6G BRAINS Topology-aware Industry-grade Network Slice Management and Orchestration

João Fonseca*†, Mohamed Khadmaoui-Bichouna‡, Bruno Mendes*†, Paulo Duarte*, Marco Araújo*,
Daniel Corujo†, Ignacio Sanchez-Navarro‡, Antonio Matencio-Escolar‡, Pablo Salva-Garcia‡,
Jose M. Alcaraz-Calero‡, Qi Wang‡
*Capgemini Engineering, Porto, Portugal; †Instituto de Telecomunicações and University of Aveiro, Aveiro, Portugal
‡School of Computing, Engineering and Physical Sciences University of the West of Scotland, Paisley, UK

***Abstract*—This paper describes the integration between the Open Network Automation Platform (ONAP) and UWS Slice Manager within the European project 6G Brains. The proposed solution allows for End-To-End(E2E) Network Slicing, enabling fine-grain and optimal traffic engineering of the Network components. This work's findings ensure an E2E connection. The solution allows external services to create slices and attach them easily. The UWS Network Slice Manager allows for detailed monitoring of the network slice's inner components. With this information, ONAP can improve the network by creating and optimising slices on demand. The validation of the integration presents the workflow to create and attach slices. These operations enable autonomous workflows for deploying E2E Services that ensure the QoS/QoE in the network.**

I. Introduction

The emergence of research work on Sixth Generation Mobile Networks (6G) networks, following the first commercial deployments of Fifth Generation Mobile Networks (5G), allows addressing of new architecture considerations tested during the 5G research and standardization that haven't been acknowledged so far. 6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections (6G BRAINS), considers the following requirements as essential for the project: 1) Inclusion of various networking paradigms for Beyond Fifth Generation Networks (B5G) networks, giving particular relevance to the management of the Radio Access Network (RAN), specifically the softwarization of this component, which paves a path to an intelligent RAN. 2) Development of automation solutions for 6G networks, leveraging AI/ML. 3) Use of Intent-based management by design. 4) Enhancing the network segment-specific control, ensuring fine-grain and optimal traffic engineering for the interoperability between the network components. A more deterministic and programmable network has also been highlighted as an essential aspect in 6G networks [1]. In this work, we present how the project has tackled these various challenges, offering the Topology-aware Industry-grade Network Slicing Manager as the solution for orchestrating Network Slices in this project.

The rest of the document is organized as follows. Section II introduces the 3rd Generation Partnership Project (3GPP) Network Slicing standardized procedures and analyzes the Network Slice Manager (NSM) landscape. Section III presents the architecture for Topology-aware Industry-grade Network Slice Management and Orchestration used in 6G BRAINS. Section IV provides insights on validating the components composing our solution and shows preliminary results of this integration. Lastly, Section V provides some conclusions on the paper and proposes future studies using this solution.

II. Related work

In this section, we review the Network Slicing concept and its adoption in the 5G and B5G/6G. Then an analysis of existent Network Slice Managers is done, evaluating their capabilities. This analysis justifies the Open-Source Software (OSS) NSM adopted in this work.

For 3GPP, Network Slice is a Logical Network tailored to a customer's needs, with specific capabilities and characteristics, in terms of Quality of Service (QOS)/Quality of Experience (QOE). Typically a 3GPP Network Slice is associated with a Communication Service (CS) to be supplied by a Communication Service Provider (CSP) to a Communication Service Consumer (CSC).3GPP defined in Technical Specification (TS) 28.530 [2], the Management aspects of Network Slicing. In specific, the 3GPP defined the procedures associated with the *Preparation* of the Network Slices and the Life Cycle Management (LCM) of a Network Slice Instance (NSI). During the *Preparation* phase, the design of the Network Slice is done, considering the requirements associated with the capacity of the network resources. It is followed by onboarding the multiple network functions that compose the Network Slice. Meanwhile, the overall environment that supports the Network Functions (NFs) part of the Network Slice is also prepared so that the resources are available for provisioning, triggered by a NSI request. The Lifecycle of a NSI has been divided into three operations: *Commissioning*, *Operation*, and the *Decommissioning*. 3GPP also defined Management Functions(Communication Service Management Function (CSMF), Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF)) as part of the Management Architecture related to the support for Network Slicing in 5G, realizing therefore, the need for the automation of the processes associated with the NSI [2].

