Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 52-72
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Coordinated Management of 5G Core Slices by MANO and OSS/BSS

Wei-Cheng Chang, Fuchun Joseph Lin

Department of Computer Science, College of Computer Science, National Yang Ming Chiao Tung University, Taiwan
Email: jefferynthu@gmail.com, fjlin@nycu.edu.tw

How to cite this paper: Chang, W.-C. and Lin, F.J. (2021) Coordinated Management of 5G Core Slices by MANO and OSS/BSS. *Journal of Computer and Communications*, **9**, 52-72.
https://doi.org/10.4236/jcc.2021.96004

Received: May 9, 2021
Accepted: June 15, 2021
Published: June 18, 2021

Abstract

To manage and orchestrate Network Slices (NSs) for 5G Core (5GC), the MANO (MANagement and Orchestration) framework is proposed by European Telecommunications Standard Institute (ETSI). In most research testbeds, MANO systems such as Tacker, OSM and ONAP are used to initiate network slices. However, this doesn't comply with the 3GPP 5G standards as MANO should only be responsible for dynamic management of NSs, and the static management such as provisioning or unprovisioning a network slice should be left to OSS/BSS (Operation/Business Support System). Thus, in our testbed, an integrated architecture was designed in which the management of network slices will be coordinated by both MANO and OSS/BSS. MANO would handle on-boarding, instantiating, scaling and terminating of network slices while OSS/BSS is responsible for static management of slices including provisioning and unprovisioning of network slices. To evaluate our system, it was compared with the management systems equipped with only OSS/BSS or MANO in order to analyze the shortfalls of those systems when used to deploy network slices. Through this analysis, this research confirms the necessity of applying both OSS/BSS and MANO for the coordinated management of 5G core slices as adopted by 3GPP.

Keywords

Network Slice, 5G, MANO, OSS/BSS, Network Management

1. Introduction

The fifth generation (5G) networks [1] are going to provide a large variety of services with diverse Quality of Service (QoS) requirements, such as Enhanced Mobile-Broadband (EMBB), Ultra-Reliable and Low Latency Communications (URLLC) and Massive Machine Type Communications (mMTC). In order to meet so many service requirements on the same hardware infrastructure, network slicing is proposed in 5G as one of its key features.

Network slicing is defined by 5G.co.uk as "providing dedicated virtual networks with functionality specific to the service or customer over a common network infrastructure". It is enabled by Network Function Virtualization (NFV) [2] technologies which allow a Network Service (NS) to be composed of multiple Virtual Network Functions (VNFs) and each VNF be executed on a Virtual Machine (VM) or container. With network slicing, a single physical network infrastructure can be sliced into multiple virtual networks so the diverse requirements of 5G network services can be satisfactorily met.

To manage and orchestrate network slices in 5G, MANO (MANagement and Orchestration) proposed by ETSI NFV was adopted by 3GPP. MANO is responsible for dynamic management such as initiating, terminating and monitoring a network slice and its VNFs for the purpose of scaling or auto-healing in real-time. On the other hand, the traditional OSS/BSS (Operation/Business Support System) still plays an important role on static management such as provisioning and unprovisioning of a network slice.

This paper shows an integrated testbed of MANO and OSS/BSS for the coordinated management of 5G core slices based on 3GPP 5G standards. Our testbed is constructed using free5GC [3], free5GMANO [4], Tacker [5] and OpenStack [6] open sources. Our research objective is to demonstrate that OSS/BSS and NFVO can work in a coordinated fashion to manage the life cycle of network slices including provisioning, instantiation, termination and unprovisioning. To evaluate our proposed system, a comparison was conducted with two compared systems: OSS only system and MANO only system respectively.

How to integrate OSS/BSS and MANO has been clarified in [7]; it explains the separate functions of OSS/BSS and MANO and the interfaces between them.

The MATILDA project [8] proposed a highly modularized OSS for managing the different components of the whole 5G infrastructure; this OSS can work with the underlying Convergence Layer to manage network slices. This convergence layer provides NFV MANO functionality; however, their OSS/BSS is not structured according to what 3GPP defined as there is no NSMF and NSSMF and the system also didn't provide EMS (NFMF).

A top-down architecture for integrating OSS/BSS and MANO is provided in [9] where an NS allocation request can be sent from Vertical Service Provider to CSMF, then CSMF to NSMF, NSMF to NSSMF, and finally NSSMF to MANO to create a NSSI (Network Slice Subnet ID). Though an integrated architecture of OSS/BSS and MANO is constructed, it is not based on 5GC but on 4G EPC and thus only supports the 5G NSA architecture.

The rest of the paper is organized as follows: Section II introduces the background and related work of the technologies used in our testbed. Section III introduces our experimental management systems of 5G network slices, including our proposed system and two compared systems. Section IV describes the implementation and qualitative evaluation of the above three systems. Finally, section V presents our conclusion and future work.

2. Background & Related Work

In order to build an OSS/BSS and MANO integrated system, not only the ETSI NFV-MANO [10] architecture and the 3GPP OSS/BSS framework but also multiple open source technologies were utilized. In this section, we present the background of these technologies.

2.1. ETSI NFV MANO Architecture

ETSI NFV-MANO is the key enabler for deploying 5G Network Services (NSs). Each NS supports a network slice subnet defined in 5G and an end-to-end network slices consists of multiple network slice subnets. Unlike Physical Network Functions (PNFs) traditionally deployed on proprietary hardware, MANO supports the deployment of VNFs on commodity hardware. Note that each network slice subnet or NS is composed of one or multiple VNFs and these VNFs are connected by Virtual Links (VLs) [11] and their interaction are presented as VNF Forwarding Graphs (VNFFGs). Operators need to provide NSs with different QoS requirements such as EMBB, URLLC and mMTC [12].

Figure 1 shows the architecture of our proposed system based on the ETSI-NFV framework. It consists of five major parts: NFV-MANO, OSS/BSS, VNFs coupled with EMSs (Element Management Systems) and NFVI (NFV Infrastructure).

1) NFV-MANO: This is shown in the right of **Figure 1** enclosed by a dashed rectangle. NFV-MANO is in charge of configuring, deploying, managing and scaling NSs and VNFs. It consists of three major components:

- NFV Orchestrator (NFVO): NFVO is in charge of the orchestration of NFVI resources, the life cycle management of NSs and the repositories of Network Service Descriptors (NSDs) [13] and Virtual Network Function Descriptors (VNFDs) [11]. Here NSD is a deployment template used by the NFVO for the life cycle management of NSs; it includes the descriptors of all NS components such as VNFDs, VLDs (VL Descriptors) and VNFFGDs (VNFFG Descriptors). On the other hand, VNFD is a template used to define various VNF properties including virtual resources used, connection points and virtual links, and lifecycle management behaviors such as scaling.
- VNF Manager (VNFM): VNFM is in charge of the life cycle management of VNFs, including their creation, modification and termination. VNFM also works with EMS to supervise every VNF.
- Virtual Infrastructure Manager (VIM): VIM is responsible for managing the NFVI resources including computing, storage and network. These resources are transformed from hardware resources to virtual resources via a virtualization layer. There are two types of VIMs: VM-based VIM and container-based VIM.

