

Adaptive Network Slicing and Slice Selection in 5G for Efficient Resource Utilization

Mohit Kumar Singh

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



Department of Computer Science and Engineering

June 2020

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.



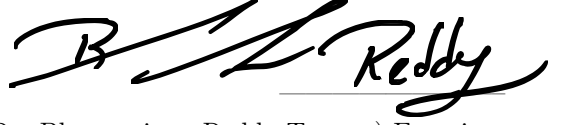
(Signature)

Mohit Kumar Singh
(Name)

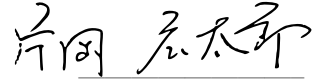
CS17MTECH11015
(Roll No.)

Approval Sheet

This Thesis entitled Adaptive Network Slicing and Slice Selection in 5G for Efficient Resource Utilization by Mohit Kumar Singh is approved for the degree of Master of Technology from IIT Hyderabad



(Dr. Bheemarjuna Reddy Tamma) Examiner
Dept. of Computer Science and Eng.
IITH



(Dr. Kotaro Kataoka) Examiner
Dept. of Computer Science and Eng.
IITH



(Dr. Antony Franklin) Adviser
Dept. of Computer Science and Eng.
IITH



(Dr. M.V. Panduranga Rao) Chairman
Dept. of Computer Science and Eng.
IITH

Acknowledgements

I would like to thank everyone whoever supported me for pursuing Master's in this research field. I specially wants to thank my guide Dr. Antony Franklin for his continuous guidance throughout this work and always motivating by his dedication, passion and enthusiasm towards the work. Also, I am very grateful to him for selecting me for M.Tech course under him and providing with all the facilities to work dedicatedly and smoothly. I am thankful to my prof. Dr. Bheemarjuna Reddy Tamma for his motivating thoughts and words which really helped me for being motivated and truthful to my work. Also, I am thankful to my lab mates to support me and having the healthy discussions on various research activities being pursued in the lab. Specially, I am thankful to Mrs. Shwetha Vittal for her continuous joint efforts in this research work. I am thankful to my lab colleagues Mehul, Venkatrammi, Suhel and Nabhasmita for standing with me in my tough times and always pushing me to work harder for achieving the best. I would like to give my warm respect to other lab mates including Yogesh, Jyoti, Supriya, Prashansa and Anshika with whom I have shared lot of sweet memories and spent some fabulous time, I wish them best of luck for their future. Last but not the least, I am very thankful to my family for being the support pillar in my career and always motivating me to achieve higher and higher milestones in life. Once again, thank you all for being so supportive and helping me in completing this research work successfully.

Dedication

To,
The Institute,
My Guide, and
My Family.

Abstract

The new upcoming radio access technology 5G is being designed to help the service provider in meeting the growing demands of the services. Software Defined Networking and Network Function Virtualization are the key enablers of the 5G, helping the monolithic hardware based existing network to evolve and develop as software based virtualized network. Study has been performed on realising the 5G architecture, proposed by 3GPP, as Reference Point Architecture and Service Based Architecture. In reference point based 5G architecture NFs perform the socket based communication. We have used nghttp2 for realising the SBA of 5G and perform the comparison with the reference point based 5G architecture in terms of various performance metrics. Network slicing is the key technique provided by 5G to support the multiple services by running different logical networks in isolation over the same physical infrastructure. Thus, the service providers can serve various services with available infrastructure by deploying network slices for each of the provided services.

In this thesis, we have proposed the novel framework for monitoring the network KPIs at the slice level. The performed monitoring at the slice level helps the service provider to perform the life cycle management of a slice and making the critical decisions like admissibility of a new slice requests, admissibility of the new user request, and slice selection for the incoming request. Study has been performed to dynamically switch in between the states of a slice for having efficient resource utilisation while keeping the response time of the request within the SLA. The novel call flows have been proposed for enabling the communication between the NSMF, Management & Orchestration unit, and slice specific NFs for managing a slice. We have studied all possible slice selection schemes for performing the task of slice selection at Network Slice Selection network function of 5G core network. The study has find the best slice selection scheme in terms of making the usage of resources to handle the current load on the slice.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	vi
1 Introduction	1
1.1 Existing Radio Access Technology	2
1.2 Evolvment of 5G	3
1.3 Key Enablers of 5G	4
1.3.1 Software Defined Networking	4
1.3.2 Network Function Virtualisation	5
1.4 Network Slicing	6
1.5 Thesis Organisation	8
2 Realization of 5G Core Network	9
2.1 Reference Point Based Architecture of 5G	10
2.2 Service Based Architecture of 5G	12
2.3 Experimental Setup	13
2.4 Results and Analysis	14
2.5 Summary	15
3 Monitoring and Orchestration of Multi Site deployed NS	16
3.1 Motivation and Related Work	16
3.1.1 Motivation	17
3.2 Network Slice Monitoring Framework	17
3.2.1 Key Performance Indicators of Network Slice	18
3.2.2 Developed Framework	18
3.2.3 Monitoring Technologies	20
3.3 MANO of a Multi Site deployed NS	21
3.3.1 Developed Multi Site Deployment Framework	21
3.3.2 Life Cycle Management of the NS	23
3.4 Results and Analysis	26
3.5 Summary	27

4	Adaptive Network Slicing in 5G for Efficient Resource Utilisation	28
4.1	Motivation	28
4.2	Background: Adaptive Network Slicing	29
4.2.1	Network Slice as a Service (NSaaS)	29
4.2.2	Role of NSSF	29
4.2.3	Network Slice Selection and Management	30
4.3	Static Network Slicing	30
4.3.1	Always-ON Network Slice	31
4.4	Adaptive Network Slicing	32
4.4.1	Controlling Slice Activation and Deactivation	32
4.4.2	Network Slice with Provisioning	33
4.4.3	Network Slice without Provisioning	34
4.5	Results and Analysis	34
4.6	Summary	35
5	SERENS: Self Regulating Network Slicing in 5G for Efficient Resource Utilization	36
5.1	Related Work and Motivation	37
5.2	Self Regulating Network Slicing (SERENS) Framework	37
5.2.1	Slice Monitoring at NSMF	38
5.2.2	Slice Analytics at NSSF	39
5.2.3	Slice Selection at NSSF	39
5.3	Implementation of SERENS Framework in 5GC	41
5.4	Performance Evaluation	43
5.4.1	Synthetic Traffic Data Generation	43
5.4.2	5G System KPIs using Slice Monitoring in SERENS Framework	44
5.4.3	Performance of Slice Selection Algorithm in SERENS Framework	44
5.5	Summary	47
6	Conclusion and Future Work	48
	References	49

Chapter 1

Introduction

5G has been evolved from the existing radio access technology which is 4G/LTE while converting the monolithic architecture running on the dedicated proprietary hardware to the software modules capable of running on any commodity hardware. These software modules are basically the network entities a.k.a NFs (Network Functions) implemented as micro services performing a specific task in the network. The NFs communicates with each other to make use of the network services provided by the other network entities in order to perform some activity. The SP (Service Provider) deploys a complete network constituting of various network entities in order to provide the essential services to the end users a.k.a UE (User Entity). 5G is designed to provide three generic services [1] which are eMBB (enhanced Mobile Broadband), URLLC (Ultra Reliable Low Latency Communication), and MMTC (Massive Machine Type Communication). Each of these services is defined by two main aspects which are QoS (Quality of Service) and SLA (Service Level Agreement). It's the responsibility of the SP to serve the users with the required QoS and specified SLA. The service providers has the complete network architecture being set up for providing the network services as shown in Fig. 1.1. The UE (User Entity) is the end user of the network service, RAN (Radio Access Network) allows the user to connect to the cloud by utilising the radio transceivers. RAN consists of the Base Stations connected with the fibre backhaul to the core network and with the fibre fronthaul to the UE. The CN (Core Network) performs the user related functionalities like authentication, access authorisation, user registration, session creation, handling the data packets to/from Data Network (DN). DN constitutes of all the data servers, cloud services/applications provided by the third party to the end-users.

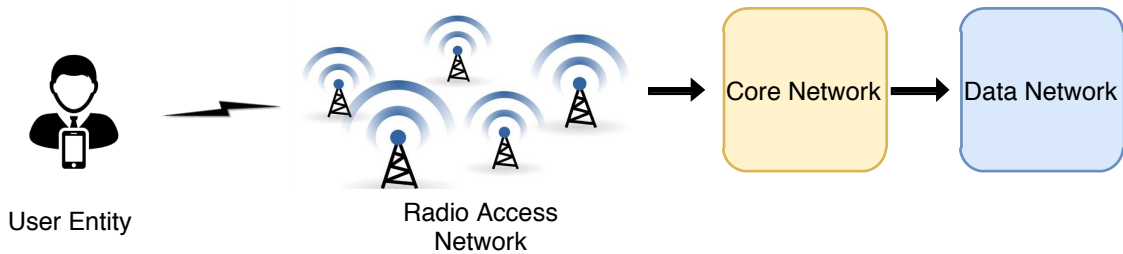


Figure 1.1: Service Provider Network Architecture.

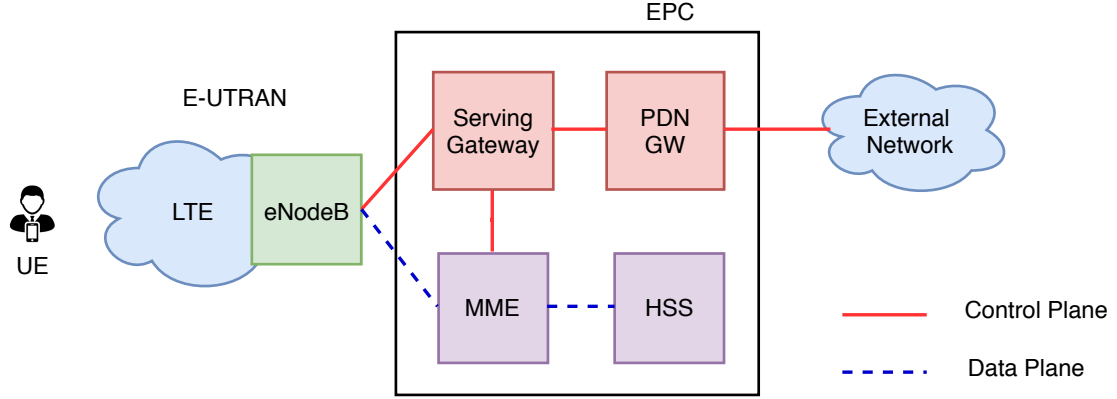


Figure 1.2: EPS Architecture with EUTRAN and EPC [2].

1.1 Existing Radio Access Technology

4G/LTE (Long Term Evolution) is the existing radio access technology providing RAN (Radio Access Network) and CN (Core Network) for building a complete network architecture. Fig. 1.2 shows the architecture of the LTE EPS(Evolved Packet System) consisting of E-UTRAN (Evolved-UMTS Terrestrial Radio Access Network) and EPC(Evolved Packet Core) [2]. E-UTRAN forms the radio access network to the architecture and has just one main entity which is the evolved base station a.k.a eNodeB or eNB. This RAN helps in performing the radio communication between users and core network (EPC) through analog and digital signal processing. The eNBs in the RAN uses S1 interface for establishing the communication with EPC, uses X1 interface for communicating with the other eNBs inside the RAN architecture. There are undergoing adaptations in the RAN architecture while moving (splitting) some of the functionalities over the cloud (C-RAN) operating in the centralised manner called as BBU-pool (Base Band Unit), while the other functionalities remains co-located with the eNodeB running in the distributed manner also called as RRHs (Radio Resource Head). In the thesis, our complete focus is on the Core Network part of the 4G/5G architecture and various technologies indulged in the 5G core network.

The EPC forms the core network of the LTE EPS architecture. EPC has two main components namely CP (Control Plane) and DP (Data Plane). The CP mainly performs the user centric activities like authentication, registration, etc. and DP performs the communication with the outside network by routing the IP data packets to/from external network. The S-GW (Serving Gateway) and PDN-GW (Packet Data Network) forms the data plane while the MME (Mobility and Management Entity) and HSS (Home Subscriber Server) together constitutes the control plane of the EPC. HSS is a database storing the network subscriber's information for performing the network related activities like session setup, user authorisation and access authentication. MME, part of CP, handles the signalling for performing security and mobility related activities. It keeps the track of the UE's location for performing the handover between the eNBs. S-GW is the connecting of EPC with RAN and helps in serving the user with the incoming/outgoing IP packets. Thus, helps in transporting the data packets between the user and the outside network. The PDN-GW a.k.a P-GW (Packet Gateway) is the connection point of EPC with the outside network. It helps in routing the IP data packet to/from the external data network. It performs some of the main activities like allocating IP

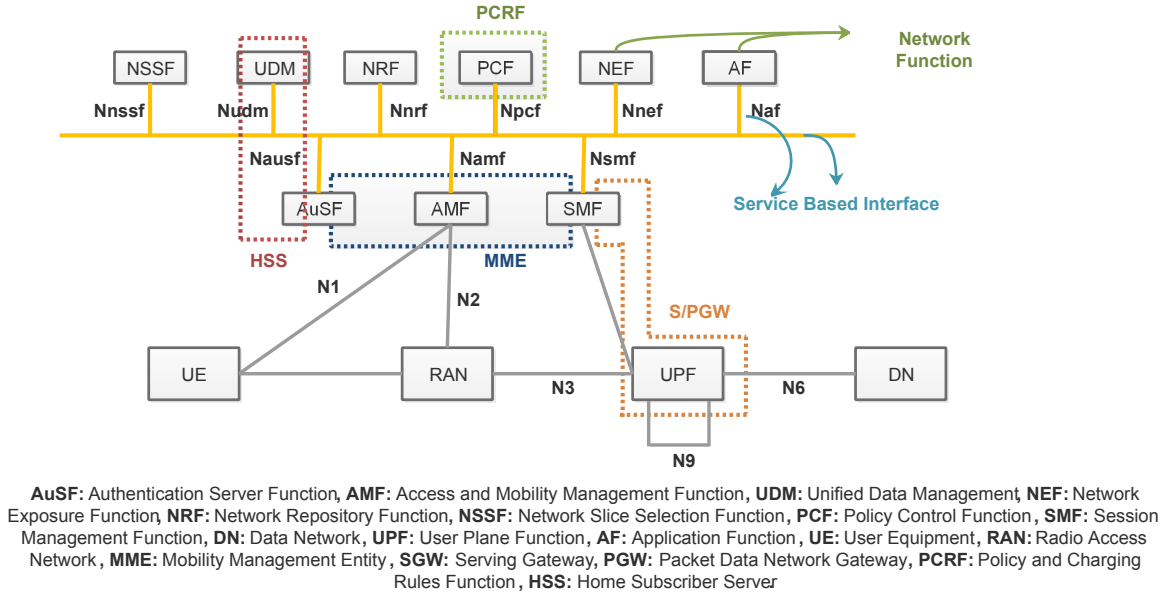


Figure 1.3: 5G evolution from the existing LTE architecture [3].

address and applying the policy and charging rules.

1.2 Evolvement of 5G

5G is being evolved from 4G/LTE, the existing radio access technology by converting the monolithic architecture into micro services based architecture capable of running on any commodity hardware, removing the hardware dependencies from the functionality of the network modules [3]. The existing LTE core network consists of network functions like MME, HSS, SGW-C, PGW-C and PCRF, now these network functions are being divided to several new network functions for performing the same set of functionalities as shown in Fig. 1.3. UDM along with AuSF NFs of 5G core network combinedly performs the functionality of HSS i.e helping in storing the UE information and performing the UE authentication. PCRF has been changed to PCF NF in 5G architecture with almost similar set of functionalities. AMF, SMF, and AuSF in together performs the functionality of MME i.e breaking down the functionality of MME into smaller sub functionalities a.k.a micro services. Similarly, SMF and UPF takes the functionalities of S/PGW where SMF is the part of control plane of core network while the UPF forms the data plane of the core network of 5G architecture. The 5G architecture has the some new NFs with the complete different functionalities from the existing 4G/LTE NFs including NSSF (Network Slice Selection Function), NRF (Network Repository Function), NEF (Network Exposure Function). These new NFs helps in supporting the new technologies and concepts being introduced in the 5G architecture.

The dedicated interfaces present in the 4G/LTE architecture for establishing the communication between the network entities has been replaced by a bus based interface in 5G architecture to which each of the NF is connected by an interface. NRF (Network Repository Function) helps a NF to discover the other NF service with the mechanism of service registration and discovery. Hence, the new upcoming 5G architecture is the more advanced in terms of network scalability, performance, manageability and performance by making the use of micro service enabled REST based core network

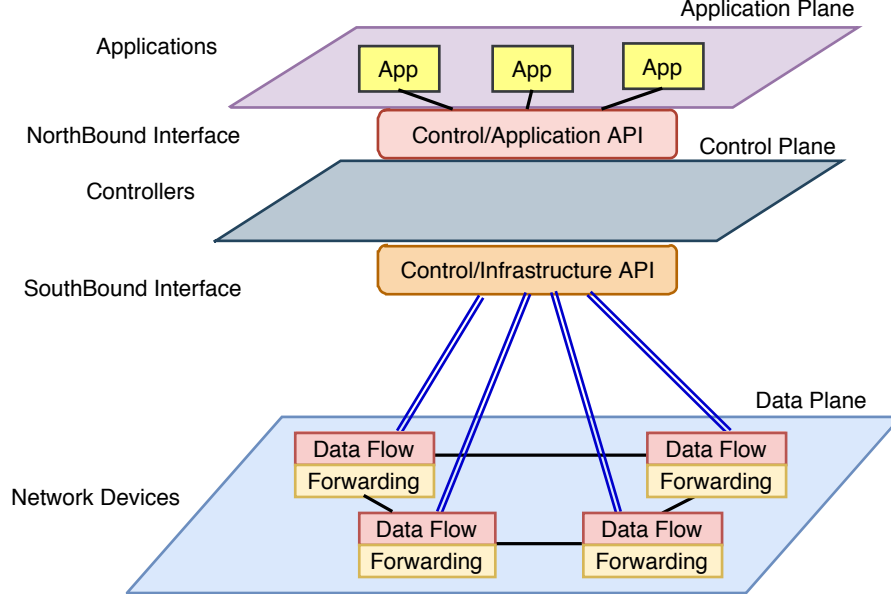


Figure 1.4: Software Defined Networking-High Level Architecture [5].

architecture.

