Enabling Multi-segment 5G Service Provisioning and Maintenance through Network Slicing

Rafael Montero, Fernando Agraz, Albert Pagès, Salvatore Spadaro
Optical Communications Group (GCO)
Universitat Politècnica de Catalunya, Jordi Girona 31, 08034 Barcelona, Spain
e-mail: rafael.montero@tsc.upc.edu

ABSTRACT
The current deployment of 5G networks in a way to support the highly demanding service types defined for 5G, has brought the need for using new techniques to accommodate legacy networks to such requirements. Network Slicing in turn, enables sharing the same underlying physical infrastructure among services with different requirements, thus providing a level of isolation between them to guarantee their proper functionality. In this work, we analyse from an architectural point of view, the required coordination for the provisioning of 5G services over multiple network segments/domains by means of network slicing, considering as well the use of sensors and actuators to maintain slices performance during its lifetime. We set up an experimental multi-segment testbed to demonstrate end-to-end service provisioning and its guarantee in terms of specific QoS parameters, such as latency, throughput and Virtual Network Function (VNF) CPU/RAM consumption. The results provided, demonstrate the workflow between different network components to coordinate the deployment of slices, besides providing a set of examples for slice maintenance through service monitoring and the use of policy-based actuations.
Keywords: 5G, Network Slicing, Network Monitoring, Sensors and Actuators.

1. INTRODUCTION

5G networks introduction comes in a period where the paradigm of network requirements is changing, driven by the move towards a fully mobile and connected society [1]. Communications in turn, start showing new traffic patterns thanks to the rise of machine-to-machine and machine-to-human applications besides the common human-centric ones, thus bringing more traffic to present-day networks. The denominated 5G services encompass this wide variety of service characteristics to address solutions for a set of market verticals and industry sectors (i.e., automotive, industry 4.0, e-health, smart city, etc.), considering between them the support of different Key Performance Indicators (KPIs) related to the requirements established for a set of 5G service types such as enhanced mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC), and massive machine-type communications (mMTC).

In this sense, pushing forward towards next-gen low latency and high capacity 5G networks has brought new challenges. The struggle to accommodate and guarantee the performance of different services over the same physical infrastructure has aroused the use of concepts such as network slicing, which entails partitioning the network either physically or virtually to isolate resources for deployed services, in a way to secure their agreed performance levels while maintaining as well a level of coexistence and isolation among them. Moreover, novel technologies as Software Defined Networking (SDN) and Network Function Virtualization (NFV) have also risen to bring configurability and flexibility to networks. In particular, the SDN concept looks forward to the separation of control and infrastructure layers, decoupling the intelligence from network devices and bringing it to a higher control level, which enables configuring and exposing all the underlying infrastructure from a single point without going through the legacy per device methodology. On the other side, NFV introduces the virtualization of network functions, providing specific functionalities deployed purely in software at a required physical location. In turn, a network service can be composed of one to many virtual network functions (VNFs), which work together to enable the overall service functionality.

All of these technologies and techniques introduced to leverage current networks performance to the 5G standards entail the use of new software modules across architectural layers in order to organize network, compute and storage related functionalities. In this regard, proposals to set up an architecture layout for 5G have been presented in [2-4], where the role of such components is defined. The challenge here, comes when services must be deployed over a set of underlying network segments (e.g., Data Centre, Core, Metro, Access). This usually means that a coordination is required between components and segments to enable instantiating or configuring resources for the provisioning of the service, as well as for being able to maintain it during its lifecycle. Such coordination has received attention from the community, as in [5-11], in order to better define how to orchestrate required configurations upon service request and how to gather specific information related to each service levels of Quality of Experience (QoE) and Quality of Service (QoS).

Regarding the 5G architecture, both ETSI and 3GPP driven views have been analysed in [12], elaborating on a consensus “meta-architecture” in a way to summarize all these efforts. Such meta architecture, despite not being necessarily precise or definitive, intends to provide a common structure for the network elements in charge of

handling the management and operation of a 5G system. Fig. 1 next, depicts a single-domain version of the 5G meta architecture focused on the use of the network slicing concept.

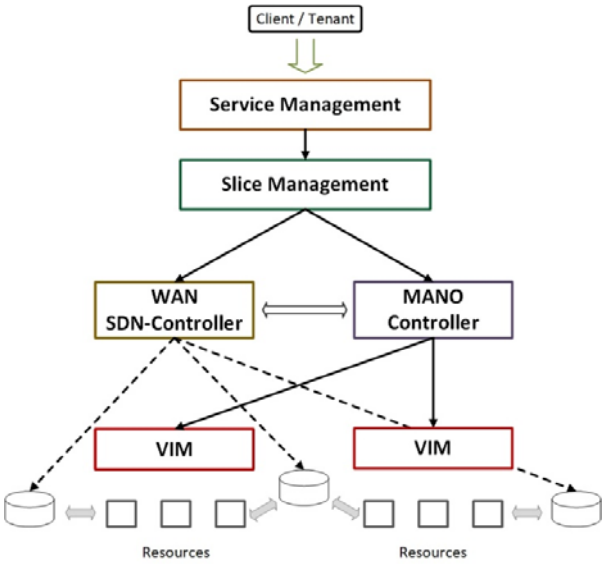

Figure 1. Single-domain 5G Meta Architecture.

The structure of the architecture, entails 5G services to be delivered in the form of network slices by means of specific management related components. In this sense, each slice is configured and maintained triggering requests at the slice-level through elements such as Management and Orchestration (MANO) controllers and Wide Area Network (WAN) SDN controllers. By these means, the required computing and storage resources can be allocated at a particular segment via its Virtual Infrastructure Manager (VIM), while network subsystems at the WAN can also be directly configured according to the necessary inter-resource connectivity.

Using as a reference these architectural approaches given by standardization groups and the efforts of the community to prove their applicability, this work introduces a network slicing-based framework for 5G service provisioning and maintenance putting emphasis on single-domain multi-segment scenarios. In particular, it takes the point of view of the Network Slice Instance (NSI) lifecycle to set up two versions of the framework. First, the one defined for the provisioning stage is presented, focusing on the use of the Slice Composition technique to manage per-segment configurations by means of a novel coordinator component. Following the 5G architecture in Fig. 1, the introduced component takes charge of the Slice Management functions and coordinates operations with the MANO and WAN SDN controllers. Then, another version related to the maintenance stage of the NS is presented. In this case, focus is on the gathering of specific parameters (i.e., latency, throughput, VNF CPU/RAM) from each slice to further analyse them and act preventively upon possible service degradation following a policy-based approach. Both stages are finally demonstrated experimentally in an emulated multi-segment testbed.

The remainder of the paper is structured as follows, Section II introduces slice provisioning and its architectural behaviour taking in consideration the NSI lifecycle and the positioning of network components against the current 5G standards. Then, Section III further analyses the architecture from the slice maintenance point of view. Section IV presents the experimental testbed that has been used for the functional validation, demonstrating the provisioning stage. Section V in turn, focuses on the maintenance of slices, presenting different scenarios for guaranteeing QoS through sensing and actuation. Section VI finally rounds up the main achievements of this work.

2. NETWORK SLICE PROVISIONING

As introduced previously, network slicing technology enables the provisioning of 5G services over a common infrastructure. To achieve this, a NSI is defined for each service instance requested by the client/tenant, containing all functionalities and resources required to support it. These are particularly arranged and configured, forming a complete logical network to fulfil specific network characteristics, compliant with required service KPIs [13].

The lifecycle of an NSI is then completely independent of the one of the service instance. In particular, it begins with a preparation stage, where the network environment is prepared and slice templates are defined. Once all arrangements have been set up, the provisioning stage begins, being the focus of this section. In this stage, all resources shared or dedicated to a particular service are created and configured. A VNF can then be

instantiated through a VIM at a specific location, while the network path to provide its connection with another VNF can be set up via a network controller at the WAN. The provisioning ends at the time the NSI becomes active, considering all configurations and actions to have been performed and that the required functionalities are reached to enable the service.

Besides containing Network Functions (NF), either physical or virtual, and the information regarding the interconnection of these NFs, a NSI can also contain or be composed of lower-level Network Slice Subnet Instances (NSSI). In this case, a NSSI usually defines resources and functionalities required at a particular network segment in a way to better orchestrate the overall slice provisioning. A more in-depth look on how NSSI given partitioning can provide NSI deployment is given in the next sub-section.