2) OSS/BSS: The upper-left part of **Figure 1** shows OSS/BSS that supports

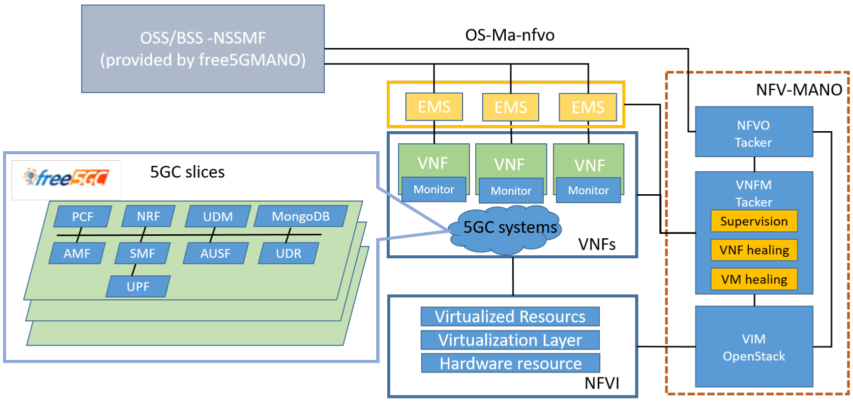


Figure 1. Proposed system architecture.

network management functions including fault management, configuration management, accounting management, performance management and security management (FCAPS).

3) VNFs coupled with EMSs: The middle part of Figure 1 shows VNFs and EMSs. VNF is a virtualized network function that can be deployed on the commodity hardware while EMS is designed to support the management of a virtual or physical network function. Each VNF is coupled with an EMS.

4) NFVI: The middle-bottom of Figure 1 shows NFVI which provides hardware resources such as computing, storage and network for VNFs. The NFVI resources are assigned to a VNF according to the requirements specified in the VNFD.

We will explain the left part of Figure 1, 5GC slices, in the next section.

2.2. 3GPP OSS/BSS Framework

MANO has a great impact on the traditional OSS/BSS architecture as the latter needs to implement new components to interact with the former. As shown in Figure 2, OSS/BSS has to have the following two functions: Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF) [14] [15]. NSMF provides the management services for Network Slice Instances (NSIs) while NSSMF provides the management services for Network Slice Subnet Instances (NSSIs). Here one NSI could consist of multiple NSSIs and one NSSI could contain multiple VNFs.

The EMS plays the role of Network Function Management Function (NFMF) [14] that provides the management services for PNFs. Also in order to manage VNFs, NFMF interacts with VNFM through the Ve-Vnfm-em interface. On the other hand, at a higher level NSSMF interacts with NFVO through the Os-Ma-NFVO interface [16]. NSSMF and NFMF coordinate with MANO to perform FCAPS network management functions defined by International Organization Standardization (ISO).

2.3. Open Source Technologies

NCTU free5GC open-source was used as our 5G core network; this open source

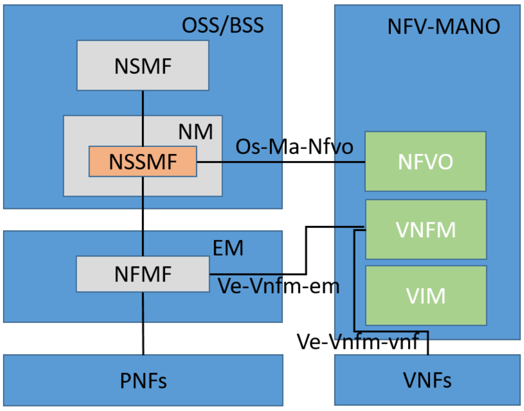


Figure 2. 3GPP OSS/BSS interact with NFV-MANO.

aims to provide 5G Core (5GC) based on 3GPP Release 15 and beyond. Unlike other 5GC open sources such as Eurecom OpenAirInterface (OAI) [17] that is based on the Non-StandAlone (NSA) 5G framework, free5GC is based on the StandAlone (SA) 5G framework [18].

As depicted in Figure 3, the components of 5GC include Network Repository Function (NRF), Access and Mobility Management Function (AMF), Session Management Function (SMF), Unified Data Repository (UDR), Policy Control Function (PCF), Unified Data Management (UDM), Network Slice Selection Function (NSSF), Authentication Server Function (AUSF) and User Plane Function (UPF).

NCTU free5GC separates Control Plane (CP) and User Plane (UP). It also adopts a modularized function design so it can be deployed in a flexible and efficient way on the MANO platform. NFs in the control plane communicate with each other through service-based interfaces while User Equipment (UE), Radio Access Network (RAN) and UPF in the user plan access the NFs in control plane through reference point interfaces including N1, N2 and N4 [1].

NTUST free5GMANO [4] is an open-source project based on 3GPP TS 28.531 [19] and TS 28.532 [20] Release 15 (R15) for orchestrating and managing 5G network slices. The NSSMF part of free5GMANO was incorporated in our testbed as the representative function of OSS/BSS. Currently, NSSMF in free5GMANO can work with both VM-based and container-based MANO. But in our research, only the former solution was used.

OpenStack/Tacker open sources were used as our MANO system: Tacker serves as NFVO/VNFM while OpenStack serves as VIM. OpenStack is an open source cloud operating system that virtualizes and manages infrastructure resources including computing, network and storage. Tacker is an official OpenStack project for orchestrating and managing NSs and VNFs. It supports multiple VIMs including OpenStack and Kubernetes.

3. OSS/BSS and MANO Integrated and Silo Design

In this section, we present our proposed system architecture and two compared

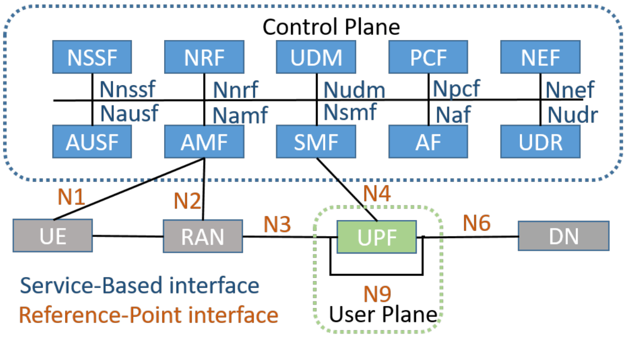


Figure 3. NCTU free5GC system architecture.

system architectures (which uses silo MANO or OSS/BSS) along with the work flows of these three systems. Next, we elaborate life cycle management and fault management mechanisms of these three systems. In the end, we would summarize the differences between three systems.

3.1. Architecture of OSS/BSS and MANO Integrated Design

Our research objective is to use the integrated system of OSS/BSS (provided by the NSSMF part of free5GMANO) and MANO (provided by OpenStack/Tacker) to demonstrate both configuration management and fault management of 5GC (provided by free5GC) network slices.

Configuration management is meant to deploy right attributes to each network function. Because our 5G network services are allocated on a virtualized environment. OSS/BSS only needs to provide NSDs and VNFDs while MANO is responsible for allocating virtualized resources required and activating VNFs according to the requirements.

As for fault management, MANO needs to detect each 5GC VNF's status to ensure every one works normally. If it detects any anomaly of VNF, it would notify OSS/BSS through the VNF's corresponding EMS.

Because free5GMANO doesn't provide EMS, we designed our own EMS to cooperate with VNFM in order to handle a VNF's fault management.

Figure 4(a) and Figure 4(b) show the flows of deployment and undeployment for a network slice, respectively. Figure 4(a) first shows that the developer should design NSDs and VNFDs that describe the attributes of NSs and VNFs via OSS/BSS. Then, OSS/BSS will do the provisioning by activating EMS and sending the NSDs, VNFDs and NS creating request to MANO.

Next, MANO will onboard the NSDs and VNFDs to NSD catalogue and VNFD catalogue respectively. After that, MANO would instantiate NS according to the NSD and VNFD. After the NS is created, MANO will start to monitor the status of each VNF and carry out fault management if needed.

On the other hand, Figure 4(b) shows that if a network service is no longer needed, OSS/BSS can do the un-provisioning by sending the NS deletion request to MANO. MANO then will delete the network slice and its related NSD and VNFDs.