To help simplify attributes related to a Network Slice, GSM Association (GSMA) proposed a Generic network Slice Template (GST) [3] that entails the characteristics associated with Network Slice/CS type. A GST is not specific to a

Network Slice deployment, being a generic structure, allows for the definition of specific NEtwork Slice Type (NEST). A NEST is a GST filled with the information associated with the CS to be provided by the CSP. The attributes related to a GST are of three types: Mandatory, Conditional, or Optional. The GSTs/NESTs allow an easy setup of the Network Slice delivered to the CSC. Network Slice Templates (NSTs) ease the automation of onboarding the Network Slices into supporting NSMs. Generally, if the Network Slice makes use of European Telecommunications Standards Institute (ETSI) Network Services, there is also the need for the onboarding of these elements, which are an onboarding dependency of the NST.

A. Network Slice Managers

The existent alternatives for NSM are pretty different. ETSI has created a document [4] for aligning 3GPP Specifications on Network Slicing and the ETSI Network Function Virtualization (NFV) architecture, as part of the NFV Release 3 effort. We will focus our study on the most well-known End-to-End (E2E) NSM OSS solutions available now. Most platforms comply with ETSI NFV.

1) Open Network Automation Platform (ONAP): Open Network Automation Platform (ONAP) has been for some years now a project of the Linux Foundation Networking (LFN) ecossystem. This project focuses mainly on Service Oriented Orchestration. ONAP follows the ETSI NFV architecture, but it is not compliant with the ETSI Management and Orchestration (MANO) Architecture. ONAP doesn't implement one of the required interfaces (os-ma-nfvo) [5] to ensure that compliance. Moreover, it doesn't follow a resource-oriented view of the network services. Nevertheless, ONAP follows the TM Forum OpenAPIs for service orchestration and can perform the automation processes related to the requested CS using closed-control loops. Focusing on the Network Slicing capabilities of ONAP, it implements the 3GPP Management Functions through specialized interfaces and the communication between them. ONAP, has developed interfaces that allow for the request of a CS to be translated into a Network Slice, which might have one or more associated Network Slice Subnets (NSSs). These templates can be onboarded directly to the platform or indirectly using intents.ONAP allows for a CSC to request a Network Slice based on Intents. ONAP can orchestrate an E2E Network Slice leveraging 5G at the RAN level if needed. Moreover, ONAP implements the GSMA NST, complies with 3GPP Network Resource Model (NRM) for Network Slices [6] ONAP implements Internet Engineering Task Force (IETF) Transport Network (TN) Slicing [7], therefore augmenting the control over the Network using Software Defined Network (SDN), in specifically leveraging the generic SDN Controller (SDNC) which exposes an API that can control and retrieve information from a specific controller.

2) Openslice: Openslice [8] was developed as part of the 5G-VINNI project and focused mostly on complying with TM Forum OpenAPIs whilst being compatible with the ETSI MANO architecture and positioning itself as an OSS/BSS solution. Moreover, it implements GSMA GST following the 3GPP NRM. This NSM can provision and orchestrate network services; for this purpose, it interfaces with Open Source Mano (OSM)(see subsubsection II-A3) at the NFV Orchestrator (NFVO) level using the SOL005 API [5]. Therefore, Openslice inherently supports managing all the resources controlled by OSM. Depending on these resources, OSM can act as a E2E NSM.