A. Slice Composition

The 5G-PPP introduced the Slice Composition concept (i.e., "slice-cum-slice") in [2]. The idea behind this concept is to enable composing a slice out of individual slices. That means, building a multi-segment NSI by combining a set of per-segment NSSIs. This allows for specific configurations and actions corresponding to a particular segment to be executed independently (i.e., NSSI provisioning), thus providing of more flexibility to the NSI in case of requested modifications related to functionality or due to required actions for maintenance. Needless to say that it also allows identifying and isolating with higher accuracy potential failures or malfunctions that could threaten the overall slice performance.

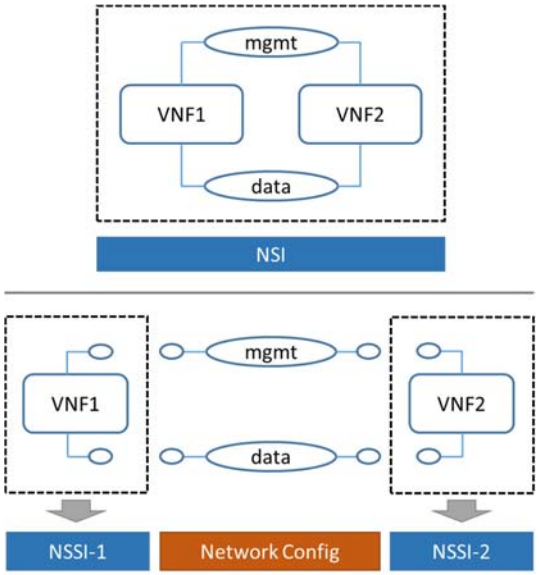

Figure 2. NSI provisioning through Slice Composition.

Fig. 2 depicts a case where a NSI, containing two VNFs interconnected via Management and Data networks, is de-composed into two different NSSIs. Once the slice is partitioned, NSSI-1 and NSSI-2 can be provisioned over different network segments, usually according to resource and functionality availability. One common use case is when a service requires specific functions to be deployed closer to the user (e.g., Edge Data Centre) while others may be present at other parts of the network (e.g., Core Data Centre). The network configuration is then finally triggered in order to provide connectivity between both NSSIs, thus concluding the NSI provisioning by enabling functions to switch to an operative state.

B. Information Models Correlation

Considering the levels of operation of network components at the 5G architecture, it is important to clarify the relation between the different information models to be used. In this regard, 3GPP introduced a 5G Network Resource Model where NSI, NSSI, and Managed Functions (MF) are defined for Network Slicing [14], while ETSI proposed an Information Model with the use of Network Services (NS) and VNFs for enabling NFV [15]. This means that a correlation between these models is required at some point of the 5G service provisioning and maintenance lifecycle in order to allow communication between components related to the 3GPP architecture and the ones aligned to ETSI NFV (e.g., the MANO controller).

Fig. 3 shows the relation of these two models given by the ETSI in [16]. In particular, the required correlation between models is set at the NSSI to NS level considering a direct mapping between these resources. In turn, the MFs and the VNFs are also directly correlated. By these means, upon the request for NSI instantiation at the Slice Management level, the NSI can be furtherly decomposed into NSSIs and triggered via the MANO controller in the form of NSs. Moreover, each NS will include the definition for the service related VNFs, thus

mapping the required MFs. The component in charge of performing this correlation in introduced in the next subsection.

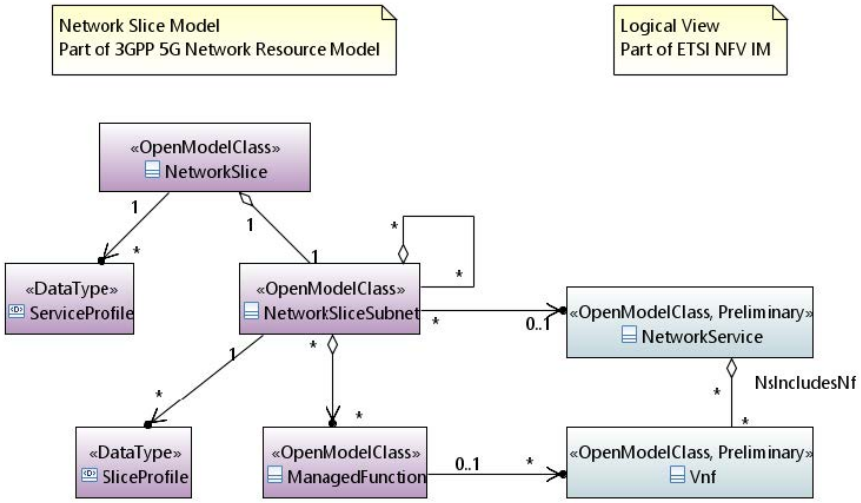

Figure 3. Relation between 3GPP 5G Network Resource Model and ETSI NFV Information Model.

C. NVF-Coordinator Component

As exposed in the introduction section, orchestration of per-segment configurations brings a challenge to the definition of the 5G-enabled architecture. In this way, and following our previous definitions on this matter [17], this work furtherly introduces a software component capable of achieving service provisioning and maintenance through network slicing by means of the slice composition technique, besides serving as the correlation entity between network slicing and NFV related information models. More specifically, the defined NFV Coordinator (NFV-C) component looks forward in serving as a middle-point between service clients/tenants connected via Operation Support Systems (OSS) / Business Support Systems (BSS) and the components of the 5G Architecture.

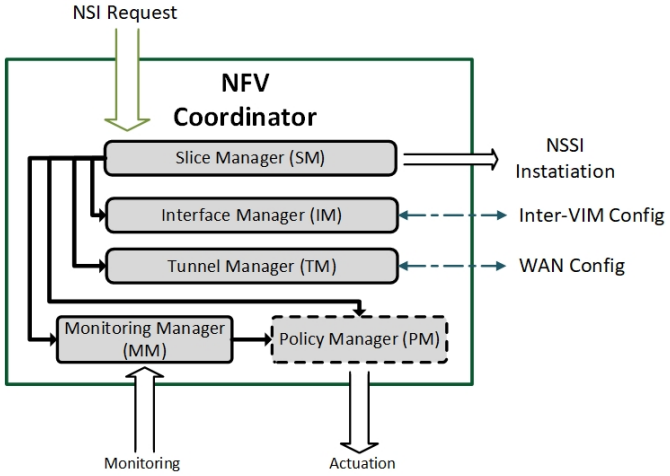

Figure 4. NVF-Coordinator Architecture.

Looking more in detail, the inner architecture of the NFV-C is provided with specific software modules as depicted in Fig. 4. These modules provide the interaction with other components (e.g., OSS/BSS, MANO controllers, WAN SDN controllers) and the behaviour necessary to fulfil tasks in NSI and NSSI lifecycles. The role of each module is described next:

- **Slice Manager (SM). -** Main coordination point between client service request and network. Handles requests for NSI instantiation and decomposes these into NSSIs, providing the required mapping to NS for triggering NSSI deployment via the MANO controller. Coordinates the connectivity set up between different

Points of Presence (PoP) through IM and TM. Performs initial configuration of monitoring and actuation frameworks via MM and PM.

- **Interface Manager (IM).** - Collects VIMs interface data. Using this information, configures routing/chaining between VNFs across different NSSIs. Interacts with the Inter-VIM Manager modules at SDN controller level.
- **Tunnel Manager (TM).** - Configures the overlay tunnel and triggers the required WAN network configurations to enable connectivity between NSSIs. Interacts with the WAN SDN-controller(s).
- **Monitoring Manager (MM).** - Handles configuration and deployment of network sensors. Collects specific parameter data, related to each service KPIs, from different layers of the architecture according to the monitoring configuration set by the SM.
- **Policy Manager (PM). -** Analyses gathered data by comparing it, for example, to pre-established safe thresholds given by the SM and based on a policy system (i.e., event-condition-action model). Eventually, it triggers actuations/actions at specific segments and through specific network components to guarantee the overall QoS of the NS.

As it is described in the modules behaviour, the IM and TM become crucial in the orchestration of the NS at the provisioning stage, while the MM and PM are the ones to guarantee its performance at the maintenance stage. The SM in turn, takes a major role in both, coordinating the whole NSI and NSSI lifecycle.