1.3 Key Enablers of 5G

5G is being evolved from the existing radio access technology, namely 4G/LTE, replacing the monolithic network with the micro service based software modules capable of running on the commodity hardware making the network programmable, cost-effective, flexible and manageable. Thus, SDN (Software Defined Networking) and NFV (Network Function Virtualization) [4] are the two key enablers of the new upcoming radio access technology, 5G. The underlying working of both these technologies with their architectural diagram is as follows:

1.3.1 Software Defined Networking

SDN (Software Defined Networking) has provided a new architecture supporting the deployed applications and making the architecture more programmable and easily manageable. The SDN architecture has separated the application from the network forwarding functions and isolated the infrastructure from the deployed application. The SDN architecture, as shown in the Fig. 1.4, consists of 3 layers which are Application layer, Control Layer and Infrastructure Layer in top-down order.

Application Layer: This layer consists of end users applications running on the network. These applications affect the behaviour of the underlying infrastructure by making the use of the SDN controllers present in the control layer. Controllers help the applications data packets to route through the best path between the end points, perform load balancing between the forwarding switches present at infrastructure layer. This layer is the open area of development for the applications making the best use of the underlying infrastructure to make applications robust, fault-tolerant and manageable.

Control Layer: This layer is been called as the brain of the SDN architecture. It consists of

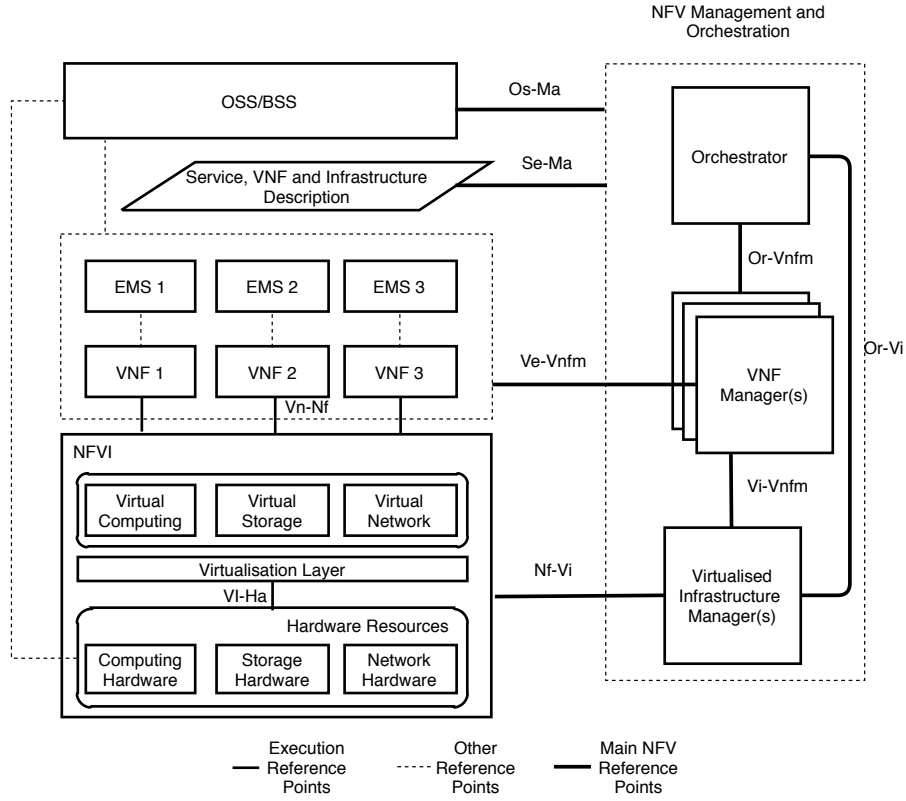


Figure 1.5: NFV reference architectural framework by ETSI [6].

the SDN controllers connecting with the southbound interface to the forwarding devices of the infrastructure layer and thus controls the complete network by implementing the routing tables, flow rules, load-balancing, etc. Performs the communication with the applications through the North-bound Interface.

Infrastructure Layer (Network Devices): This layer consists of the forwarding devices that performs the communication with the SDN controller through Southbound Interface and performs the routing of the data packets with the flow tables and performs the actions on the received packets on the basis of the flow entries of the flow table.

1.3.2 Network Function Virtualisation

NFV (Network Function Virtualisation) has isolated the hardware binded network entities, running on the firmware, to run on any commodity hardware in the resource virtualized environment. NFV is the term used with SDN, enabling the software defined NFs to run on commodity hardware and delivers the same network functionalities independent of the underlying physical resources. NFV has make the network more manageable, accessible, fault-tolerant and efficient by allowing the SP to provide the required resources virtually and dynamically. Since, each of the VNF has been virtualised with the actual available physical resources, the SP can make higher revenue by supporting more services with the limited available physical resources.

The ETSI (European Telecommunication Standards Institute) has provided the infrastructural

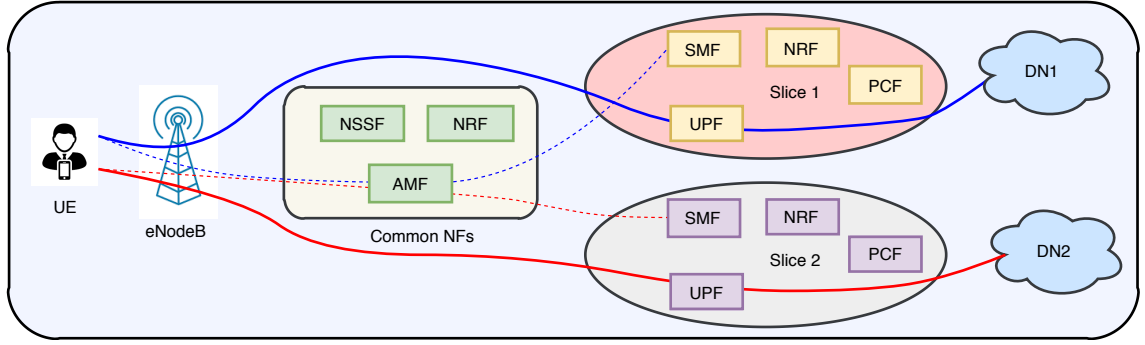


Figure 1.6: Performed Network slicing in the build network architecture.

framework for NFV as shown in the Fig. 1.5. NFV MANO (Management and Orchestration), NFVI (NFV Infrastructure), and VNFs (Virtual Network Function) are the key components of the ETSI NFV architecture. The architecture in together handles the management, orchestration of resources and manages the LCM (Life Cycle Management) of the deployed VNFs.

The NFV architecture has some main building blocks which provides the complete overall support to the deployed NF over the infrastructure. Some of them are listed as follows:

NFV Management and Orchestration (MANO): The MANO entity performs the complete management and orchestration of the deployed NF on the architecture. It internally has three modules namely Orchestrator, VNF Manager, and VIM (Virtualized Infrastructure Manager). The Orchestrator is in charge of orchestration and management of the NFV infrastructure and the resources available for realising the services. The VNF Manager handles the LCM of the VNFs which includes instantiation, update, query, scale, etc. VIM provides the actual resources for the computation of the deployed VNFs in the virtualized manner. VIM controls the interaction of the VNFs with the underlying computing, storage like available resources. VIM helps in performing the resource management and operations like fault detection, monitoring and optimisation.

Virtualized Network Function (VNF): The VNF is the virtualization of the Network Functions running on the legacy non-virtualized network and are bind to the firmware for their operation. VNF are capable of running on any commodity hardware and delivers the same set of functionality as of running on their dedicated firmware.

NFV Infrastructure: NFVI (NFV Infrastructure) constitutes of all the hardware and software resources required for the functioning of the deployed VNFs on the infrastructure. It provides the abstraction layer to the VNFs from resources and ensure the hardware independent management of the VNFs. It performs partitioning of the resources and enables the usage of the hardware and software resources in the virtualised manner.

1.4 Network Slicing

The existing radio access technology (4G/LTE) has the network entities (like S-GW, P-GW, HSS, etc.) co-exists with the hardware devices making the legacy network monolithic and rigid. Such a

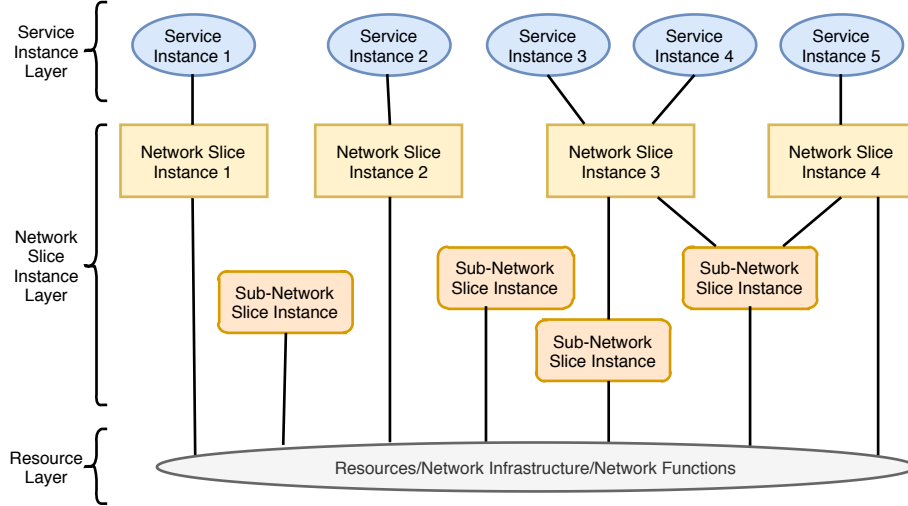


Figure 1.7: A network slice architectural framework [8].

deployed network creates difficulty to the SP in meeting the diverse network service requirements. The new upcoming radio access technology (5G) has made it possible for the SP to meet the ever growing network service requirements with the inherent new technologies like SDN, NFV, Network Slicing, etc. by making the network more robust, scalable, manageable, fault-tolerable and accessible. Network slicing [7], one of the main functionality of 5G, helps the SP in creating logical networks over the same physical infrastructure. These logical networks serves a specific type of service to the user and helping the SP to overcome the difficulty of serving the different types of services in a network. These logical networks are known as a NS (Network Slice) and run in complete isolation with other logical networks using the shared physical infrastructure.

The Fig. 1.6 shows a sample 5G CN (Core Network) consisting of two slices (Slice1 and Slice2) having the dedicated NFs (like SMF, NRF, PCF, UPF) specific to the slice and a slice having the common NFs (like NSSF, AMF, and NRF) shared between both the slices. The slice specific NFs performs the task specific to the slice and performs the communication with the NFs present in the same NS. The AMF NF is the entry point for the incoming request from RAN to the CN. AMF discovers the NSSF with the help of NRF and gets the target slice for the incoming USR (User Service Request) and communicates with the SMF of the selected slice. NRF NF present in each of the slice helps the slice specific NFs to discover each others service and communicate. Each of the slice has the specific UPF for performing the data plane activities with the outside DN (Data Network). Thus, we can imagine a network slice as a network function chain deployed for supporting a network service in the network. These NFs are the virtualised Network Functions (VNFs) capable of being deployed over any commodity hardware and providing the same functionality as that of the legacy firmware based network entities in non-virtualised networks. Thus, a NS incorporates both the key 5G technologies namely, SDN and NFV.

The architectural view of a network consisting of network slices have three main layers which are Infrastructure Layer, Network Slice Instance Layer, and Service Instance Layer as shown in Fig. 1.7. All three layers in together provided the network services to the end user. The functionality of each of the layer is described as follows:

1. **Infrastructure Layer:** The infrastructure layer consists of the physical resources, network infrastructure of the switches/controllers and the deployed NFs. This layer provides all the underlying resources and functionalities required by the NFs to operate.
2. **Network Slice Instance Layer:** This layer of the network consists of all the NSIs (Network Slice Instances) currently being deployed in the network to meet the service requirement. A network slice supporting a specific type of service (say eMBB) can have multiple instances running to meet the traffic load on the slice, each deployed instance is called as Network Slice Instance. A NSI (Network Slice Instance) can have a single or multiple NSSIs (Network Sub Slice Instances) running in beneath. A NSSI can be a part of multiple NSIs to make the better usage of the underline physical infrastructure to serve various services to the end users.
3. **Slice Instance Layer:** This layer is the top most layer of the typical network slice architectural framework, consisting of all the NSs offered by the SP to the end users. These NSs consists of one or multiple NSIs to perform the network specific activities and provide the end users with a specific service.

1.5 Thesis Organisation

The rest of the thesis is organised as follows:

Chapter 2: Realising SBA of 5G In this chapter we have described our study on complete 5G architecture and means of realising the SBA of 5G. Studied and compared reference point based 5G architecture with REST based SBA of 5G.

Chapter 3: 5G Network Slice Management and Monitoring In this chapter, we focused on performing the management and orchestration of the NSIs and performing the slice level monitoring through our proposed architectural framework.

Chapter 4: Adaptive Network Slicing in 5G Core In this chapter, we have highlighted the importance of performing adaptive network slicing and proposed the BDA based algorithms for efficiently managing the network slice to make best of the available resources.

Chapter 5: Self Regulating Network Slicing with Analytics and Selection in 5G Core This chapter focuses on the slice selection mechanisms for an incoming user request. We have studied all the possible slice selection schemes and compared with the proposed slice selection scheme.

Chapter 2

Realization of 5G Core Network

5G is the new upcoming the radio access technology evolving from the existing radio access technology namely (4G/LTE) while making the network more programmable, manageable and scalable. As discussed in the chapter 1, SDN and NFV are two key technologies inherent by 5G replacing the network entities running on the propriety and dedicated hardware with the software modules capable of running on the commodity hardware while not having any hardware dependency for their performance. Network Slicing helps the network to offer different services to the end-users by achieving the service level isolation with same hardware infrastructure. Each of the network slice offers a network service with specified QoS/SLA. The 5G architecture by 3GPP [10], consists of Radio Access Network (RAN) and Core Network (CN). RAN performs the communication of the UE (User Entity) with the eNBs and then to the core network. NFV extends to the RAN of 5G also, allowing the components of the network to get virtualised and provide better Quality of Service (QoS) as the network requirement grows in demand. In 5G, the functionalities performed at each of the eNB, distributed across an area, are being moved to a central unit called as BBU pool while some of the radio functionalities remain co-located to the eNB a.k.a RRH (Radio Resource Head). Our focus in the thesis is not to study the RAN part of the 5G architecture rather we keep our focus on the Core Network of the 5G architecture.

The Core Network (CN) of the 5G architecture consists of NFs, each performing a set of function-

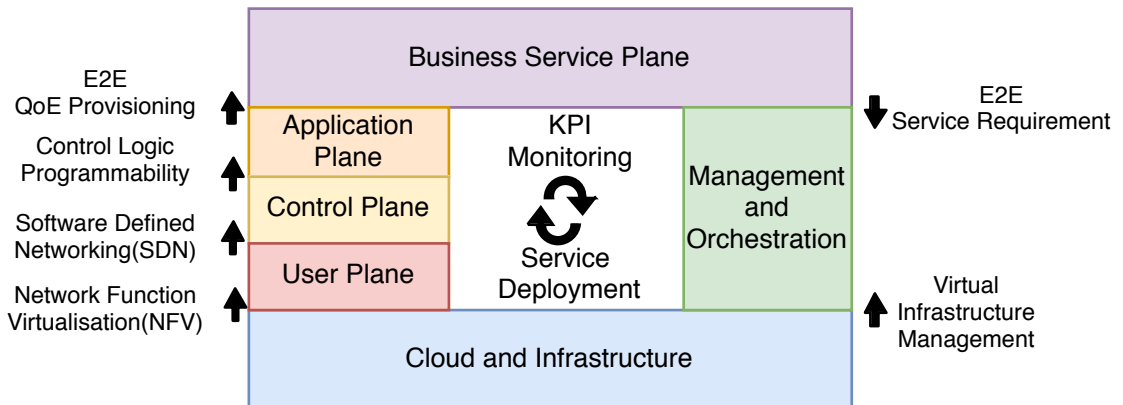


Figure 2.1: An overview of 5G Architecture [9].

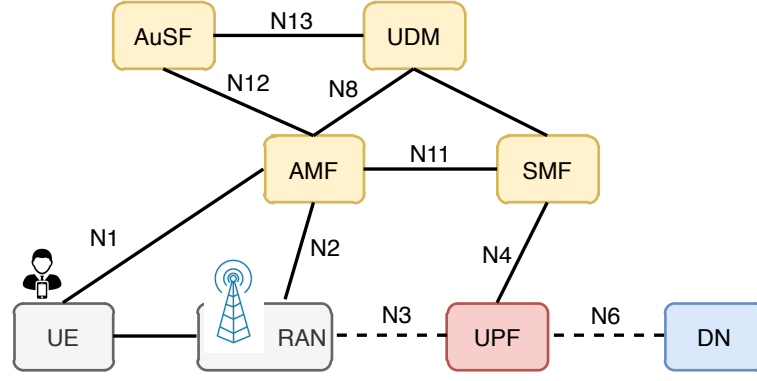


Figure 2.2: Reference Point based 5G Architecture.

activities while utilising the services offered by the other NFs. CN performs the user centric activities like UE registration, authentication, de-registration, charging policies, data packets forwarding to the outside Data network (DN). The Fig 2.1 shows the overview of the 5G architecture. The cloud and infrastructure provider forms the bottom layer of the architecture providing the underlying physical resources to the deployed network entities. The user plane handles the IP packet forwarding to the outside network consisting of only single NF namely UPF. The other NFs constitutes the control plane performing the user centric activities like that of user authentication, access authorisation, user registration, session creation and user de-registration. The application plane is the part of OSS/BSS monitoring the E2E QoE and the deployment of the network services offered to the end users using the network. The Management and the orchestration unit performs the overall resource allocation and orchestrate the NFs deployed in the core network. Also, monitors the network KPIs of the NFs for their management of the provided resources and their performance.

In this chapter, we have discussed the reference point and SBA of 5G. We have realised the SBA of the 5G by using the HTTP2 REST based API (nghttp of c++). Then, we have compared the studied realisation methods of 5G architecture on some of the metrics.

2.1 Reference Point Based Architecture of 5G

5G CN (Core Network) consists of Control Plane(CP) and Data Plane (DP) a.k.a User Plane (UP). CP has NFs performing the user related activities like network authentication, access authorisation, registration, session creation, maintaining the user profile, charging policies, modify session, de-registration, etc. DP performs the functionality of handling the incoming and outgoing IP data packets in the network. The NFs like AMF (Access Management Function), UDM (Unified Data Management), AuSF (Authentication Server Function), SMF (Session Management Function), UPF (User Plane Function), NRF (Network Repository Function), NSSF (Network Slice Selection Function), NSMF (Network Slice Management Function), NWDAF (Network Data Analytics Function) forms the complete CP of the 5G CN. Whereas, DP has a single NF which is UPF (User Plane Function).

