ONAP
OPEN NETWORK AUTOMATION PLATFORM

ONAP E2E Network Slicing

Technical Overview

Providing End-to-end 5G Network Slicing Capability

OVERVIEW:

- Tenants and network operators can order slice-based services
- Enables end-to-end network slice creation as well as reuse
- Supports many of the slice lifecycle management operations

CURRENT CHALLENGES:

- Use case diversity in 5G necessitates Network Slicing
- Shaping the network to cater to industry verticals requirements while optimizing cost (CAPEX and OPEX)
- Simple interface to request network slices by slice consumers, enterprises and industry verticals by simply providing service requirements
- Network resource configuration and performance management from multi-vendors from E2E perspective.

SOLUTION:

- ONAP Guilin enhances the basic capabilities for Network Slice Orchestration
- Supports Network Slice lifecycle operations of E2E Slice Design and Creation, Activation, Deactivation and Termination, as well as basic Network Slice subnet design and creation of Slice subnet instances in the RAN, Core and Transport domains
- Provides 3GPP-defined Communication Service Management Function (CSMF) and Network Slice Management Function (NSMF) functionality within ONAP. In addition, provides the basic functionality of Network Slice Subnet Management (NSSMF) functionality for RAN, Core and Transport domains within ONAP.
- Supports E2E Slice design including design of Communication Service, Service Profile, Slice Profiles, Network Slice Template (NST) and Network Slice Subnet Templates (NSST)
- Supports selection of suitable NST, Network Slice Instance (NSI) and Network Slice Subnet Instance (NSSI)covering the scenario of new NSI creation by providing suitable slice profiles, which may also result in new NSSI creation
- Interacts with external Core and RAN Network Slice Subnet Management Functions (NSSMFs)
- Develop RAN NF simulators and Core NF simulators which can be directly managed and configured by ONAP and explore open source NFs at the same time.

Overview

Network Slicing is one of the key features of 5G. The essence of Network Slicing is to be able to effectively cater to of a diverse set of use case categories and customers in an optimal manner. 5G is expected to fulfill the requirements of a wide variety of use cases including urban mobile broadband, massive machine-type communications, and ultra-reliable low latency communications.

Further, 5G should be able to meet the QoS and quality of experience of different consumers, and industry verticals including connected home, autonomous vehicles, smart cities, remote healthcare, in-stadium experience, rural broadband, factory automation, and smart enterprises. Each of these categories have a different set of performance requirements (e.g., latency, throughput, reliability, availability, etc.) and characteristics (e.g., mobility, security, resource sharing level, etc.). End-to-end Network Slicing enables addressing these in an optimal manner through sharing of resources and shaping the network dynamically to meet the demands of various services. This, in turn, saves significant CAPEX and OPEX for the communication service provider. Further slice orchestration functionality is another enabler for network automation and shall play an important role in quick service rollouts, as well as service assurance and SLA adherence, leading to an improved end user experience.

An End-to-End (E2E) Network Slice consists of RAN, Transport, and Core network slice subnets, with each of these subnets possibly consisting of further subnets, for example, RAN subnet decomposed further into fronthaul, mid haul, and RAN Network Functions.

The objective of this use case is to realize End-to-end 5G Network Slicing using ONAP. This use case intends to demonstrate the modeling, orchestration (life cycle and resources) and assurance of a network slice in alignment with relevant standards. The key highlights of this use case include:

- Modular architecture providing loosely coupled building blocks that provides flexibility for various deployment scenarios

- Functionality aligned with 3GPP and other relevant standards such as ETSI and IETF.
- Interfaces and APIs aligned with relevant standards (3GPP, IETF, ETSI, TM Forum) while enabling easy customization through use of appropriate adaptors or plug-ins. This would enable easier interoperability of slice management functions realized within ONAP with 3rd party slice management functions, as well as northbound and southbound systems.
- Taking a step-by-step approach to realizing different architectural options being both backward and forward compatible.
- Providing flexibility in network slice selection by providing an option of manual intervention, as well as abstracting the network internals as needed.
- The use case implementation team is composed of service providers, software and hardware vendors, solution providers, and system integrators thereby taking into consideration different perspectives and requirements while being a true open source community effort.

This use case is a multi-release effort in ONAP with the first steps taken in the Frankfurt release, and significant functions and capabilities added in the Guilin release. It will continue to expand in scope both in breadth and depth, and along the journey it shall also align with updates to the relevant standards which are also currently evolving. This use case has also started collaboration with other open initiatives such as O-RAN Alliance to enable wider adoption and use.

In the Frankfurt release, this use case covered the following aspects:

- E2E Network Slice design
- E2E Network Slice creation and activation, deactivation, and termination
- Selection of appropriate Network Slice instance and mapping it to a service
- Selection of appropriate core network slice subnet instance
- Interfacing with an external entity to create a new core network slice subnet instance or to map an existing network slice instance to a network slice instance

It also took the first step towards alignment with relevant standard bodies (e.g., 3GPP, ETSI, TM Forum) w.r.to both interfaces, information models as well as functional aspects.

