Mobile Computing

WS 2022-23

Provisioning of a 5G Stand-Alone system using Open5GS

With

srsRAN & UERANSIM

by Armin Lehmann

Masters in Engineering
Information Technology

Project Members

Gurunag Sai Udaykumar *1387227*
Harish Palanivel *1392283*
Chinmaya Nithin Dimmiti *1386312*

Table of Contents **Page**

Introduction

Open5GS

It is an open-source project that provides 5G mobile packet core network functionalities with an AGPLv3 or commercial license. It can be used to build private 5G telecom networks by individuals or telecom network operators.

Open5GS contains a series of software components and network functions that implement the 5G NSA and 5G SA core functions.

5G Core Network (5GC)

5G core network (5GC) holds a key role in realizing the full potential of 5G services. Without 5GC, fully-fledged NR services cannot be obtained. 5G Core (5GC) is the heart of a 5G network, controlling data and control plane operations. The 5G core aggregates data traffic, communicates with UE, delivers essential network services and provides extra layers of security, among other functions. The 5G core is responsible for a variety of functions within the mobile network that makes communication possible. Authentication, authorization, and data and policy management are just a few services that run on a 5G core.

5G SA Core

The 5G SA core works in a different way, it uses a Service Based Architecture (SBI). Control plane functions are configured to register with the NRF, and the NRF then helps them discover the other core functions.

The 5G SA core user plane is much simpler, as it only contains a single function. The UPF carries user data packets between the gNB and the external WAN. It connects back to the SMF too.

With the exception of the SMF and UPF, all config files for the 5G SA core functions only contain the function's IP bind addresses/ local Interface names and the IP address/ DNS name of the NRF.

5G SA uses a 5G core to manage connectivity and user authentication

Benefits of 5G SA

- MNOs can launch new enterprise 5G services such as smart cities, and smart factories
- It is fully virtualized, cloud-native architecture (CNA), which introduces new ways to develop, deploy and manage services
- The architecture enables end to end slicing to logically separate services
- Automation drives up efficiencies while driving down the cost of operating the networks.
- By standardizing on a cloud-native approach, MNOs can also rely on best of breed innovation from both vendors and the open-source communities
- By choosing a cloud-native microservices-based architecture, MNOs can also decide on a variety of deployment models such as on-prem private cloud, public cloud, or hybrid to meet their business objectives.

srsRAN

srsRAN is a free and open source 4G and 5G software radio suite. Featuring both UE and eNodeB/ gNodeB applications, srsRAN can be used with third-party core network solutions to build complete end-to-end mobile wireless networks.

Network Slicing

Network slicing overlays multiple virtual networks on top of a shared network domain, that is , a set of shared network and computing resources. Network slicing is used most often in discussion of 5G networks, in part because the 5G specification calls for network slicing as a fundamental capability.

Components of 5G Core

5G Core is considered as the brains of the monile network, as the core manages many different requests and services. The key services of the 5G core provides are below:

User Plane Function (UPF)

The User Plane Function (UPF) represents the data plane evolution of a Control and User Plane Separation (CUPS) strategy, which is a fundamental component of the 3GPP 5G core network(5GC).

The UPF plays the most critical role in the process of data transfer. It provides the interconnect point between the mobile infrastructure and the Data Network (DN), i.e., encapsulation and decapsulation of GTP-U.