D. Slice Provisioning Architecture

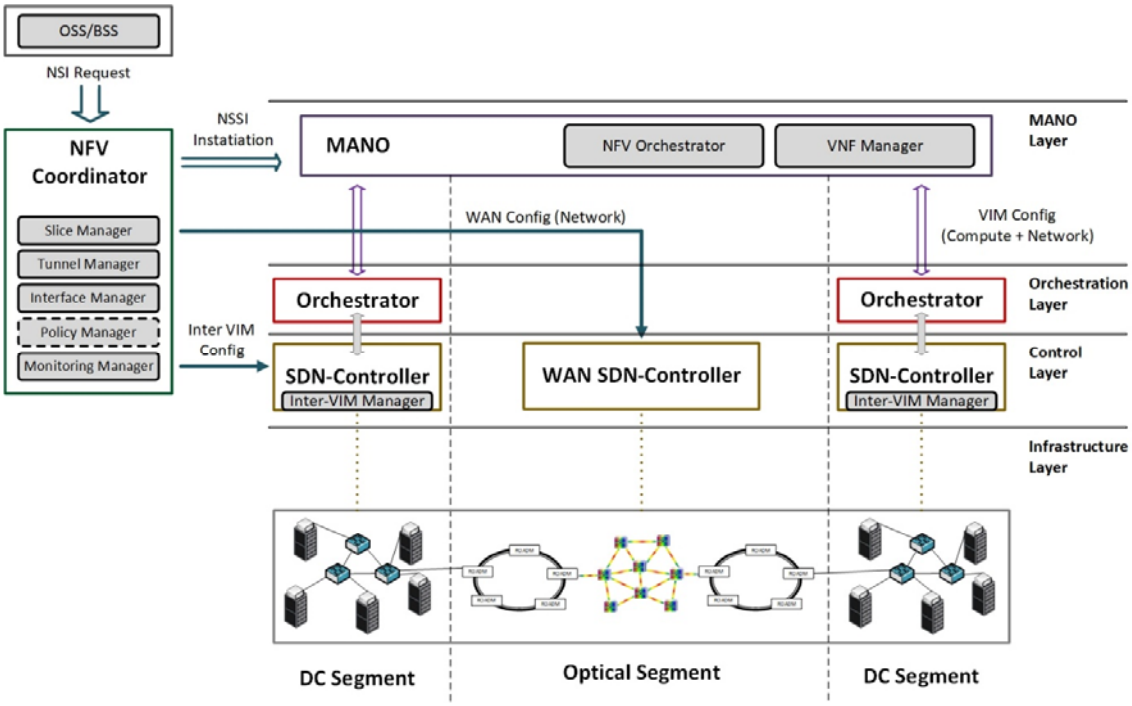

Figure 5. Architecture for Multi-segment Network Slice Provisioning.

Fig.5 depicts the framework presented in this work tailored for the NSI Provisioning stage. The architecture is layered in different levels according to the role defined for each of its components in a 5G single-domain multi-segment scenario. Looking from a bottom-up perspective the layers are:

- **Infrastructure Layer. -** It comprises all the computational and network resources at the physical level, distributed across different network segments. These resources expose their capabilities to upper layers via southbound protocols, opening a communication path so configuration actions can be requested on them, as well as allowing the gathering of real-time monitoring data.
- **Control Layer.** - At this level reside the components capable of providing the intelligence to configure underlying network resources. The SDN-Controller in turn, contains a set of plugins to enable communication with physical devices at the southbound, besides defining the Application Programmable Interfaces (API) used to expose network topology and functionalities to applications at the northbound. Finally, it also defines specific software modules designed to enable network configurability, providing as well databases and data buses to allow inter-module communication and data storage.

- **Orchestration Layer.** - An Orchestrator entity is capable of handling the provisioning of computational resources, enabling the instantiation of network functions in form of a VNF, VM, container, etc. Moreover, it is also in charge of requesting the underlying SDN-Controller to provide the required network connectivity between deployed resources. This kind of component is usually present in segments allocating both computational and network resources (e.g., Data Centres).
- **Management and Orchestration (MANO) Layer.** - A MANO entity is responsible of orchestrating NSSI level resources, in the form of NS requests, towards the registered VIMs and configuring the deployed NFs, either at deployment or runtime. In this way, it is the one in charge of managing NS and VNF lifecycles, in addition to providing the required templates for their definition, given as Network Service descriptors (NSD) and VNF descriptors (VNFD) [13]. In particular, VNFDs contain pre-defined charms or cloud-init files that describe the required configurations to be done at VNFs [18]. Developments on this level are pushing forward towards adding more functionalities, such as handling the triggering of WAN connectivity set up between NSSIs, supporting hybrid network services (i.e., VNFs and PNFs), improving monitoring, providing policy-based slice performance guarantees, among others [19].

Upon the event of a new NSI request by the OSS/BSS, the SM at the NFV-C takes charge of the NSI instantiation, by analysing the required resources for the provisioning of the service. Then, it decomposes it into NSSIs, according to the demanded resources, and triggers their deployment via the MANO entity. This one in turn, deploys the NSs by contacting the requested VIMs to set up VNFs instantiation and configuration, considering as well the required network connectivity in these segments (i.e., Intra-VIM Config). As VNFs are instantiated, the IM collects their interface data and configures necessary routing/chaining between functions (i.e., Inter-VIM Config). In parallel, the TM sets up the overlay tunnel and network configurations at the WAN. After VNFs have completed their boot up and connectivity has been enabled, the end-to-end service functionality given by the combination of NSSIs becomes operative. The provisioning stage ends with the SM passing the specifics on QoS parameter monitoring and established safe thresholds to the MM and PM.

E. Mapping to the 3GPP / ETSI NFV Framework

After analysing the introduction of the NFV-C and the architecture for 5G service provisioning, it is important to understand its mapping towards the existing standards. With this purpose, and following the discussion given at the introduction section regarding the definition of a meta architecture, Fig. 6 depicts the relation of the proposed architecture towards the 3GPP and ETSI NFV framework.

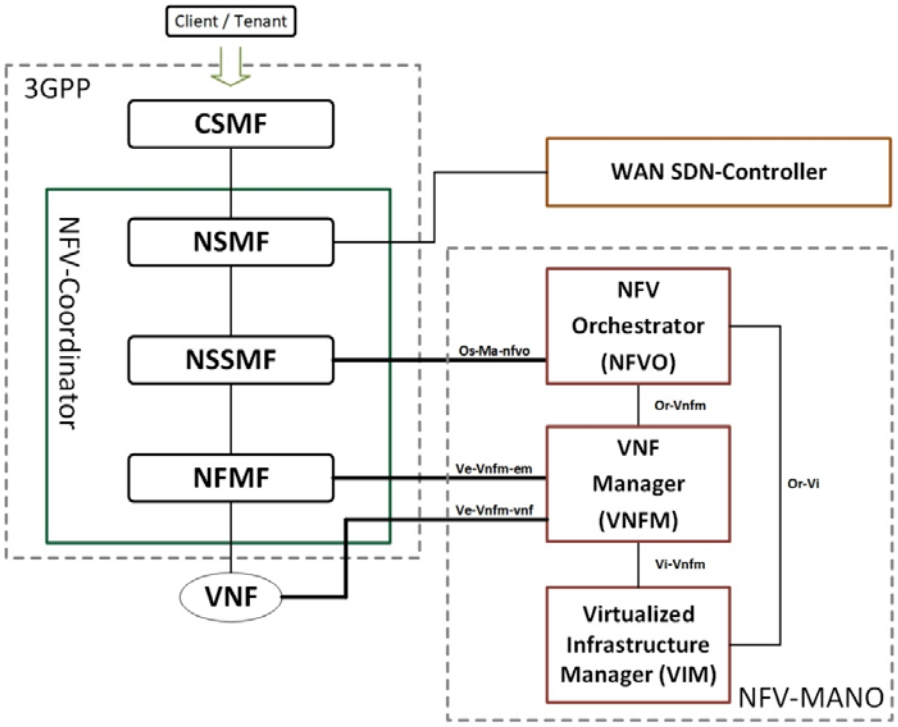

Figure 6. Architecture Mapping to the 3GPP / ETSI NFV Framework.

The figure bases on the common study from the 3GPP and ETSI given in [20], where a first approach for the relationship between both frameworks is presented, considering the communication interfaces between modules

of the 3GPP and NFV-MANO. In this way, the role of the NFV-C and its internal components can be mapped according to provided functionalities.

Starting from the client/tenant request for new service provisioning, the Communication Service Management Function (CSMF) module takes charge of analysing the demands and requesting for a new NSI. The CSMF in turn, could be represented in the form of an OSS/BSS being in charge of Service Management, and of triggering the NSI request towards the Network Slice Management Function (NSMF) module. This one in turn, handles the NSI lifecycle, and decomposes each slice it into smaller segment-specific NSSIs to be instantiated via the Network Slice Subnet Managed Function (NSSMF) module.

The NSSMF, is the one responsible of triggering and configuring NSSIs via the Os-Ma-nfvo interface that connects to the NFV Orchestrator (NFVO) component of the NFV-MANO, following the interface specification [21]. For this, the NSSI to NS mapping should be performed by the NSSMF, so the NFVO can process each NS request and coordinate their instantiation with the VNF Manager (VNFM) and the underlying VIMs. Finally, the Network Functions Managed Function (NSMF) module at the 3GPP framework, can also take charge of providing configurations for VNFs and gathering specific monitoring data from them for fault and performance management.