3) ETSI Open Source Mano (OSM): OSM is both a NFVO and a NSM. This project is currently being developed with ETSI's supervision, therefore ensuring compliance with the ETSI NFV and MANO architectures. Moreover, the inputs given by the 5G Tango project not only follow the 3GPP NRM but also provide the interfaces for the communication with the NFVO via SOL005[5], [4]. OSM supports the management of Network Services and the SDN that may be associated with the Virtual Infrastructure Manager (VIM) being used. The Network Services might leverage Core and RAN Virtual Network Functions (VNFs), allowing for the deployment of a E2E Network Slice. However, this solution may not be able to cover the TN between VIMs. The SDNC support is limited to ONOS.

4) 5GR-VS: 5GROWTH Vertical Slicer (5GR-VS)[9] is a NSM that allows for the deployment of E2E Network Slices, following the 3GPP NRM. It complies with the ETSI MANO architecture and can make use of OSM as an NFVO. Moreover, working with the rest of the 5Growth Stack can automate some of the CS lifecycle procedures across the several Layers of this Stack, leveraging Closed-control loops. The platform can also leverage a SDNC for controlling the traffic in the WAN, therefore providing TN Slices. A similar approach was used to control a specific RAN, enabling RAN Network Slices. The 5GR-VS is a plugin NSM, but due to the scope of the project is oriented to interact mainly with specific plugins.

Our analysis shows that except for ONAP, 3GPP compliant NSM lack automation features. Moreover, the existing solutions besides this one have interfaces that are not easy to use for a CSC who is not savvy in Network Slicing and Network Orchestration. The analysis on the orchestration of E2E Network Slices discovered a gap concerning the TN between Core and RAN, as only ONAP presents an agnostic interface to communicate with all sorts of SDNC. Moreover, the existing slicing solutions, including ONAP, don't offer a fine-grain view of the components and devices part of the E2E Network Slice. The lack of information may allow a potential vector of attack as the integrity of the information cannot be guaranteed. The 6G BRAINS project aims to solve these problems while ensuring an Automation Industry-grade solution. Therefore, ONAP was chosen as a base solution to be enhanced under the project's scope.

III. Proposed architecture

Following the requirements of 6G BRAINS, section I, and the analysis of the current NSMs landscape in section II, we present in this section the architecture of our solution.

A. Overall Architecture

From our study on the different NSM capabilities, we proceeded to use ONAP in the scope of the 6G BRAINS project. In figure 1, we present how we improved the capabilities of ONAP, enhancing ONAP capabilities regarding topology awareness. The CSC can request a new CS via the Industrial Virtual Assistant (IVA) [10] that translates it into an Intent or directly via ONAP. This request specifies the type of QOS/QOE needed in the desired Network Slice. ONAP can realize the E2E 5G Network Slicing decomposing the request into different NSS: Core, TN, RAN. In figure 1, we can see how the Network Slice request is mapped in the Service Orchestrator (SO), passing through the 3GPP compliant Management Functions to interface with USM finally.

B. UWS Network Slice Manager (USM)

The USM is one of the main components of the architecture. It provides an abstraction layer for the E2E Service Orchestrator to instantiate E2E Network Slices across the different network segments of the mobile network infrastructure. To this end, the USM must provide the network topology along with the list of available network functions to program the data plane, allowing the E2E SO to define where to instantiate new Network Slices without having deep knowledge of how this will be done. The USM is supported by two key components, depicted in the bottom-left side of Fig. 1, that provide the requirements mentioned above: the TIA and the SCA.

1) Topology Inventory Agent (TIA): Discovers and updates the complete inventory of physical and virtual machines and their inter-networking and topological information. This component is deployed in every applicable physical host and can discover the topology using network discovery protocols (e.g., CDP, LLDP, and SNMP), hypervisor APIs (e.g., libvirt and libpci), and running daemons (e.g., OpenStack nova-compute). More details of this component's architecture can be found in [11], [10]. TIA uses two different entities to define the topology structure. The first one represents a network interface ("resourceType": "DEVICE_PORT" in figure 2a), and the second one represents a connection between interfaces ("resourceType": "CONNECTION" in figure 2a). The complete topology can be represented using the information contained in both entities. Figure 2a shows an example of the topology structure discovered by TIA and how it can be represented. Interfaces also contain information about the device (physical host, virtual host, software switch [1]) where they are located. In Figure 2a, a big grey box represents a physical computer. It has two colored boxes inside. The yellow box represents a virtual computer (or virtual host), and the blue represents a software switch. There is a difference in the terms described as *deviceHostName* and *hostName*. It can be noticed especially in the interfaces of the virtual computer and the software switch where the values are different. The *hostName* refers to the device where they are located, and the *deviceHostName* term refers to the location of such device. This allows the allocation of virtual computers and software switches in their respective physical computers and interfaces in their respective devices. Regarding the connection entities, *srcResourceId* and *dstResourceId* refers to the *resourceId* of the source and destination interfaces that are connected.