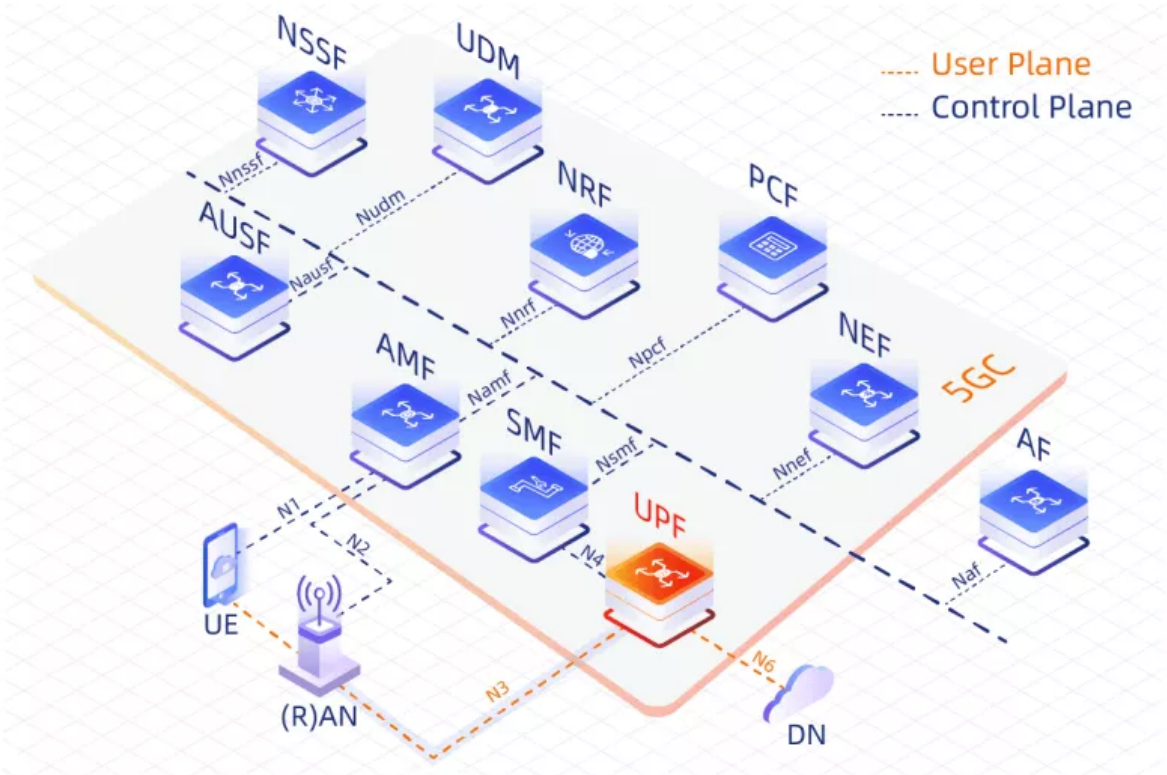

Figure 1. 5G network

Session Management Function (SMF)

SMF is a fundamental element of the 5G Service-Based architecture (SBA). The SMF keeps trace of PDU sessions and QoS flows in the 5GC for UEs and make sure their states and status are in sync between network functions in control and user planes. It also receives PCC (Policy and Charging Control) rules from PCF (Policy Charging Function) and convert PCC rules into SDF templates, QoS profiles and QoS rules for UPF, gNB and UE respectively for QoS flows establishment, modification and release.

Access and Mobility Management Function (AMF)

AMF terminates the control plane of different access networks onto the 5G core network(5GC) and control which UEs can access the 5GC to exchange traffic with DNs. It also manages the mobility of UEs when they roam from one gNB to another for session continuity, whenever possible.

Unified Data Management (UDM) / Authentication Server Function (AUSF)

UDM: UDM us a centralized way to process network user data in 5G through Nudm interfaces to provide services for AMF, SMF, SMSF, AUSF, NEF and GMLC. It also provides services such as authorization of accessing, registration, uninterrupted services.

AUSF: As a major part of 5GC to facilitate security processes, AUSF performs the authentication function of identifying UEs and storing authentication keys.

Policy Control Function (PCF)

The 5GC PCF performs the same function as the PCRF in 4G networks.

- Provides policy rules for control plane functions. This includes network slicing, roaming and mobility management.
- Accesses subscription information for policy decisions taken by the UDR.
- Supports the new 5G QoS policy and charging control functions.

Network Slice Selection function (NSSF)

Network slicing is a key native capability of 5G that can maximize the performance of communication networks and reduce network construction and operation and maintenance costs, and it has become an industry consensus that slicing is capability and slicing is product in the 5G era.

The core network, as a key anchor point for network differentiation and operation, has emerged as a more important network in the 5G area, providing the ability to deploy network functions.

Network Repository Function (NRF)

NRF, one of the network functions of the 5G core network (5GC0. It supports the service discovery feature, which receives NF discovery requests from NF instances and provides information about the discovered NF instance (discovered) to another NF instance. Registration information includes NF type, address, service list, etc.