Reference Point based development of 5G Core network uses the legacy standard of communication between the NFs which is point-to-point communication using the connected back haul network. Each pair of NF who want to communicate for the exchange of some services has the

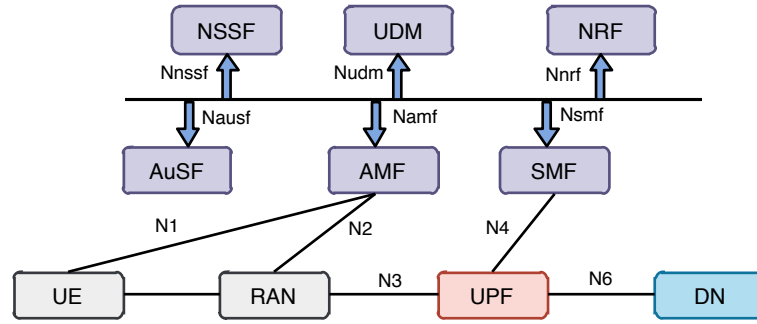


Figure 2.3: Service Based Architecture of 5G.

dedicated point-to-point connection between them. Reference Point based communication makes the network more prone to the network failure as any back haul network link failure will lead to malfunctioning of the end-to-end network. The reference point based 5G architecture is shown in the Fig. 2.2. The AMF NF is the first point of contact by the incoming USR (User Service Request) over the N1 interface. AMF makes use of the services offered by the AuSF over the N12 interface for performing the authentication and authorisation of the incoming user in the network. AuSF internally uses the user information stored in the data repository using the N13 interface. On the successful authentication of the incoming user the AMF makes the communication with the SMF for the session creation/flow setup using the N11 interface. SMF makes communication with the UPF over the N4 interface to create a session for the user, routing the data packets to outside Data Network (DN) using the N6 interface. On successful session creation for this user, the data packets of the user are then forwarded directly to UPF by the RAN over the N3 interface. On the de-registration of the user the created session is deleted and the complete procedure follows on the next visit of the user. The key functionalities of developed NFs of 5GC are as follows:

- Access Management Function (AMF): AMF supports user registration management, mobility management, connection management, user authentication and authorisation. It also marks the end of NAS signalling.
- Unified Data Management (UDM): Handles user identification, authorisation, subscription information and also generates the Key agreement credentials for the user.
- Network Repository Function (NRF): Handles network service registration and discovery. Enabling the NFs to discover each other services and perform their specific task.
- Authentication Server Function (AuSF): This network entity is the authentication server, being utilised to perform the authentication of the user by making use of UE's information from the UDM.
- Network Slice Selection Function (NSSF): This network entity helps the AMF to get the target slice for setting up the session for the incoming user. It selects the target network slice from the set of candidate network slices.
- Session Management Function (SMF): This network entity does the task of creating a session for the user with the help of UPF and is being contacted by the AMF using the service bus

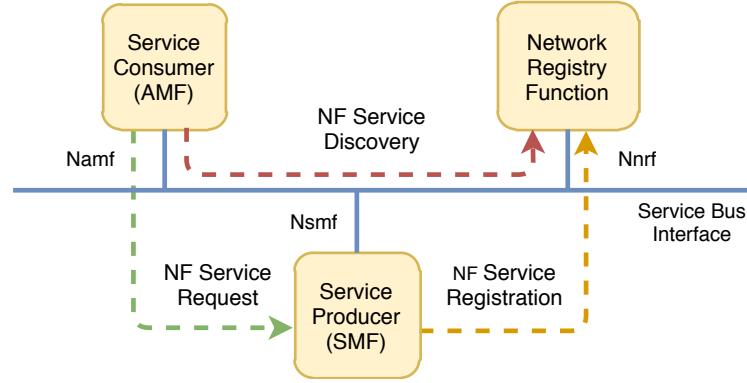


Figure 2.4: Service Registration and Discovery at NRF.

interface.

- User Plane Function (UPF): This NF handles the communication of the uplink/downlink data packets to/from the outside Data Network using N6 interface. The data packets are directly send to UPF for forwarding to the DN.
- Network Exposure Function (NEF): Provides exposure of the services and secure provision of the outside functions to make connection with the 3GPP defined network.
- Application Function (AF): Controls the policy framework, access NEF and performs traffic routing.

2.2 Service Based Architecture of 5G

The 3rd Generation Party Project (3GPP) has proposed the Service Based Architecture (SBA) of the 5G [11]. In the SBA of 5G, the point-to-point interaction of NFs have been replaced with a common message bus to which all the entities are being connected with their respective interfaces called as Service Bus Interfaces (SBIs). The common message bus is the open area of development for the SP to implement and enable the REST based communication of the NFs. There are available open source tools like Kafka, DPDK, HTTP2 based libraries, etc, to realise the SBIs (Service Bus Interfaces) of the SBA architecture. In the SBA, the UE and RAN are connected to the AMF with the dedicated interfaces which are N1 and N2 respectively. The RAN module communicates with the data plane NF UPF with the dedicated point-to-point N3 interface and UPF then communicates with the outside network over N6 interface. All the Control Plane entities communicates with each other using the SBIs and discovers each other using the NRF NF of the CP.

NRF NF, absent in reference point architecture, plays a key role in removing the dedicated interfaces of the reference point architecture. NRF helps the NFs to discover each other services by knowing the physical address of the target NF. The Fig. 2.4 shows the service registration and discovery mechanism being followed by the NFs to find and utilise the services offered by the other NF. In the example communication depicted in Fig. 2.4 AMF NF is the service consumer while SMF acts as the service producer. SMF first register its NF service with NRF, then AMF does the

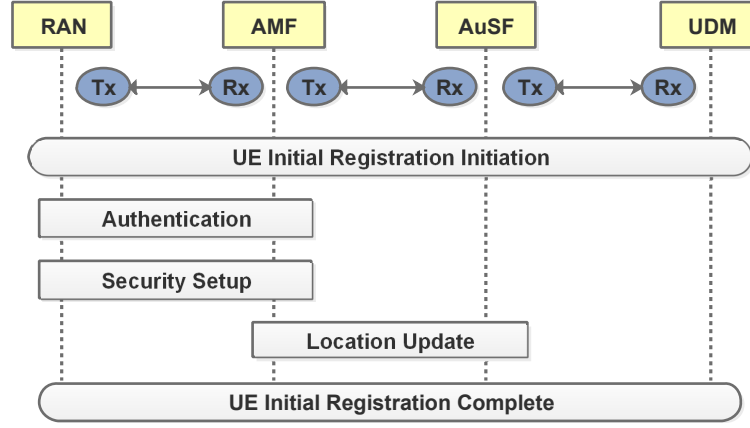


Figure 2.5: Call flow involved in performing the UE registration in ONVM-5G.

service discovery for SMF and NRF return back the address of the service helping the AMF to send the NF service request to SMF.

2.3 Experimental Setup

The performance of the reference point and SBA of 5G has been studied and compared on the developed test bed setup. The latency involved in activities like UE registration, UE session creation, UE de-registration time along with the plane throughput are being used to study and make the comparison between both the studied architectures.

Point-to-point dedicated network interfaces are being utilised for setting up the communication between the NFs in the reference point based 5G architecture. While, we have developed the common bus interface using the HTTP2 REST based API provided by the nghttp2 [12] library of C++ for SBA of 5G core network. The library provides APIs for establishing the HTTP2 based communication between the two entities, here NFs, and exchange the information. The library is based on the client-server model while using the TLSv1.2 for the HTTP2 connection establishment. The NRF network function is being realised using the third party open source software named as Consul [13]. It works on the principle of registry and discovery i.e it helps the NFs to register in the network and helps them to find/discover the other present NFs in the network.

In the test bed setup, the machines are Intel Xeon Gold having the 48 cores with two 1G NICs having Ubuntu 16.04 LTS as the Operating System (OS). In the test bed set up we have run the individual network functions in the docker containers where each of the dockerized network function runs separately in the server machine of above mentioned specification having the dedicated interfaces between them. The NFs in the SBA of 5G are using the client-server model for the communication, while utilizing the nghttp library for having the REST based communication. For simulating the UE and RAN, we have developed the RAN+UE simulator which triggers the UE related activities and uses Iperf [14] for sending the uplink traffic in the network. Similarly, we have developed a NF named as SINK acting as the outside network, running the Iperf for receiving the incoming traffic from the data plane and calculating the uplink throughput. SINK NF also sends the downlink traffic back to the RAN+UE simulator for calculating the downlink throughput at RAN+UE simulator.

We have tested the control plane functionalities as well as data plane functionalities of the 5G

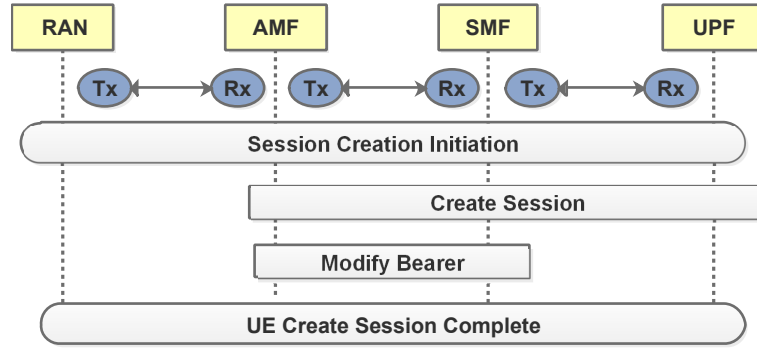


Figure 2.6: Call flow involved in performing UE session creation in the ONVM-5G.

architecture being realised in the test bed set up. Some of the control plane activities being realised are **user registration**, which emulates the user requests and performs the user initial attach along with the user network authorisation/authentication along with security setup and location update as shown in the Fig. 2.5. Then the second activity is **session creation** depicted in the Fig. 2.6, in this activity a session/flow is being created for the user and private IP address is being allocated to the user for the communication with the outside network. The last activity is **de-registration**, in which the user's created session is being deprecated and user is no longer being able to access the network to reach the data network. The UE, performing above mentioned functionalities is being simulated using the developed RAN+UE simulator where each thread invoked starts the UE activities and the load on the network is being varied using the number of threads being performing the UE related activities.

2.4 Results and Analysis

The time taken in performing the mentioned UE centric activities is being observed and compared between both the type of implementation of the 5G architecture. The implementation of the SBA involves the HTTP2 based client-server communication between the Network Functions involves a bit higher amount of time as compared to the dedicated point-to-point interface based communication in Reference Point implementation of 5G. The infrastructure on which both the models are studied compromises of the very efficient backhaul connectivity, i.e, the server machines are interconnected with each other with highly efficient fiber optical cable able to make the communication with the other servers very efficiently. While the implementation of the SBA of the 5G incurs some extra overhead in terms of service discovery with the help of NRF and then after getting the address of the target NF from NRF, it uses the HTTP2 based SBI to make the communication with the other NFs which is slower than the point-to-point dedicated interface based communication.

The observed delay in the performed user centric activities are depicted in Fig. 2.7, Fig. 2.8, Fig. 2.9. The Fig. 2.7 shows the time incurred in performing the user registration in both the 5G architectures. The Fig. 2.8 shows the time taken to create a session for the UE while the Fig. 2.9 shows the time involved in de-registering the user from the network which involves the deletion of the session created for the user. The above three metrics are being studied with the increase in the number of UE requests for the corresponding action to complete. As the number of users performing the specific metric operations increases the observed average time value for performing the activity increases.

The Fig. 2.10 shows the observed data plane throughput for the with the increase in the number of users. We have observed the data plane throughput for a user starts dropping as the number of users increases. The reason is being the CPU computation bottleneck at UPF NF of the data plane to process the incoming data packets and send it to the SINK and vice-versa.

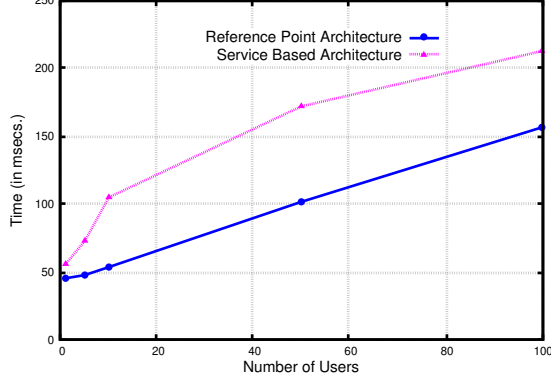


Figure 2.7: User registration time with the number of users.

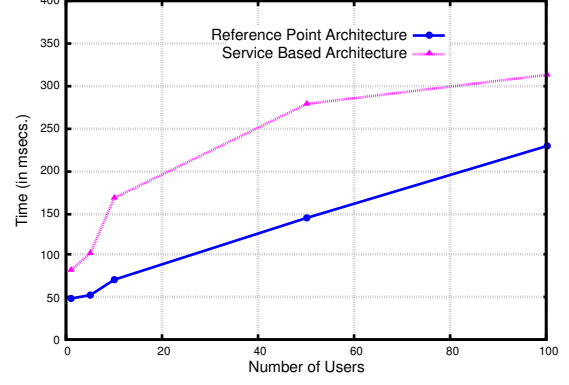


Figure 2.8: Session creation time with the number of users.

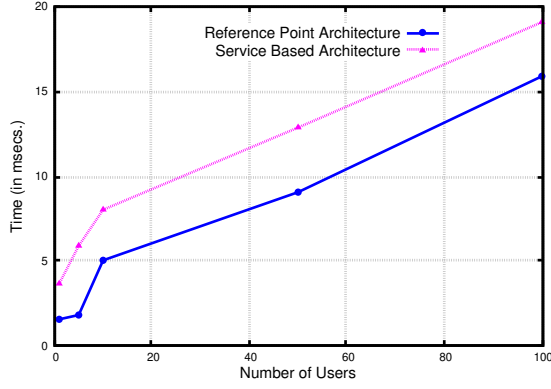


Figure 2.9: De-registration time with the number of users.

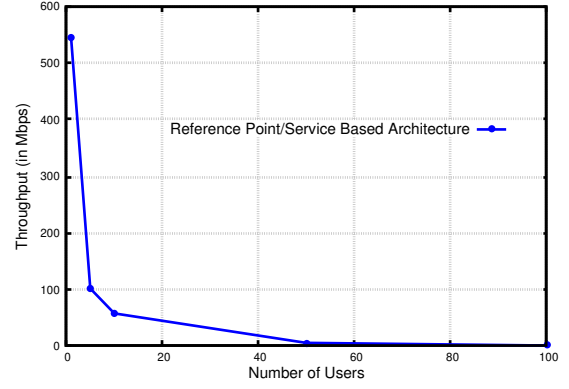


Figure 2.10: Data Plane throughput with the number of users.

2.5 Summary

We have successfully studied the Reference Point and Service Based Architecture (SBA) of 5G. The REST based communication is being achieved by using HTTP2 based API provided by nhttp2 library. Both of the 5G architectures which are Reference Point and SBA are studied and compared in terms of latency incurred in performing the UE activities and the achievable data plane throughput. The reference point based implementation results in bit lower latency while performing the UE activities as compared to the SBA of 5G. On the other hand, the reference point architecture is more prone to the network faults, also making the network rigid and not scalable. The UPF entity is the one which reached to the throughput saturation due to computational bottleneck. But, we believe given the higher resources to the UPF the data plane throughput can be increased to the order of Gbps.

Chapter 3

Monitoring and Orchestration of Multi Site deployed NS

5G networks have the capability to support a wide variety of services and requirements using network slicing [1]. Operators could customize their network for different applications and customers using slices. Slices furnish the flexibility in providing services as they differ in functionality (e.g., priority, and policy control), in performance requirements (e.g. latency, data rates, and availability), or in serving only specific kind of users (e.g., public safety users, industrial users, corporate users). A network slice can provide the functionality of a complete network, including radio access network and core network functions. One network can support one or several network slices. Additionally, the provision of the services by slices is boosted by Network Function Virtualization (NFV), for the 5G architecture, involving the virtualization of various network function services constituting control plane and data plane of 5G Core (5GC) and thereby forming the key enabler here. 3GPP [15] defines Network Data Analytic Function (NWDAF) for data collection and data analytics in centralized manner, making it a critical entity in the 5GC. 3GPP in [16] specifies how an NWDAF may be used for analytics as potential solutions to address various key issues on one or more network slices.

In this chapter, we have addressed the functionality of NWDAF co-located with the Network Slice Management Function (NSMF) entity of our deployed framework enabling the monitoring of the deployed slices and their corresponding VNFs by measuring some of the KPIs like throughput, CPU and memory usage. Also, we have illustrated the orchestration of the network slices being deployed over the multiple sites (physical machines/servers) by providing the detailed call flows for each of the Life Cycle Management (LCM) phase. The time for each phase of the LCM is being observed and comparison has been made between the single site and the multi site deployments.

3.1 Motivation and Related Work

In [17], authors detail on an integrated analytics architecture with respect to Radio Access Network (RAN) and core network architectures individually, by listing various use cases, the enhancements of the current architecture of the 3GPP 5G System (5GS) and key design directions where it could be useful. Authors in [18] discuss the applicability of exploiting data analytics for supporting the oper-

ation of the Radio Resource Management (RRM) algorithms embedded within the Next Generation-RAN (NG-RAN) nodes by conceiving RAN Data Analytic Function (RANDAF) as an execution platform for Data Analytics (DA) applications. In [19] authors propose a context based framework towards RRM using three mechanisms: Compass, Context Extraction and Profiling Engine (CEPE) and Context Information Processing (CIP) for optimizing the RAT selection, traffic steering and switching operations in 5G network environments along with evaluating one of these to target the minimization of the information collected and used by the data analytics engine..

In [20], the authors presented the state-of-art methods in service discovery with reflections on the specific needs of micro-service architecture in the context of telecom applications. The authors in [21] conclude that NFV based implementation is better suited for networks with high signaling traffic. This motivated us to realize the importance and use of NFV in 5G network slicing with micro-service architecture using lightweight dockerized framework. The authors in [22] discuss the creation phase of network slicing, by providing an insight into the procedures and mechanisms required to make the deployment of Network Slice Lifes (NSL) more flexible, agile, and automated from the perspective of both the NSL provider and the tenant. In [23], the authors focused on run time phase of a network slice by proposing a SDN/NFV-based architecture enabling operation of NSL instances with recursiveness, multi-tenancy, and multi-domain support. These architectural solutions address the isolation properties necessary in slicing namely performance, security, privacy, and management isolation, in compliance to ETSI NFV information model. The authors in [24] discuss the various types of multi-domain supportive slice orchestrators to handle instantiation, management and deployment of a network slice over multiple administrative domains and using resources from different technological domains.