2) Slice Control Agent (SCA): Abstracts a set of heterogeneous programmable data plane technologies (wireless and wired) and provides a unified control interface for network slicing. The SCA also provides key performance metrics about the status of the network slices at a specific datapath point (e.g., a Network Interface). This component is deployed in each applicable physical machine in the architecture to communicate with their networking devices in a technology-dependent manner (e.g., OVS). For more details on the architecture of the SCA, the mechanisms that it abstracts from different technologies, and the control functions that it provides for network slicing, refer to [12] [10]. Its architectural design, shown in Figure 2b, consists of three main parts. The first part is the Northbound Interface (NBI), which acts as a technology-agnostic access point from/to upper layers of the infrastructure (e.g., USM). It consumes intent-based messages to program the data plane by translating generic directives into technology-specific instructions. When the SCA is instantiated, as a first step, it publishes its capabilities (e.g., create a new slice, remove slice, etc.) described in its actions catalogue. After that, it waits for new intent messages to be processed. Additionally, the acknowledge module publishes information on whether the action has been successfully enforced, while the metrics module reports performance information at defined intervals. The second is the SCA Core, which processes incoming messages and locally manages the life cycle of the slices. It contains an overlay flows dissector module with the logic to dissect the packet representation structure into different sub-flows associated with each overlay network. Additionally, a slice builder module creates generic instructions to satisfy the slice requirements but delegates their implementation to other underlying modules containing a specific technology's logic. It also includes a metrics engine module for calculating and reporting slice metrics. The third and final part is the Data Plane Abstraction Service (DPAS), which provides an abstraction layer to tackle the network traffic control technologies (slice enablers) registered as plug-ins. By doing so, the SCA Core does not need to know how each of these Slice enablers works, thereby achieving a modular approach where the DPAS communicates within a common language with upper layers of the architecture and in a technology-dependent manner with any plug-able underlying slice enabler. This module aims to select a proper technology to program slicing policy rules in the host computer's specific hooking point (programmable network point).

The information received by the Network USM from TIA and SCA is then exposed via a REST NBI. The primary purpose of the REST interface is to allow upper layers of the architecture (e.g., E2E Service Orchestrator) to make use of such information to determine where the E2E Network Slice

[1]Software switch in this context refers to Linux Bridge or Open vSwitch (see https://www.openvswitch.org)