Network Exposure Function (NEF)

NEF, located between the 5G core network and external third-party application functionaries (and possibly some internal Afs), is responsible for managing the external open network data, and all external applications that want to access the internal data of the 5G core must pass through the NEF.

5G Design and Planning Considerations

Why 5G network? What is the use and purpose when we have good 4G network? 4G network supports almost all the needs like internet calling, video calling, fast streaming. What does 5G network provides?

The design is considered to support highly complex applications. For example, there is no one-size-fits all approach, the range of applications requires data to travel distances, large data volumes, etc. These use cases are derived or provided by 3GPP. Many more of use cases are found under 3GPP description. So 5G architecture must support low, mid and high-band spectrum – from licensed, shared and private sources – to deliver the full 5G vision.

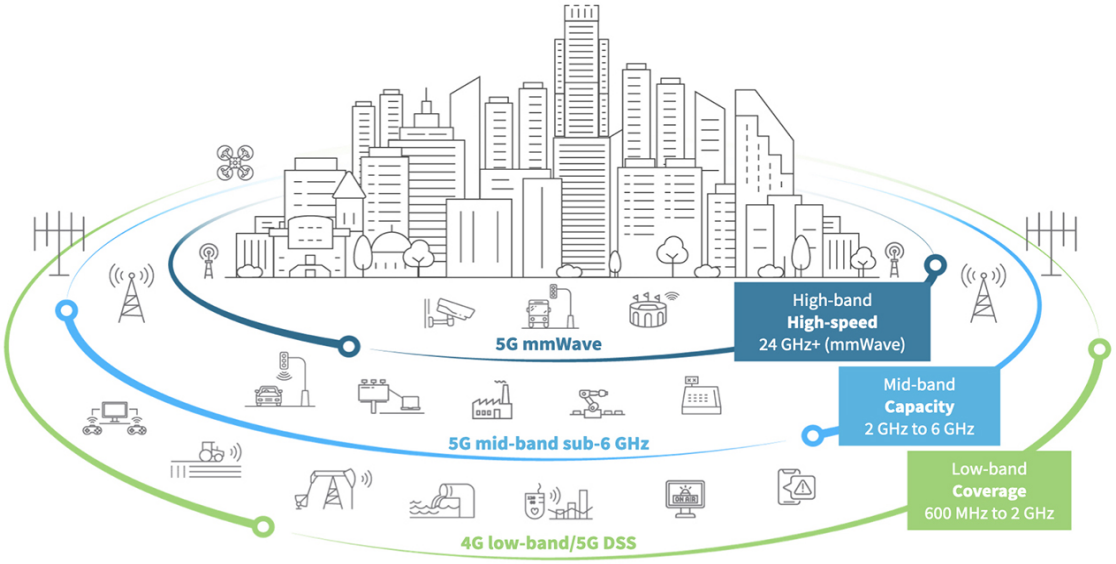

Figure 2. 5G Design & Planning

5G is architected to run on radio frequencies ranging from sub 1 Ghz to extremely high frequencies called 'millimeter wave' (or mmWave). The lower the frequency, the farther the signal can travel. The higher the frequency, the more data it can carry.

There are three frequency bands at the core of 5G networks:

- 5G high-band (mmWave) delivers the highest frequencies of 5G. These range from 24 GHz to approximately 100 GHz. Because high frequencies cannot easily move through obstacles, high-band 5G is short range by nature. Moreover, mmWave coverage is limited and requires more cellular infrastructure.
- 5G mid-band operates in the 2-6 GHz range and provides a capacity layer for urban and suburban areas. This frequency band has peak rates in the hundreds of Mbps.
- 5G low-band operates below 2 GHz and provides a broad coverage. This band uses spectrum that is available and in use today for 4G LTE, essentially providing an LTE 5g architecture for 5G devices that are ready now. Performance of low-band 5G is therefore similar to 4G LTE, and supports use for 5G devices on the market today.

Requirements

The main purpose is to deal with 5G core, including RAN (Radio Access Network), Data Networks (DN) and Network Slicing. The project must be implementated based on virtualization technology such as VirtualBox, LXC, or a Docker. Also in practical, we need to implement the design principle of network softwarization i.e, the use of Network Function Virtualisation (NFVs) and Software Defined Networking (SDN) including network slicing concepts.