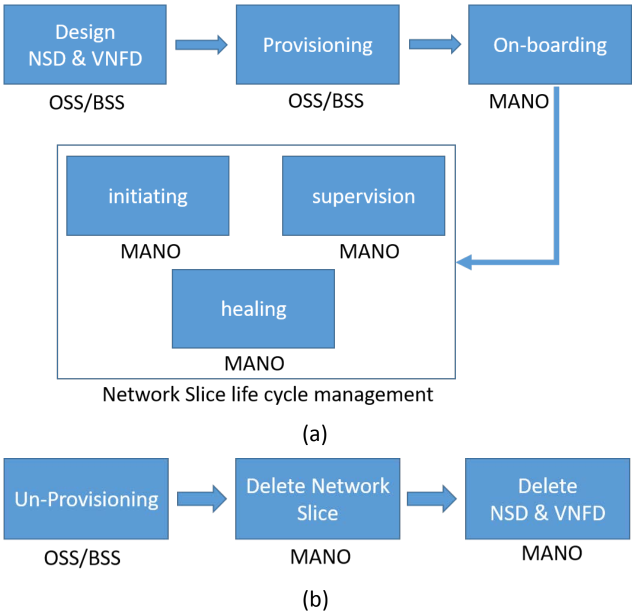


Figure 4. OSS/BSS and MANO integrated system flow (a) deployment (b) undeployment.

3.2. Architecture of OSS/BSS and MANO Silo Design

Next, we are going to present our two compared systems. The first compared system is the OSS/BSS only system. All the 5G core network functions are built as PNFs, and all the orchestration and management jobs would be handled by OSS/BSS and EMS, Figure 5 shows the architecture of this system. Note that in the OSS/BSS block, only the Command Line Interface (CLI) part of free5GMANO was used because only this interface was needed to operate OSS/BSS. Moreover, NSSMF wasn't needed because there was no need to work with MANO in this system.

The flow of OSS/BSS only system is shown in Figure 6. If we want to activate a 5G core network service, OSS/BSS needs to activate all the network functions and their corresponding EMSs on the physical hardware resource. The EMS would start supervising all the PNFs for configuration and fault management. Finally, if no longer needed, OSS/BSS would send terminating requests to all the PNFs to shut down this 5G service.

The second compared system is the MANO only system as shown in Figure 7. In this architecture, MANO is the orchestrator of the whole system. Since there is no OSS/BSS, it has to also perform static management normally done by OSS/BSS such as provisioning and unprovisioning 5G services. The network slice deployment and undeploymemt flows are shown in Figure 8(a) and Figure 8(b) respectively. In these two flows, we can observe that MANO needs to handle not only dynamic management jobs such as slice creation and deletion, but also static management jobs like provisioning and unprovisioning.

3.3. Life Cycle Management Design

Next, we talk about the life cycle management of our three systems one by one.

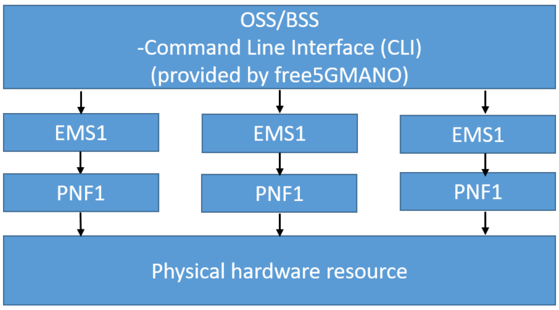


Figure 5. OSS/BSS only system architecture.

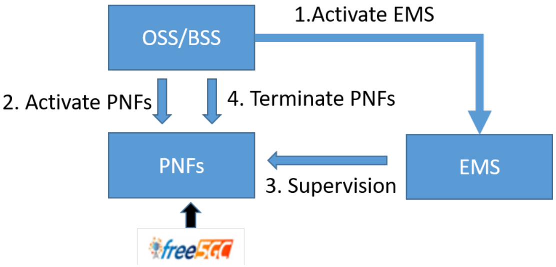


Figure 6. OSS/BSS only system flow.

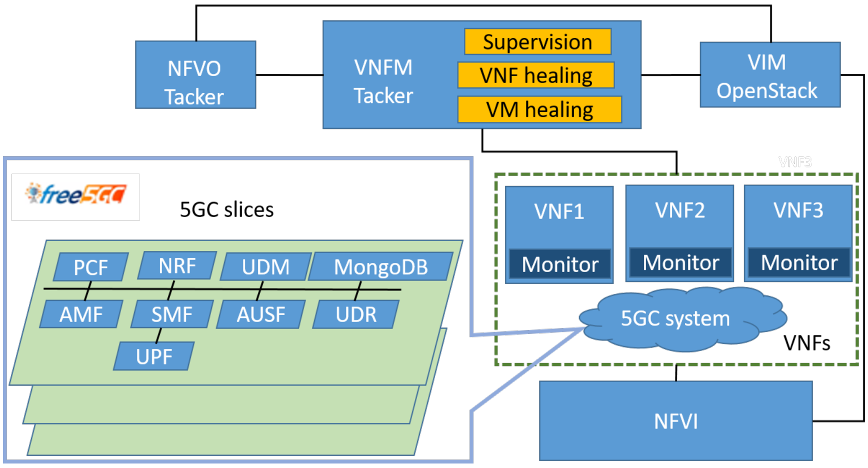


Figure 7. MANO only system architecture.

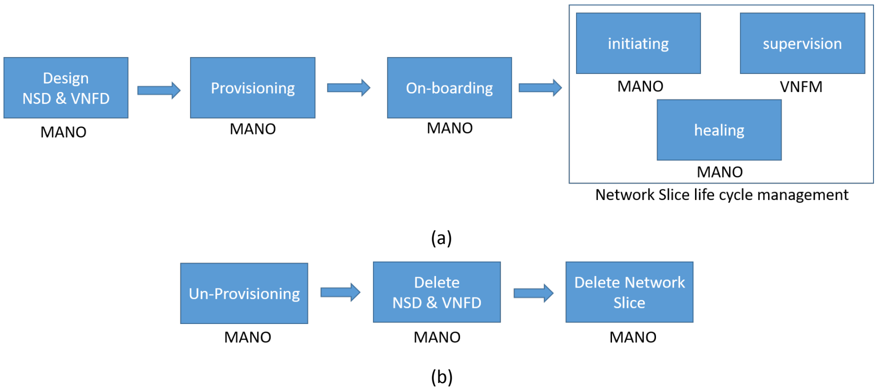


Figure 8. MANO only system flow (a) deployment (b) undeployment.

First is our OSS/BSS with MANO integrated system in which OSS/BSS is responsible for static management such as provisioning and unprovisioning of NS and VNF via designs of NSD/VNFD templates and MANO is responsible for dynamic management such as allocating virtualized resource and activating/supervising/deactivating VNFs. The information like how many virtualized resources should be given to the network slices and how to activate the VNFs are specified in NSD/VNFDs. MANO would just follow the instructions in the templates. The detailed flows have been shown previously in Figure 4.

Second is the OSS/BSS only system. In this case, network functions are built as PNFs on bare machines. Without MANO's help to do configuration management, manual effort is required to pre-configure free5GC configuration files on the server. OSS/BSS also needs to activate/deactivate PNFs one by one by itself. The detailed flow has been shown previously in Figure 6.

Third is the MANO only system, without OSS/BSS's help, MANO has to do both static management and dynamic management, so MANO not only needs to allocate virtualized resources to network slices and activate VNFs but also support design of NSD/VNFD templates. The detailed flow has been shown in Figure 8.