3.1.1 Motivation

All these works highlight on the challenges faced by operators and the things the operators have to bear in mind in designing architectural solution for successful deployment and operation of network slices, while meeting the SLA requirements. Though these factors are quite important, addressing the effective utilization of usage of VNF resources by a slice and the need of associating them to a fully operational end to end network slices. However, it is worth mentioning that little attention has been placed to date to detail the actual implementation on the proposed framework or architecture involving actual components of it to solve the different issues faced by operator in building network slice supportive self organizing network. Hence in this chapter, we focus on building a full prototype integrated framework of network slice monitoring and analytics capturing both 5G network slice and network function specific metrics to help achieve self optimization in an end to end 5G network slicing. We envision on addressing those needs of VNF resources across life cycle of various network slices an operator has to nourish for successful deployment and operation of network slices.

3.2 Network Slice Monitoring Framework

As defined by 3GPP, a Management Data Analytics Service (MDAS) [25] provides data analytics for the network. MDAS can be deployed at different levels, for example, at domain level (e.g., Radio Access Network (RAN), Core Network (CN), Network Slice Subnet Instance (NSSI)) or in a centralized manner (e.g., in a PLMN level). A domain-level MDAS provides domain specific

analytics, e.g., resource usage prediction in a CN or failure prediction in a NSSI, etc. A centralized MDAS can provide end-to-end or cross-domain analytics service, e.g., resource usage or failure prediction in an NSI, optimal CN node placement for ensuring lowest latency.

5G management system can therefore benefit from management data analytics services for improving performance of the network and efficiency of network slices to accommodate and support the diversity of services and requirements. The management data analytics utilize the network management data collected from the network including e.g. service, slicing and/or network functions related data and make the corresponding analytics based on the collected information. For example, the information provided by Performance Management Data Analytics Function (PMDAF) can be used to optimize the network performance.

3.2.1 Key Performance Indicators of Network Slice

5G system needs to support stringent KPIs for latency, reliability, throughput, etc. Enhancements in the core network contribute to meeting these KPIs using network slicing, in-network caching, scalable assignment of network resources and hosting services closer to the end points. Therefore, measuring the KPIs is very crucial to cater the various requirements of 5G Core (5GC) enhancements, flexibility, and optimization.

3GPP defines KPIs through the measurement of key parameters of input and output of internal network system. KPIs are considered to be primary metrics to evaluate process performance as indicators of quantitative management. As service serve-ability performance is a significant factor here, it falls into one of the three related categories: service accessibility, retain ability and integrity performance. Pertaining to these KPIs, we focus on 5G core performance monitoring and quality bench-marking. Table 3.1 lists a few set of KPIs in 5G system for various KPI categories specified by 3GPP.

Table 3.1: KPI categories

KPI Cate- gory	Example KPI
Accessibility	Registered Subscribers of Network and Network Slice Instance through AMF
Integrity	End-to-end Latency of 5G Network
Retainability	Quality of Service (QoS) flow Retainability
Utilization	Mean number of PDU sessions of network and network slice instance

3.2.2 Developed Framework

The developed framework consists of 5G SBA as the System Under Test, orchestrated using the NFV MANO functions provided by OSM [26] Rel.5. On its northbound interface there is a Network

Table 3.2: Metrics exposed by AMF

Metric Name	Description
ue_registration_requests_total	Total number of registrations requested by all UE
ue_registrations_success_total	Total number of successful registrations
ue_session_creation_total	Total number of sessions created
ue_deregistration_requests_total	Total number of deregistrations requested by all UE
ue_deregistrations_success_total	Total number of successful deregistrations
active_ue_total	Number of active UE

3.2.3 Monitoring Technologies

1. **Prometheus:** Prometheus [27] is an open source, pull-based, service monitoring system. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.
2. **cAdvisor (container Advisor):** cAdvisor [28] is an OSS which is a running daemon that collects, aggregates, processes, and exports information about running containers. We use this to get CPU utilization, memory usage, and network usage of the VNFs which runs as docker containers. cAdvisor exposes these metrics at a HTTP endpoint which can be scraped by Prometheus.

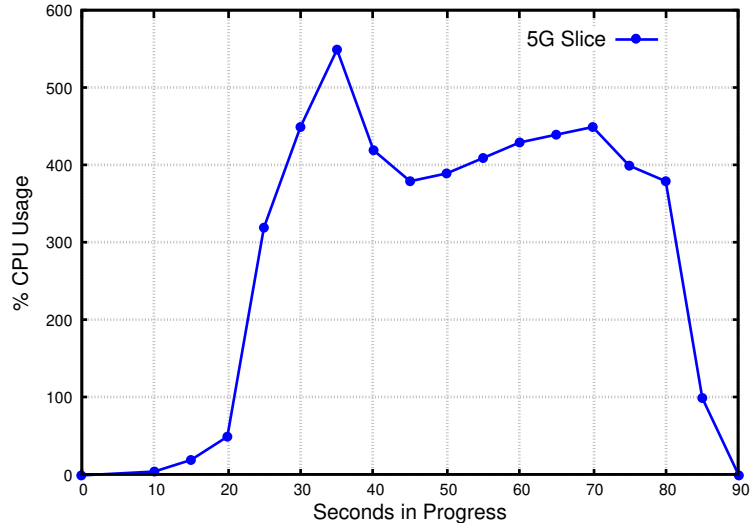


Figure 3.2: CPU utilisation recorded for the slice.

3. **node_exporter:** node_exporter [29] is a Prometheus exporter for hardware and OS metrics exposed by *NIX kernels. We use this to get CPU utilization, memory usage, and other OS metrics of the actual hardware as opposed to individual containers from cAdvisor.
4. **Prometheus C++ Client:** Prometheus C++ Client [32] is a C++ open source library,

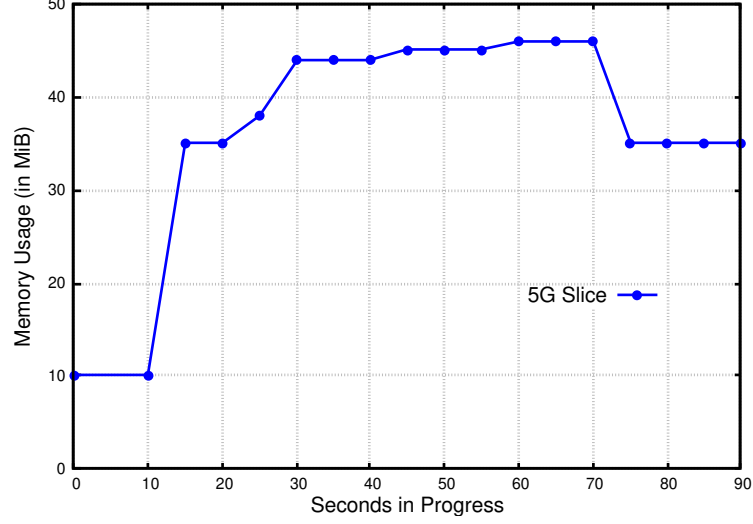


Figure 3.3: Memory usage recorded for the slice.

which we use to instrument our 5G SBA code. It exposes custom defined metrics at a HTTP endpoint to be scraped by Prometheus. We use this in AMF to expose the metrics listed in table 3.2.

5. **Grafana:** Grafana [33] is a leading open source dashboard and graph editor which helps us in visualizing the metrics from Prometheus and creating dashboard with required graphs.

Figures Fig. 3.2 and Fig. 3.3 show the Grafana snapshots of CPU and memory metrics, consumed by the slice, when 10 UEs perform UE registration, end to end uplink and downlink data exchange and UE de-registration when served by a *single thread* at each of the core network functions respectively.

3.3 MANO of a Multi Site deployed NS

MANO (Management and Orchestration) unit of the network infrastructure proposed by 3GPP helps in managing the life cycle of the network slice and provide complete orchestration in terms of infrastructure, resources and communication with other network entities. In this section, we have mentioned about the development of the framework supporting multi site deployment of a NS along with managing the complete life cycle of the deployed NSs. With the observed high memory and CPU utilisation by some of the NFs of a slice, multi site deployment helps the SP (Service Provider) to run the highly loaded network modules or NSSIs (Network Sub Slice Instances) on a high end servers able to meeting the high demand of the network.

3.3.1 Developed Multi Site Deployment Framework

Here, we have detailed on the complete emulated implementation of the end-to-end network slice management, NSMF, NWDAF, NSSF interactions, and orchestrating the slice by retrieving the real time status information from NSMF. We have emulated network slice orchestration and its life cycle management functionalities by developing an NSMF acting as OSS/BSS using the NFV

MANO functions provided by Open Source MANO (OSM) Rel.5 on its North Bound Interface. OSM provides the Orchestration (NFVO) and VNFM functionalities that supports communicating with different VIMs. We have picked a light weight VIM-Emulator which emulates the Openstack functions for VIM named as vim-emu. Vim-emu allows the execution of real network functions packaged as Docker containers in emulated network topologies running locally on the developer's machine. As shown in Fig. 3.4, our framework supports deploying end-to-end network slices on eMBB, uRLLC, and mMTC across different sites hosted by respective vim-emulator.

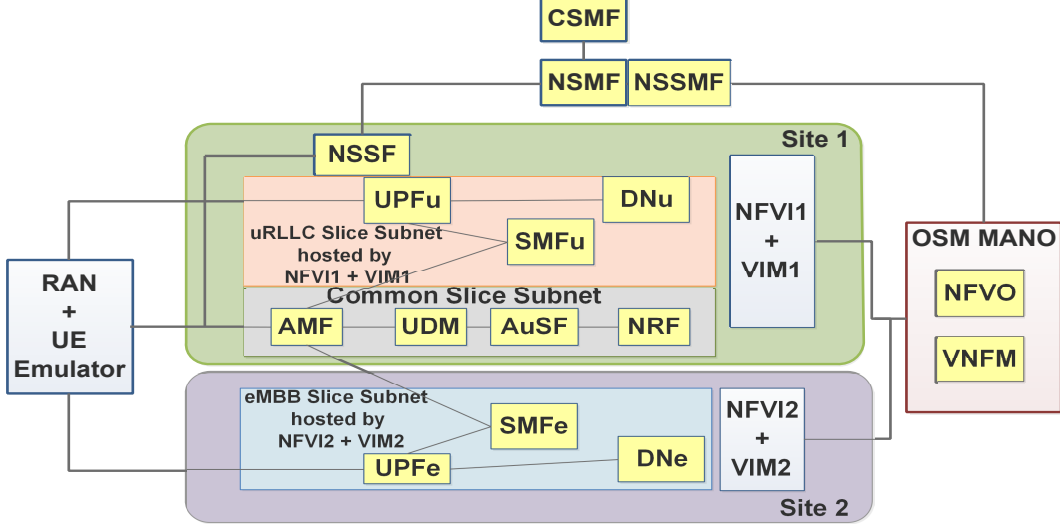


Figure 3.4: Deployed framework for the multi site deployment of the network slices.

The developed prototype model of the 5G SBA comprised of 5GC control plane entities listed as AMF, AUSF, UDM, SMF, and NSSF implementing REST API message interaction model using HTTP2 library API provided by nhttp2. The Network Resource Function (NRF) runs as a Consul server providing service registration and discovery services to other network functions in the framework which run Consul client inside them. The data plane entities listed as UPF, Data Sink entity on N3 and N6 interfaces. For testing purposes, we developed a light weight RAN with embedded UE NAS function terming it as RAN + UE Emulator. All these network functions including RAN + UE Emulator are developed as virtualized docker containers each intended to provide micro service functionalities such as UE registration, de-registration, and end-to-end uplink and downlink data exchange over different network slices. We used the framework shown in Fig. 3.4, in order to emulate the orchestration and selection of different network slices. We deployed two different sample network slice subnets named as uRLLC slice subnet and eMBB slice subnet in 5GC along with a common slice subnet across two different sites. *Site1* represents a local site as it is hosting uRLLC slice subnet, common slice subnet, and RAN + UE Emulator along with OSM. *Site2* represents a remote site as it consisted of only an eMBB slice subnet exclusively.

We used the system configuration shown in Table 3.1 for *site1* and Table 3.2 for *site2*.

Common slice subnet comprised of AMF, AUSF, UDM, and NSSF hosted on a local site by vim-emu1 (VIM1+NFVI1). uRLLC network slice subnet consisted of SMFu, UPFu and Data Network(u) hosted on a local site (*site1*) by vim-emu1 (VIM1+NFVI1). eMBB network slice subnet consisted of SMFe, UPFe, and Data Network(e) hosted on a remote site (*site2*) by vim-emu2 (VIM2+NFVI2).

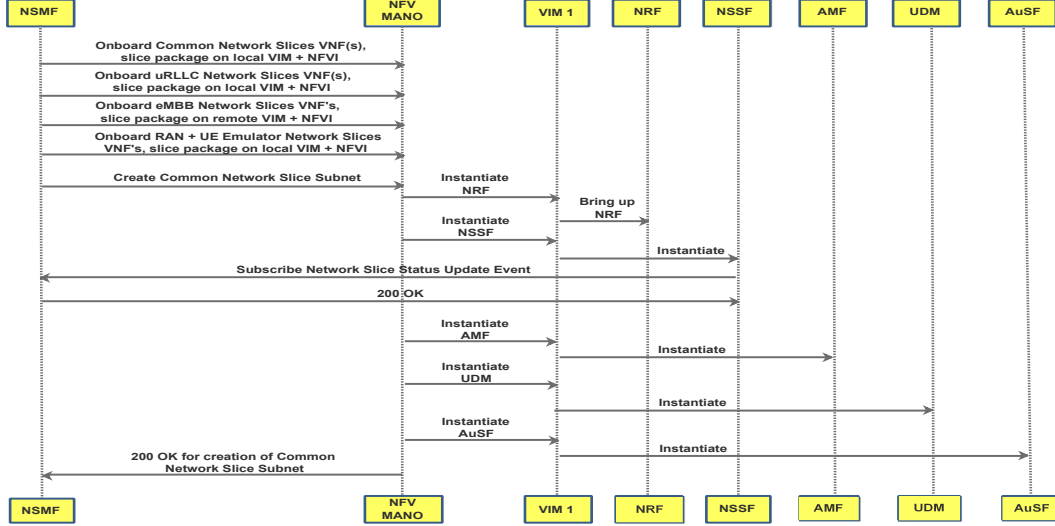


Figure 3.5: REST API message exchange in preparation phase & commissioning phase of common network slice subnet.

Table 3.3: Local site (site1) system configuration

Architecture	Intel(R) Xeon(R) CPU E5-2640 v4
Total Number of CPU Cores	40
Thread(s) per core	2
Clock Speed	1199 MHz, with max capacity 3400MHz

3.3.2 Life Cycle Management of the NS

A network slice goes from several phases starting from the time its being prepared till its being removed from the deployed system. Following are the four phases which constitutes the complete life cycle of a slice in the network.

- **Preparation Phase** NSMF on boot-up, on boarded different slice subnet templates namely common network slice subnet template with NSSF, AMF, AUSF, and UDM VNFs on local site using VIM1+NFVI1, uRLLC network slice subnet template with SMFu, UPFu, and DNu VNFs on local site using VIM1+NFVI1. It then on-boarded eMBB NS template consisting SMFe, UPFe, and DNe VNFs on remote site using VIM2+NFVI2 as shown in fig. 3.5. Finally, the RAN+UE Emulator Network Slice template consisting of RAN Emulator VNFd is on-boarded on local site using VIM1+NFVI1.
- **Commissioning Phase**
 - **Commissioning of Common Network Slice Subnet** This phase involves instantiating and activating Common Network Slice Subnet on local site. Fig.3.5 shows the REST API message exchange on respective interfaces in our implementation framework during preparation and commissioning phase of common network slice subnet where this slice subnet gets successfully instantiated and activated. Upon activation, each VNF in this slice subnet (AMF, AUSF, UDM, and NSSF) registers itself at NRF by sending its type

Table 3.4: Remote site (site2) system configuration

Architecture	Intel(R) Xeon(R) Gold 6126
Total Number of CPU Cores	48
Thread(s) per core	2
Clock Speed	1499 MHz, with max capacity 3400MHz

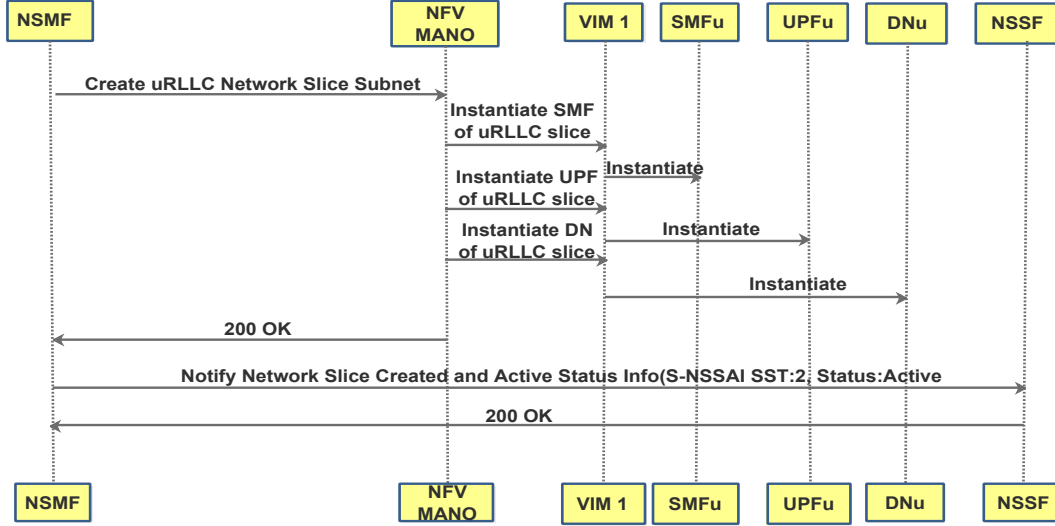


Figure 3.6: REST API message exchange in commissioning phase Of uRLLC network slice subnet.

of service, IP Address, and port number. It is now ready to serve the traffic of 5GC control plane at their respective SBIs. NSSF would subscribe to NSMF for network slice(s) status update events.