In the Guilin release, enhancements to this use case include:

- Enhancements to E2E Network Slice creation and activation, deactivation, and termination
- Selection of appropriate RAN and Transport slice subnet instance, and enhancements to Network Slice instance and core network slice subnet instance selection
- RAN/Core/Transport NSSI creation and activation, deactivation, and termination
- Update of existing RAN and Core NSSIs when reusing them for a new NSI

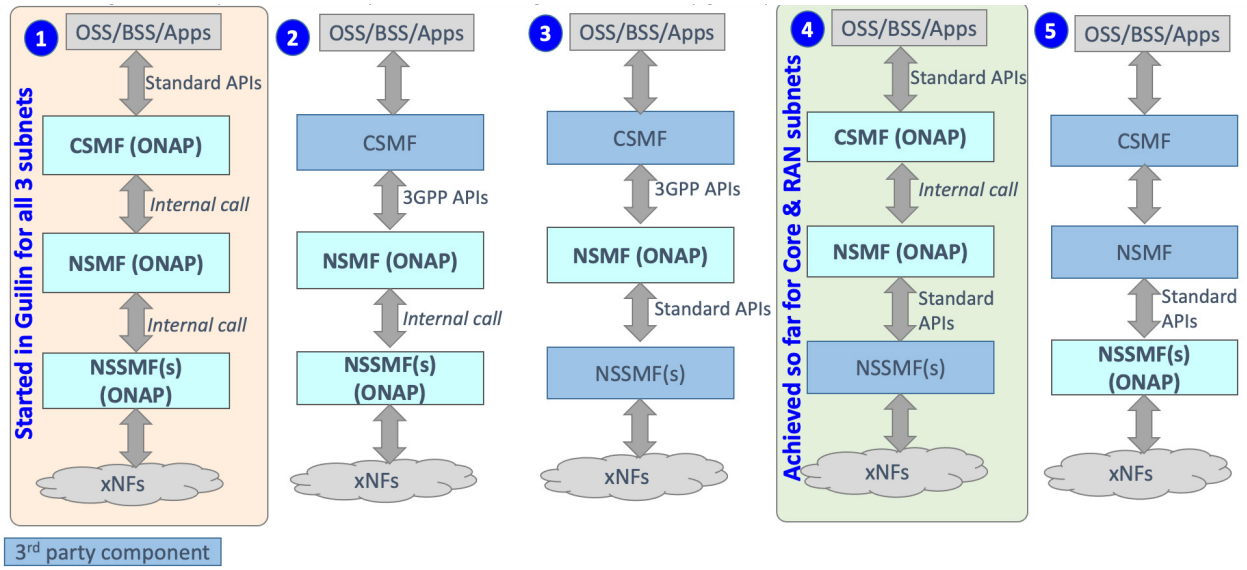


Figure 1: Architectural Options and Scope of ONAP Frankfurt Release

From a RAN and Transport Slicing perspective, two deployment options are foreseen, and Guilin release functionality has taken the first steps towards supporting both options. This, in turn, will enable CSPs and other ONAP users to cater to a wider range of deployment configurations depending on their need. An overview of RAN and Transport Slicing is illustrated in Figure 2, and the two deployment options are shown in Figures 3-4.