The requirement has been explained in detail below:
Setting up a 5G-SA (stand-alone) system using Open5GS in accordance with the specifications given.

1. Open5GS with srsRAN

A 5G network with 5GC components configured to connect with srsRAN for 5G-RAN to establish a PDU session. Since there is no support on network slicing from srsRAN, the requirement has been changed and hence a 5GC with srsRAN with no network slicing has been implemented.

2. Open5GS with UERANSIM

The following specific requirements must be adhered to when implementing the project network:
- UERANSIM for 5G-RAN
- Control Plane: Multiple SMF (one for each APN/DNN)
- User Plane: Per-SST UPF (for each SST)
- Additionally: Integration of TAC/TAI for nearby UPF

The network slices configuration has explained in the table below

SST	SD	APN/DNN	Description
1	1	Internet	Access to a file sharing platform and streaming platform for Tenant Group A
1	2	Internet2	Access to a file sharing platform and streaming platform for Tenant Group B
2	1	Voip	Access to SIP call server for Tenant Group A
2	2	Voip2	Access to SIP call server for Tenant Group B

These slices are to be maintained within a data network showing the below mentioned service functionality
1. Streaming Platform using OwnCast (Owncast, 2022)
2. File Sharing Platform using NextCloud (NextCloud, 2022)
3. SIP Call Server using Kamailio (Kamailio, 2022)

Project Planning

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1: Planning and Research Survey				
Mobile Wireless Communication Technology Journey(0G, 1G, 2G, 3G, 4G, 5G)	Chinmaya Nithin	100%	12/14/22	12/16/22
Research about Open5GS, Network Slicing, SBI, srsRAN	Harish	100%	12/15/22	12/20/22
Interpretation and Analysis on Terminologies (Network slicing, VNF, NF srsRAN, SA, NSA etc)	Gurunag Sai	100%	12/14/22	12/17/22
Difference between SA vs NSA	Chinmaya Nithin	100%	12/17/22	12/21/22
WINTER HOLIDAYS			12/24/22	1/3/23
Phase 2: 5G SA Architechture, Analysis and Tools				
Design of 5G SA Architechture	Chinmaya Nithin	100%	1/4/23	1/12/23
Specifications of CPF and UPF	Harish	100%	1/5/23	1/12/23
VM Initialisation and Tools Installation (Open5GC, srsRAN etc)	Gurunag Sai	100%	1/5/23	1/13/23
Phase 3: Configuration and Implementation of Network Frameworks				
Installing open5gs and confirguring 5GC Components	Gurunag Sai	100%	1/13/23	1/18/23
Installing srsRAN and connecting to 5GC	Harish	100%	1/18/23	1/31/23
Connecting to Data Network & Testing	Chinmaya Nithin	100%	1/31/23	2/6/23
Phase 4: Implementation of Open5gs-5GC-UERANISM				
Initializing different VMs for specific components 1. Open5gs - Control Plane 2. UPF - Uplane 1 3. UPF - Uplane 2 4. UE/GNB	Gurunag Sai	100%	2/7/23	2/9/23
Configuring AMF,SMF of VM1 Configuring UPF1,UPF2	Harish	100%	2/9/23	2/11/23
Configuring UE and GNB	Chinmaya Nithin	100%	2/10/23	2/11/23
Connecting Open5gs - Control Plane to UPF1 and UPF2	Gurunag Sai	100%	2/12/23	2/15/23
Connecting UE/RAN to Open5gs	Harish	100%	2/12/23	2/20/23
Implementation for Service Functionality: 1. File Sharing Platform 2. Streaming Platform 3. SIP call Server	All	73%	2/15/23	2/22/23
Testing and Refactoring	All	100%	1/11/23	2/22/23
Phase 5: Testing, Refactoring and Deployment				
Final Testing and Refactoring	All	100%	2/22/23	2/23/23
Phase 6: Documentation				
Report .pdf/.docx	All	100%	2/21/23	2/28/23
Presentation .ppt	All	100%	2/21/23	2/28/23
Presentaion Video .mp4	All	100%	2/27/23	2/28/23

Implementation

Open5GS with srsRAN

Network Architecture :