3.4. Fault Management Design

In our proposed system, a mechanism was designed to handle the VNFs' failure. There are two kinds of fault that may occur on VNFs [21]: one is the virtualized resource fault, the other is the VNF fault. Our system can deal with both situations. We will go through these two cases one by one.

First is the virtualized resource fault management. The VMs run on OpenStack platform may be paused, suspended or shut down. Therefore, it needs some method to automatically detect VMs' status and do recovery actions when needed. The virtualized resource fault detection flow is shown in Figure 9. Because Tacker support few fault detecting functions, a monitor service was implemented as a plugin of VNFM. After VNFM deploys a new VNF, it would ask the monitor service to create a new monitor which is designed to detect VM's status using OpenStack's built-in API. If the monitor detects the fault of a VM, VNFM would notify the corresponding EMS.

The detailed interaction flow between EMS and MANO is shown in Figure 10. After EMS is activated by OSS/BSS, it would subscribe to the notification

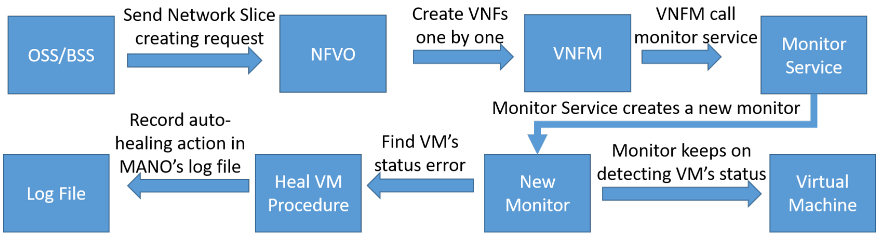

OSS/BSS
Send Network Slice creating request
NFVO
Create VNFs one by one
VNFM
VNFM call monitor service
Monitor Service
Monitor Service creates a new monitor
Log File
Record auto-healing action in MANO's log file
Heal VM Procedure
Find VM's status error
New Monitor
Monitor keeps on detecting VM's status
Virtual Machine

Figure 9. Virtualized resource fault management.

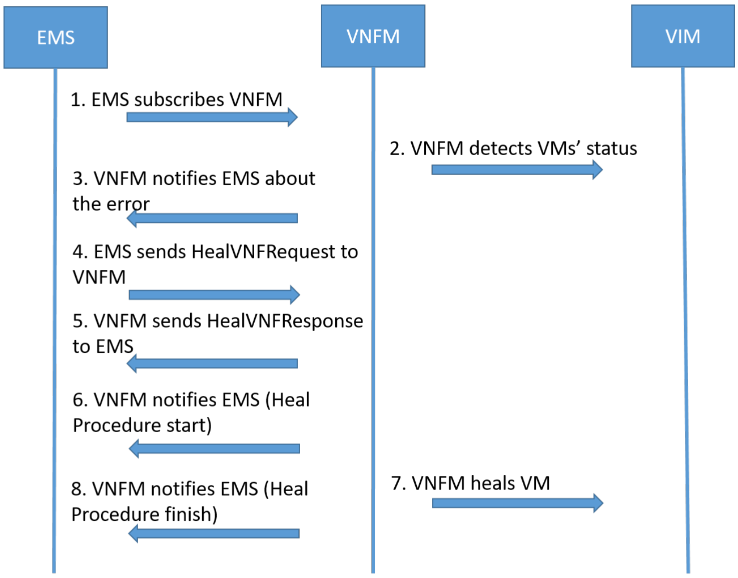

EMS
VNFM
VIM
1. EMS subscribes VNFM
2. VNFM detects VMs' status
3. VNFM notifies EMS about the error
4. EMS sends HealVNFRequest to VNFM
5. VNFM sends HealVNFResponse to EMS
6. VNFM notifies EMS (Heal Procedure start)
7. VNFM heals VM
8. VNFM notifies EMS (Heal Procedure finish)

Figure 10. Virtualized resource fault detection and healing procedure.

service of VNFM. Note that each VNF has its own EMS. Every time the EMS receives an error report from VNFM, it should notify OSS/BSS and let OSS/BSS decide whether to recover the VM failure. If OSS/BSS decides to do recovery, VNFM would use OpenStack's API to heal the failed VM.

Next is the VNF fault management which means the underlying VM operates normally, but the program running on VM fails. The fault detection flow is shown in Figure 11. Unlike the previous case which the monitors are running outside the VMs. There is a monitor running inside each VM and keeps detecting VNF's status and reports to VNFM by using RESTful API if it finds the process terminated. VNFM needs to notify EMS every time it receives the fault report. OSS/BSS also needs to decide whether to heal the VNF failure as in the previous case. The detailed interaction flow between MANO and EMS is shown in Figure 12.

Fault management on two compared systems was also implemented. In the OSS/BSS only system, OSS/BSS activates EMS before PNFs. Each EMS needs to supervise the corresponding PNF's status. But unlike in the virtualized environment, the failure of PNF cannot be healed automatically. As for the MANO only system, the fault management mechanism is almost the same as that in the proposed system. The difference lies in that without OSS/BSS's help, MANO needs to decide whether to heal the failure of VNF/VM or not. However, this is normally out of MANO's capability. Consequently, MANO may have to heal all failed VNFs/VMs. But, this may not be necessary in some cases. For example, if a network slice is going to be terminated, it won't be necessary to heal the failure of any VNF/VM belonging to this network slice.

3.5. Summary of Architectural Designs

In summary, the proposed system consists of both OSS/BSS and MANO and

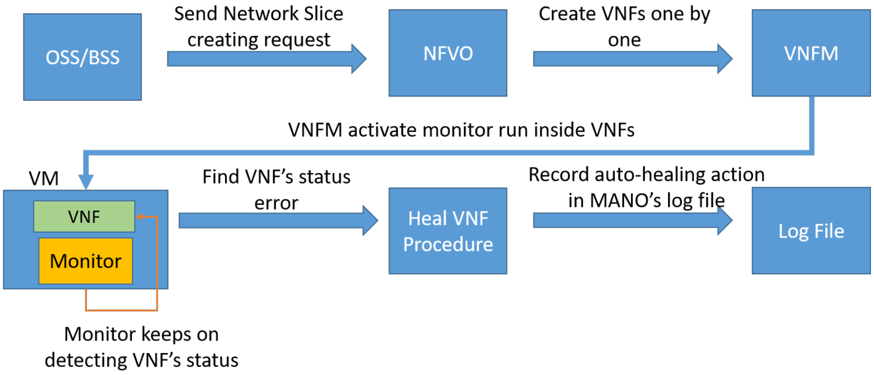

OSS/BSS
Send Network Slice creating request
NFVO
Create VNFs one by one
VNFM
VNFM activate monitor run inside VNFs
VM
VNF
Monitor
Find VNF's status error
Heal VNF Procedure
Record auto-healing action in MANO's log file
Log File
Monitor keeps on detecting VNF's status

Figure 11. VNF fault management.