- **Commissioning of uRLLC Network Slice Subnet** This phase involves instantiating uRLLC slice subnet on local site. Fig.3.6 shows the REST API message exchange on respective interfaces in this commissioning phase of uRLLC network slice where this slice gets successfully instantiated and activated. Similar to common slice subnet, every VNF in this subnet would register itself at NRF. All the VNFs in this slice subnet (SMFu, UPFu, and DNu) would be active now and ready to serve the traffic of 5GC control plane and user plane at their respective interfaces. Point to note here is that NSMF would notify the NSSF (through co-located NWDAF) about the successful activation of this uRLLC slice with respective SST value of 2 (as NSSF would have already subscribed for this status update event).
- **Commissioning of eMBB Network Slice Subnet** This phase involves instantiating eMBB slice subnet on remote site which took SMFe, UPFe, and SINKe to active state. Here, NSMF would notify the NSSF (through co-located NWDAF) about the successful activation of the eMBB slice, with respective SST value of 1 (as NSSF would have already subscribed for this status update event).

• Operation Phase

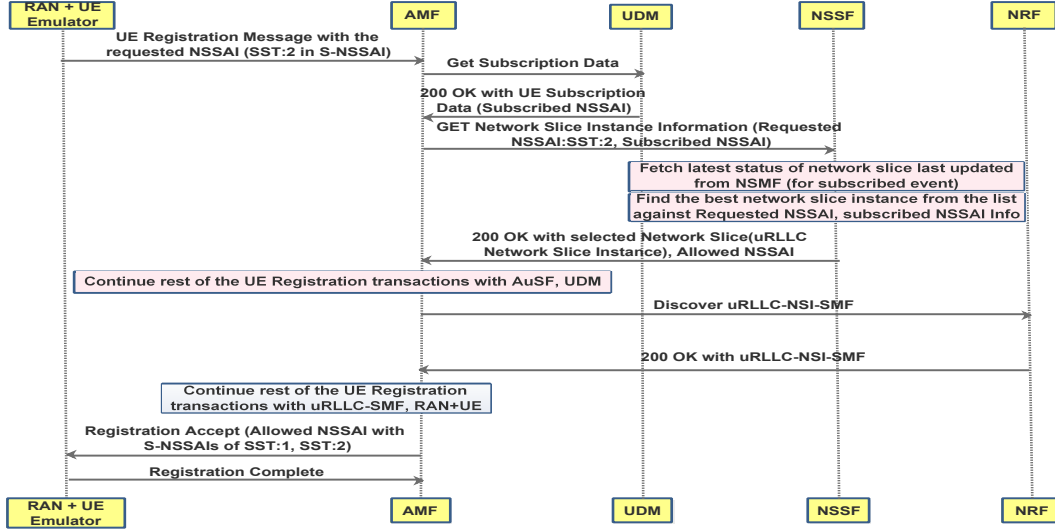


Figure 3.7: End-to-End message interactions for UE registration with default PDU session request for uRLLC slice.

- **UE Registration and PDU Session Establishment for uRLLC Slice** Once the RAN+UE emulator function gets active, the UE starts registering to the 5GC with default PDU session establishment request for uRLLC slice requesting NSSAI with the SST value of 2. Here, AMF seeks help from NSSF asking for network slice instance and *Allowed NSSAI* information. Now, NSSF already has the real time status of the available network slice instances from NSMF. So, it chooses the most appropriate slice instance on uRLLC SST value of 2. NSSF then provides the final *Allowed NSSAI* and respective slice instance id to AMF.
- **PDU Session Establishment for eMBB Slice** Now the RAN+UE emulator requests for additional PDU session establishment for eMBB slice with the SST value of 1. Here, AMF doesn't contact NSSF again as it has the updated *Allowed NSSAI* list consisting of both active slices with SST value of 1 and 2. AMF then discovers the matching SMF from NRF by providing this eMBB service using respective slice instance id. After this, rest of the transactions on the PDU session establishment continues with SMFe and UPFe.

- **Decommissioning Phase** We verified the slice decommissioning phase by triggering the slice deactivation for these active slices from NSMF over NBI interface to OSM MANO.

The Fig. 3.7 shows the interaction between the NFs involved in performing the UE registration to the network. The activity starts with the UE registration message from the RAN+UE emulator to the AMF with the requested NSSAI, having the SST value. AMF communicates with the UDM for fetching the UE subscription data, having the UE subscribed NSSAI. AMF communicates with the NSSF for getting the best slice instance from the list of requested and subscribed NSSAI on the basis of the latest status of the network slice instances. NSSF reverts back with the selected slice instance for this UE. AMF then discovers the corresponding slice instances with the help of NRF for continuing the rest of the registration process.

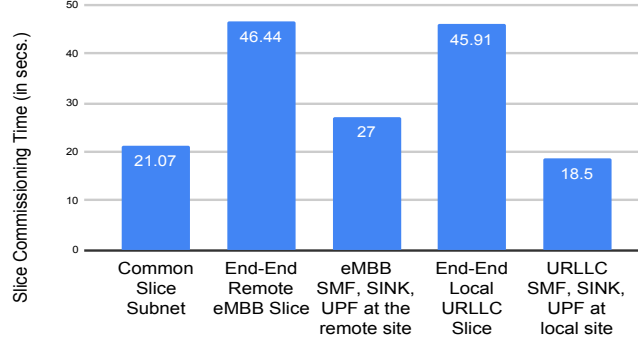


Figure 3.8: Commissioning Time for different network slice instances.

3.4 Results and Analysis

We have collected the time for each of the network slice LCM phase for both of the studied single site and multi site slice deployment. The Fig. 3.7 shows the preparation time for the different slice instances. We can observe high preparation time for the end-to-end slice deployment as compared to the multi site deployment in which we deploy the network sub slices over the multiple sites parallelly. As shown in Fig.3.8 and Fig.3.10, we measured commissioning time and decommissioning time of different end-to-end slice(s) versus slice subnets across different sites respectively. The observed commissioning time of the end-to-end slice is high as compared to the commissioning time of the dedicated slice subnets like of eMBB and URLLC. These results show that, in order to serve on different 5G network slicing use cases for uRLLC, eMBB, mMTC, and edge computing scenarios, an operator should carefully choose the systems (VMs) to host these network functions and place the network functions appropriately across different sites when managed by a single MANO entity. We can intelligently deploy a slice having multiple slice subnets over the multiple sites with the less commissioning time. Thus with the multi site deployment of a network service we observe less time for each of the network slice life cycle phase as compared to that of the end-to-end deployment of the network slice on a single site.

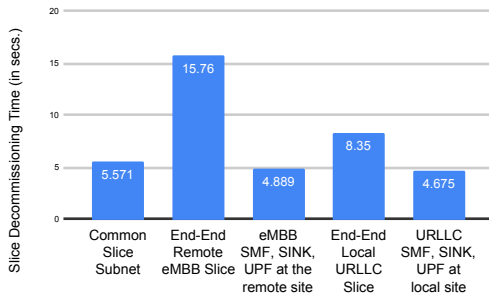


Figure 3.9: Preparation Time for different network slice instances.

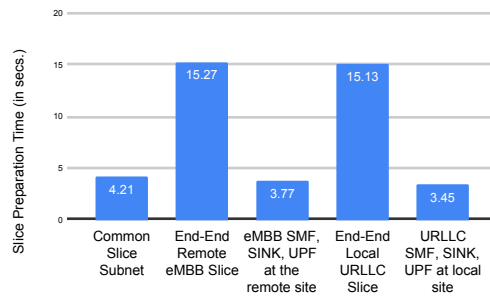


Figure 3.10: De-commissioning Time for the different network slice instances.

3.5 Summary

We have studied the life cycle of various types of network slices hosted across different sites managed by single MANO in a multi-site environment with micro service based docker containerized framework, in the context of 5G networks. We have measured the time of different phases of slice life cycle on NSI(s) for the deployed network slices with the deployed infrastructure. As a proof of concept we built an end-to-end network slice management framework by developing an NSMF, NWDAF, and NSSF using REST API based HTTP2 and the orchestrated environment provided by ETSI NFV MANO aligned OSM Rel.5. We focused on the 5GC side entities with our 5G SBA prototype model using REST API based HTTP2. The real time status information of different network slices at NSSF from NSMF instantaneously is then used to cater the different slice requests from an UE. Work encompasses mainly on addressing the requirements of latency critical applications of uRLLC slice like remote health monitoring systems and remote control of dense traffic areas.

Chapter 4

Adaptive Network Slicing in 5G for Efficient Resource Utilisation

In this chapter, we have acknowledged the previous work by proposing the algorithms for managing the state of a network slice acquired during its life time. We have analysed the importance of our proposed mechanism in terms of low request response time and efficient resource utilization. The response time measures here is the time the request has to wait before being allocated to a network slice for using the network services. While, the resource utilization has been compared between the studied mechanisms in terms of running the network slice. Here, the adaptive means we are dynamic to the state of the running slice which means we are dynamically changing the state of the network slices as per the incoming load to the network. While in static network slicing, we are static to the state of the running slice which means we are not switching the state of the slice once activated with the incoming traffic to the network.

4.1 Motivation

In the work mentioned in chapter 3, we have measured the time taken by each of the phase of the life cycle of the network slice which are Preparation Phase, Commissioning Phase and De-commissioning phase. The duration for the Operation Phase depends on the amount of the time the slice is serving the load of the traffic, so the time for this phase is not fixed rather depends on when the state is being changed from the active to de-active or de-commissioned. We have observed an high amount of time involved in bringing up the end-to-end network slice, where the preparation phase and the commissioning phase of Network Slice (NS) life cycle is taking around 15 seconds and 45 seconds respectively. The incoming requests have a certain SLA (Service Level Agreement) and QoS (Quality of Service) which permits an requests a maximum delay to complete the registration and session creation. While having such a high amount of time involved in the slice life cycle phases, we left with the pre-honoring the slices so as to have least request response time for the incoming requests. Thus, our previous work motivates in performing the intelligent operation/management of the slices so as to have low waiting time of the incoming requests and on the same time not wasting the resources available to the service provider.

4.2 Background: Adaptive Network Slicing

Network Slicing in 5G helps in meeting the adverse requirements of the users in terms of services offered by the SP (Service Provider). Enables the SP to cater the needs of various services by composing a NS (Network Slice) for each of the service in isolation with each other on the same physical infrastructure. Thus, helping the SP to maximise the revenue by meeting more network requirements with the same amount of limited infrastructure. We have explained some key terminologies and concepts associated with a network slice in this section.

4.2.1 Network Slice as a Service (NSaaS)

Network slicing allows a network operator to provide dedicated virtual networks with functionalities specific to the service over a common network infrastructure. It facilitates mapping of service demands from a customer to functionalities, topology, policies, and parameters of a network slice automatically. Thus, it will be able to support the numerous and varied services envisaged in 5G.

ETSI standard defines Os-Ma-Nfvo a reference point interface between Operations Support System/Business Support System (OSS/BSS) and NFVO as the North Bound Interface of NFV MANO, detailing Network Service Descriptor (NSD) Management, Network Service (NS) life cycle management, LCM of VNF, Life Cycle Change Notification over NS, NS Performance Management, NS Fault Management, Fault Management (FM) on resources supporting VNF, and flavours of these. Since we realize the network slice as network service, all the functions and features defined by ETSI on this interface could be very well mapped to network slice.

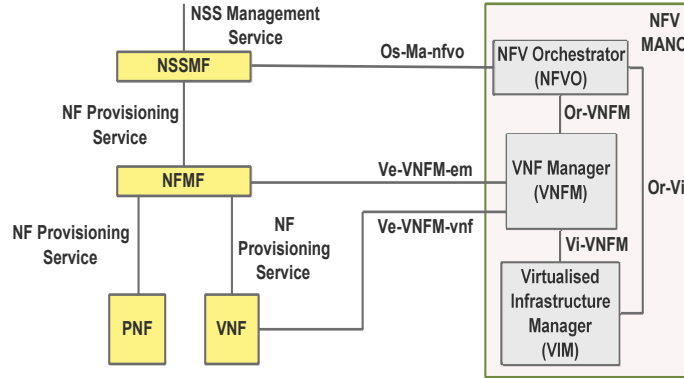


Figure 4.1: Deployment Scenario for NSSI.

3GPP network slice management system [34] shall be capable to consume the services provided by NFV MANO interface i.e., over the Os-Ma-nfvo reference point. Hence in this context, NSSMF shall assume the role of OSS/BSS entity. Fig. 4.1 shows a deployment scenario for NSSI (Network Slice Sub Instance) management with the interface to NFV MANO.

4.2.2 Role of NSSF

3GPP defines network slice selection as the process of selecting an instance of a network slice. In this context, NSSF plays an effective role in providing information on a network slice instance to an Access Management Function (AMF).

To enable an UE to use different services provided by different network slice instances, there is an assistance information parameter named as Network Slice Selection Assistance Information (NSSAI). This parameter assists the network to allow an UE to camp on a particular instance of a network slice. An NSSAI is a collection of S-NSSAIs, where S-NSSAI stands for Single Network Slice Selection Assistance Information. Each S-NSSAI assists the network in selecting a particular NSI. An S-NSSAI is comprised of a Slice/Service Type (SST) which refers to the expected network slice behaviour in terms of features and services, with optional Slice Differentiator (SD). 3GPP [35] assigns SST value of 1 to eMBB, 2 to uRLLC, and 3 to mMTC.

During the registration procedure, when the UE context in the AMF does not include an *Allowed NSSAI* for the corresponding access type, the AMF queries the NSSF to fetch every such detail of the NSI. Also, during the PDU Session Establishment, if the AMF is not able to determine the appropriate Session Management Function (SMF) to serve the S-NSSAI provided by the UE, the AMF may query the NSSF with this specific S-NSSAI, location information, and PLMN ID of the Subscriber Permanent Identity (SUPI).

4.2.3 Network Slice Selection and Management

When multiple network slices are supported by the 5GC, NSSF helps in effective slice instance selection during the UE registration and PDU session establishment. 3GPP specifications [35], [11], and [36] define Network Data Analytics Function (NWDAF) as the service producer in providing network slice instance load information to NSSF. But, how NWDAF could obtain the slice instance load information is not defined in the standard as it is not a slice management function by itself. In this context, NSMF is the entity who controls the overall orchestration and life cycle management of different network slices effectively. All the real time information of the slices like current load and the status of the network slices like active, decommissioned, or configured would be available at NSMF.

In our work, we have considered that the NWDAF be co-located with NSMF. NSSF could retrieve the instantaneous status of different network slice(s) from NWDAF co located with NSMF or NSSMF for that matter and then add a local intelligence to select a particular slice instance, when a request arrives from AMF at NSSF. In the context of 5G network, every slice typically would consist of common slice subnet VNFs like AMF, NSSF, Authentication Selection Function (AUSF), and Unified Data Management (UDM), as well slice specific VNFs like SMF and User Plane Function (UPF). Our study and experiments on slice life cycle relate to this specific set of entities in 5GC. Thus, NSSF and NSMF are the two key NFs helping in achieving the adaptive network slicing where NSSF helps in monitoring and analyzing the incoming load and taking the decision of switching in between the states of network slice while NSMF helps in performing the communication with the MANO entity for improvising the decision of change in the slice state .

4.3 Static Network Slicing

We define the term "Static Network Slicing" for the network slice which is prepared, created and activated with successful deployment and being operational. This refers to the step by step phases of slice life cycle defined by ETSI. Once the service of slice is utilized, the slice would be considered as no longer in use and is decommissioned thereafter. In 5GC NFV framework, when VNFs execute

Algorithm 1: Algorithm to serve the request using With/Without provisioning mode.

Result: Serving the request using With/Without provisioning algorithm.

```
1  $load[] \leftarrow inputLoad$ ;
2 while  $i$  in  $load$  do
3    $CurrentRequests \leftarrow load[i]$ ;
4    $CurrentRequests == 0$  Wait for  $monitoringPeriod$  and Deactivate/Decommission the
   slice;
5   if  $CurrentState == Active$  then
6     | Requests will be served with service time.
7   else
8     | if  $CurrentState == Deactive$  then
9       | Trigger activation of slice;
10      | Wait till state becomes active;
11    else
12      | Trigger commissioning of the slice.
13      | Wait till state becomes active;
14    end
15  end
16 end
```

on VMs, the total time of getting a slice into operational phase would be in the order of minutes. These VNFs also need to be configured before being operational. Fig. 3.4, shows our development framework consisting of common slice subnet, uRLLC slice subnet and eMBB slice subnet. Common and uRLLC slice subnets are deployed in our local site (NFVI1 + VIM1) and eMBB slice subnet is deployed over remote site (NFVI2 + VIM2). Common slice subnet consists of common network functions for eMBB and uRLLC slice. As shown in the Fig. 3.5, we compared the commissioning time of different end-to-end slice(s) versus slice subnets across different sites. These results show that, in order to serve on different 5G network slicing use cases for uRLLC, eMBB, mMTC, and edge computing scenarios, an operator should carefully choose the systems (VMs) to host these network functions and place the network functions appropriately across different sites when managed by a single MANO entity. These experiments persuaded us to study and monitor the life cycle of network slice in depth.

4.3.1 Always-ON Network Slice

When a network slice exists always to serve the various verticals for both high and low number of devices at various circumstances, we define it as "Always-ON network slice". The operator pre-deploys the slice before an UE requests to connect to it. The slice instance information is available at NSSF from NSMF. The operator provisions the UE subscription with a set of configured NSSAIs from which the UE selects a subset as requested NSSAIs. When the UE requests to connect to a slice, the AMF with the help of NSSF in the network accepts or denies individual S-NSSAIs and returns the set of Allowed NSSAIs. This slice always exists irrespective of the users connected to it. This type of slice is typically useful to cater the requirements of eMBB scenario. However the problem with this type of slicing is that, it consumes the resources of different VNFs involved in it, even when there are no users using it. This is not encouraging and cost effective to the operator for deploying mMTC slices or URLLC slices. However, such type of network slices (Always ON network

slices) results in very low waiting time for the incoming user requests as the slice is always active to serve the incoming user request.

4.4 Adaptive Network Slicing

In this section we propose the concept of "Dynamic Network Slicing". In the mMTC/URLLC slice scenarios where there is very less often communication between the devices, it is wise to create the network slice when needed or just in time of use. In this angle we study the traffic of network slice selection in estimating the future traffic on different slices and use it for controlling the activation and deactivation of slice. This is motivated by the fact the slice commissioning phase time is in the order of minutes, even with VNFs running on light weight containers.