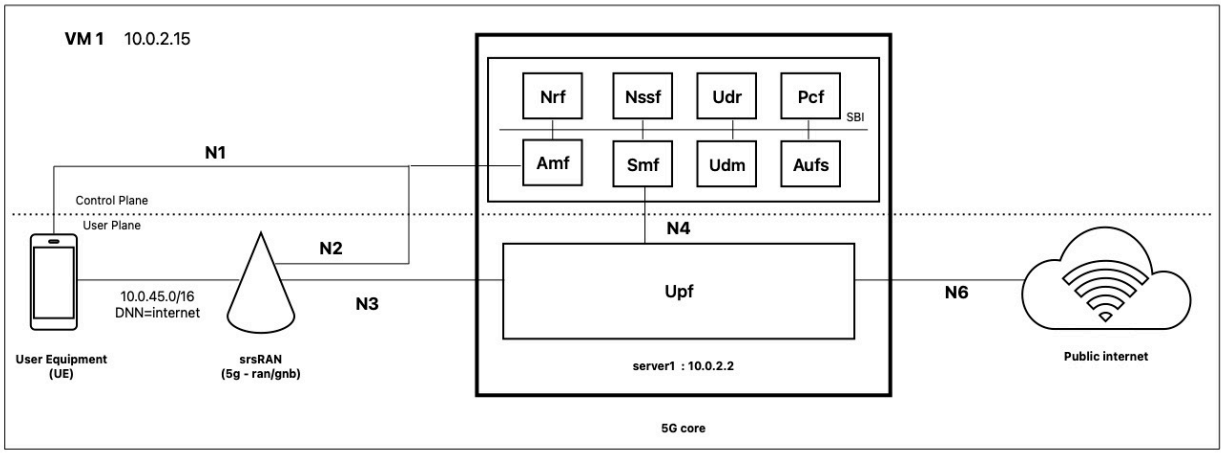

Figure 3. Open5GS with srsRAN Architecture

The above architecture has been implemented by changing the configuration files of Open5GS 5C and srsRAN.

Configuring changes are mainly dealt on AMF, UPF, SMF components and srsRAN (enb/gnb & ue). The set up of 5GC is done by cloning the Open5GS from the source which has inbuilt configuration files of the all components. As mentioned we focus on specific components and the changes are made as follows.

Open5GS 5GC components changes - AMF, UPF, SMF

AMF configuration	SMF Configuration	UPF Configuration
amf: …. ngap: - addr: 127.0.0.2 metrics: addr: 127.0.0.5 port: 9090 …. plmn_support: - plmn_id: mcc: 901 mnc: 70 s_nssai: - sst: 1	smf: sbi: - addr: 127.0.0.4 port: 7777 pfcp: - addr: 127.0.0.4 gtpc: - addr: 127.0.0.4 - addr: ::1 gtpu: - addr: 127.0.0.4 - addr: ::1 metrics: addr: 127.0.0.4 port: 9090 subnet: - addr: 10.45.0.1/16 - addr: 2001:db8:cafe::1/48 ….	upf: pfcp: - addr: 127.0.0.7 gtpu: - addr: 127.0.0.2 subnet: - addr: 10.45.0.1/16 - addr: 2001:db8:cafe::1/48 metrics: - addr: 127.0.0.7 port: 9090

srsRAN configuration changes

ENB Configuration	UE Configuration	RRC Configuration

[enb]
….
mcc = 901
mnc = 70
mme_addr = 127.0.0.2
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
…..
	[rf]
	…..
	device_name = zmq
	device_args = tx_port=tcp://
	*:2001,rx_port=tcp://
	localhost:2000,id=ue,base_srate=11.52e6
	….
	[usim]
	mode = soft
	algo = milenage
	opc =
	000102030405060708090a0b0c0d0e0f
	k =
	00112233445566778899aabbccddeeff
	imsi = 901700123456789
	imei = 353490069873319
	
	[nas]
	apn = internet
	apn_protocol = ipv4v6
	
	[gw]
	netns = ue1
		nr_cell_list =
		(
		{
		rf_port = 0;
		cell_id = 1;
		root_seq_idx = 1;
		tac = 7;
		pci = 500;
		dl_arfcn = 368500;
		coreset0_idx = 6;
		band = 3;
		}
		);

Once the configuration changes are made on Open5GS and srsRAN , connect the two by performing network settings like IPv4/IPv6 forwarding and adding NAT rule for TUNnel interface for the user equipment.