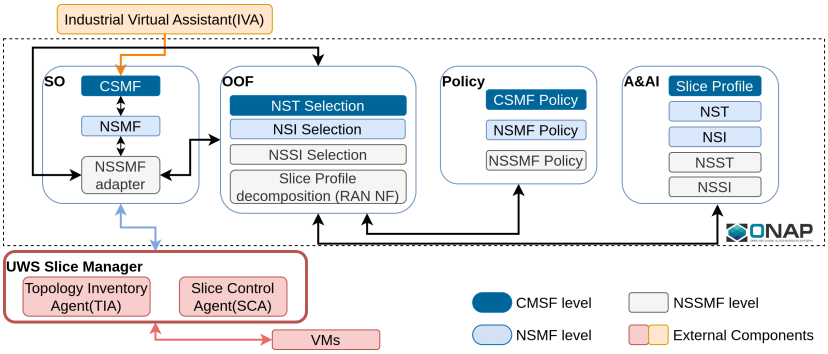
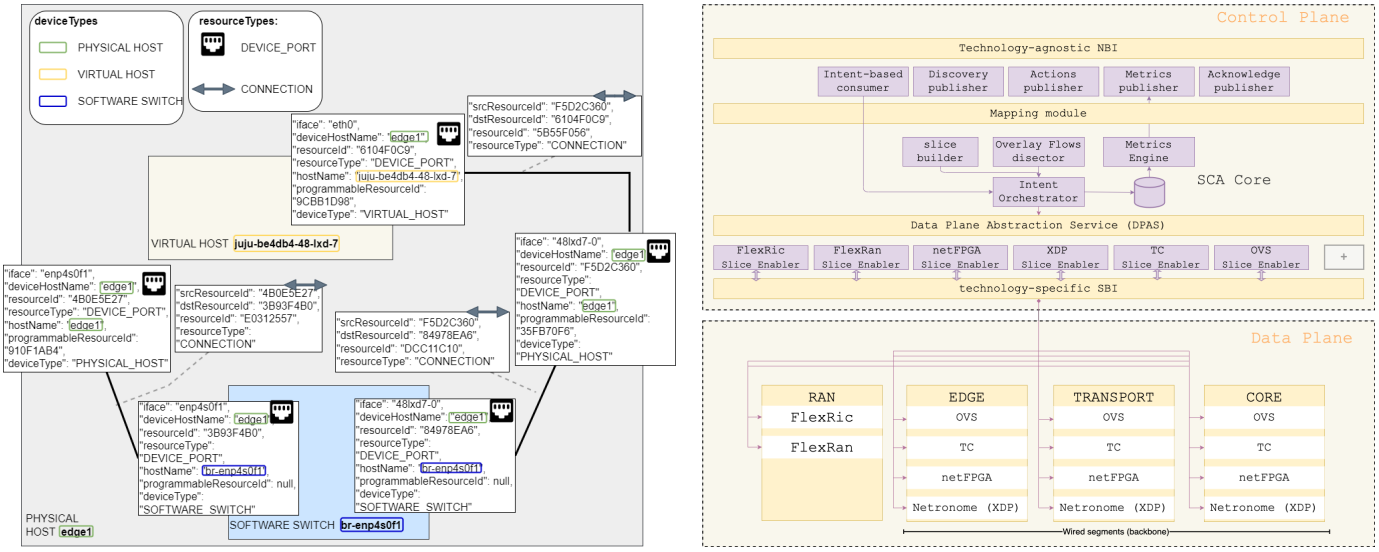


Fig. 1: Architecture with integration between ONAP and UWS Network Slice Manager (USM)



(a) Topology Inventory Agent (TIA) discovery example.

(b) Slice Control Agent (SCA) logical architecture.

Fig. 2: Internal Architecture of the UWS Network Slice Manager (USM) components

will be instantiated and which control plane modules will be involved during that process.

IV. Empirical Validation

A. Design Time

ONAP's Service Design and Creation (SDC) enables on-boarding and composition of the necessary services. In this work, creating custom Network Slices was necessary to assume the necessary resources. A customer segment template and a network service template were created to define the segment's network service, and the external services will then use the service to request a segment instance. A resource template was modelled, comprising resources that map to the TN. To aggregate, these resources are considered to create a service template composed of resources to be created and slice properties that ensure the QOS: the maximum Bandwidth, the minimum Bandwidth, the slice priority, and its name. These resources should correspond to the current network connectivity in the USM's architecture.

After establishing services and resources, SDC distributes them to catalogue databases and run-time components, as

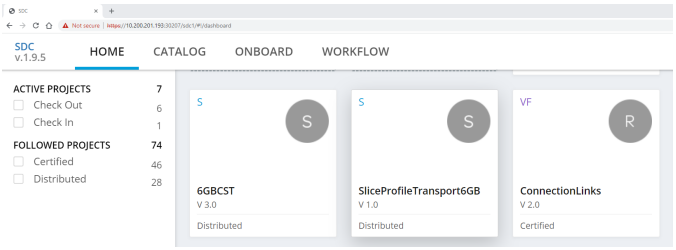


Fig. 3: ONAP's Catalogue of Customized Services.

shown in Figure 3. Those templates should be available in the catalog GUI, and OSS/BSS systems can use them via the catalogue APIs to request a service slice creation.