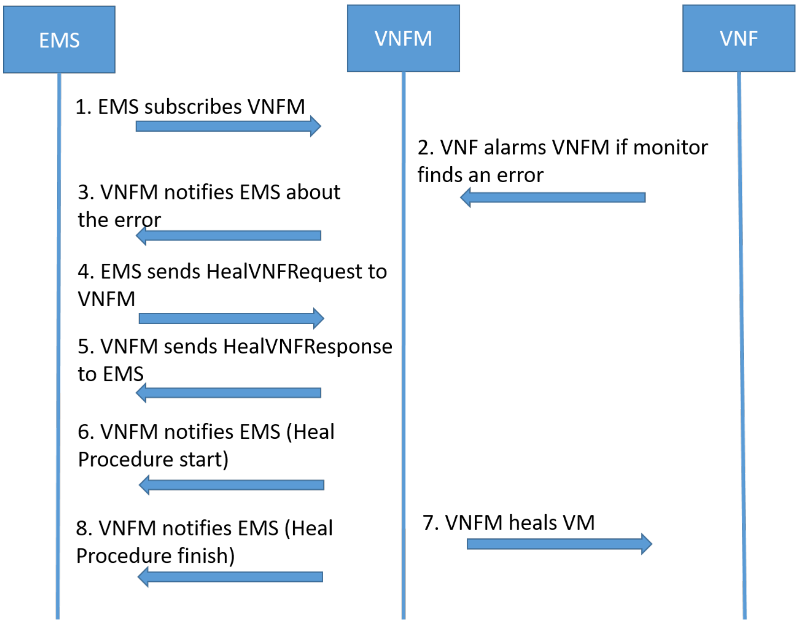

EMS
VNFM
VNF
1. EMS subscribes VNFM
2. VNF alarms VNFM if monitor finds an error
3. VNFM notifies EMS about the error
4. EMS sends HealVNFRequest to VNFM
5. VNFM sends HealVNFResponse to EMS
6. VNFM notifies EMS (Heal Procedure start)
7. VNFM heals VM
8. VNFM notifies EMS (Heal Procedure finish)

Figure 12. VNF fault detection and healing procedure.

these two functional blocks would work together to manage 5G services on a cloud platform. All the activation and deactivation of 5G services would be handled by MANO while OSS/BSS only deals with provisioning and un-provisioning of 5G services. On the contrary, the two compared systems only have either OSS/BSS or MANO but not both. As a result, either MANO or OSS/BSS needs to perform both static and dynamic management by itself. As for life cycle management and fault management, proposed system also performs better than the two compared systems. More detailed comparison of these three systems will be present in the next section.

4. Implementation and Evaluation

In this section, we first show the environment setup of our proposed system and two compared systems. Next, we show the testing results of all three systems. Finally, we do a qualitative comparison between these three systems.

4.1. Environment Setup

In our OSS/BSS with MANO integrated system, three servers were used to build the system, one for OSS/BSS, another for NFVO and VNFM, yet another for VIM. The detailed server configurations are shown in Table 1.

In the OSS/BSS only system, two servers were used to build the system, one for OSS/BSS, another for allocating PNFs. The detailed server configurations are shown in Table 2.

In the MANO only system, two servers were used to build the system, one for NFVO and VNFM, the other for VIM. The detailed server configurations are shown in Table 3.

4.2. Testing Results of Three Systems

In this part, we elaborate the details of the three system implementations and their testing results one by one.

Table 1. Specification of implmentation environment OSS/BSS with MANO integrated system.

Entity	Operating System	Processor	RAM	HDD
Tacker (NFVO, VNFM)	Ubuntu 18.04	Intel E5-2678V3 2.5 GHZ 10 Cores	128 GB	S4510 960 GB SSD
OpenStack (VIM)	Ubuntu 18.04	Intel E5-2678V3 2.5 GHZ 10 Cores	128 GB	S4510 960 GB SSD
free5GMANO (OSS/BSS)	Ubuntu 18.04	Intel i7-8700 3.20 GHZ 6 Cores	64 GB	512 GB

RAM (Random Access Memory), HDD (Hard Disk Drive).

Table 2. Specification of implmentation environment OSS/BSS only system.

Entity	Operating System	Processor	RAM	HDD
Server for PNFs	Ubuntu 18.04	Intel E5-2678V3 2.5 GHZ 10 Cores	128 GB	S4510 960 GB SSD
free5GMANO	Ubuntu 18.04	Inteli7-8700 3.20 GHZ 6 Cores	64 GB	512 GB

RAM (Random Access Memory), HDD (Hard Disk Drive).

Table 3. Specification of implmentation environment MANO only system.

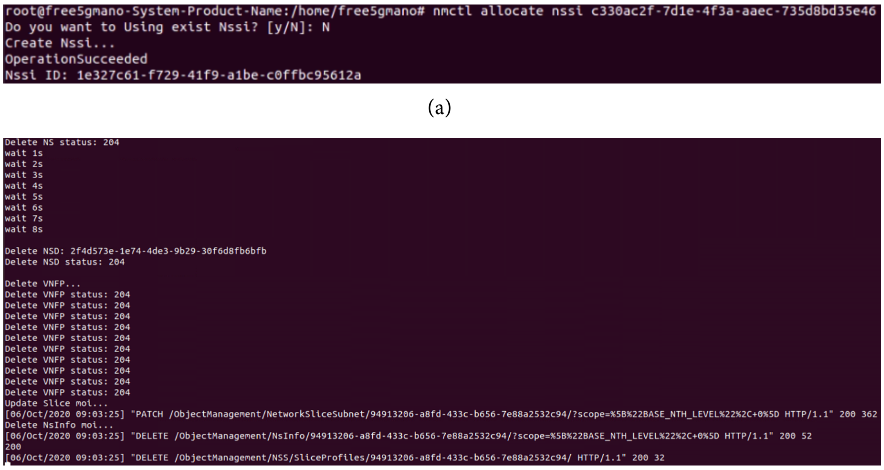
Entity	Operating System	Processor	RAM	HDD
Tacker (NFVO, VNFM)	Ubuntu 18.04	Intel E5-2678V3 2.5 GHZ 10 Cores	128 GB	S4510 960 GB SSD
OpenStack (VIM)	Ubuntu 18.04	Intel E5-2678V3 2.5 GHZ 10 Cores	128 GB	S4510 960 GB SSD

RAM (Random Access Memory), HDD (Hard Disk Drive).

In our proposed system, OSS/BSS asks MANO to allocate network slices. In the provisioning stage, the NSD template and VNFD template were needed to be built first as illustrated in Figure 4(a). Next, these two templates were combined into an NSST (Network Slice Subnet Template) [19]. The NSST then can be used to allocate a new NSSI (Network Slice Subnet Instance) [19]. After an NSSI is allocated, an NSSI ID would be created. This ID will be used to deallocate the NSSI if it is no longer needed.

The snapshots of NSD template, VNFD template and NSST are shown in Figure 13 & Figure 14. In the free5GMANO OSS/BSS, CLI can be used to create, onboard and inspect our NSD, VNFD and NSST Template. The snapshots of using OSS/BSS to allocate/deallocate NSSI are shown in Figure 13 & Figure 14. After OSS/BSS asks MANO to allocate the NSSI, MANO would place the network slice and its VNFs over the OpenStack virtual machines as indicated in Table 1.