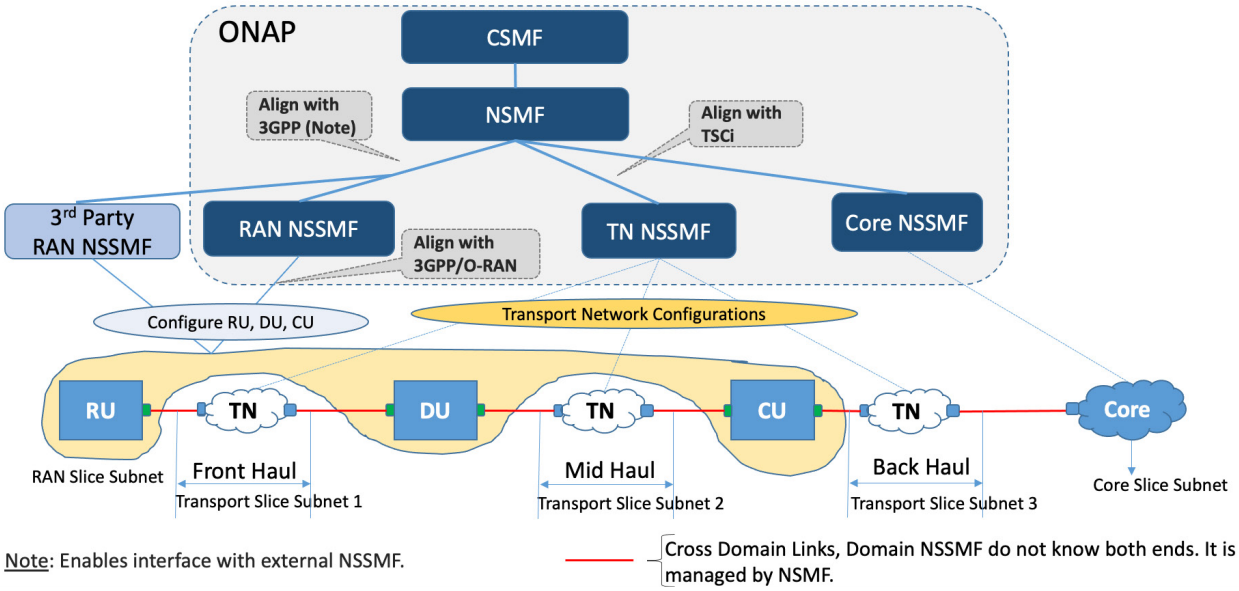


Figure 2: RAN and Transport Slicing overview

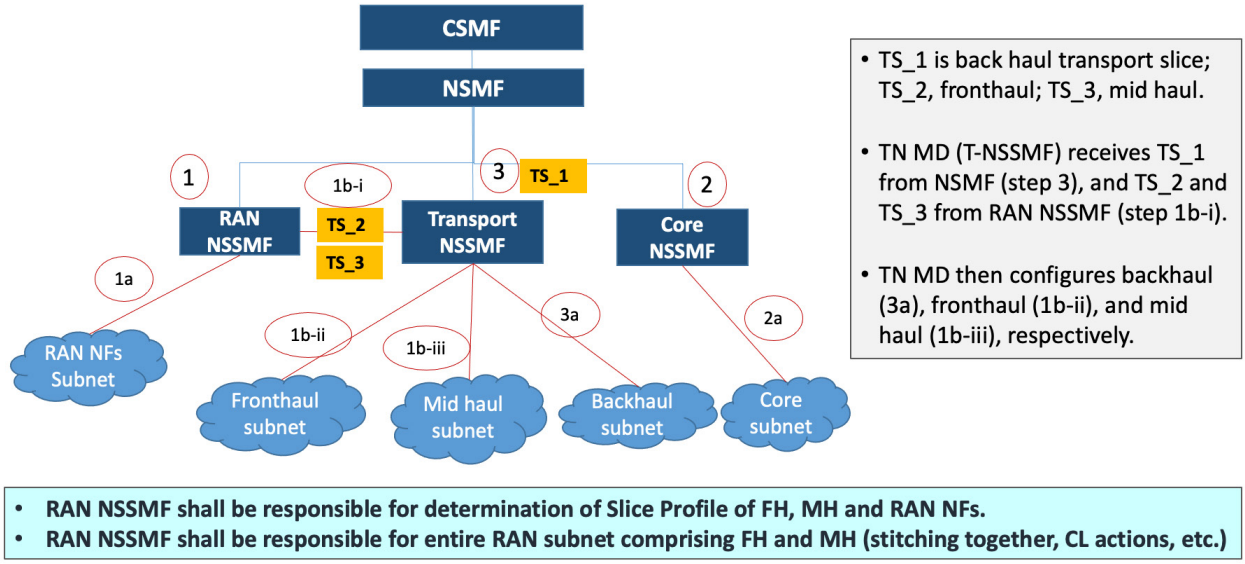


Figure 3: RAN and Transport Slicing: Deployment option 1

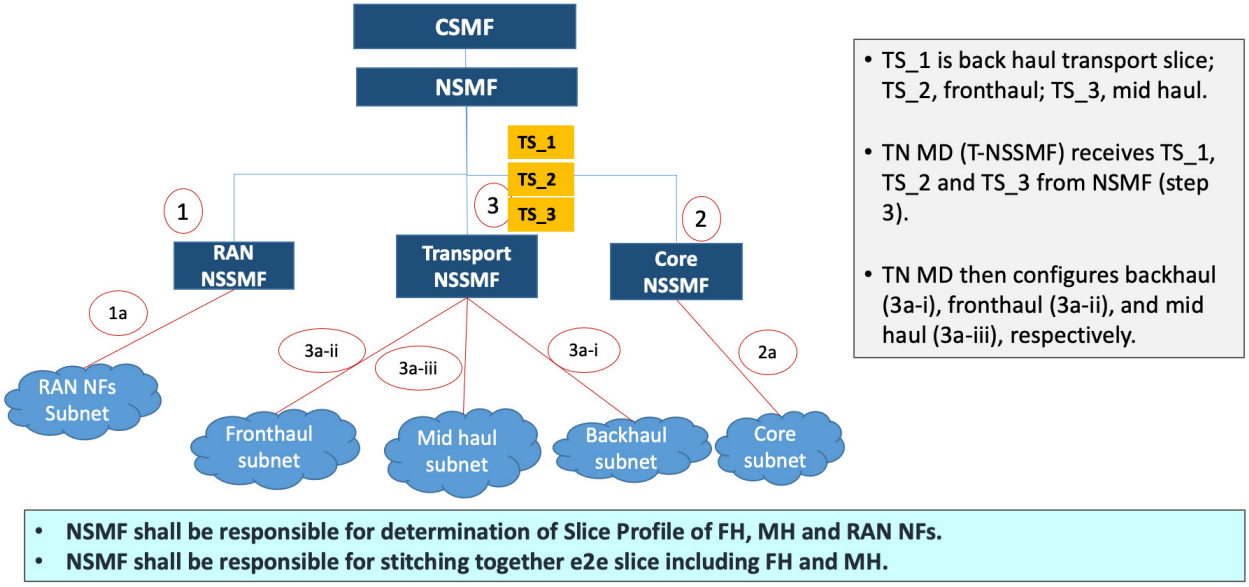


Figure 4: RAN and Transport Slicing: Deployment option 2

Problem Statement

Use case diversity and the need to address various industry vertical requirements necessitates Network Slicing. The following challenges exist in the realization of Network Slicing:

- A simple user interface for slice consumers, enterprises and industry verticals to request network slices by just providing service requirements
- A framework for network slicing design that aligns with service and resource models
- Providing the appropriate network slice as required for the service, including creation of a new slice (and its constituents) or reuse of a suitable existing slice, based on the user request, and active network slice inventory
- Dynamic network slice orchestration and lifecycle management to effectively cater to the needs of diverse services in an optimal manner
- Mechanism to intervene in the automatic network slice selection process to ensure correctness and making adjustments as needed

Requirements

The problem statement discussed above can be translated into the following high-level requirements:

  - Communication Service and Network Slice Design (aligning with 3GPP where relevant)
  - Design of communication service template (to capture service requirements from user)
  - Design of Service Profile
  - Design of Slice Profiles (for RAN, Core and Transport subnets)
  - Design of Network Slice Template (NST)
  - Design of Network Slice Subnet Template (NSST) for RAN, Core and Transport slice subnets
- Network Slice Instance allocation
  - Generate a new S-NSSAI
  - Determine the slice characteristics and requirements needed to support the service requested by the user
  - Based on user input on sharing the resources to be used for the service with other users/services:

- Sharing not allowed: Determine the properties (i.e., Slice Profiles of the subnets) of a new slice to be created
- Sharing allowed: Determine a suitable existing network slice instance for reuse, if none exists or not suitable, then determine the properties (Slice Profiles of the subnets) of a new slice to be created

- Network Slice Instance creation/update

  Based on the outcome of the previous step:

  - **New slice instance to be created:** Create a new network slice by triggering the NSSMFs, each of which, in turn, does the following:
    - Determine if an existing Network Slice subnet instance can be reused, or if a new Network Slice subnet instance has to be created, based on whether sharing is allowed, and the new Slice Profile's match with existing network slice subnet instances and
    - Create/Update network slice subnet instance. Based on the previous step if:
      - A new slice subnet instance to be created: Create a new slice subnet instance by creating/configuring the constituents as applicable.
      - An existing slice subnet instance to be reused: Update the existing slice subnet instance's configuration (link the new Slice Profile, update new S-NSSAI, etc.).
  - **Existing slice instance to be reused:** Update the configuration of the existing slice instance (link the new Service Profile, update new S-NSSAI, etc.), and instruct the NSSMFs to update the configuration - link the new Slice Profile generated for the respective subnets to the existing slice subnet instance (which is a constituent of the slice instance that is being reused), update the S-NSSAI list, etc.

- Service and Network Slice activation and deactivation
- Network Slice de-allocation and termination

Solution

The Open Network Automation Platform (ONAP) project is an open source project that provides a platform for designing, implementing, and managing different kinds of network services. ONAP enables service orchestration including automated life cycle management and resource orchestration of services. It has a robust service design framework, and orchestration includes FCAPS functionality, ETSI specified Management and Orchestration (MANO) layer functionality, and the ability to manage and control Software Defined networks—both IP and optical. It is also focused on management of software-defined Radio Access Network (RAN).

The Frankfurt release of ONAP introduced the Network Slice orchestration functionality by implementing two of the three 3GPP-specified Network Slice Management functions, namely Communication Service Management Function and the Network Slice Management Function, and an (external) interface to the third one, namely Network Slice Subnet Management Function (NSSMF) for the 5G core subnet. The Guilin release has taken the next significant steps by implementing the basic aspects of the NSSMF for the RAN, Core and Transport subnets, and also connecting to an external RAN NSSMF for the RAN subnet. Enhancements were also made to NSMF functionality in stitching together an end-to-end slice. A couple of basic Closed Loop scenarios (one with the help of Machine Learning done offline) involving the RAN subnet, and initial steps to monitoring of KPIs by the operator/slice tenant have been implemented - these will be further enhanced in upcoming releases.