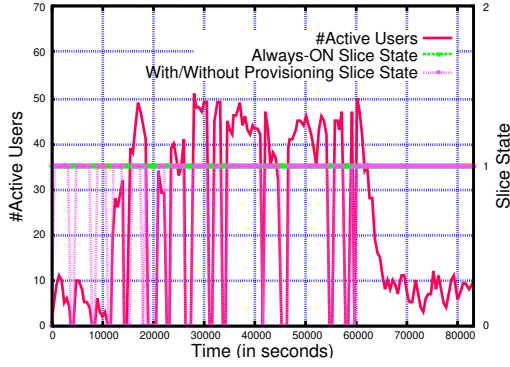


Figure 4.2: Input load and state of slice when no user present for 20% of the time.

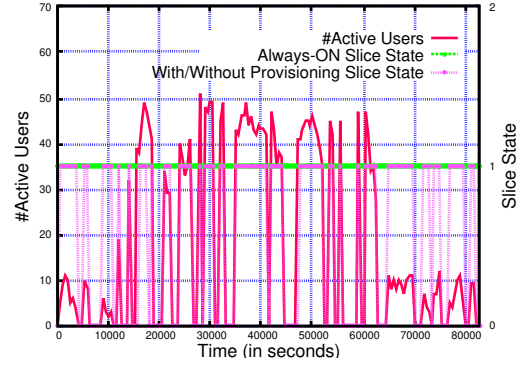


Figure 4.3: Input load and state of slice when no user present for 50% of the time.

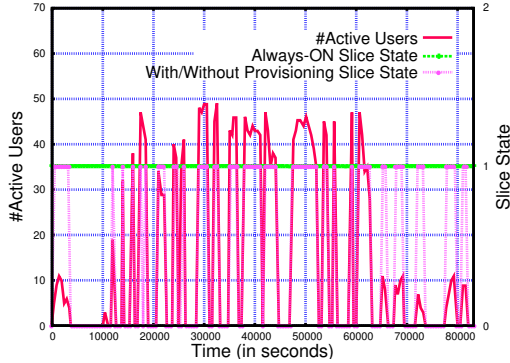


Figure 4.4: Input load and state of slice when no user present for 70% of the time.

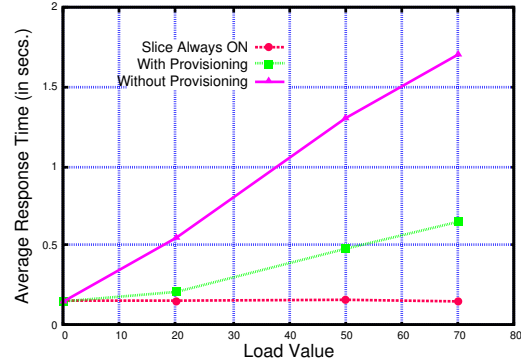


Figure 4.5: Average response time of With/Without provisioning versus Always-ON network slice.

4.4.1 Controlling Slice Activation and Deactivation

Network slice commissioning including instantiation, configuration and activation consumes quite an amount of time as studied and quoted in the section 3.4. Triggering the slice activation on request from user (during PDU session establishment) through AMF, would not help. At the same time,

keeping the slices always operational would not help the operator in effective utilization of VNF resources involved in the life of a slice. Hence, we propose an intelligent technique of tracking the activation and deactivation of slice from NSSF. The proposed algorithms have been studied on three different load patterns as shown in Fig. 4.2, 4.3, and 4.4. Load pattern in Fig. 4.2 shows the heavily loaded slice with no active user present 20% of the time while Fig. 4.3 and Fig. 4.4 represents load patterns where no active user present 50% and 70% of the time respectively.

Algorithm 2: Metric calculation for With/Without provisioning algorithm.

Result: Metric calculation for With/Without provisioning algorithm.

```

1  $load[] \leftarrow inputLoad;$ 
2  $totalRT \leftarrow 0;$ 
3  $totalRequest \leftarrow 0;$ 
4  $currentState \leftarrow 0;$ 
5  $serviceTime \leftarrow 150msec.;$ 
6  $switchingOnTime;$ 
7 while  $i$  in  $load$  do
8    $CR \leftarrow load[i];$ 
9    $totalRequests \leftarrow totalRequests + CR$ 
10  if  $currentState == 1$  then
11     $currentRT \leftarrow CR \times serviceTime;$ 
12     $currentState \leftarrow 1;$ 
13  else
14     $currentRT \leftarrow switchingOnTime + (CR \times serviceTime);$ 
15     $currentState \leftarrow 1;$ 
16  end
17   $totalRT \leftarrow totalRT + currentRT$ 
18   $RT[i] \leftarrow currentRT;$ 
19   $CR == 0$   $currentState \leftarrow 0;$ 
20 end
21  $AvgRT \leftarrow totalRT \div totalRequests;$ 

```

4.4.2 Network Slice with Provisioning

In the naive method of switching the state of slice, we decommission the slice when the number of active users drop to 0 and starts the deactivation timer in which slice does not serve the traffic as it is moving to decommission state.

In With-Provisioning mode shown in Algorithm 1, when the active users in slice drops to 0, we simply don't decommission the slice, rather we monitor the load pattern for monitoring period and then either deactivate/decommission the slice or keep the slice as active. Now, when the slice state is deactivate and a user request arrives for the slice then the request will have the additional activation time in getting the response from NSMF and sending that response from NSSF to AMF. If the slice is decommissioned then each request has to wait for the commissioning time to get the request served.

If the slice state is active then the algorithm will behave like "Always-ON" algorithm with no

additional latency incurred for the total response as shown in the Algorithm 1.

4.4.3 Network Slice without Provisioning

In this mode of algorithm, when the number of active users on the slice drops to 0, we change the state of the slice from active state to decommissioned state, rather than de-active state. Decommissioning the slice releases all the underlined reserved resources required to store the state or information of the VNFs and running them. This mode of algorithm incurs higher latency in getting the final response from the NSMF to NSSF and then to AMF, when the slice is OFF (decommissioned). When the slice state is OFF and the new user request arrives for the slice then this user needs to bear the additional latency of commissioning (including instantiation, configuration plus activation time) to the overall response from the NSSF which is of around 45 seconds as shown in the tables 3.1 and 3.2.

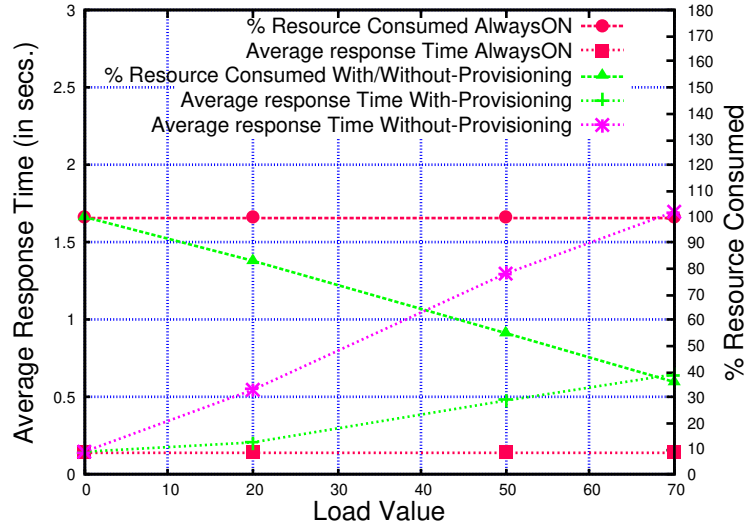


Figure 4.6: Comparing the resource consumed with average response time across different modes of slices.

4.5 Results and Analysis

Fig. 4.5 shows the average response time for requests from AMF to get back the response from NSSF. We can observe that for lightly loaded slice (70% no active user) there is high response time as many of the times the state of the slice is deactivated/decommissioned. We have observed low response time for the highly loaded slice i.e at the load value of 20% (20% time no user is present) because there is high chance of the slice to be in active state and will be allocated to the incoming user request by NSSF with no extra delay being incurred.

Fig. 4.6 shows the resources consumed versus average response time of various modes of slicing as discussed in the Algorithm 1. We can see that for the slices like (mMTC) where no user is active for quite an high amount of time, we can opt “Without Provisioning” mode of algorithm where we can reserve our resources with marginal increment in the response time for the user request. The Algorithm 2 shows the calculation of the metrics recorded for various studied load patterns. The

variable *switchingOnTime* in the algorithm takes the value of activation time (1.8 secs.) for “With Provisioning” algorithm and commissioning time (45 secs.) for “Without Provisioning” algorithm. Fig. 4.7 shows the Cumulative Distribution Function (CDF) of response times for different slice modes at various input loads. We infer that “Without Provisioning” slice mode accounts for higher response time compared to “With Provisioning” slice mode. However, as the input load reduces with respect to number of users, total response time gets higher in this slice mode. While for the slices like eMBB having some of users active majority of the time, we can opt “With Provisioning” mode of the algorithm serving the users with higher response time than “Always-ON” mode but saving some of the slice specific resources. For the slices like uRLLC we should choose “Always-ON” mode for serving the user with the minimal latency. However depending on the need of the hour, the requirements for eMBB and uRLLC could be satisfied with either of these modes of slice algorithm we proposed here. These experiments help an operator choose the type of slice based on the criteria of deployment while meeting the SLA requirements, but also benefit with effective utilization of VNF resources across varied set of slices in parallel.

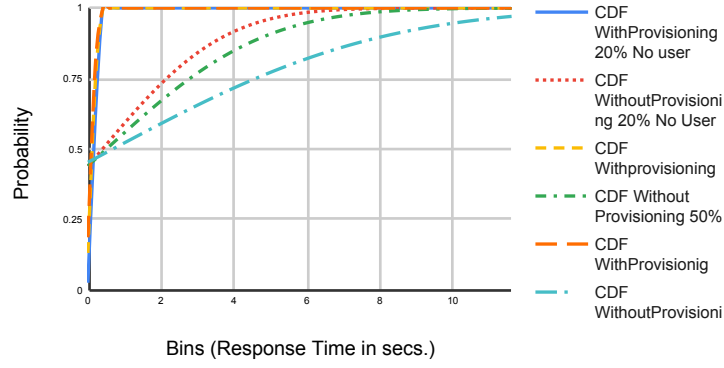


Figure 4.7: CDF of response time for With/Without provisioning algorithm on different load patterns.

4.6 Summary

Here we proposed different methods to control the activation and deactivation of slice. With static slicing method, we measured the time of different phases of slice life cycle on NSI(s). We proposed adaptive network slicing method, where we studied the slice selection traffic and used it to estimate the use of slice in order to control the slice activation and deactivation process consequently. Our techniques help an operator to decide on pre-deployment of network slices depending on the type and need of the use scenario like eMBB (Always-ON network slice), mMTC and uRLLC (Slice with provisioning and without provisioning). Further it helps in effective utilization of VNF resources by balancing them across various slices with respect to slice activation and deactivation.

Chapter 5

SERENS: Self Regulating Network Slicing in 5G for Efficient Resource Utilization

Network slicing is one of the key feature of 5G which helps in setting up the different virtual networks, each providing a specific type of service with defined QoS and SLA, running on the same physical infrastructure and in isolation with each other. Static slicing, reserving the resources for the deployed slice(s), may not be making the best usage of the available resources, rather dynamic slicing is the better suited method for CSP to maximise their revenue by dynamically adjusting the available resources to a slice as per the load on the slice. Our point of focus in this chapter is to propose scheme for selecting a best target slice from the set of candidate slice(s) to serve an incoming User Service Request (USR) in 5GC with most efficient usage of the deployed resources.

Demand of high availability and reliability on the service provision makes CSP tempting and envision to have multiple slice instances of same type [24] leading to challenging and complex situation of selecting a specific slice instance to serve USR. Hence, the 5GC network can have multiple candidate slices capable of serving the incoming USR. This leaves us with an open research problem to select one of the appropriate candidate slices to serve incoming USR, while maintaining slice SLA at the same time. 3GPP [35] defines network slice selection as the process of selecting an instance of a network slice for the incoming USR in 5GC. Control Plane (CP) and Data Plane (DP) together constitute the 5GC. The incoming USR has to be allocated a slice instance to carry forward the CP and DP centric activities on the 5GC slice. In this context, the NSSF provides information on the required network slice instance to Access Management Function (AMF). Performing dynamic slicing and getting the candidate slices for a USR urges for designing a complete framework, consisting of NSSF and NSMF along with MANO entities, which help in monitoring, analyzing the life cycle and load of multiple slice instances and finally selecting one or more of them to cater the USRs. Hence, in this chapter we proposes a novel SERENS framework for achieving the Self Regulating Network Slicing (SERENS) by performing the slice monitoring, slice analytics, and slice selection in a closed loop automation manner.

5.1 Related Work and Motivation

In the literature, sincere efforts have been made in making the decision of admissibility of a new slice request and a new user request. In [37], the authors proposed the admission control algorithm to perform the slice selection for the incoming tenant request based on its required SLA and available slice resource. While, granting the request to the least loaded candidate slice. In the same context, authors in [38] calculates the bandwidth and End-to-End (E2E) delay of the provisioned slice(s) and compared against the SLA requirement to decide on the admissibility of the new slice request. Authors in [39] predict the resource requirement of the incoming slice request and admit the new slice request only if it does not result in degradation of the new slice and the provisioned slice(s). Authors in [40] have proposed Machine Learning (ML) based model trained on the network Key Performance Indicator (KPI)s to predict the load on the network and selecting the slice type from the pre-defined slice categories like eMBB, uRLLC, and mMTC. In [41], the authors have proposed a slice admission control framework which makes the decision on the admissibility of the new slice request on the basis of the available resource capacity.

The decision of admissibility for the new slice request and user request urges the close monitoring of slices at various levels in 5GC. Additionally, no attention has been placed in literature so far, on slice selection for an incoming user request at NSSF in 5GC, after identifying the candidate slices. In continuation to our previous work [42] on controlling slice activation and deactivation in 5GC to achieve dynamic slicing, in this paper we focus on achieving self regulation of network slicing in 5GC with Closed Loop Automation (CLA) mechanism shown in Fig. 5.1. The proposed SERENS-MAS Architectural is the complete framework to facilitate self regulation of network slicing in 5GC with Monitoring, Analytics and Selection of the best slice to serve a USR. The framework helps in monitoring the provisioned slices at NSMF, by tracking the real-time status of the resource utilisation of slices and to facilitate further analytics and selection on slices at NSSF. The slice Analytics inherits Big Data Analytics (BDA) to serve the incoming USRs without incurring any additional delay by performing dynamic slicing. In slice selection mechanism we proposes a novel slice selection scheme. The proposed scheme helps CSP to meet the current demand of the service with the least number of provisioned slices as compared to the other studied slice selection schemes. Following are the key research contributions achieved by the proposed work in this chapter:

1. A framework to facilitate the self regulation of network slices called SERENS using the Closed Loop Automation (CLA) for achieving the slice monitoring, slice analytics, and slice selection in the 5GC.
2. Algorithms for slice monitoring, slice analytics, and slice selection in order to study the proposed SERENS framework to optimize the usage of underlying resources.
3. Implementation of the proposed SERENS framework in a 5G test-bed system as a proof of concept and evaluate the effectiveness of the proposed solutions.

5.2 Self Regulating Network Slicing (SERENS) Framework

In order to manage the 5G system which will have number of slices operating on the same network infrastructure, we need slice monitoring, slice analytics and slice selection in a closed loop manner.

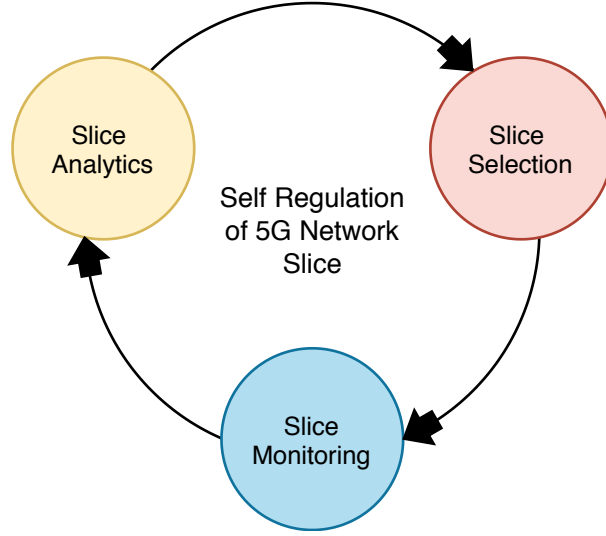


Figure 5.1: The proposed SERENS Framework.

Thus, the Fig. 5.1 shows the proposed Self Regulating Network Slicing (SERENS) framework in 5GC having the CLA mechanism. The figure depicts the close loop working of the slice monitoring, slice analytics, and slice selection functions. The monitoring includes various network and 5G KPIs such as number of users, latency, reliability, and throughput. The measurement of the slice level KPIs will be done at the Network Slice Management Function (NSMF) in 5GC. These measured KPIs are fed to the slice analytics module at NSSF of 5GC and updated with the instantaneous load information of all the available slice instances. The main functionalities of all the three components of the CLA mechanism of self regulatory network slicing are described below.

5.2.1 Slice Monitoring at NSMF

5G system needs to support stringent KPIs for latency, reliability, throughput, etc. Typically NFV MANO units have the provision of reporting the performance management metrics mainly CPU and memory of individual VNFs and Network Services (NSs), as part of NFV MANO NS performance management. However monitoring and reporting the 5GC slice and end to end slice specific KPIs through NFV MANO adds to the complexity of overall slice performance management, as MANO is transparent to 5G system and core network function specific metrics. Hence, the proposed framework proposes a tightly integrated framework of network slice monitoring and analytics, capturing both the network slice and network function specific metrics to achieve self optimization and regulation in 5GC slice selection. In this context, we measure the slice level KPIs in CLA at NSMF, to feed the analytics function at NSSF, with instantaneous load information of all the available slice instances.

The SERENS framework has the slice monitoring functionality at NSMF entity of the 5GC, responsible for monitoring all the deployed slice instances. Below mentioned are the key modules implemented in the NSFM entity performing the task of slice monitoring in the proposed framework.

- **Slice LCM Handler:** The Slice Life Cycle Management (LCM) Handler module performs the communication with the orchestrator entity over its North Bound Interface (NBI) and controls the life cycle of a network slice.

- **Slice Monitoring Module:** The Slice Monitoring module fetches the required KPIs of all the available slices such as number of registered UEs, total number of registration requests from different UEs, active UEs, and de-registered UEs at every slice instance, along with data plane throughput of the slice instance.