Taking into account the NFV-C operational level, its role can be defined at the Slice Management level when considering the consensus architecture layout of Fig.1, Then, analysing each of its internal modules, the SM would take the NSMF and NSSMF functionalities, coordinating the slice provisioning and maintenance, while the IM and TM will be the ones triggering connectivity setup towards the WAN SDN-Controller(s). The MM and PM in turn, would take part of the NFMF functions related to VNF monitoring to enable policy-based slice maintenance. In this sense the NSI and NSSI lifecycle will be handled by the NFV-C, while the NS and VNF lifecycles will be delegated to the MANO controller, which in turn plays the role of the NFVO and VNFM.

F. Architecture Comparison to the State of the Art Works

Given the architecture presented in this work, and after analysing its functionality and the scope of operation of its main components. It becomes crucial to make a comparison with other efforts from the community [5-11] set to prove the applicability of the standards [2-4]. In this regard, Table 1 summarizes the main characteristics of some of these approaches with the focus set on comparing the drawbacks/advantages between them.

Table 1. Characterization of community-driven approaches towards the 5G Architecture

	Domain of Operation	**Highest level of Operation**	**3GPP 5G Framework Compliant**	**ETSI NFV Framework Compliant**	**3GPP to ETSI Information Model Support**
Proposed in this work	Single-domain	Slice Management	Yes	Yes	Full mapping support
5G-Network Slice Broker	Single-domain	5G-Service Management	Yes	No	Not mentioned (slices only)
5G-Crosshaul	Multi-domain	Slice Management	No	Yes	Not mentioned
5G-ICN	Multi-domain	5G-Service Management	Yes	No	Not mentioned (slices only)
SDN/NFV MANO Architecture for Dynamic VNF Services	Multi-domain	MANO	No	Yes	Not mentioned (net. services only)
CogNet	Single-domain	MANO	No	Yes	Not mentioned (net. services only)
SELFNET	Single-domain	MANO	No	Yes	Not mentioned (net. services only)

The comparison shown in the table, describes characteristics of these proposals such as their domain(s) and highest level of operation taking into account the standards. Besides this, the focus is also set on analysing the compliance with both 3GPP and ETSI frameworks and the support for mapping data between them following the

reference information models [14-15]. Based on this, the architecture proposed in this work, although considering its operation reserved to a single-domain on first hand, it achieves management of resources at the Slice level and provides the necessary mapping to correlate the 3GPP and the ETSI NFV models. In this way, it allows handling both NSI/NSSI and NS/VNF resources, presenting a valuable contribution to the state of the art.

3. SENSORS & ACTUATORS FOR SLICE MAINTENANCE

Continuing the lifecycle of the NSI, once its functionality is fully provisioned, it begins the runtime or maintenance stage. During this phase, the NSI is capable of handling traffic necessary to support a particular service. As previously introduced, the stage includes the monitoring of specific parameters relevant to the slice KPIs in a way to trigger required actions / actuations (e.g., reconfigurations, NSI modifications, upgrades) to guarantee service operation in line with the agreed Service Level Agreements (SLA).

Focusing on the monitoring task, network sensors allow gathering real-time data from network segments so this can be furtherly aggregated to the NSI level and processed. A sensor in turn, could be defined as any component capable of retrieving information regarding the current state of the service. In practice, it can be present in many forms (e.g., VNF, VM, container, application) and at different levels of the architecture (i.e., infrastructure layer, control layer, orchestration layer). Depending on the case a sensor could be also instantiated or configured at the provisioning stage or during runtime. In this section, we analyse the gathering of specific network parameters by the MM. More specifically, detail is given on how such data is exposed and whether it requires of specific configurations or component instantiations to make it available.

A. Latency Sensor

Latency is identified in 5G as one of the most crucial requirements, considering service demands with times below 1 millisecond. In this regard, we have presented in [22] a mechanism based on the use of a latency sensor, to test and gather the end-to-end latency between two VNFs with working traffic between them. The sensor is deployed in the form of a VNF and its strategically placed between the service VNFs to be measured. The idea is that the sensor would capture packets flowing through itself so it can use the arrival times of these to calculate the Round Trip Time (RTT) on both sides. With this data it can finally get the latency between functions and store it at a local database, to be exposed to higher-layer components.

Fig.7 gives a more in-detail look to the mechanism. The sensor in turn, uses the PACKET-LEFT arrival time (TPL) coming from VNF1 to calculate the round-trip time between sensor and VNF2 (RTT-R). It basically compares this time with the PACKET-LEFT-ACK arrival time (TPLA) corresponding to the same packet. On the other side, the same technique is used to get RTT-L between sensor and VNF1, using the PACKET-RIGHT (TPR) and PACKET-RIGHT-ACK (TPRA) arrival times.

Then, once RTT-R = (TPLA – TPL) and RTT-L = (TPRA – TPR) are calculated, it is possible to get the end-to-end latency (L), VNF1 to VNF2, by using L = ((RTT-L + RTT-R) / 2).

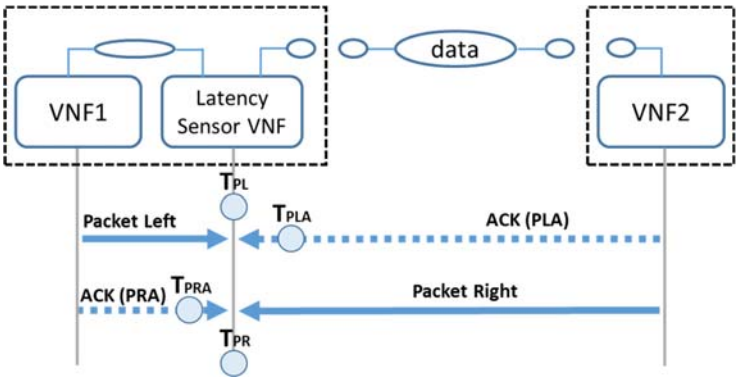

Figure 7. Latency Sensing Mechanism.

B. Throughput Sensing

The throughput between NFs deployed at different segments of the network can also provide significant value to test the current state of a service. In this case, in order to retrieve this data from the network the MM at the NFV-C should contact components at different levels of the architecture.

A first approach will be triggering the collection of NF metrics at the orchestrator level. By reading such data, it would be possible to get the amount of bytes/packets flowing through the interfaces of each VM, VNF or container involved in the operation of a particular service. By knowing the amount of data transferred in a specific period of time, the throughput between NFs can be calculated. On the other hand, if metrics collection is triggered at the controller level, then statistics on network subsystems (e.g., switches, routers, access-points) interfaces can

be retrieved. This would allow getting real-time statistics on the data flowing through specific links or segments of the network.

The hop-to-hop throughput can be measured then thanks to this data, making it possible to identify potential bottlenecks and network failures that could compromise the service operation.

C. CPU/RAM Gathering

Checking the state of allocated compute resources (e.g., VM VNF, container) can also help in analysing the current operation status of a slice. In this case, the component in charge of managing these resources (i.e., the orchestrator), is the one collecting all this data and making it available to other components, such as in [23].

The MM at the NFV-C in turn, must contact and request DC segments orchestrators, managing resources belonging to a particular service, to gather and store CPU/RAM metrics related to these resources. The idea is that the MM will be able to check whether there is a potential problem at the slice due to insufficient compute resources or if it is more related to a problem at the network. In the case that an abnormality is found at any VM, VNF or container, then corrective actions (e.g., VNF migration, VNF scaling, VM resizing) could be performed to guarantee the service agreed performance.

D. BER Monitoring

Focusing on the network part of the slice, the monitoring of the Bit Error Rate (BER) on optical network segments would be an important variable to check potential slice performance issues related to any malfunction or traffic saturation at the WAN.

The component enabling the collection of this metric would be the SDN-Controllers operating at optical segment level (e.g., Metro/Access, Core). A controller must gather and store this data at a local database, to make it available to other components such as the MM. Which in turn, should be able to make use of this information to trigger preventive actions in case BER reaches levels above the established safe thresholds. In particular, a rise of the BER could significate a physical problem at a specific network subsystem or could also be related to saturation due to overused paths by currently operative services. In this regard, actions triggered to overcome such events are usually related to path reconfigurations across the WAN.

E. Actuations over Monitored QoS

An actuation in 5G, is considered as any action triggered to solve any particular problem at a slice that could compromise its overall QoS. The components executing or orchestrating these actions are called actuators, and its main responsibility is to accommodate a slice to the current state of the network to guarantee its performance. As sensors, actuations could consider either deploying new components (e.g., VNF, sensor, application) or triggering configurations at different levels of the 5G architecture. More detail on this is given in the next sub-section in view of the whole maintenance framework depiction.