With this functionality, an end-to-end Network Slice Instance (NSI) containing a suitable RAN, Core and Transport Network Slice Subnet Instances (NSSIs) can be allocated to fulfill a service's requirements. Based on the service requirements, and the currently active network slice and slice subnet instances, it could involve both reuse of existing NSI, or creation of a new NSI, wherein the creation of a new NSI could involve reuse of an existing Core/RAN NSSIs or Creation of a new Core/RAN/Transport NSSIs. The reuse/creation of RAN NSSI could also involve reuse/creation of Transport Fronthaul (FH), Mid-haul (MH) and RAN Network Functions (NF) NSSIs, in case of deployment option 1 as shown in Figure 3. The allocated NSI is then presented to the operator for confirmation that the choice is indeed appropriate or if any modifications are required. This functionality enables creation of services which can run on network slices. Further, basic activation/deactivation/deallocation of NSI and NSSIs are also supported.

Thus, the Guilin release takes yet another significant step towards realizing full-fledged end-to-end network slice orchestration functionality covering all network segments, as well as the entire life cycle management and resource orchestration of slices and slice subnets.

Implementation Details

The design of communication services to run on network slices is done by using the existing ONAP design constructs such as allotted resources. The steps for creating network slice template, its constituents and a service that can run on it is illustrated below.

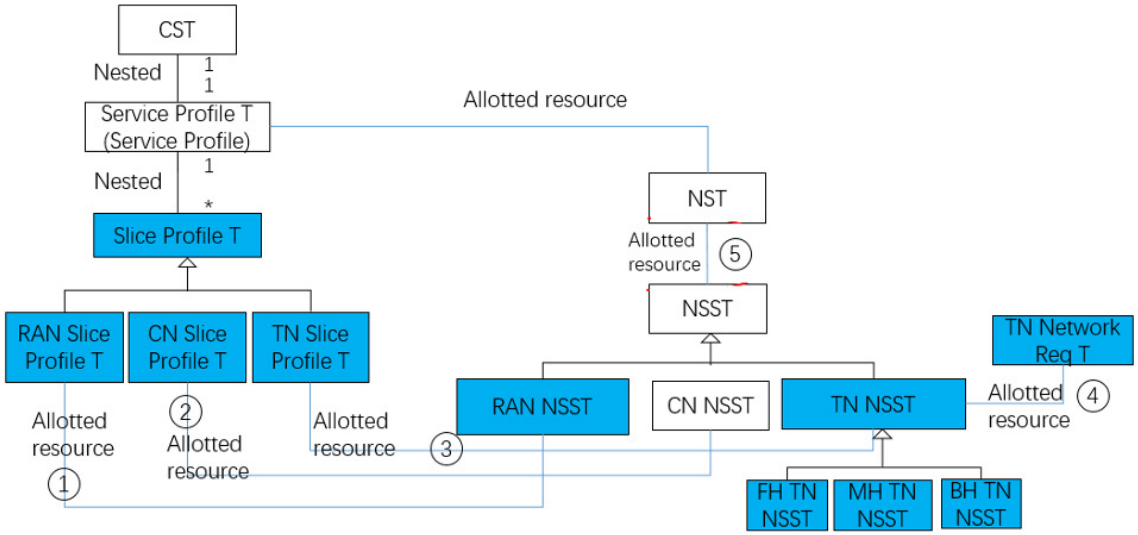


Figure 5: Design-time actions

Further details on this can be found here: https://wiki.onap.org/display/DW/Modeling+enhancements.

The run-time actions for instantiating a new service that runs on a network slice is depicted in Figure 6 below.

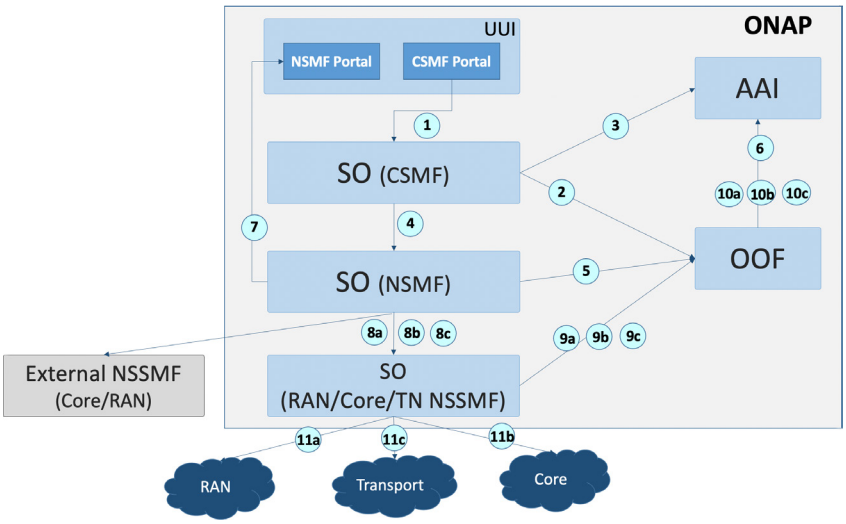


Figure 6: Simplified illustration of steps in service instantiation

The steps in Figure 6 are explained below:

1. The slice consumer requests a communication service with relevant inputs on the service's characteristics and performance requirements via the CSMF portal in UUI. The CSMF portal sends a request to SO acting as CSMF denoted as SO (CSMF).
2. SO (CSMF) converts the communication service request into a Service Profile and triggers OOF for providing a suitable Network Slice Template (NST).
3. SO (CSMF) updates AAI with the communication service details. OOF provides the suitable NST back to SO.
4. SO (CSMF) then triggers SO (NSMF) for allocating a suitable NSI.
5. SO (NSMF) triggers OOF for selecting a suitable NSI (if allowed to be shared and available, or Slice Profiles of the constituent NSSIs for a new NSI to be created)..
6. OOF checks Policy inputs and existing inventory (AAI), and then selects the suitable NSI and provides it to SO (NSMF). In case no suitable NSI exists or if the service request required a non-shared service to be instantiated, then OOF provides suitable RAN/Core/TN Slice Profiles in order to create a new NSI.
7. SO (NSMF) shares details of the selection made by OOF with NSMF Portal in UUI for the operator to check and make any modifications if needed.
8. SO (NSMF) triggers RAN/Core/TN NSSMFs (in Steps 8a/8b/8c respectively) for allocating/updating the RAN/Core/TN NSSIs. Note that the RAN/Core NSSMFs could be within ONAP or external to ONAP also.
9. SO (RAN/Core/TN NSSMF) triggers OOF for selecting a suitable NSSI with the corresponding Slice Profile.
10. OOF checks Policy inputs and existing inventory of NSSIs. If sharing is allowed and an existing NSSI matches the Slice Profile, then OOF returns the existing NSSI that can be reused (in case of RAN and Core), otherwise it informs SO (NSSMF) to create a new RAN/Core/TN NSSI.
11. SO (RAN/Core/TN NSSMF) then creates/updates RAN/Core/TN NSSI and its constituents.

The inventory (AAI) is also updated with the Service Profile, NSI, NSSIs, Slice Profiles and S-NSSAI, and appropriate links are created (not shown in above sequence diagram). For more detailed flows, refer to https://wiki.onap.org/display/DW/Use+case+flows.

Let us take a look at the implementation done in various ONAP components to realize the above functionality. The description below does not distinguish between what was implemented in Frankfurt and Guilin.

User interface (CSMF Portal and NSMF Portal)

The user interface (UUI) provides the following functionality:

a. **CSMF portal:** The customers fill the create communication service form in CSMF portal to create a network service that uses a slice, and then they can see the created network service in the list, and execute operations of activating, deactivating or terminating the network service

b. **NSMF portal:** This consists of:

- Slicing Task management in which network operators can find all the slicing tasks created by customers in CSMF component and executing proper operations according to different task status. This is also used to verify and modify slice options suggested by OOF if needed.
- Slicing Resource management which provide the functions of displaying and processing the existing NS, NSI and NSSI.

Service Orchestrator (SO)

In SO, there are separate BPMN workflows for CSMF ,NSMF and various NSSMFs. CSMF workflow processes the service request that comes from CSMF portal (UUI) and saves the order information into a communication service instance in AAI. CSMF workflow sends network slice request to NSMF workflow, and NSMF then creates service profile, NSI, and NSSI. Service profile is a logical concept which exists only in AAI—it contains two AAI instances, one is a profile instance that will hold the slice parameters, and the other is a service instance which will be used to organize the NSI. NSI is also a service instance in AAI which will be used to organize NSSI. NSSI is the actual entity which will be created by NSSMF and an AAI service instance will also be created to represent NSSI in an ONAP context. NSI and NSSI can both be shared.

SO queries OOF for slice template selection and then NSI and NSSI selection. In response to slice instance selection query, OOF may return an existing slice instance or may recommend SO to create a new slice instance. A new process called Orchestration Task is created to manage recalibration of NSI&NSSI selection with manual intervention by the operator via NSMF Portal in UUI. AnNSSMF adapter in SO interacts with internal/external NSSMFs for NSSI orchestration.

The NSSMF functionality has a common part which responds to subnet capability query from SO (NSMF), and it invokes the domain (RAN/Core/TN) specific NSSMF functionalities. The domain-specific NSSMF workflows are responsible for performing necessary actions for creating/updating the NSSI based on the feedback from OOF (for NSSI creation/reuse). In case of RAN NSSI, for deployment option 1 (Figure 3), the SO (RAN NSSMF) further invokes SO (TN NSSMF) via the NSSMF adaptor for creation/updation of TN Fronthaul and TN Mid haul NSSIs.

Creation of a RAN NSSI involves configuration of RAN Network Functions (NFs) by SO (RAN NSSMF) as the RAN NFs are assumed to be already instantiated. Creation of a Core NSSI involves invocation of the Macro workflow in SO (via ExtAPI) by SO (Core NSSMF) resulting in creation of new Core NFs. Reuse of a Core NSSI Core NSSI involves invocation of the Macro workflow in SO, by SO (Core NSSMF) resulting in update of the S-NSSAIs served by the Core NFs. Creation of a TN NSSI involves setup/configuration of appropriate transport paths driven by SO (TN NSSMF) through the SDN-C.