A traffic generator is used to test whether the 5GC network slices are working normally or not. The traffic generator can simulate 5G UE and RAN, and establish PDU sessions between RAN and UPF. After the PDU session is established, traffic generator can send ICMP and UDP packets to DN (Data Network) through UPF. The traffic generator and DN are placed on the same machine as OpenStack. Two VMs were launched, one for traffic generator, the other for DN



```
root@free5gmano-System-Product-Name:/home/free5gmano# nmctl allocate nssi c330ac2f-7d1e-4f3a-aaec-735d8bd35e46
Do you want to Using exist Nssi? [y/N]: N
Create Nssi...
OperationSucceeded
Nssi ID: 1e327c61-f729-41f9-a1be-c0ffbc95612a
```

(a)

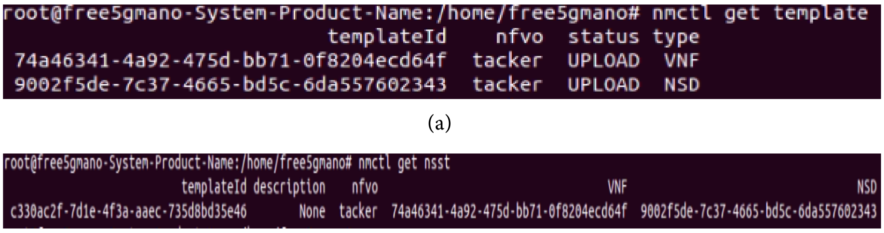
```
Delete NS status: 204
wait 1s
wait 2s
wait 3s
wait 4s
wait 5s
wait 6s
wait 7s
wait 8s

Delete NSD: 2f4d573e-1e74-4de3-9b29-30f6d8fb6bfb
Delete NSD status: 204

Delete VNFP...
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Delete VNFP status: 204
Update Slice moi...
[06/Oct/2020 09:03:25] "PATCH /ObjectManagement/NetworkSliceSubnet/94913206-a8fd-433c-b656-7e88a2532c94/?scope=%5B%22BASE_NTH_LEVEL%22%2C+0%5D HTTP/1.1" 200 362
Delete NsInfo moi...
[06/Oct/2020 09:03:25] "DELETE /ObjectManagement/NsInfo/94913206-a8fd-433c-b656-7e88a2532c94/?scope=%5B%22BASE_NTH_LEVEL%22%2C+0%5D HTTP/1.1" 200 52
200
[06/Oct/2020 09:03:25] "DELETE /ObjectManagement/NSS/SliceProfiles/94913206-a8fd-433c-b656-7e88a2532c94/ HTTP/1.1" 200 32
```

(b)

Figure 13. Snapshots of NSSI Management. (a) Allocate NSSI; (b) Deallocate NSSI.



```
root@free5gmano-System-Product-Name:/home/free5gmano# nmctl get template
                              templateId    nfvo   status type
 74a46341-4a92-475d-bb71-0f8204ecd64f  tacker  UPLOAD  VNF
 9002f5de-7c37-4665-bd5c-6da557602343  tacker  UPLOAD  NSD
```

(a)

```
root@free5gmano-System-Product-Name:/home/free5gmano# nmctl get nsst
                              templateId description    nfvo                                    VNF                                    NSD
 c330ac2f-7d1e-4f3a-aaec-735d8bd35e46        None  tacker  74a46341-4a92-475d-bb71-0f8204ecd64f  9002f5de-7c37-4665-bd5c-6da557602343
```

(b)

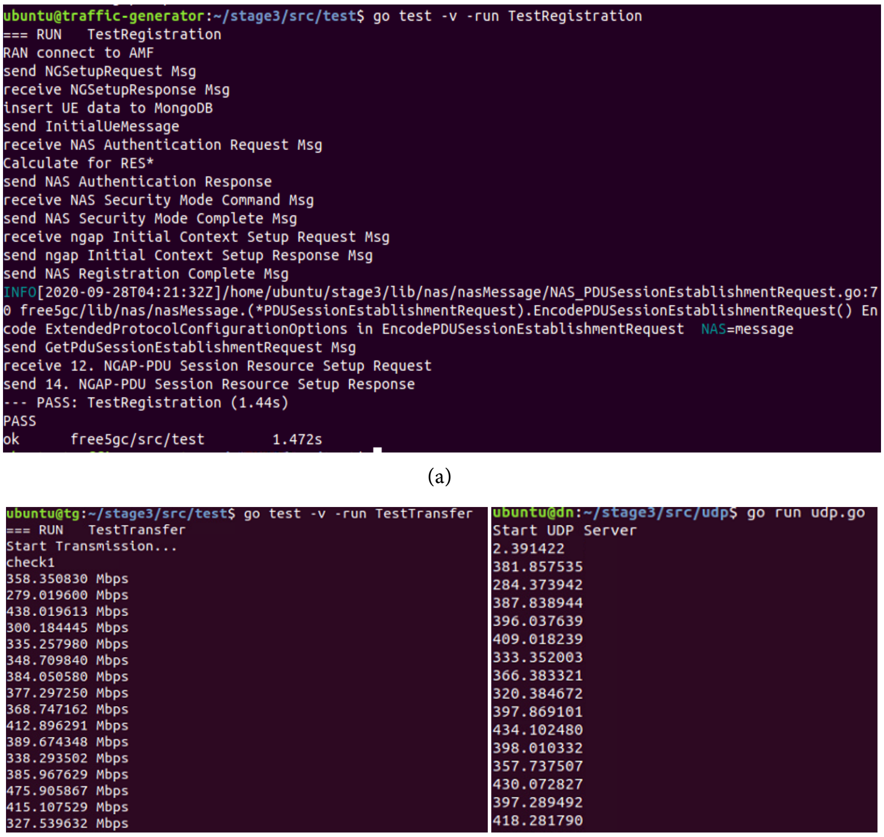
Figure 14. Snapshots of (a) VNFD template and NSD template; (b) NSST Template.

Server, so traffic generator can send packets to UPF through OpenStack's intranet, and UPF forwards the packets to the DN server. The snapshots of traffic generator establishing PDU Session and sending packets and DN Server receiving packets are shown in Figure 15.

In the OSS/BSS only system, OSS/BSS is responsible for activating/deactivating PNFs, the free5GC PNFs are placed on a bare machine which is the server for PNFs entity in Table 2. Because free5GMANO doesn't support activating/deactivating PNFs functionalities originally, free5GMANO CLI was modified for our requirements. After that, OSS/BSS can be used to provision and unprovision the 5G core network service in the form of PNFs. However, unlike in the proposed system where MANO can help to configure VNFs' attributes automatically, PNFs' attributes need to be preconfigured, such as free5GC configuration files, manually.

The traffic generator and the DN server are also used to test our 5GC PNFs. They are installed on the server for PNFs.

In the MANO only system, there is no OSS/BSS's support, so NSD and VNFD need to be provided to MANO and allocate/deallocate network slice manually by using Tacker's CLI or GUI. Again, the traffic generator and the DN server are



```
ubuntu@traffic-generator:~/stage3/src/test$ go test -v -run TestRegistration
=== RUN   TestRegistration
RAN connect to AMF
send NGSetupRequest Msg
receive NGSetupResponse Msg
insert UE data to MongoDB
send InitialUeMessage
receive NAS Authentication Request Msg
Calculate for RES*
send NAS Authentication Response
receive NAS Security Mode Command Msg
send NAS Security Mode Complete Msg
receive ngap Initial Context Setup Request Msg
send ngap Initial Context Setup Response Msg
send NAS Registration Complete Msg
INFO[2020-09-28T04:21:32Z]/home/ubuntu/stage3/lib/nas/nasMessage/NAS_PDUSessionEstablishmentRequest.go:7
0 free5gc/lib/nas/nasMessage.(*PDUSessionEstablishmentRequest).EncodePDUSessionEstablishmentRequest() En
code ExtendedProtocolConfigurationOptions in EncodePDUSessionEstablishmentRequest  NAS=message
send GetPduSessionEstablishmentRequest Msg
receive 12. NGAP-PDU Session Resource Setup Request
send 14. NGAP-PDU Session Resource Setup Response
--- PASS: TestRegistration (1.44s)
PASS
ok      free5gc/src/test        1.472s
```

(a)

```
ubuntu@tg:~/stage3/src/test$ go test -v -run TestTransfer
=== RUN   TestTransfer
Start Transmission...
check1
358.350830 Mbps
279.019600 Mbps
438.019613 Mbps
300.184445 Mbps
335.257980 Mbps
348.709840 Mbps
384.050580 Mbps
377.297250 Mbps
368.747162 Mbps
412.896291 Mbps
389.674348 Mbps
338.293502 Mbps
385.967629 Mbps
475.905867 Mbps
415.107529 Mbps
327.539632 Mbps
```

```
ubuntu@dn:~/stage3/src/udp$ go run udp.go
Start UDP Server
2.391422
381.857535
284.373942
387.838944
396.037639
409.018239
333.352003
366.383321
320.384672
397.869101
434.102480
398.010332
357.737507
430.072827
397.289492
418.281790
```

(b) (c)

Figure 15. Snapshots of traffic generator and data server operations. (a) Traffic generator establishing PDU session; (b) Traffic generator sending packets; (c) DN server receiving packets.

used to test the 5GC network slices in this system. Similar to the proposed system, two VMs were launched on OpenStack, one for the traffic generator and the other for the DN server. If the network slice works, UPF would receive ICMP and UDP packets from the traffic generator and forward to the DN server.