Open5GS with UERANSIM

Network Architecture :

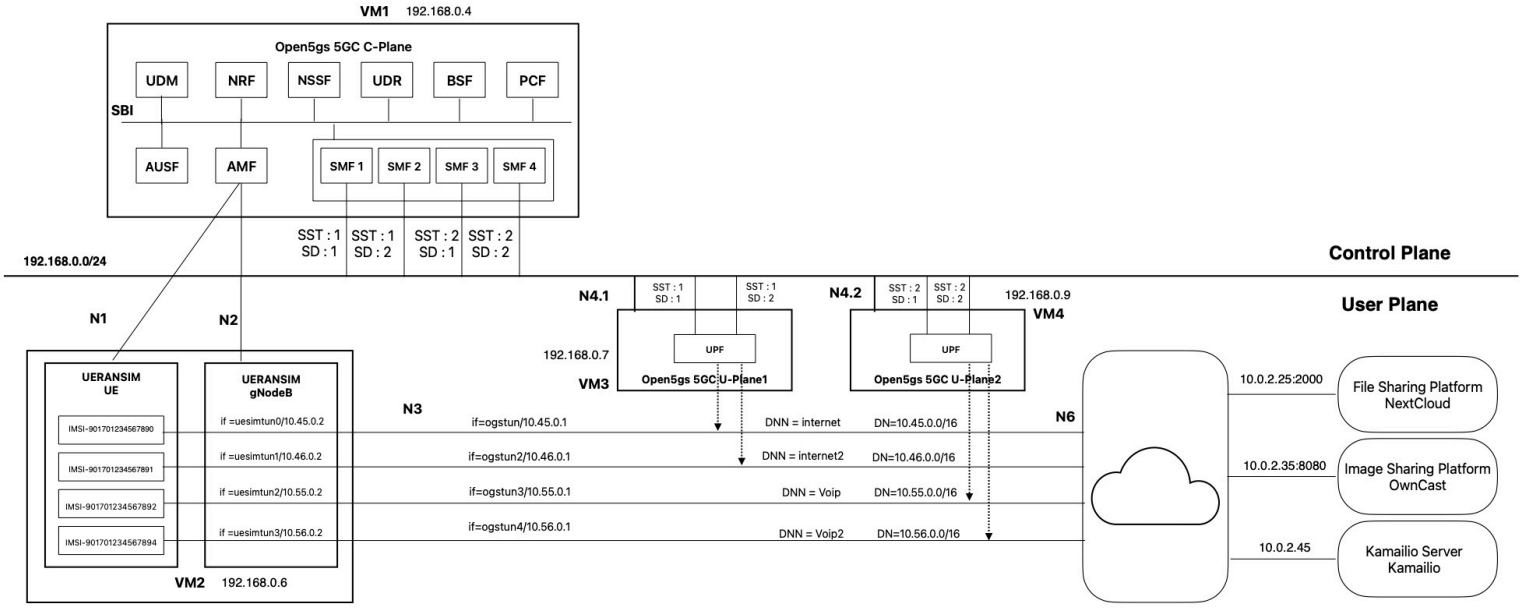

Figure 4. Open5GS with UERANSIM architecture

The above architecture has been implemented by changing the configuration files of Open5GS 5C and UERANSIM.

From the architecture, it is clear that network slicing has been implemented and the configuration changes are at multiple components.
Starting with VM implementation. The below table explain the requirement and the division of the network components in each VMs

VM #	SW & Role	IP address	OS	Memory (Min)	HDD (Min)
VM1	Open5GS 5GC C-Plane	192.168.0.4	Ubuntu 22.04	1GB	20GB
VM2	UE&RAN (gNodeB & UEs)	192.168.0.6	Ubuntu 22.04	1GB	10GB
VM3	Open5GS 5GC U-Plane1	192.168.0.7	Ubuntu 22.04	1GB	20GB
VM4	Open5GS 5GC U-Plane2	192.168.0.9	Ubuntu 22.04	1GB	20GB