The framework presented in this work, follows a policy-based approach to perform actuations when required [24]. In particular, an ECA (i.e., event-condition-action) model is used to define specific behaviours and thresholds to be monitored at the slice, with their corresponding preventing actions. More specifically, the policy model defines an event (e.g., high latency), a condition to be met (e.g., greater than 1ms) and an actuation to be enforced (e.g., reconfigure service path). The PM module at the NFV-C is defined as the main component for the coordination of these tasks, providing a connection point between defined service requirements, monitoring data coming from the MM and the other network components.

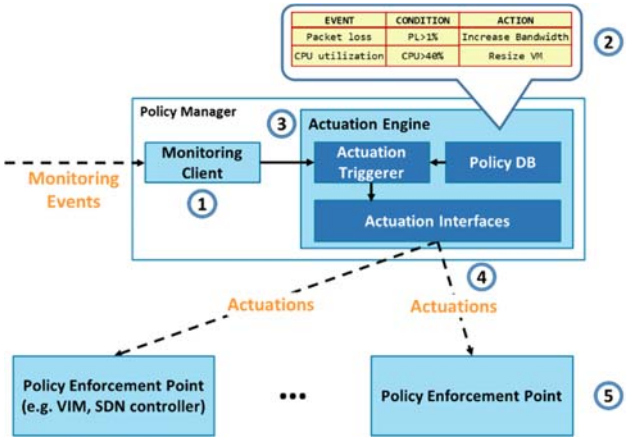

Figure 8. Policy Manager Architecture.

The architecture of the PM is depicted in Fig. 8. Its behaviour focused on the slice runtime stage, following policies set up during provisioning, is described in five steps:

1. **Receive monitoring events** - The Monitoring Client at the PM receives all monitoring data relevant to the KPIs of the slice, analysing specific parameters.
2. **Available policies for actuation (ECA model) -** Policy database, initialized during slice deployment, contains all the specifications required for the maintenance of the service.
3. **Determine policy to be applied -** The monitoring data is compared to the established thresholds at the DB. Following model instructions, upon violation of any of these thresholds, preventive actions/actuations are selected.
4. **Distribute actuations across enforcement points -** Set of required actions are coordinated through one to many components of the architecture.
5. **Apply desired (re-)configuration -** Once components receive instructions, configurations are pushed towards network or compute resources to modify specific characteristics of the slice, in a way to guarantee its proper operation.

It is important to consider that policies set for slice maintenance could also change during runtime due to service client requests or based on identified changes of the network performance. Besides, actuations can also be defined using other data analysis techniques such as Machine Learning.

F. Slice Maintenance Architecture

If the focus is set on the slice operation during runtime, then it is possible to analyse a different version of the presented provisioning framework. In this approach, the slice maintenance architecture puts emphasis on the components, such as the PM or MM, related to guaranteeing the normal operation of the slice.

Fig. 9 depicts the layered architecture. As it can be seen, the MM enables monitoring of the previously discussed parameters by requesting and gathering data from different components at different levels of the architecture. In line with this, the PM has also direct contact towards many layers, setting up required configurations when policies are executed. Besides these, other components are also involved in the process.

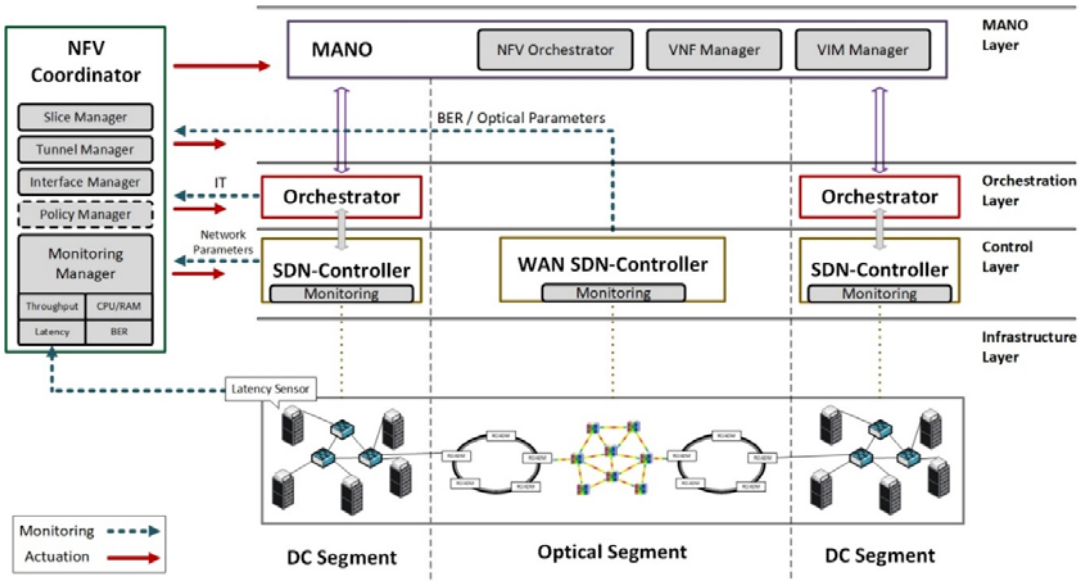

Figure 9. Architecture for Multi-segment Network Slice Maintenance.

In order to give a better understanding of the whole maintenance procedure, sensing and actuations tasks have been characterized according to the layer in which they operate:

- **Infrastructure Layer. -** Direct sensing at this level requires a deployed resource (e.g., Sensor) to gather and expose data straight from the data layer. Latency sensor is a good example for getting information of a running service without interacting with any other high-layer component. In case of actuations, reconfigurations of a sensor could be done directly, but in most cases changes at this level should be done via other network components (e.g., orchestrator, controller).
- **Control Layer. -** The SDN control level represents the main sensing point for network resources metrics. A segment controller in turn, is able of retrieving statistics data from network subsystems, either at the optical or electrical domain, via southbound protocols. This information is then stored at a local database and

usually exposed to clients using Representational State Transfer (REST) interfaces. As these components have wide control of the underlying infrastructure, they provide the tools for reconfiguring network resources in case any reconfiguration is required for slice maintenance. Then, the PM would contact any controller in case of identifying potential issues at any network segment (e.g., DC network, WAN), so they can configure the requested changes.

- **Orchestration Layer. -** Orchestration components on the other hand, gather and provide statistics data from the deployed compute resources (e.g., VM, VNF, container). CPU/RAM gathering would be an example of metrics that could be retrieved by the MM at this level. In this way, the PM can analyze this data and recognize potential performance issues of the service due to the lack of compute resources. In this case, actuations could represent the resizing or scaling of a VM, or even the live migration of VNF resources.
- **MANO Layer. -** At this level, it is not expected to retrieve many data metrics. In any case, mostly information regarding the correct provisioning of the NSSIs is present at the MANO component, which may not be relevant at runtime. Nevertheless, from the actuation point of view, actions through this component would enable making high-level reconfigurations of the slice. Meaning that the entire composition of a NSSI can be changed, for example by adding VNFs or modifying how these are chained together. Moreover, changes to the entire NSI could also be executed, instantiating new NSSIs or eliminating them to rearrange the overall behaviour of the service and how their resources are mapped over the underlying network infrastructure

During the maintenance stage, the PM and MM modules would maintain a constant interaction towards components at the layered architecture. These interactions, depending on the level and component, would use communication interfaces such as REST, Remote Procedure Calls (RPC) or direct remote access to gather monitoring data and trigger actions when necessary. At the end of the slice lifecycle and once the runtime stage is finished, the decommissioning stage begins. It comprises the deactivation of the NSI, where high-level components should coordinate all required configurations to reclaim all the resources previously allocated for the slice (e.g., termination of network functions). Additionally, it should also consider reconfiguring or removing dependencies of shared resources to guarantee the normal operation of other slices and services [12]. Upcoming sections present how both provisioning and maintenance architectural versions are proven experimentally, setting up different scenarios for KPI-related parameter monitoring and preventive policy-based actuations.

4. EXPERIMENTAL TESTING – PROVISIONING

To experimentally validate and test the previously discussed frameworks, an emulated multi-segment testbed based on the use of open source software components has been set up, as depicted in Fig. 10.

