might include the necessary network services and how are they related. On Step C, OSSs request the network slice instantiation. In our testbed, the NST contains a single network service, and we can observe the deployment time for the complete NSI of around 58s (Step D). This is highly dependent on the underlying VIM (Virtual Infrastructure Manager) and NFV Infrastructure (NFVI).

5. CONCLUSIONS

We have presented a general scenario for multi-site NFVI-PoP that supports network slicing and multi-tenancy. We have validated the proposed Network Slice Manager and measured the deployment time of a simple Network Slice, based on a single network service.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from EC H2020 5GPPP project 5GTANGO (761493) and MINECO project DESTELLO (TEC2015-69256-R).

REFERENCES

[1] Mschner K. et all Hedmar, P. Description of network slicing concept. NGMN Alliance document, January 2016.

[2] 3GPP. 3GPP TR28.801 study on management and orchestration of network slicing for next generation network, 2017.

[3] NFVISG ETSI. NFV-EVE012 report on network slicing support with ETSI NFV Architecture Framework, 2017.

[4] Dräxler, Sevil, et al. "Sonata: Service programming and orchestration for virtualized software networks." Communications Workshops (ICC Workshops), 2017 IEEE International Conference on. IEEE, 2017.

[5] NFVISG ETSI. NFV-IFA013 Os-Ma-nfvo reference point -interface and information model specification, 2016.

[6] SONATA GitHub, https://github.com/sonata-nfv/tng-schema.