SO also provides Service activation, deactivation and service termination functionalities, for which the SO (CSMF),SO (NSMF) and SO (NSSMF) are involved.

ONAP Optimization Framework (OOF)

OOF is invoked by SO for Network Slice Template (NST) selection, and NSI/NSSI selection. In case of NSI selection, as described earlier, OOF may return:

- Existing NSI, if the service request indicates it is shareable and if a suitable NSI exists
- Slice Profiles, if the if the request indicates it is non-shareable or no suitable NSI exists. The Slice Profiles for the RAN, Core and TN subnets shall be generated to conform to the Service Profile in accordance with the capability of the corresponding subnet.

In case of NSSI selection, OOF may return:

- Existing NSSI, if the service request indicates it is shareable and if a suitable NSSI exists
- Constituent Slice Profiles (in case of RAN for deployment option 1 – Figure 3), or no solution

In addition, OOF is also used to determine if an NSI/NSSI has to be terminated when it is being deallocated for a service.

External API (ExtAPI)

A new value "CST" for the serviceType attribute in the TMF 641 based Service Order API has been introduced. The relatedParty attribute in the Service Order is set according to the Customer, where relatedParty.id will map to the AAI "global-customer-id" in the "customer" object. The serviceSpecification.id is to be set to the UUID of the CST from SDC (i.e., this is the template for the Service we are ordering from CSMF). The action field will be set to "add" to indicate creation of a new service instance, "modify" for activation/deactivation of a service instance (the attribute serviceState = active/inactive will indicate if it is activation or deactivation), and "delete" for termination of a service instance. Upon sending a POST with the JSON body to /{api_url}/nbi/api/v4/serviceOrder/, ExtAPI will generate a Service Order ID and send it in the response—this ID can be used to track the order. ExtAPI will then invoke SO's API for creating the service. This is a step towards standardizing the external interfaces for slice orchestration.

Active & Available Inventory (A&AI)

A&AI module has 3 new nodes (Communication-service-profile, Service-profile and Slice-profile), modified service-instance nodes, added 3 new nodes as new attributes of service-instance node.
To map to SDC templates (Communication Service Template/Service Profile Template/Slice Profile Template/NST/NSST), run-time instances of this use case have Communication Service Instance/Service Profile Instance/Slice Profile Instance/NSI/NSSI. Note that the Slice Profile Templates for the 3 subnets (RAN, Core and Transport) are different. Further, for the 2 deployment options for RAN and Transport as shown in Figures 3-4, different RAN NSSTs are designed.

To align with ONAP's model-driven approach, this use case reuses "service-instance" for all run-time instances. The relationship between service-instances use the existing attribute "relationship-list" or "allotted-resources". Communication-service-profile holds the original requirement of Communication-

service-instance, such as latency, data-rate, and mobility-level. Service-profile holds the slice parameter info of Service-profile-instance. Slice-profile holds the slice subnet parameter info of different network domain NSSIs, such as (Radio) Access Network (AN), Transport Network (TN), and Core Network (CN) NSSI.

A&AI provides query APIs to CSMF and NSMF, such as:

- Query Communication-service-instances/Service-profile-instances/NSI/NSSI
- Query Service-profile-instance by specified Communication-service-instance
- Query NSI by specified Service-profile-instance, query NSSI by specified NSSI

A&AI also supply creation APIs to SO, such as:

- Create Communication-service-profile/Service-profile/Slice-profile
- Create relationship between service-instances, between Service Profile and Slice Profiles

For Transport Slicing, AAI is updated with the TSCi information model.

CCSDK/SDN-C

When invoked by SO (TN NSSMF) for TN NSSI creation, CCSDK/SDN-C sets-up/configures the new transport NSSI. This also involves mapping from TSCi model to the relevant southbound models with the help of ACTN MPI adaptor on the southbound towards the optical domain controller. Similarly, CCSDK/SDN-C updates the transport network during NSI reuse, activation/deactivation and termination.

When invoked by SO (RAN NSSMF) for RAN NSSI creation, CCSDK/SDN-C (SDN-R) configures the RAN NFs, namely the CUs, DUs and Near-Real-time Radio Intelligent Controllers (Near-RT RICs) that have already instantiated. In case of RAN NSSI reuse or NSI reuse, CCSDK/SDN-C re-configures the RAN NFs appropriately. When triggered by Policy for Closed Loop actions in RAN for RAN NSSI(s), CCSDK/SDN-C sends appropriate configuration updates to the Near-RT RICs.

Details of RAN configuration is obtained from Config DB which is not a part of ONAP. In future ONP releases, the configuration information stored in this DB is intended to be moved to the Configuration Persistence Service (CPS).