Configurations on Open5GS 5GC C-plane (VM1)

amf.yaml

```
amf:
  sbi:
    - addr: 127.0.0.5
      port: 7777
  ngap:
    - addr: 192.168.0.4
  metrics:
      addr: 127.0.0.5
      port: 9090
  guami:
    - plmn_id:
        mcc: 901
        mnc: 70
      amf_id:
        region: 2
        set: 1
  tai:
    - plmn_id:
        mcc: 901
        mnc: 70
      tac: 1
  plmn_support:
    - plmn_id:
        mcc: 901
        mnc: 70
      s_nssai:
        - sst: 1
    - plmn_id:
        mcc: 901
        mnc: 70
      s_nssai:
        - sst: 1
          sd: 000001
    - plmn_id:
        mcc: 901
        mnc: 70
      s_nssai:
        - sst: 1
          sd: 000002
    - plmn_id:
        mcc: 901
        mnc: 70
      s_nssai:
        - sst: 2
          sd: 000001
    - plmn_id:
        mcc: 901
        mnc: 70
      s_nssai:
        - sst: 2
          sd: 000002
```

nssf.yaml

```
nssf:
  sbi:
    - addr: 127.0.0.14
      port: 7777
  nsi:
    - addr: 127.0.0.10
      port: 7777
      s_nssai:
        sst: 1
        sd: 000001
    - addr: 127.0.0.10
      port: 7777
      s_nssai:
        sst: 1
        sd: 000002
    - addr: 127.0.0.10
      port: 7777
      s_nssai:
        sst: 2
        sd: 000001
    - addr: 127.0.0.10
      port: 7777
      s_nssai:
        sst: 2
        sd: 000002
```

As per the requirement, we have implemented multiple SMFs. Hence we have created four SMFs in the same place.

smf.yaml	smf2.yaml	smf3.yaml	smf4.yaml
smf: sbi: - addr: 127.0.0.31 port: 7777 pfcp: - addr: 192.168.0.21 gtpc: - addr: 127.0.0.31 gtpu: - addr: 192.168.0.21 metrics: addr: 127.0.0.31 port: 9090 subnet: - addr: 10.45.0.1/16 dnn: internet dns: - 8.8.8.8 - 8.8.4.4 - 2001:4860:4860::8888 - 2001:4860:4860::8844 mtu: 1400 ctf: enabled: auto freeDiameter: /etc/freeDiameter/smf.conf info: - s_nssai: - sst: 1 sd: 000001 dnn: - internet	smf: sbi: - addr: 127.0.0.32 port: 7777 pfcp: - addr: 192.168.0.22 gtpc: - addr: 127.0.0.32 gtpu: - addr: 192.168.0.22 metrics: addr: 127.0.0.32 port: 9090 subnet: - addr: 10.46.0.1/16 dnn: internet2 dns: - 8.8.8.8 - 8.8.4.4 - 2001:4860:4860::8888 - 2001:4860:4860::8844 mtu: 1400 ctf: enabled: auto freeDiameter: /etc/freeDiameter/smf2.conf info: - s_nssai: - sst: 1 sd: 000002 dnn: - internet2	smf: sbi: - addr: 127.0.0.33 port: 7777 pfcp: - addr: 192.168.0.23 gtpc: - addr: 127.0.0.33 gtpu: - addr: 192.168.0.23 metrics: addr: 127.0.0.33 port: 9090 subnet: - addr: 10.55.0.1/16 dnn: voip dns: - 8.8.8.8 - 8.8.4.4 - 2001:4860:4860::8888 - 2001:4860:4860::8844 mtu: 1400 ctf: enabled: auto freeDiameter: /etc/freeDiameter/smf3.conf info: - s_nssai: - sst: 2 sd: 000001 dnn: - voip	smf: sbi: - addr: 127.0.0.34 port: 7777 pfcp: - addr: 192.168.0.24 gtpc: - addr: 127.0.0.34 gtpu: - addr: 192.168.0.24 metrics: addr: 127.0.0.34 port: 9090 subnet: - addr: 10.56.0.1/16 dnn: voip2 dns: - 8.8.8.8 - 8.8.4.4 - 2001:4860:4860::8888 - 2001:4860:4860::8844 mtu: 1400 ctf: enabled: auto freeDiameter: /etc/freeDiameter/smf4.conf info: - s_nssai: - sst: 2 sd: 000002 dnn: - voip2
freeDiameter : ListenOn = "127.0.0.31"	freeDiameter : ListenOn = “127.0.0.32"	freeDiameter : ListenOn = “127.0.0.33"	freeDiameter : ListenOn = “127.0.0.34"

Configuration changes on U-Planes (VM3-U-Plane1 and VM4-U-Plane2)

These VMs also include the 5GC components, but here we activate only the UPF based on their machine IP address. While running or restarting the components, here we need to stop all the other components and run/restart only the UPF component on both the VMs.