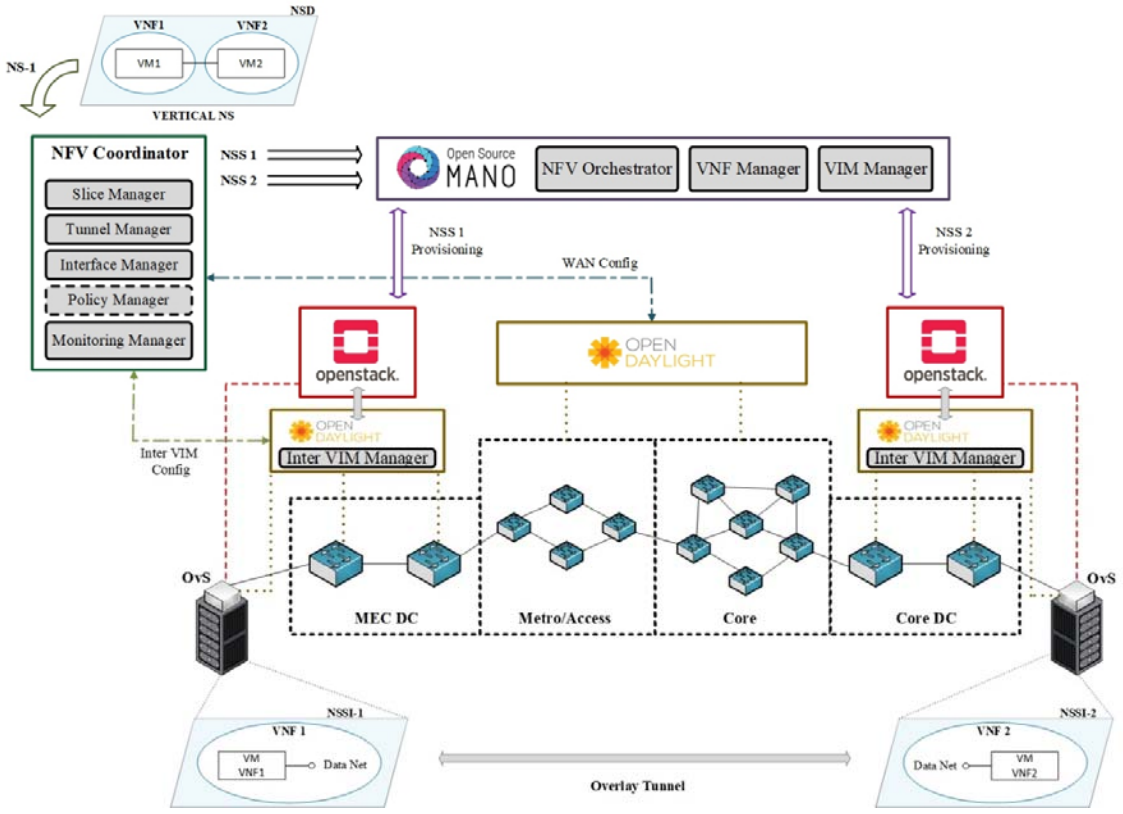

Figure 10. Experimental Testbed for 5G Service Provisioning and Maintenance.

The testbed represents some of the segments of 5G networks; in particular, in this case, the Multi-access Edge Computing (MEC) DC, Metro/Access, Core and Core DC segments are considered. Regarding the network resources at the infrastructure level, these are emulated using the MININET tool [25], which allows designing specific network topologies and interconnecting them, enabling thus the multi-segment scenario emulation.

In order to provide the required connectivity at the network, meaning the triggering of flow configurations at Open Virtual Switches (OVS) [26] via southbound protocols [27], network resources are controlled by per-segment SDN-Controllers. At the testbed, an instance of the OpenDaylight [28] controller is considered at each segment (i.e., MEC DC, WAN, Core DC). These provide the intelligence necessary to set up point-to-point connections between deployed compute resources (e.g., VM, VNF, container), besides enabling as well the direct gathering of statistics from network subsystems. At last, monitoring data and network capabilities are exposed via REST interfaces and RPCs, so these become available to clients at higher layers.

Regarding computational resources, these are managed from the orchestration layer. In this scenario, two instances of OpenStack [29], one at each DC segment, are set up to enable resource allocation/instantiation. Then, considering the slice provisioning stage in particular, all VMs, VNFs or containers required to enable service operation, can be deployed through these components. Orchestrators as well, provide the gathering of specific metrics (e.g., CPU, RAM, throughput) from resources, to finally expose monitoring data and available functionalities via REST interfaces.

On top of the controllers and orchestrators, an instance of the Open Source MANO (OSM) [30] component is deployed to orchestrate NSSI/NS level configurations. OSM in turn, registers underlying VIMs (i.e., OpenStack instances), to make these available for NSSI/NS instantiation. In this way, OSM can contact the orchestrators directly upon requests for the provisioning of specific per-segment configurations. Fig. 7 in particular, shows how OSM is able to provision both NSSI-1 and NSSI-2 on MEC DC and Core DC segments. That said, OSM is also the one on charge of providing the templates for NS and VNF definition.

Finally, and in charge of coordination across architectural layers and network segments, the NFV-C component is introduced as a set of Java-based modules and scripts running on software containers. As it can be seen in the figure, the coordinator de-composes the incoming service request into NSSI-1 and NSSI-2 and triggers its configuration via OSM. Then, it requests the controller at the WAN to set up network connectivity between both segments. The NFV-C as well, uses Inter-VIM Manager modules at DC segments to configure routing between NSI computing resources. Once everything is instantiated and configured, the maintenance stage begins.

A Management Network is employed in order to provide communication across all software components. In turn, at the infrastructure level, all service traffic flows through a Data Network, which connects segments end-to-end.

Fig. 11 shows the provisioned NSSI-1 and NSSI-2 over both OpenStack VIMs (i.e., openstack-left, openstack-right) as seen at the OSM dashboard (a). Each NSSI instantiation in turn, considers OSM requesting compute and network resources allocation at a DC segment via OpenStack and OpenDaylight. On (b), the instantiated VMs corresponding to VNF1 and VNF 2 are seen at the OpenStack dashboard. Each VM is connected to the Data Network (i.e., data) for internal connectivity and to the Management Network (i.e., provider) for external access.

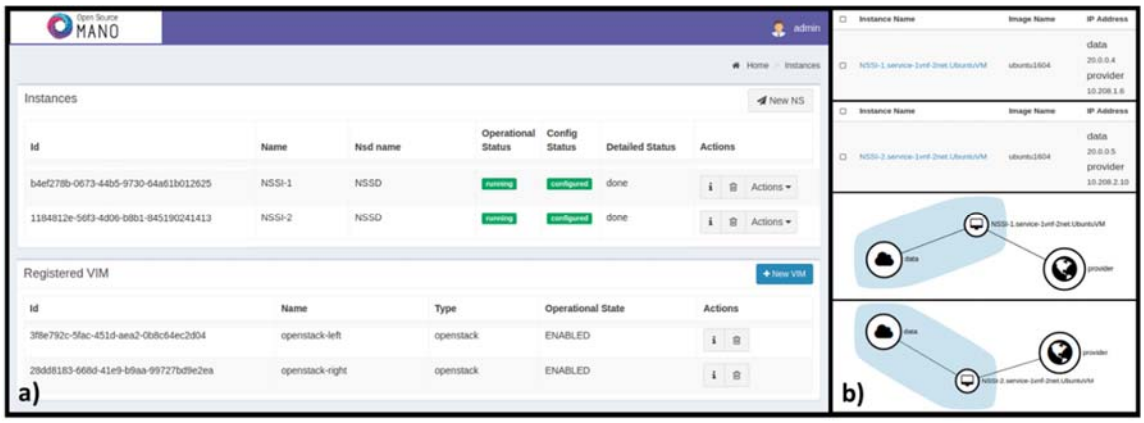

Figure 11. OSM (a) and OpenStack (b) dashboards with deployed NSSIs.

5. EXPERIMENTAL TESTING – MAINTENANCE

At the maintenance stage, sensors and actuations are required to guarantee the overall slice QoS in case of potential performance issues. In order to test this experimentally, a set of different use cases are presented, considering an ECA model defined for each. The idea is to trigger the gathering of parameters relevant to the slice KPIs for further analysis by comparing them to the established safe thresholds, then acting when these are surpassed/violated.

Use cases may require the monitoring of one or more specific parameters, depending on the case and on the defined model. Actions to be executed in turn, would consider different levels of actuation (e.g., NSI level, NSSI/NS level, VM/VNF level), which would depend on the identified affected resources.