For all three systems, we demonstrate that 5G core network services can be successfully provisioned. Below we focus on qualitative comparison of these three systems instead of their quantitative comparison due to the lack of a fair common base for the latter.

4.3. Qualitative Comparison

In this part, we do a qualitative comparison between our three systems to demonstrate that an integrated OSS/BSS and MANO architecture is better than the other two architectures. We elaborate our arguments from four perspectives: life cycle management, system capacity, OPEX/CAPEX and FCAPS management.

We first start from life cycle management. In this perspective, there are four management operations: provisioning, unprovisioning, instantiation and termination. We elaborate what shortfalls may happen if one of OSS/BSS or MANO is missing.

- Provisioning:

1) OSS/BSS with MANO integrated system: OSS/BSS needs to support evaluation of tenants' requirements, design of NSD/VNFD, then sends NSD/VFND to MANO to create the corresponding network slice.

2) OSS/BSS only system: Without MANO, OSS/BSS cannot use virtualized resources. So, network functions can only be provisioned on physical resources, and there is no support of network slice. Thus, diverse QoS requirements of 5G services cannot be met.

3) MANO only system: Without OSS/BSS, MANO needs to take care of provisioning of network slices in addition to their instantiation such as taking the order of network slices, supporting specifications of network slices based on NSD and VNFD. These are beyond MANO capabilities at the moment.

- Unprovisioing:

1) OSS/BSS with MANO integrated system: OSS/BSS would ask MANO to terminate the network slice when the tenant no longer needs it. MANO would take care of the rest.

2) OSS/BSS only system: Without MANO, OSS/BSS cannot use virtualized resources. So, only physical resources need to be unprovisioned and there is no support of network slice. Thus, diverse QoS requirements of 5G services cannot be met.

3) MANO only system: Without OSS/BSS, MANO needs to take care of unprovisioning of network slices after their termination such as removing customers' subscription and charging. These are beyond MANO capabilities at the moment.

- Instantiation:

1) OSS/BSS with MANO integrated system: After MANO receives NSD,

VNFD and network slice instantiation request from OSS/BSS. It would start to create the network slice, allocate virtualized resources and activate all VNFs.

2) OSS/BSS only system: Without MANO, OSS/BSS only activates PNFs, not VNFs. PNFs cannot allow being instantiated and terminated dynamically.

3) MANO only system: Though MANO can take care of the instantiation of network slices but the provisioning of network slices such as the order of network slices, the creation of network slices based on NSD and VNFD will be left unattended.

- Termination:

1) OSS/BSS with MANO integrated system: After MANO receives network slice terminating request from OSS/BSS, it would terminate all the VNFs and release all virtualized resources.

2) OSS/BSS cannot easily terminate PNFs. This makes the life cycle management of PNFs and their delivered network services more difficult and less automated.

3) Though MANO can take care of the termination of network slices but unprovisioning of network slices after their termination such as removing customers' subscription and charging will be left unattended.

Next is the system capacity part. In this perspective, there are also four system characteristics: scalability, portability, flexibility and efficiency. We elaborate that an integrated OSS/BSS and MANO architecture is more advantageous than the other two architectures.

- Scalability:

1) OSS/BSS with MANO integrated system: Due to the support of MANO, VNFs can be scaled in/out or scaled up/down as defined by NSD and VNFDs created during the provisioning.

2) OSS/BSS only system: Only PNFs are supported by this system and they cannot be easily scaled.

3) MANO only system: Though VNFs can be scaled in/out or scaled up/down theoretically by MANO but without the support of NSD and VNFD creation by OSS/BSS, the scaling path is not well-defined.

- Portability:

1) OSS/BSS with MANO integrated system: With MANO, service providers have the ability to instantiate their network functions anywhere in the world using virtualization. With OSS/BSS, a global strategy can be defined to best allocate network functions in different parts of the 5G network.

2) OSS/BSS only system: Without MANO, network functions can only exist as PNFs. So they won't be easily portable.

3) MANO only system: Without OSS/BSS, a global strategy to best allocate network functions in different parts of the 5G network is not possible.

- Flexibility:

1) OSS/BSS with MANO integrated system: OSS/BSS can ask MANO to initiate or scale VNFs at any time. This would allow operators to flexibly create and update network services because its constituent network functions are virtu-

alized and can be allocated anywhere in the cloud platform.

2) OSS/BSS only system: Without MANO, network functions cannot be virtualized, so PNFs are bundled to proprietary hardware. Network services that are composed of PNFs are hard to provision and upgrade.

3) MANO only system: Without OSS/BSS, MANO needs to decide when to initiate VNFs and set the policy of how to scale VNFs by itself. These are beyond MANO capabilities at the moment.

- Efficiency:

1) OSS/BSS with MANO integrated system: Better utilization in hardware resource because multiple network slices can share the underlying infrastructure resources. MANO can ask NFVI for a proper amount of resources as instructed by OSS/BSS.

2) OSS/BSS only system: Without MANO, network functions can only exist as PNFs. Resource utilization of PNFs could be poor because their underlying hardware resources cannot be easily shared.

3) MANO only system: Without OSS/BSS, MANO needs to evaluate tenants' requirements and optimizes virtualized resource usage. However, these are beyond MANO capabilities at the moment.

Next is the OPEX and CAPEX perspective. In this perspective, we elaborate how service providers' capital investment and operation cost would be increased if one of MANO or OSS/BSS is missing.

- OPEX:

1) OSS/BSS with MANO integrated system: Because MANO can supervise, heal and scale VNFs in real time, it would take less cost for network operators to manage VNFs. OSS/BSS only needs to decide whether to heal the VNF or not, so network operators' operating cost is low.

2) OSS/BSS only system: OSS/BSS needs to supervise each PNF's status by using EMS. Each EMS maps to one PNF, and if EMS finds the PNF's status is abnormal, OSS/BSS would notify network operators to dispatch maintenance workers to repair the PNF manually. This increases network operators' operating cost.

3) MANO only system: Without OSS/BSS, MANO needs to be enhanced with OSS/BSS functionalities such as taking customer's orders, deciding whether to heal/scale the VNF or not. MANO after enhancement requires a long period of operations to become stable. This increases network operators' operating cost.

- CAPEX:

1) OSS/BSS with MANO integrated system: Since VNFs can run on commodity servers, MANO can flexibly allocate virtualized resources when a new network slice is created. Moreover, OSS/BSS can estimate the amount of virtualized resources required by a network slice optimally. Hence the hardware cost could be lowered.

2) OSS/BSS only system: Without MANO, PNFs can only run on proprietary hardware, so there will be higher cost on capital expense for network operators.

3) MANO only system: Without OSS/BSS, it is beyond MANO capabilities to evaluate how many virtualized resource should be distributed to a newly created network slice optimally. So, hardware expense would increase.