B. Topology Synchronization and Inventory

A Network Topology method was introduced in ONAP to facilitate network resource onboarding. The USM is called by ONAP SO to start the topology discovery .USM responds with the topology information in JSON format. On the ONAP side, the data is read, analysed, and verified by SO before being

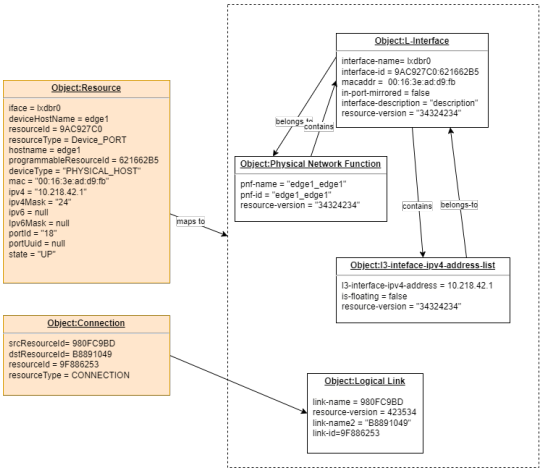


Fig. 4: TIA discovered resources mapped into AAI schema

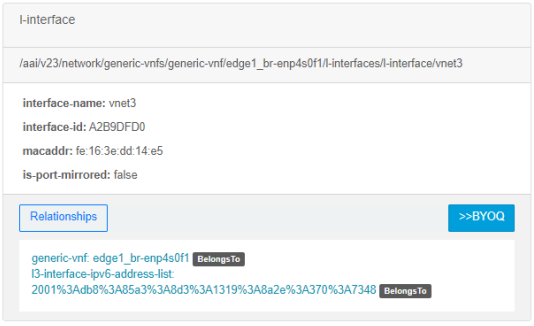


Fig. 5: VNF saved on ONAP Inventory.

mapped to inventory schemes of Available Inventory (AAI). This ONAP component has a unique schema for each resource, network, service, etc. AAI delivers a logically centralised view of inventory data, incorporating updates from orchestrators, controllers, and assurance systems. Finally, AAI is updated with the resources JSON file. An example of the resources can be seen in Figure 2a, which distinguishes the several components based on a hierarchy. This led to the creation a system for translating the hierarchical topology of USM to the AAI schema. The result is available in Figure 4.

To synchronize the network topology between the ONAP's SO and the USM, a new procedure was developed. ONAP starts by requesting the topology from the USM. Then it is needed to process the incoming information and categorise it based on whether the resource is virtual or physical. The process will then check the new network information with the one on the AAI. If the information is different, it will be changed. Figure 5 depicts an analysis of the stored data. This procedure keeps the topology on ONAP up to date, allowing optimal slice creation and resource use.

C. Network Slicing Orchestration integration with the USM

The Network Slice Creation method allows external applications to request the creation of slices using ONAP to prepare the slice and the 3rd party USM to perform it on

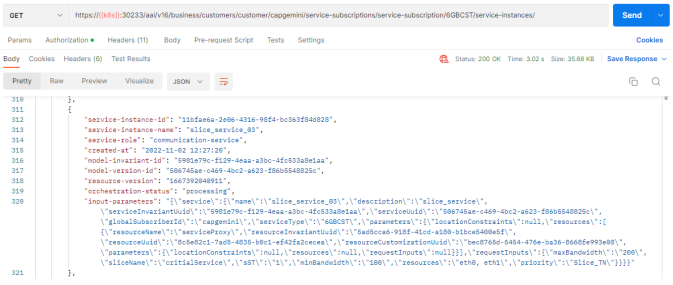


Fig. 6: ONAP create Service Instance.