In the SERENS framework, upon activation, Slice Selection entity subscribes to Slice Monitoring entity to fetch the status and KPIs (CPU, memory, 5GC KPIs) of every available slice instance, at specific periodicity. We have the slice monitoring function at NSMF, responsible for monitoring all the deployed slice instances, using Prometheus Time Series Database (TSDB) [27] plugged in it along with Grafana [33] as our main slice visualization tool. Prometheus TSDB collects the various 5GC slice specific accessibility KPIs from AMF like number of registered UEs, total number of registration requests from different UEs, active UEs and de-registered UEs at every slice instance, along with data plane throughput from respective UPF of slice instance, exported by Prometheus Client at AMF, and NF specific metrics from cAdvisor [28] running in the system which hosts slice instances.

5.2.2 Slice Analytics at NSSF

The slice analytics functionality of SERENS framework resides in the Slice Analytics module of the NSSF entity. At a given point of time, this module performs data analytics functionality on the number of USRs received for better handling the future requests on the candidate slices. For this it continuously analyses the slice status and load information it receives from NSMF, to ensure controlling the slice activation and deactivation of the required slices on a need and timely basis. Hence, the slice information maintained by this module helps in getting the set of candidate slices for the incoming USR and effectively helps to achieve dynamic slicing.

5.2.3 Slice Selection at NSSF

In the SERENS framework, NSSF fetches the status and infrastructure KPIs such as CPU and memory usage along with 5GC KPIs of every available slice instance, at specific periodicity, from NSMF. NSSF performs the slice selection for the incoming USR, using the received slice specific information from NSMF. AMF is the first NF being contacted by the new incoming USR requesting for a specific slice. However AMF further contacts NSSF if it does not have sufficient information on the slice for this request. Thus, AMF, NSSF, NSMF take part in deciding the appropriate slice instance for the incoming USR.

The Slice Selection entity of the proposed SERENS framework has four main functional modules as described below.

- **Slice Request Handler:** The Slice Request Handler module of the NSSF, collects the concurrent USRs arriving to the network and assigns them to the Slice Selection Algorithm module for selecting a suitable network slice instance.
- **Slice Profile Registry:** This registry of NSSF maintains the slice instance information it obtains from NSMF, along with slice SLA and user's record of slices in a database, available to other modules of NSSF to use this stored slice information.

- **Slice LCM Controller:** This functional module of NSSF helps in managing the network slice life cycle and put the received information from Slice Selection Algorithm module into effect by informing the NSMF to trigger the corresponding change in the network slice life cycle.
- **Slice Selection Algorithm:** The Slice Selection Algorithm module of NSSF implements the slice selection scheme for making the decision of selecting the best candidate slice. It informs the Slice LCM Controller module for triggering the activation/deactivation of a new/existing slice instance dynamically, achieving dynamic slicing, while making use of the Big Data Analytics (BDA) analysis of the incoming requests.

Algorithm 3: The proposed slice selection algorithm.

```

1 Incoming Request( $R_i$ ) at time  $i$  Result: Target Slice
2  $InstanceLoad[], RequestsServed[]$ 
3  $NumberInstances[], ExcessInstances[]$ 
4  $ActiveUsers[], AptSlices[]$ 
5  $InstanceId \leftarrow 1$ 
6 if  $i > StayDuration$  then
7    $\leftarrow removeUSR(i - StayDuration)$ 
8 for  $t \leftarrow i$  to  $i + TTL - 1$  do
9    $\leftarrow ActiveUsers[t] \leftarrow ActiveUsers[t] + 1$ 
10 while  $R_i > 0$  do
11    $S \leftarrow InstanceIds$  in decreasing order of Load
12   while  $s$  in  $S$  do
13      $t \leftarrow \min(R_i, SliceCapacity - InstanceLoad[s])$ 
14      $R_i \leftarrow R_i - t$ 
15      $\leftarrow RequestsServed[i] \leftarrow (s, t)$ 
16   if  $R_i > 0$  then
17      $InstanceLoad[] \leftarrow (InstanceId + 1, 0)$ 
18      $InstanceId \leftarrow InstanceId + 1$ 
19  $AptSlices[i] \leftarrow ActiveUsers[i] \div SliceCapacity$ 
20  $NumberInstances[i] \leftarrow Size(InstanceLoad[i])$ 
21  $ExcessInstances[i] \leftarrow NumberInstances[i] - AptSlices[i]$ 

```

The proposed slice selection algorithm is shown in the Algorithm 3. This proposed Most Loaded Slice Selection (MLSS) scheme maps the incoming requests to the slice instances in decreasing order of their active users. With the aim of having the minimum number of instances running to serve the current traffic, the proposed algorithm ensures more users leave the network from the *least loaded instance*, leading to de-commissioning of an active slice instance with no load. All the provisioned slice instances are capable of serving maximum of *SliceCapacity* number of users with the required SLA which means all the active slice instances are the candidate slices for the incoming USR. The algorithm makes use of some of the following global data structures.

InstanceLoad[]: storing the information of load on a slice instance in the form of pair having *InstanceId* with the current load of the slice instance.

RequestsServed[]: stores the number of requests served by each of the slice instance at any point

of time of the simulation. It's being utilised by the function *removeUSR* at line 6, which takes one parameter donating the time of which the algorithm has to remove all active users that came at time i -*StayDuration* and correspondingly offloads their slice instances. The parameter *StayDuration* donates the TTL value for the requests of this slice.

NumberInstances[]* and *ExcessInstances[]: These data structures records the number of active slice instances at any point of time along with the the excess slice instances as compared to the ground truth value.

ActiveUsers[]: It stores the number of active users present on this particular slice may be served by more than one slice instance of same slice type at a time i .

AptSlices[]: It stores the number of apt slices required at any point of time to handle the current active users on the slice, which is *ActiveUsers[i]*. We have used a variable *InstanceId* for providing the unique instance Id to each of the activated slice instance.

The algorithm described in Algo. 3 works on very simple principle of having the least possible slices running to handles the current load of the active users by improvising the best method of user placement on the slice instances. At line 9, the active user count is incremented from time i to time $i+TTL$. Then from line 10 to 18 in the algorithm, we allocate the received R_i number of requests to the slice instances in Most Loaded Slice first manner or activate a new slice instance if all the active slice instances got completely occupied. Then, from the line 19 to 21 we have stored the metric values in the corresponding data structures for the final evaluation of the followed slice selection scheme.

5.3 Implementation of SERENS Framework in 5GC

In order to study the performance of the proposed solution, we have implemented the proposed SERENS framework in the NSMF and NSSF network entities of the 5GC, performing slice monitoring, slice analytics, and slice selection functionalities of various deployed slices. The Fig. 5.2 shows the deployed architecture for realising this proposed SERENS framework having the NSSF and NSMF performing self regulation of network slices.

The deployed architecture consists of NSMF, 5GC slices with Service Based Architecture (SBA) on Control Plane (CP) and with N3 and N6 interface on User Plane/Data Plane (UP/DP), orchestrated using the NFV MANO functions provided by OSM [26] Rel.5. Here OSM provides the NFV Orchestration (NFVO) and Virtual Network Function Management (VNFM) functionalities that supports communicating with different Virtual Infrastructure Management (VIM)s. We have picked a light weight VIM-Emulator [31] which emulates the Openstack [43] functions for VIM named as vim-emu. Vim-emu allows the execution of real network functions packaged as docker [30] containers in an emulated network topology.

NSMF utilizes OSM's North Bound Interface (NBI) taking the role of OSS/BSS, responsible for creating, deploying one or more slice instances of various types like eMBB, uRLLC, and mMTC. For simplicity we consider a slice instance formed by Session Management Function (SMF), User Plane Function (UPF) and SINK network functions for all types of slices. Here SINK represents Data Network (DN) on N6 interface with UPF. AMF, NSSF, Network Repository Function (NRF), Authentication Server Function (AUSF), Unified Data Management (UDM) are shared among the slice instances and hence form a common slice subnet. For testing purposes, we developed a light

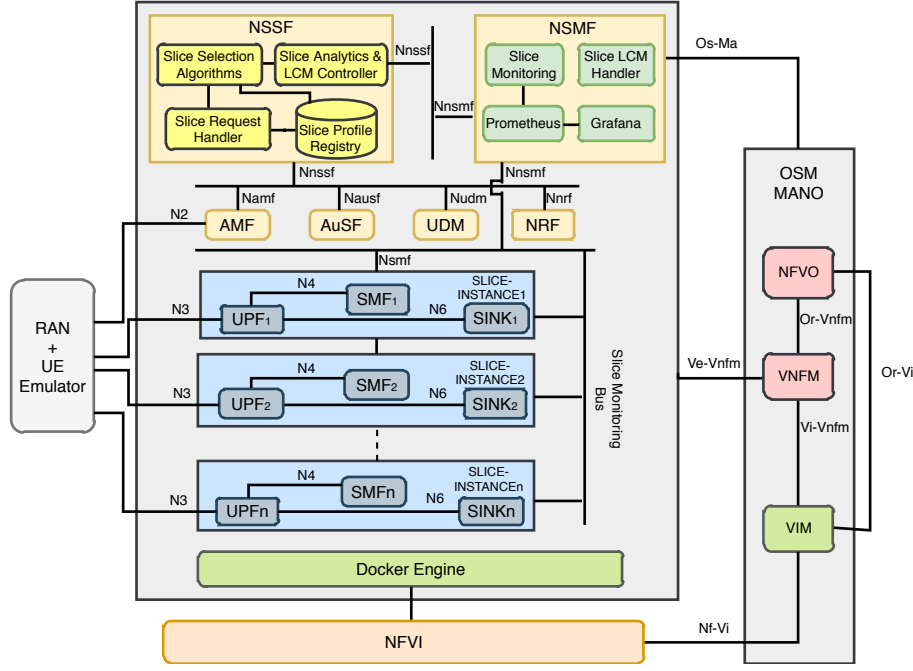


Figure 5.2: Proof of concept system realising the proposed SERENS framework.

weight RAN with embedded User Equipment (UE)'s Non Access Stratum (NAS) function terming it as RAN + UE Emulator. The complete framework on 5GC is designed and implemented as per 3GPP [35]. Service Based Interaction is realized with REST API using HTTP/2 library from nghttp2 [12]. All the network functions including RAN + UE Emulator are developed as virtualized docker containers each intended to provide functionalities as micro services such as UE registration, de-registration, and end-to-end uplink and downlink data exchange over different network slices.

On startup NSMF prepares and instantiates a common core network slice subnet Network Service Descriptor (NSD) with AMF, AUSF, UDM, and NSSF VNFDs. Upon activation, each VNF in the common slice subnet registers itself at NRF. Once this slice subnet is activated, each NF is now ready to serve the traffic of 5GC control plane at their respective Service Based Interface (SBI). NSMF then on boards and instantiates a set of required slice instances of various slice types, defined using respective Network Slice Templates (NSTs). NSMF on boot up, on boards the slice template consisting of data network slice subnet NSD with data sink Virtual Network Function Descriptor (VNFD), Once all the slice instances are active and are in run time phase of their life cycle, NSMF instantiates RAN slice subnet with RAN+UE emulator VNFD. When the RAN+UE emulator function gets active, the UEs start registering to the 5GC requesting for a specific slice service. Further, AMF contacts the NSSF to find an appropriate target slice instance.

To enable an UE to use different services provided by different network slice instances, there is an assistance information parameter named as Network Slice Selection Assistance Information (NSSAI). This parameter assists the network to allow an UE to camp on a particular instance of a network slice. An NSSAI is a collection of S-NSSAIs, where S-NSSAI stands for Single Network Slice Selection Assistance Information. Each S-NSSAI assists the network in selecting a particular NSI. An S-NSSAI comprises of a Slice/Service Type (SST) which refers to the expected network slice behaviour in terms of features and services, with optional Slice Differentiator (SD). 3GPP [35]

assigns SST value 1,2,3 to eMBB, uRLLC, and mMTC respectively.

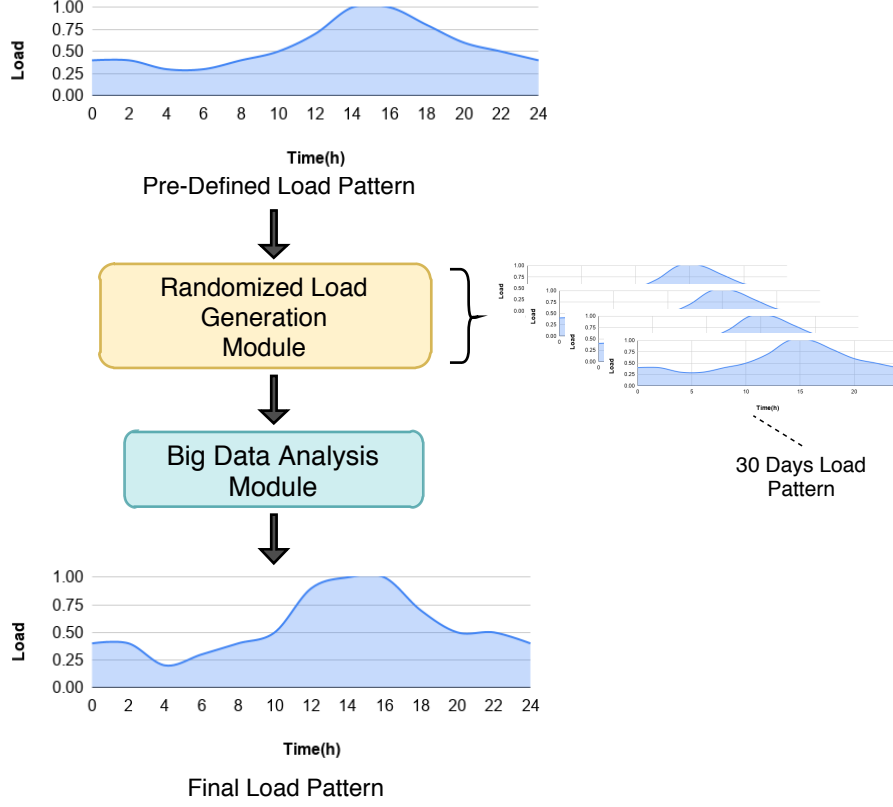


Figure 5.3: BDA enabled simulation model for data generation.

5.4 Performance Evaluation

In this section of the chapter, we have evaluated the performance of the proposed SERENS framework by building a complete prototype of the framework. The three functional entities of the framework which are Monitoring, Analytics, and Selection are being studied on the implemented prototype. For Slice Monitoring evaluation, we have collected the 5G n/w KPIs at the slice level and for Slice Analytics Slice Selection, we have provided the complete evaluation of the proposed slice selection scheme comparing the performance with the other possible slice selection schemes. For studying the slice selection schemes, we have performed the data generation for each of the slice type (mMTC, URLLC, and eMBB) and compare the performance of the schemes on each of the slice type load.

5.4.1 Synthetic Traffic Data Generation

We have used a BDA enabled simulation model for generating the data set, simulating the actual load pattern of slice requests coming to the network. Since it is difficult to get the real data set from the service providers we have used the BDA based model for generating an incoming requests load pattern as shown in Fig. 5.3. The Randomized Traffic Generator module of the model generates the load pattern of 30 days by using the input traffic profile. The BDA module of the model generates

the traffic load pattern profile by performing the average value based analysis on the provided 30 days load profile. The predicted load (in range of 0-1) from the model is scaled with a constant factor to get the numeric value of the incoming requests and observed at a time interval of every 8 minutes. Each of the incoming slice request has a Time-To-Live (TTL) field [40] which specifies the time duration for which a request stays in the network. The TTL field value is specific to the type of service being requested. We have considered the TTL value for eMBB, URLLC, and mMTC services in the range of 160-300 Secs, 150-200 Secs, and 60-100 Secs, respectively. The observed incoming requests pattern with their TTL value are used for generating the number of active users for each of the generic slice types (eMBB, URLLC, and mMTC) as shown in the first plot of the Fig. 5.6. The performance of the slice selection schemes has been studied on the aforementioned slice specific load patterns. For this performed study we have considered the slice capacity in terms of number of users (in this case, 1000 users).

5.4.2 5G System KPIs using Slice Monitoring in SERENS Framework

Slice monitoring function fetches the required metric values of all available slices at regular intervals and notifies NSSF with all this real time status of the slice instances. The slice monitoring mechanism of the proposed SERENS framework monitors the 5G systems KPIs with the deployed architecture depicted in the Fig. 5.4 and 5.5. These figures show the CPU utilization and memory utilization captured for the active slice instances by the Slice Monitoring module using Prometheus [27] at regular intervals. Here, the slice1 represented by *5g_sba_e2e_slice-1* had higher number of users using it compared to other slices and hence is showing high CPU and memory consumption. In the complete proposed SERENS framework's CLA functionality this is the first stage with the crucial slice information notified to slice analytics unit of NSSF at a specific periodicity.

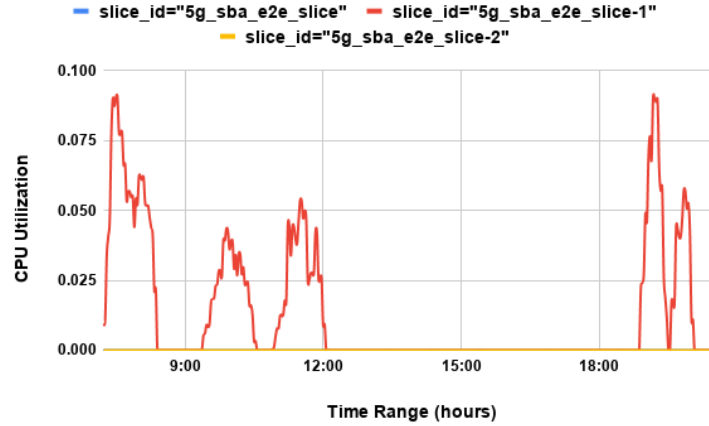


Figure 5.4: Slice level monitoring showing CPU usage of slices.

5.4.3 Performance of Slice Selection Algorithm in SERENS Framework

The slice analytics and selection functionalities of the SERENS framework is studied with the slice selection schemes running in the Slice Selection Algorithm module, while making use of the BDA based analytics performed at Slice Analytics Module of NSSF to achieve the dynamic slicing. The

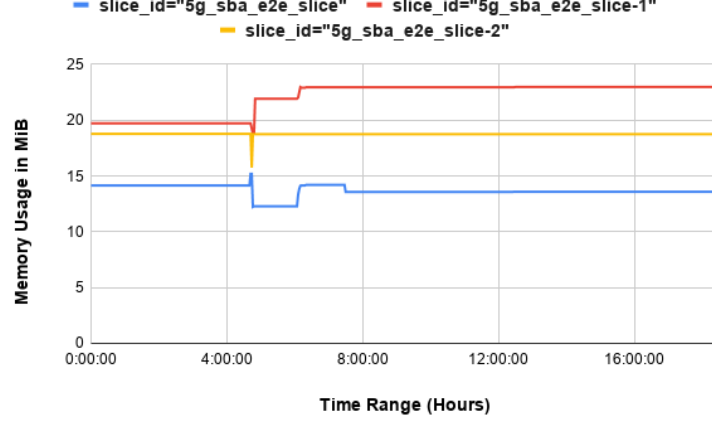


Figure 5.5: Slice level monitoring showing the memory usage of the slices.