Final is the FCAPS perspective. FCAPS is the abbreviation of Fault management, Configuration Management, Account Management, Performance Management and Security Management. There are fundamental operation and management functionalities of telecommunication networks. We elaborate below that an integrated OSS/BSS and MANO architecture also performs better than the other two architectures in terms of FCAPS.

- Fault Management: Because VNFs are run on virtualized infrastructures, OSS/BSS cannot supervise them directly. However, MANO can help to monitor VNFs' status in real time and coordinate with OSS/BSS for the high-level management strategy such as to decide whether to heal a failed VNF or not. In traditional communication networks, OSS/BSS network operators can only heal broken PNFs manually because they are bundled to proprietary hardware. But with MANO, VNFs can be repaired automatically.
- Configuration Management: Because network slices and VNFs are run on virtualized infrastructures, it is beyond the capability of OSS/BSS to manage their configurations at real time. OSS/BSS is only responsible for evaluating tenants' requirements and supporting the creations of NSD and VNFDs with configuration parameters such as deployment flavors and instantiation levels. On the other hand, MANO would allocate virtualized resource, create network slices, activate all VNFs according to deployment flavors and perform scaling actions according to instantiation levels.
- Account Management: Because VNFs are run on virtual infrastructures which are isolated from underlying physical infrastructure. Thus, it is beyond the capability of OSS/BSS to collect VNFs' events such as UE registration with AMF or packet delivering by UPF for accounting purposes. On the other hand, MANO is able to collect VNFs' events and report them to OSS/BSS. Network operators can use these data to do charging and billing.
- Performance Management: VNFs are run on virtualized infrastructures, so OSS/BSS cannot collect performance data from them directly. The MANO framework is thus needed to collect these data from virtualized NS, VNF and infrastructures in order to detect any anomaly in system performance.
- Security Management: Because the states of VNFs change from time to time, and OSS/BSS only deals with static management. Virtual Security Functions (VSF) [22] (which are special types of VNF tailored with security functionalities) are needed to protect other VNFs and set security rules for them. Because VSFs are also VNFs running on virtualized infrastructures, OSS/BSS depends on MANO to initiate and activate them.

5. Conclusions

In our research, the integrated OSS/BSS and MANO system was used to accomplish both configuration management and fault management of the 5G core network slices. A qualitative comparison between the integrated system and both the OSS/BSS and the MANO silo systems is also performed to analyze the shortfalls of the latter when used to deploy network slices. Through this analysis, this research confirms the necessity of applying both OSS/BSS and MANO for the coordinated management of 5G core slices as adopted by 3GPP. Neither OSS/BSS nor MANO can do both static management and dynamic management by itself.

With the integrated OSS/BSS and MANO system, OSS/BSS can be used to provision or un-provision 5G services. Then with NFV MANO technologies, NFs can be virtualized and placed on cloud platforms, so CAPEX can be reduced. Moreover, the status of a VNF can be detected dynamically to allow OSS/BSS to perform traditional FCAPS functions and reduce OPEX.

In 3GPP standards for fault management, after EMS receives VNF/VM fault notification, it should inform OSS/BSS and let OSS/BSS decide whether to heal the failed VNF or not. However, our proposed system has not yet implemented this flow but let EMS notify MANO to heal every failed VNF or VM. In our future effort, we plan to add more OSS/BSS functions so that the decision of whether to heal failed VNF/VM or not can be based on the global view of OSS/BSS.

We only focus on configuration management (C) and fault management (F) of FCAPS in our research. Accounting management, performance management and security management are still left unaddressed. Hence another potential is to address these areas.

Finally, another future effort could be adding free5GMANO Web UI into our proposed system. We use CLI to operate OSS/BSS in current stage. However, it is not user-friendly. With a Web UI, our proposed system would be easier to use.

Acknowledgements

This work was financially supported by the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] 3GPP (3rd Generation Partnership Project) (2018) Technical Specification Group Services and System Aspects; System Architecture for the 5G system, V15.3.0, 3GPP TS23.501.

[2] Tipantuña, C. and Yanchapaxi, P. (2017) Network Functions Virtualization: An Overview and Open-Source Projects. *Proceedings of the IEEE Second Ecuador Technical Chapters Meeting* (ETCM), Salinas, 16-20 October 2017, 1-6. https://doi.org/10.1109/ETCM.2017.8247541

[3] free5GC.org (2019) https://www.free5gc.org/

[4] free5GMANO.org (2020) https://github.com/free5gmano/free5gmano

[5] OpenStack Tacker Project (2010) https://docs.openstack.org/tacker/latest/

[6] OpenStack (2010) https://www.openstack.org/

[7] Devadetta, V.S. (2013) Impacts of NFV on Classical OSS. https://www.researchgate.net/publication/303489759_Impacts_of_NFV_on_Classical_OSS

[8] MATILDA Project (2019) OSS for 5G-Ready Applications. https://private.matilda-5g.eu/Publications/DownloadPublication/22

[9] Messaoudi, F., Bertin, P. and Ksentini, A. (2021) Towards the Quest for 5G Network Slicing. *Proceedings of the* 2020 *IEEE* 17*th Annual Consumer Communications & Networking Conference* (*CCNC*), Las Vegas, 10-13 January 2020, 1-7. https://doi.org/10.1109/CCNC46108.2020.9045327

[10] ETSI (2014) Network Functions Virtualisation (NFV); Management and Orchestration. V1.1.1, ETSI GS NFV-IFA 001.

[11] ETSI (2019) Network Functions Virtualisation (NFV) Release 3; Management and Orchestration. VNF Descriptor and Packaging Specification, V3.3.1, ETSI GS NFV-IFA 011.

[12] Popovski, P., Trillingsgaard, K.F., Simeone, O. and Durisi, G. (2018) 5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View. *IEEE Access*, **6**, 55765-55779. https://doi.org/10.1109/ACCESS.2018.2872781

[13] ETSI (2019) Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models. Network Service Descriptor File Structure Specification, V2.6.1, ETSI GS NFV-SOL 007.

[14] 3GPP (3rd Generation Partnership Project) (2019) Management and Orchestration; Architecture Framework. Release 15, 3GPP TS 28.533.

[15] Chahbar, M., Diaz, G., Dandoush, A., Cérin, C. and Ghoumid, K. (2019) NESSMA: Network Slice Subnet Management Framework. *Proceedings of the* 10th *International Conference on Networks of the Future* (*NoF*), Rome, 1-3 October 2019, 54-57. https://doi.org/10.1109/NoF47743.2019.9015010

[16] ETSI (2016) Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-nfvo Reference Point—Interface and Information Model Specification, V2.1.1, ETSI GS NFV-IFA 013.

[17] EURECOM OpenAirInterface (2014). https://openairinterface.org/

[18] Liu, G., Huang, Y., Chen, Z., Liu, L., Wang, Q. and Li, N. (2020) 5G Deployment: Standalone vs. Non-Standalone from the Operator Perspective. *IEEE Communications Magazine*, **58**, 83-89. https://doi.org/10.1109/MCOM.001.2000230

[19] 3GPP (3rd Generation Partnership Project) (2018) Management and Orchestration; Provisioning. Release 15, 3GPP TS 28.531.

[20] 3GPP (3rd Generation Partnership Project) (2018) Management and Orchestration; Generic Management Services. Release 15, 3GPP TS 28.532.

[21] 3GPP (3rd Generation Partnership Project) (2018) Telecommunication Management; Fault Management (FM) for Mobile Net-Works That Include Virtualized

Network Functions; Procedures. Release 15, 3GPP TS 28.516.

[22] ETSI (2017) Network Functions Virtualisation (NFV) Release 3; Security; Security Management and Monitoring Specification. v3.1.1, ETSI GS NFV-SEC 013.