DCAE

- **Data Exposure Service (DES):** This new micro-service provides an easy interface to query current and historical PM/KPI data by the operator/slice tenant or any other ONAP component. It abstracts the database or the data lake (which may be within ONAP or external to ONAP) interface by providing a flexible API interface. The API to database query mappings can be realized with the help of templates which can be loaded to DES.
- **Slice Analysis MS:** This new micro-service performs two functions:

a. Analyzes PM data received from the RAN (Downlink and Uplink PRBs used for data traffic for each S-NSSAI) via the PM-Mapper micro-service, and determines the updates (if any) to the aggregate throughput for all the cells under a Near-RT RIC's coverage area. Subsequently, it activates a Control Loop by sending an appropriate DMaaP message to Policy containing the configuration updates to be sent to the RAN (via SDN-C (R)). This is illustrated in Figure 7.

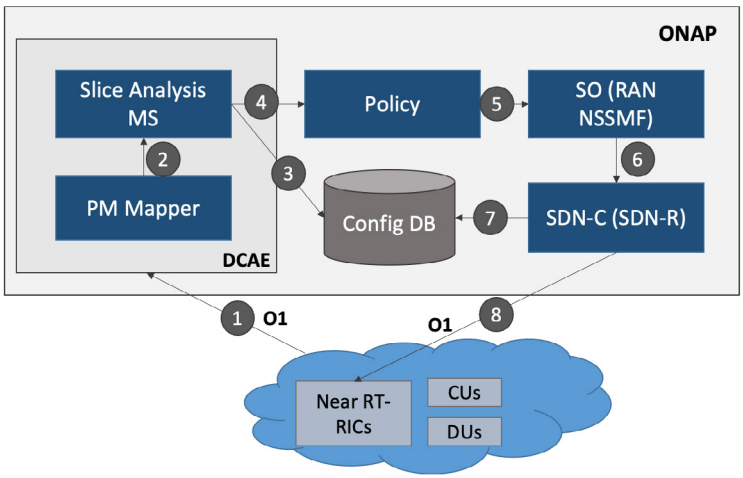
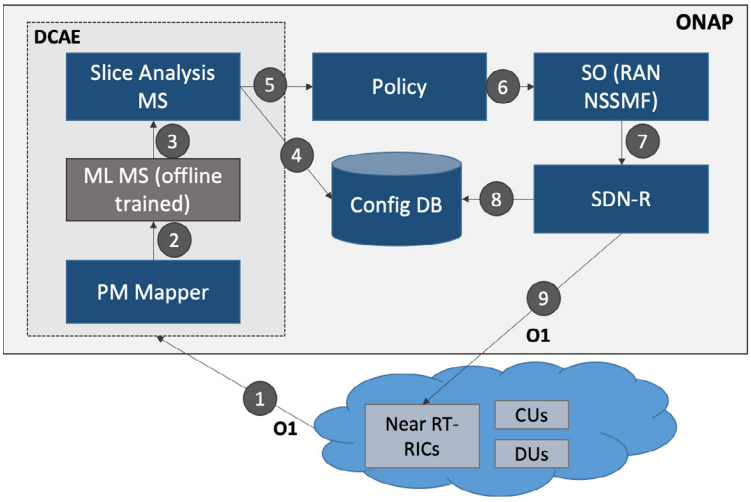


Figure 7: Simplified illustration of Closed Loop flow

b. Upon reception of configuration updates from an offline trained ML-model (onboarded as an ML microservice) the maximum number of connections to be supported by each cell for every S-NSSAI, , it activates a Control Loop by sending an appropriate DMaaP message to Policy containing the configuration updates to be sent to the RAN (via SDN-C (R)). This is illustrated in Figure 8. Note that the ML microservice is not an ONAP component and is onboarded to DCAE for demonstrating the functionality.



The details of the current RAN configuration are obtained by the Slice Analysis MS from the Config DB (similar to SDN-C (R) as explained above).

Figure 8: Simplified illustration of steps in ML-based Closed Loop flow

Summary

Network Slicing in 5G is taking off in real-world deployments and is on the cusp of playing a key role in catering to the needs of industry verticals apart from the traditional mobile broadband service needs in an optimal manner. The scope of this use case and the functionality that was started in Frankfurt and then enhanced further in Guilin are important steps towards realizing true end-to-end network slicing.

As this use case continues to evolve in the upcoming releases in both breadth, i.e., covering all network segments, adopting standard external interfaces and enabling different architecture options, as well as depth, i.e., implement the entire lifecycle of slice orchestration, slice and service assurance, it shall become one of the differentiating features from an ONAP platform perspective. This shall provide immense value to CSPs in providing robust and flexible slice orchestration capabilities, as well as in decreasing the time-to-market to offer differentiated capabilities and features with regard to Network Slicing.

Resources

1. E2E Network Slicing wiki page in Guilin release
   https://wiki.onap.org/display/DW/E2E+Network+Slicing+Use+Case+in+R7+Guilin
2. E2E Network Slicing wiki page in Frankfurt release
   https://wiki.onap.org/display/DW/E2E+Network+Slicing+Use+Case+in+R6+Frankfurt

Contributors

Lin Meng, China Mobile

Swaminathan Seetharaman, Wipro Limited

Shankaranarayanan Puzhavakath Narayanan, AT&T

Min Zhang, China Mobile

Borislav Glozman, Amdocs