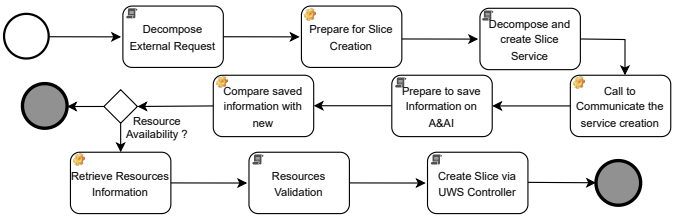
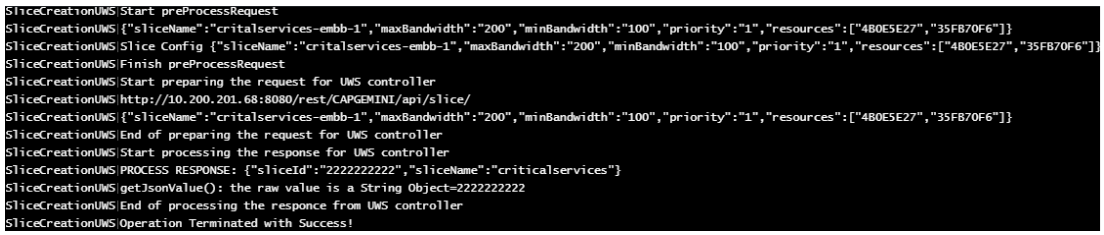


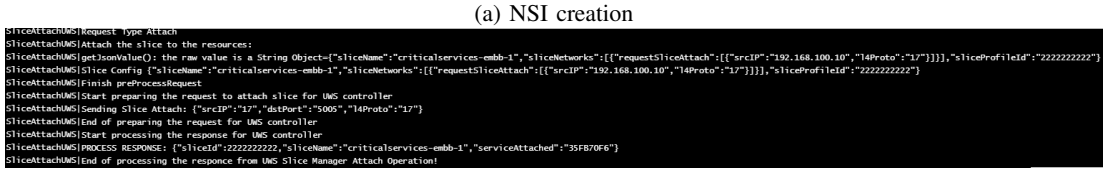
Fig. 7: Create Slice Workflow.

the network. Creating a Network Slice involves several steps. First, a request is made by an external service via the NBI. Next, the current network topology is reviewed. Then, the user can request a new slice, specifying the requirements to ensure the QOS for that CS. The CSMF converts the request into a slice profile, which includes performance requirements such as maximum latency, bandwidth, and packet loss, and stores it in the Active and AAI, Figure 6. The CSMF then uses the ONAP Optimisation Framework (OOF) to provide a suitable NST, then requests the NSMF to allocate an NSI and forwards the request to the OOF. OOF selects the most suitable NSI and Network Slice Subnet Instance (NSSI). Then the NSMF will trigger the NSSMF adapter, and it will request the UWS controller via the Southbound interface to generate the required slice. At the SO. The workflow to create a new slice starts. It begins by receiving an initial request, which gathers the necessary information upon processing, Figure 7. Then SO updates ONAP, searching its inventory for any existing slices that match the information provided. If a match is found, it is used. Otherwise, a new slice is created. Once these steps are completed, the workflow prepares the request to send a creation request to the USM (Use Case-Specific), which includes the slice's identification, characteristics, and resources.

The SO can also test the procedure to generate a request to apply and attach the slice to the resources, Figure 9. It begins by receiving the initial request, which is processed to obtain information about the slice to which it must be linked. The following stages evaluate the information saved on the AAI side, compare it to the information received, and validate the new fields to be added. Depending on the circumstance, it will update and then prepare the request to submit to USM to attach the network slice to the resources with the request characteristics.