Slice Analytics module performs average value based analysis on the number of incoming USRs, from Slice Request Handler module, to predict the incoming USR pattern. The proposed Most Loaded Slice Selection (MLSS) scheme mentioned in the Algorithm 3 is studied and compared with Least Loaded Slice Selection (LLSS) scheme which maps the incoming user requests to the slice instances in the increasing order of the number of active users and Random Slice Selection (RND) scheme which picks a random candidate slice instance for serving the incoming USR. The schemes select a candidate slice in the resource optimised manner and hence we study their performance on three slice specific load patterns which are eMBB, URLLC, and mMTC, generated using the aforementioned simulation model (Section 5.4.1). The schemes have been compared on the basis of metrics collected using the Algorithm 3. All three schemes make use of dynamic slicing to trigger the slice activation of the already provisioned slice instance(s) by making use of the predicted incoming USR pattern at Slice Analytics module. We have assumed that the predicted user request pattern is correct and can be used by the Slice Selection Algorithm module to dynamically scale up the slice instance. In case of inaccurate prediction, the incoming user request might face a delay in getting the target network slice instance.

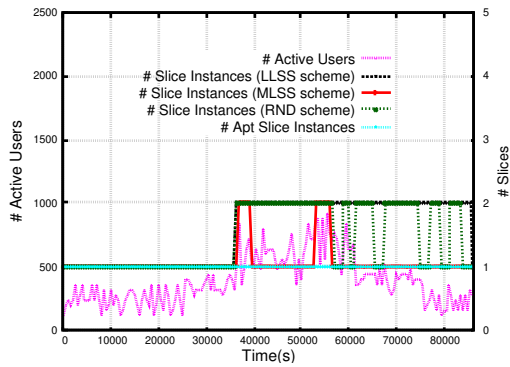


Figure 5.6: Number of active slice instances for mMTC slice type load.

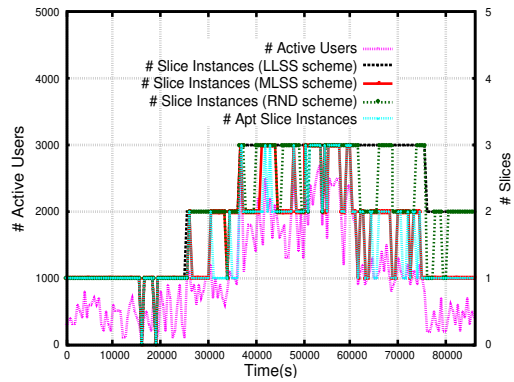


Figure 5.7: Number of active slice instances for URLLC slice type load.

The figures 5.6, 5.7, and 5.8 shows the recorded number of active slice instances running to

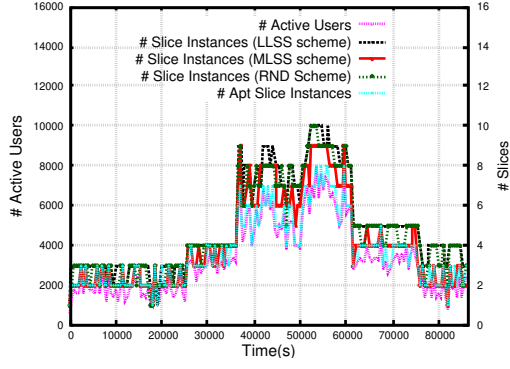


Figure 5.8: Number of active slice instances for eMBB slice type load.

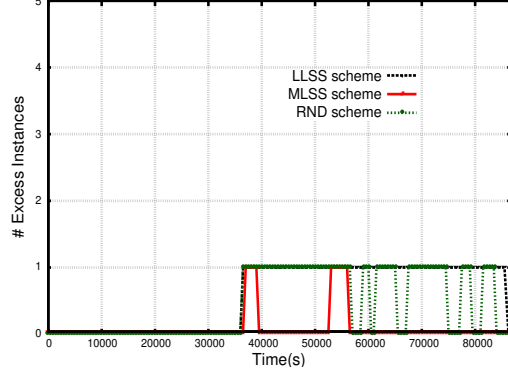


Figure 5.9: Number of excessive slice instances for mMTC slice type load.

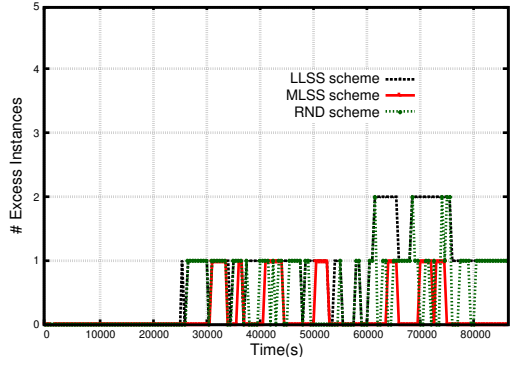


Figure 5.10: Number of excessive slice instances for URLLC slice type load.

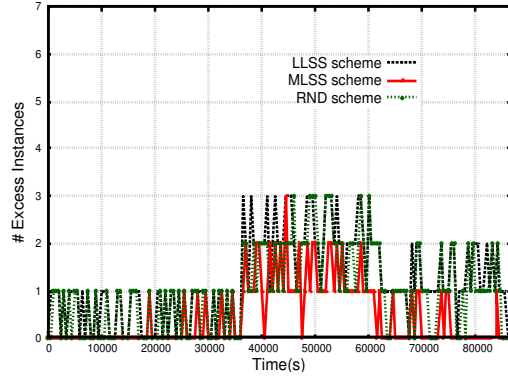


Figure 5.11: Number of excessive slice instances for eMBB slice type load.

serve the active users on the each of the studied slice specific loads respectively. The plots shows the observed metric value for the slice specific load patterns. Due to higher TTL value for the incoming eMBB slice request, there are higher number of active users on the eMBB slice as compared to URLLC and mMTC slices at a given point of time. Presence of higher number of active users demands the requirement of running higher slice instances, in order to meet the requirements of the user traffic. We can observe that the proposed MLSS scheme has less number of active slice instances than the other two schemes (LLSS and RND) because MLSS scheme places the incoming USR in the resource optimised manner, allowing SPs to decommission an active slice instance (having no active user) and make better use of the available resources. On the other hand, LLSS scheme and RND scheme are observed to run more number of slice instances as they possess poor slice selection techniques from the active slice instances.

The figures 5.9, 5.10, and 5.11 shows the number of excessive slice instances running for each of the studied slice specific load pattern, as compared to the ground truth value, which represents the minimum slice instances needed to support the active users on the slice. We can observe that the proposed MLSS scheme is using excessive slice instances for less amount of time compared to other two schemes. LLSS scheme and RND scheme used an excessive slice instance for 45% and 30% of the time, respectively for mMTC and URLLC slices, and 60% and 50% of the time respectively for eMBB slices as compared to the proposed scheme. For the mMTC slices, we observe that the proposed

MLSS scheme runs an excessive slice for just about 6% of the time as compared to the ground truth value, while the same value goes to 12% and 17% for URLLC and eMBB slices, respectively.

The figures 5.12, 5.13, 5.14 shows the Cumulative Distribution Function (CDF) for excessive slice instances across different slices. We can observe that the proposed MLSS scheme shows high probability of having less excessive slice instances running across all the slice types as compared to LLSS and RND schemes. Also, we can observe the right shift in the curves as we go to higher loaded slice patterns because on the higher loaded slices the algorithms tend to run higher number of excessive slice instances as compared to the less loaded slice instances.

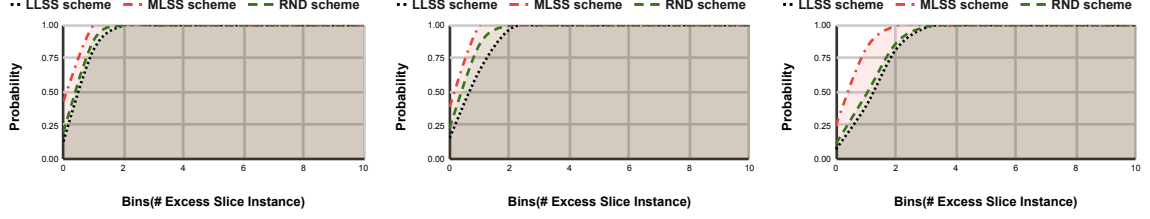


Figure 5.12: CDF plot of excessive slice instances for mMTC slice type load.

Figure 5.13: CDF plot of excessive slice instances for URLLC slice type load.

Figure 5.14: CDF plot of excessive slice instances for eMBB slice type load.

5.5 Summary

In this research work, we have successfully proposed a Self Regulating Network Slicing (SERENS) framework performing the Monitoring, Analytics and Selection of the slice instances in a closed loop automation to serve incoming user requests for efficient resource utilization. Our proposed SERENS framework helps the service provider to monitor the 5GC function's KPIs at the slice level and use it to select the candidate slices for the incoming user requests. Our study has shown that the proposed MLSS scheme outperforms the LLSS and RND schemes. This proposed MLSS scheme makes the best use of the available resources by handling the incoming slice traffic with minimum slice instances and thus, benefiting the service provider in terms of making higher revenue with the available resources. In future, we plan to introduce the Machine Learning (ML) based model(s) for performing the task of predicting the number of slice instances required at a given point of time and evaluate the performance of the model(s) with our proposed MLSS scheme.

Chapter 6

Conclusion and Future Work

In this thesis work, we have successfully studied various aspects of network slicing in 5G Core network. We have realised the REST enabled SBA of 5G Core using HTTP2 based API provided by the nhttp library. We have realised NRF using Consul. We have demonstrated the multi-site deployment of a network slice and how it is being beneficial to offload highly loaded network functions to a separate server for avoiding the computational bottleneck. The main emphasis of this thesis work is on the Algorithmic aspects for controlling various network slicing related activities. We have provided the algorithm for performing adaptive network slicing in 5G core network. In adaptive network slicing, we dynamically controls the states of the network slice in order to have the best usage of the resources and on the other hand does not increases response time for the incoming user request. The proposed Network Slicing with Provisioning methods helps in saving upto 70% of the resources while making the marginal increase in the response time. Also, we have shown that the random(RND) and least loaded slice selection (LLSS) schemes, selecting target slice from the set of candidate slices, does not make good use of the resources. Our proposed slice selection scheme (MLSS) saves upto 60% of the resources in terms of running extra slice instances. Also, MLSS schemes makes the best use of the resources to serve the slice traffic and runs extra slice instance for just about 6% of the time as compared to the ground truth value.

Following are the points which can be work upon in future in continuation of this thesis work:

1. **Dynamic scaling of the resources:** Managing the resources of the VNFs as per the need of the current load and providing more resources to the highly loaded NFs while using ML technique for traffic prediction.
2. **ML Models for best slice prediction:** ML model trained on some of the network metrics and future load can be used to select the best target slice and can give better results then the proposed MLSS scheme.
3. **RL based modelling for adaptive slicing:** Reinforcement Learning can be used to train model for penalising the SLA violation while performing adaptive network slicing to save the resources.
4. **Inter/Intra slice resource congestion Model:** The proposed slice monitoring framework can further be utilized for performing resource congestion in between the NFs of a slice or the NSIs.

Publications

1. Shwetha Vittal, **Mohit Kumar**, and Antony Franklin, "Adaptive Network Slicing with Multi-Site Deployment in 5G Core Networks", in press of 6th IEEE Conference on Network Softwarization (NETSOFT) 2020, Ghent, Belgium.
2. Madhura A, **Mohit Kumar**, and Bheemarjuna Reddy Tamma, "ONVM-5G: a framework for realization of 5G core in a box using DPDK ", CSI Transactions on ICT, May, 2020."
3. **Mohit Kumar Singh**, Shwetha Vittal, and Antony Franklin, "SERENS: Self Realising Network Slicing in 5G for Efficient Resource Utilization", under review in 3rd IEEE Conference on 5G World Forum 2020, Virtual Event.

References

- [1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi. 5g wireless network slicing for embb, urllc, and mmhc: A communication-theoretic view. *IEEE Access*, 6:55765–55779, September 2018.
- [2] 3GPP. Technical specification group services and system aspects; general packet radio service (gprs) enhancements for evolved universal terrestrial radio access network (e-utran) access. Technical Report TS 23.401, 3GPP, 2017.
- [3] Adeppady M., Singh M.K., and Tamma B.R. A framework for realization of 5g core in a box using dpdk. In *CSIT Volume 8*, pages 77–84, 2020.
- [4] C. Bouras, A. Kollia, and A. Papazois. Sdn nfv in 5g: Advancements and challenges. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 107–111, 2017.
- [5] Dr. Bhargavi Goswami. Revolution in existing network under the influence of software defined networks (sdn). 03 2017.
- [6] Yacine Rebahi, Majid Ghamsi, Nicolas Herbaut, Daniel Negru, Paolo Comi, Paolo Crosta, Pascal Lorenz, Evangelos Pallis, and Evangelos Markakis. Virtual network functions deployment between business expectations and technical challenges: The t-nova approach. *Recent Advances in Communications and Networking Technology*, 5:49–64, 12 2016.
- [7] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung. Network Slicing Based 5G and Future mobile networks: Mobility, resource Management, and Challenges. *IEEE Communications Magazine*, 55(8):138–145, Aug 2017.
- [8] Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies and solutions. *IEEE Communications Surveys and Tutorials*, Vol.PP, NÂ°99, March 2018, ISSN: 1553-877X, 03 2018.
- [9] Chia-Yu Chang. Cloudification and slicing in 5g radio access network. (cloudification et découpage des réseaux d’accès radio de cinquième génération). 2018.
- [10] 3GPP. System architecture for the 5g system. Technical Report TS 23.501, section 6.2, 2018.
- [11] 3GPP. Procedures for the 5G System. Technical Report TS 23.502, 3GPP, 2018.
- [12] Nghttp2: HTTP/2 C Library. <https://nghttp2.org/>.

- [13] Consul. <https://www.consul.io>.
- [14] iperf3. <https://iperf.fr/>, 2014.
- [15] 3GPP. Architecture enhancements for 5G System (5GS) to support network data analytics services. Technical Report TS 23.288, 3GPP, 2019.
- [16] 3GPP. Study of Enablers for Network Automation for 5G. Technical Report TS 23.791, 3GPP, 2019.
- [17] E. Pateromichelakis, F. Moggio, C. Mannweiler, P. Arnold, M. Shariat, M. Einhaus, Q. Wei, Ö. Bulakci, and A. De Domenico. End-to-end data analytics framework for 5g architecture. *IEEE Access*, 7:40295–40312, 2019.
- [18] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti. Data analytics in the 5g radio access network and its applicability to fixed wireless access. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–6, April 2019.
- [19] Sokratis Barmounakis, Alexandros Kaloxylas, Panagiotis Spapis, Chan Zhou, Panagis Magdalinos, and Nancy Alonistioti. Data analytics for 5g networks: A complete framework for network access selection and traffic steering. 11 2018.
- [20] C. Rotter, J. Illés, G. Nyíri, L. Farkas, G. Csatari, and G. Huszty. Telecom strategies for service discovery in microservice environments. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 214–218, March 2017.
- [21] A. Jain, Sadagopan N S, S. K. Lohani, and M. Vutukuru. A comparison of sdn and nfv for re-designing the lte packet core. In *2016 IEEE Conference on NFV-SDN*, pages 74–80, Nov 2016.
- [22] José Ordóñez, Oscar Adamuz-Hinojosa, Pablo Ameigeiras, P. Muñoz, Jorge Ramos, Jesus Folgueira, and Diego Lopez. The creation phase in network slicing: From a service order to an operative network slice. 07 2018.
- [23] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira. Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87, May 2017.
- [24] 5g pagoda end-to-end network slice. https://5g-pagoda.aalto.fi/assets/demo/attachement/delivrables/5G!Pagoda-D4.3-End-to-end%20Network%20slice_1.0.pdf.
- [25] 3GPP. Management and Orchestration; Architecture Framework, 2018.
- [26] OSM. https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE.
- [27] Prometheus. <http://prometheus.io>.
- [28] cAdvisor. <https://github.com/google/cadvisor>.
- [29] node_exporter. https://github.com/prometheus/node_exporter.
- [30] Docker. <https://www.docker.com>.

- [31] H. Karl M. Peuster and S. v. Rossem. Medicine: Rapid Prototyping of Production-Ready Network Services in Multi-PoP Environments. In *IEEE Conference on NFV-SDN*. IEEE, 2016.
- [32] Prometheus C++ Client. <https://github.com/jupp0r/prometheus-cpp>.
- [33] Grafana. <https://grafana.com/>.
- [34] 3GPP. Study on Management and Orchestration of Network Slicing for Next Generation Network. Technical Report TR 28.801, 3GPP, 2018.
- [35] 3GPP. System Architecture for the 5G System. Technical Report TS 23.501, 3GPP, 2018.
- [36] 3GPP. Network Data Analytics Services. Technical Report TS 29.520, 3GPP, 2018.
- [37] A. Kammoun, N. Tabbane, G. Diaz, and N. Achir. Admission control algorithm for network slicing management in sdn-nfv environment. In *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, pages 1–6, May 2018.
- [38] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin. Resource allocation with admission control for gbr and delay qos in 5g network slices. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 213–220, Jan 2020.
- [39] B. Han, A. DeDomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H. D. Schotten. Admission and congestion control for 5g network slicing. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, Oct 2018.
- [40] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard. Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5g networks. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0762–0767, Oct 2019.
- [41] M. R. Raza, A. Rostami, L. Wosinska, and P. Monti. A slice admission policy based on big data analytics for multi-tenant 5g networks. *Journal of Lightwave Technology*, 37(7):1690–1697, April 2019.
- [42] Shwetha Vittal, Mohit Kumar, and Antony Franklin A. Adaptive Network Slicing with Multi-Site Deployment in 5G Core Networks (in press). In *IEEE Conference on Network Softwarization (NETSOFT)*, 2020.
- [43] Openstack. <https://www.openstack.org>.