A. VM Resizing

The first presented use case considers the enhancement of resources for a VM upon recognition of high CPU or RAM consumption that could affect service operation. More specifically the ECA model defined at the slice provisioning stage is the following:

- **EVENT (high CPU/RAM consumption). -** VM1 and VM2 are deployed in MEC DC and Core DC segments, respectively. The MM triggers CPU and RAM consumption metrics gathering via the orchestrators. The PM then, analyses this data to identify high levels of this parameters.
- **CONDITION (CPU > 40%, RAM > 80%). -** Thresholds defined at the model consider normal CPU consumption level to be below 40 percent and 80 percent in case of RAM consumption. Any identified measure above these levels should trigger preventive actions over the VMs.
- **ACTION (Resize VM).** - Actions in this case entail the PM requesting the orchestrator related to the affected VM, to re-provision it (i.e., resize) with additional compute resources, which basically means augmenting the allocated RAM and number of virtual CPUs for the VM.

Fig. 12 depicts the results obtained using Grafana platform for data visualization [31]. The graph on top shows monitoring data related to CPU consumption percentage of running Virtual Data Units (VDU), in this case VM1 and VM2, while the bottom one details the results on their RAM consumption percentage. The graphs are furtherly divided in 3 states according to the computational resources assigned to the VMs:

- 1st State (left):

- VM1&2 @ 1vcpu, 512MB RAM, 5 GB.

- 2nd State (middle):

- VM1 @ 2vcpu, 512MB RAM, 5GB.
- VM2 @ 1vcpu, 512MB RAM, 5GB.

- 3rd State (right):

- VM1 @ 2vcpu, 512MB RAM, 5GB.
- VM2 @ 2vcpu, 512MB RAM, 5GB.

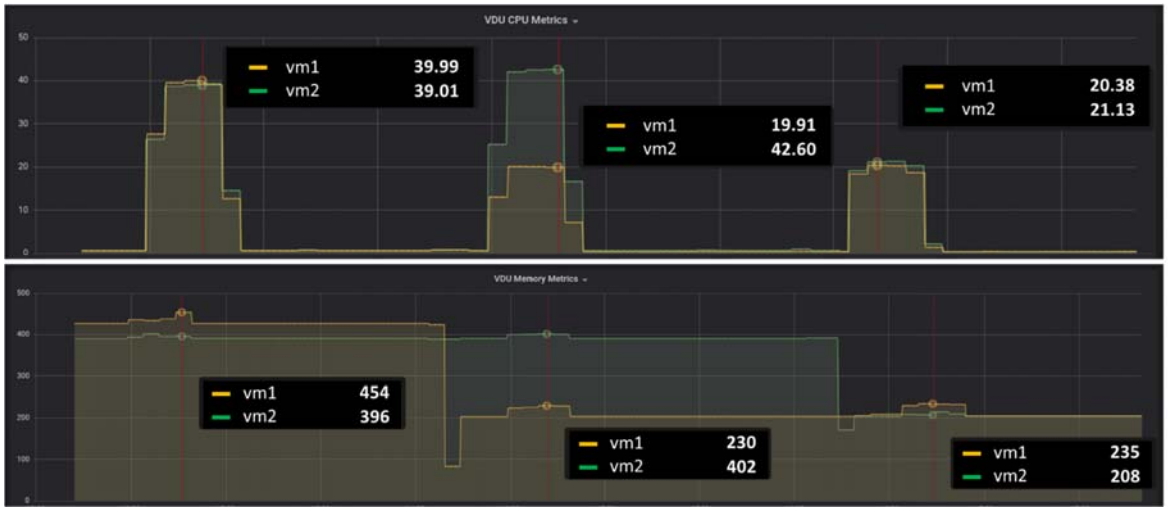

Figure 12. CPU (top) average usage [%] and RAM (bottom) consumption [MB] on VM1/VM2 as seen in Grafana dashboard.

At the first state, VM1 and VM2 are running and work traffic is emulated between them using the iPerf network tool [32]. Then, it is possible to analyse how CPU usage reaches around 40 percent while RAM goes up to 80 percent. This consumption levels meet the condition set at the previously defined ECA model. Then, to check if augmenting computational resources of the VM would lower CPU/RAM levels, VM1 resizing is triggered, providing it with an additional virtual CPU. Results of this modification are depicted at the second state, where after work traffic is started again between VMs, it is possible to see how VM1 lowers usage of CPU to 20 percent while RAM goes down to 40 percent. The third and final state, shows results after the same modification is triggered at VM2, depicting both VMs resource usage to similar levels. Thanks to this exercise, it has been proved how performing a VM resizing actuation can lower CPU/RAM consumption levels back to the safe standards (i.e., below the established threshold). In other cases, actions could also entail the addition of RAM or disk memory. As a final consideration, the VM resizing time that impacts service downtime has not been analysed in this exercise as there are already previous studies in this subject as in [33].

B. Throughput Shaping

The second use case presents a policy-based actuation based on the monitoring of throughput between VM1 and VM2. More specifically, by calculating and considering the Packet Loss Ratio (PLR) between them. The model defined for this case is the following:

- **EVENT (high PLR). -** Similar to the previous use case, VM1 and VM2 are provisioned and work traffic is emulated between them. The MM then, via the orchestrators, gathers data regarding the VMs amount of transferred/received bytes and packets. With this data it is possible for the PM to calculate the PLR to check for losses due to channel bandwidth saturation.
- **CONDITION (PLR > 1%). -** A threshold of 1 percent of PLR is set, in a way to protect service operation. In any case that this threshold is surpassed, the PM manager should perform preventive actions to maintain this variable between safe levels.
- **ACTION (increase bandwidth).** - The required actions to overcome high PLR, consider the PM asking the involved SDN-controllers for an enhancement of network bandwidth at the established data path for inter VM communication. The controllers then, should either reconfigure network subsystems to increment assigned bandwidth for this particular service or modify both VM interfaces configuration in case the bandwidth limit is set at this level.

In [17], we analysed this particular use case by emulating work traffic between VMs by using the iPerf network tool. The idea was to try to saturate the allocated communication channel, so to reach the PLR threshold of 1 percent. In this scenario, the bandwidth limitation was set at the VMs interfaces to control inbound traffic. As Fig. 13 exposes, the exercise considered two threshold violation events where actuations needed to be triggered to lower the PLR:

- **1st Threshold Violation (left):**
  - At the initial state, the throughput starts rising from 10 Mb/s while the VMs interfaces are limited to 25 Mb/s of allowed inbound traffic. Once traffic transfer rate reaches this level, the PLR starts rising. At the time 1 percent is surpassed, the PM triggers interfaces reconfiguration, setting up a new inbound traffic limit of 40 Mb/s and lowering the PLR back to 0 percent.
- **2nd Threshold Violation (right):**
  - While the PLR is normalized, the throughput continues to rise, now above the previous 25 Mb/s limit. Once it reaches the new 40 Mb/s inbound limit, the PLR starts to rise again to meet the 1 percent threshold. In this case, VMs interfaces are configured once more setting up a new bandwidth capacity limit of 50 Mb/s. PLR then goes back to safe boundaries, thus stabilizing service operation according to agreed performance standards.

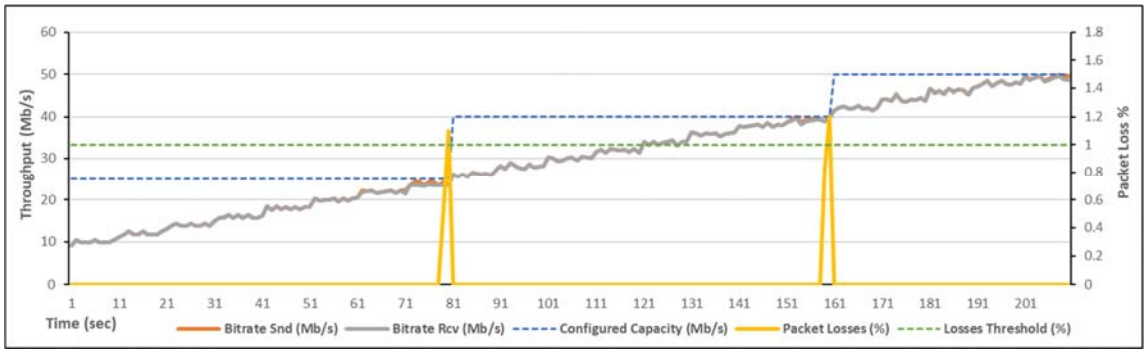

Figure 13. Throughput [Mb/s] vs Packet Loss Ratio [%] with policy-based actuations.

C. Path Reconfiguration

The third use case presents a scenario where the running service must maintain a required standard for end-to-end latency between VMs/VNFs. Then, a latency sensor is deployed along VM1 and VM2 at provisioning to analyse traffic flowing between them and calculate its latency. Once this data is gathered by the MM, the following model comes into action:

- **EVENT (high latency). -** The latency sensor, previously introduced in the slice maintenance section of this work, stores and exposes the measured latency between VMs. The MM reads this data, so the PM can check if it rises up to service operation threatening levels.
- **CONDITION (latency > 1ms). -** The limit set in this case, as in performance standards for 5G communications, is 1 millisecond for end-to-end latency. This means VM1 to VM2 or the other way around.
- **ACTION (reconfigure datapath).** - In case condition is met, the PM contacts the controllers to demand for a datapath reconfiguration. In this way, required instructions are sent to network subsystems to change the path set for traffic flow, thus maintaining latency levels below the threshold in front of a rise due to physical malfunctions or to overutilization of datapaths.