(a) NSI creation



(b) NSI attach

Fig. 8: ONAP's Logs for NSI Orchestration operations using the USM

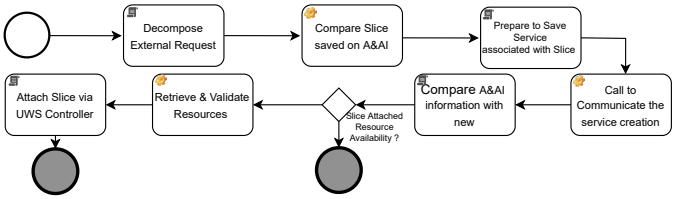


Fig. 9: Attach Slice Workflow.

ONAP's logs available in Figure 8a and Figure 8b, respectively result of the `creation` and `attach` operations using the USM.

V. Conclusion

In conclusion, the research presented in this paper confirms the significance of a Topology-aware Network Slice Management and Orchestration in the Industry. The analysis presented in the Related Work, section II, demonstrates the importance of the research in this field. This document presents the necessary workflows to integrate the USM with ONAP. Moreover, the NSTs needed by ONAP allows for fast synchronization of the USM topology. The NST enables the NSM developed within the 6G BRAINS project to gather all information and make the best decisions at any time. Finally, this document explains the integration of dedicated external services in ONAP for requesting the creation of slices and applying them to new network services and resources.

Future studies can explore the mechanisms for improving the performance of this solution and how it would behave in stressful network situations. Ultimately, this work highlights the enhanced capabilities of USM working with ONAP to manage and orchestrate E2E Network Slices.

Acknowledgment

This work is funded by the European Commission Horizon 2020 5G-PPP Program under Grant Agreement Number H2020-ICT-2020-2/101017226 "6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections". It is noted that Mohamed Khadmaoui-Bichouna is the co-lead author of this paper.

References

[1] C. Bernardos, M. Uusitalo, C. Anton, Artuñedo, and et al., "European vision for the 6g network ecosystem," 5G Infrastructure Association, Tech. Rep., 06 2021.

[2] G. TSG, "Management and orchestration; concepts, use cases and requirements," 3rd Generation Partnership Project, Technical Specification 28.530, dec 2022.

[3] G. A. N. Group, "Generic network slice template," GSM Association, Tech. Rep. 116, nov 2021.

[4] E. N. I. EVE, "Evolution and ecosystem; report on network slicing support with etsi nfv architecture framework," European Telecommunications Standards Institute, Tech. Rep. 012, dec 2017.

[5] E. N. I. SOL, "Protocols and data models; restful protocols specification for the os-ma-nfvo reference point," European Telecommunications Standards Institute, Tech. Rep. 005, aug 2022.

[6] G. TSG, "Management and orchestration; 5g network resource model (nrm); stage 2 and stage 3," 3rd Generation Partnership Project, Technical Specification 28.541, jan 2023.

[7] X. Geng, L. M. Contreras, R. Rokui, J. Dong, and I. Bykov, "IETF Network Slice Application in 3GPP 5G End-to-End Network Slice," Internet Engineering Task Force, Internet-Draft draft-gcdrb-teas-5g-network-slice-application-01, Oct. 2022, work in Progress.

[8] C. Tranoris, "Openslice: An opensource oss for delivering network slice as a service," 2021.

[9] X. Li, A. Garcia-Saavedra, X. Costa-Perez, C. Bernardos, and et al., "5growth: An end-to-end service platform for automated deployment and management of vertical services over 5g networks," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 84–90, 2021.

[10] A. Kazmierowski, J. Kodjabachian, P. Salva-Garcia, and et al., "D5.2 Preliminary integration for AI-based E2E network slicing control and MANO," Dec. 2022.

[11] I. Sanchez-Navarro, A. Serrano Mamolar, Q. Wang, and J. M. Alcaraz Calero, "5gtoponet: Real-time topology discovery and management on 5g multi-tenant networks," *Future Generation Computer Systems*, vol. 114, pp. 435–447, 2021.

[12] M. B. Weiss, A. Gavras, P. Salva-Garcia, J. M. Alcaraz-Calero, and Q. Wang, "Network management - edge and cloud computing the slicenet case," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, 2020, pp. 1–6.