The work presented in [22], follows this same model criterion to test the datapath switching based on the collection of up-to-date latency information. In this case, VM1-Sensor-VM2 are connected via internal data networks, so the sensor can measure work traffic latency. By augmenting the traffic volume at some points of the emulated infrastructure, then creating network congestion by saturating the current datapath used links, latency can be risen up to the established threshold limit. Then, the PM becomes in charge of triggering the switching from the current datapath to the alternative one via the controller, thus performing a reconfiguration of the underlying OVS. Once the alternative datapath is operative, the end-to-end latency between VMs goes back to the standards defined at the provisioning stage. Through this exercise, exemplified in Fig. 14, it was possible to prove slice maintenance when considering running services which are highly dependent on latency for its normal operation.

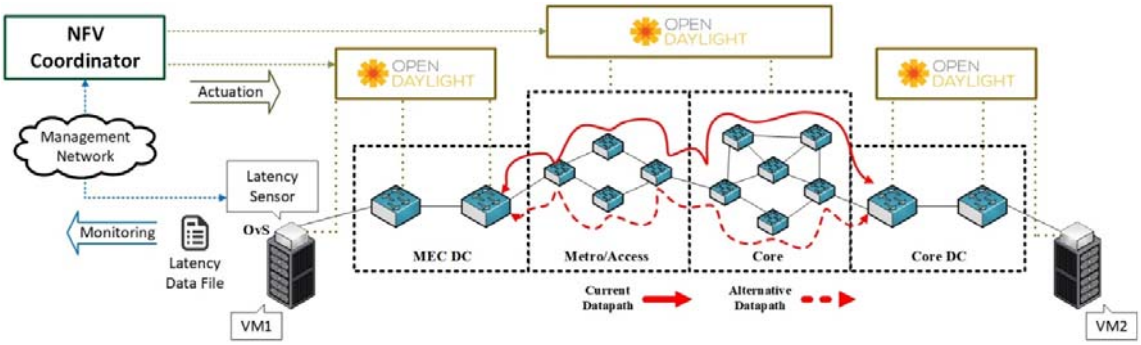

Figure 14. QoS guaranteeing on latency-sensitive services through path reconfiguration.

6. CONCLUSIONS

Current deployments of 5G scenarios, both at the industry and research field, are going through a crucial stage regarding the definition of the network architecture, where a set of control, management and orchestration components are introduced to allow the use of 5G enabling technologies such as Network Slicing, SDN, NFV, among many others.

In the presented work, an overall analysis on a well-coordinated network architecture capable of enabling service provisioning and maintenance through network slicing, besides the use of other techniques, has been provided. In this regard, its relation with the current standards set for 5G architecture has been analysed as well as its support towards the correlation between 3GPP and ETSI reference information models. Additionally, the framework has been validated experimentally, setting up an emulated multi-segment testbed for slice deployment on a single-domain scenario, and defining as well use cases for policy-based service maintenance. The characterization of this architecture looks forward to serve as proof of concept for the state of the art in 5G deployments. As for the future work in this subject, this work opens the path towards taking the proposal up to the 5G-Service Management level of operation, so the architecture capabilities can be proven on multi-domain scenarios.

ACKNOWLEDGEMENTS

This work has been supported by the H2020 5GPPP SLICENET project (H2020-ICT-2016-2/761913) and the Spanish Government through project ALLIANCE-B (TEC2017-90034-C2-2-R) with FEDER contribution.

REFERENCES

[1] NGMN Alliance 5G White Paper, Version 1.0, February 2015.
[2] 5G-PPP 5G Architecture White Paper, Version 2.0, December 2017.
[3] ETSI GS NFV-MAN 001 V1.1.1, December 2014.
[4] ONF TR-526, Applying SDN Architecture to Network Slicing, Issue 1, April 2016.
[5] K. Samdanis, X. Costa-Perez and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," in IEEE Communications Magazine, vol. 54, no. 7, pp. 32-39, July 2016.
[6] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna and A.Moored, "5G-Crosshaul: An SDN/NFV Integrated Fronthaul/Backhaul Transport Network Architecture," in IEEE Wireless Communications, vol. 24, no. 1, pp. 38-45, February 2017.
[7] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin and G. Wang, "5G-ICN: Delivering ICN Services over 5G Using Network Slicing," in IEEE Communications Magazine, vol. 55, no. 5, pp. 101-107, May 2017.
[8] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," in IEEE Vehicular Technology Magazine, vol. 13, no. 2, pp. 65-74, June 2018.
[9] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowiec, A. Autenrieth, V. Lopez and D. Lopez, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks [invited]," in IEEE/OSA Journal of Optical Communications and Networking, vol. 7, no. 11, pp. B62-B70, 1 November 2015.
[10] CogNet deliverable D2.2, “CogNet final requirements, scenarios and architecture”, April 2017.
[11] P. Neves, R. Cale, M. R. Costa, C. Parada, B. Parreira, J. M. A. Calero, Q. Wang, J. Nightingale, E. ChirivellaPerez, W. Jiang, H. D. Schotten, K. Koutsopoulos, A. Gavras and M. J. Barros, The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm, IJDSN, 2016.
[12] 5G-PPP 5G Architecture White Paper, Version 3.0, June 2019.

[13] 3GPP TR 28.801, Study on management and orchestration of network slicing for next generation network, Version 15.1.0, January 2018.
[14] 3GPP TS 28.541: “Management and orchestration of networks and network slicing; NR and NG-RAN Network Resource Model (NRM); Stage 2 and stage 3”
[15] ETSI GR NFV-IFA 015: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on NFV Information Model".
[16] ETSI GR NFV-IFA 024: "Network Functions Virtualisation (NFV) Release 3; Information Modeling; Report on External Touchpoints related to NFV Information Model".
[17] R. Montero, A. Pagès, F. Agraz and S. Spadaro, “Supporting QoE/QoS-aware end-to-end network slicing in future 5G-enabled optical networks”, PW 2019, San Francisco (United States), 2-7 February 2019.
[18] ETSI Open Source MANO, “OSM VNF Onboarding Guidelines”, http://osm-download.etsi.org/ftp/Documentation/vnf-onboarding-guidelines/.
[19] ETSI Open Source MANO, OSM Release FIVE Technical Overview, 1st Edition, January 2019.
[20] ETSI TS 128 533 V15.0.0 (2018-10), 3GPP TS 28.533 version 15.0.0 Release 15, “5G; Management and orchestration; Architecture framework”.
[21] ETSI GS NFV-SOL 005 V2.4.1 (2018-02), “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point”.
[22] R. Montero, F. Agraz, A. Pagès and S. Spadaro, “End-to-end Network Slicing in Support of Latency-sensitive 5G Services”, 23rd Conference on Optical Network Design and Modeling, ONDM 2019, Athens (Greece), 13-16 May 2019.
[23] Gnocchi - Metric as a Service, Gnocchi 4.2.1.dev96 documentation, https://gnocchi.xyz/.
[24] R. Montero, F. Agraz, A. Pagès and S. Spadaro, "Actuation Framework for 5G-enabled Network Slices with QoE/QoS Guarantees," 2019 21st International Conference on Transparent Optical Networks (ICTON), Angers (France), 2019, pp. 1-4.
[25] Mininet, https://mininet.org.
[26] Open Virtual Switch, Linux Foundation, https://www.openvswitch.org/.
[27] McKeown N. et al., “OpenFlow: Enabling innovation in campus networks”. ACM Communications Review, vol. 38, num. 2, pp. 69-74, April 2008.
[28] OpenDaylight, https://www.opendaylight.org.
[29] OpenStack, https://www.openstack.org.
[30] Open Source MANO, https://osm.etsi.org.
[31] Grafana Monitoring & Data Visualization Platform, https://grafana.com/.
[32] iPerf Measurement Tool, https://iperf.fr/.
[33] A. Pages, F. Agraz, R. Montero, G. Landi, R. Monno, J.I. Aznar, A. Viñez, C. Jackson, D. Simeonidou and S. Spadaro, "Experimental Assessment of VDC Provisioning in SDN/OpenStack-based DC Infrastructures with Optical DCN," ECOC 2016; 42nd European Conference on Optical Communication, Dusseldorf, Germany, 2016, pp. 1-3.