The changes made on both VMs are as follows

U-Plane1 (upf.yaml) | VM 3	**U-Plane2 (upf.yaml) | VM 4**
upf: pfcp: - addr: 192.168.0.7 gtpu: - addr: 192.168.0.7 subnet: - addr: 10.45.0.1/16 dnn: internet dev: ogstun - addr: 10.46.0.1/16 dnn: internet2 dev: ogstun2	upf: pfcp: - addr: 192.168.0.9 gtpu: - addr: 192.168.0.9 subnet: - addr: 10.55.0.1/16 dnn: voip dev: ogstun3 - addr: 10.56.0.1/16 dnn: voip2 dev: ogstun4

Configuration changes on UERANSIM (gNB & UEs) VM2

Changes needed to be performed on gNB is as follows

gNB

```
mcc: '901'          # Mobile Country Code value
mnc: '70'           # Mobile Network Code value (2 or 3 digits)

nci: '0x000000010'  # NR Cell Identity (36-bit)
idLength: 32        # NR gNB ID length in bits [22...32]
tac: 1              # Tracking Area Code

linkIp: 192.168.0.6   # gNB's local IP address for Radio Link Simulation (Usually same with local IP)
ngapIp: 192.168.0.6   # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 192.168.0.6   # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
#  - address: 10.0.2.15
   - address: 192.168.0.4
     port: 38412

# List of supported S-NSSAIs by this gNB
slices:
  - sst: 1
    sd: 1
  - sst: 1
    sd: 2
  - sst: 2
    sd: 1
  - sst: 2
    sd: 2
```

Subscriber Information (other information is the same) is as follows. These User informations were registered with Open5GS WebUI

UE #	IMSI	DNN	OP/OPc	SST/SD
UE0	901701234567890	internet	OPc	1/1
UE1	901701234567891	internet2	OPc	1/2
UE2	901701234567892	voip	OPc	2/1
UE3	901701234567893	voip2	OPc	2/2

Configuration changes with UEs (UE0, UE1, UE2, UE3)

UE0	UE1	UE2	UE3
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits) supi: 'imsi-901701234567890' # Mobile Country Code value of HPLMN mcc: '901' # Mobile Network Code value of HPLMN (2 or 3 digits) mnc: '70' # Permanent subscription key key: '11223344556677889 9aabbccddeeff00' # Operator code (OP or OPC) of the UE op: '0102030405060708090a0b0c0d0e0f00' # This value specifies the OP type and it can be either 'OP' or 'OPC' opType: 'OPC' # Authentication Management Field (AMF) value amf: '8000' # IMEI number of the device. It is used if no SUPI is provided imei: '356938035643800' # IMEISV number of the device. It is used if no SUPI and IMEI is provided imeiSv: '4370816125816151' # List of gNB IP addresses for Radio Link Simulation gnbSearchList: - 192.168.0.6 …… # Initial PDU sessions to be established sessions: - type: 'IPv4' apn: 'internet' slice: sst: 1 sd: 000001 # Configured NSSAI for this UE by HPLMN configured-nssai: - sst: 1 sd: 000001 # Default Configured NSSAI for this UE default-nssai: - sst: 1 sd: 000001	# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits) supi: 'imsi-901701234567891' # Mobile Country Code value of HPLMN mcc: '901' # Mobile Network Code value of HPLMN (2 or 3 digits) mnc: '70' # Permanent subscription key key: '11223344556677889 9aabbccddeeff01' # Operator code (OP or OPC) of the UE op: '0102030405060708090a0b0c0d0e0f01' # This value specifies the OP type and it can be either 'OP' or 'OPC' opType: 'OPC' # Authentication Management Field (AMF) value amf: '8000' # IMEI number of the device. It is used if no SUPI is provided imei: '356938035643801' # IMEISV number of the device. It is used if no SUPI and IMEI is provided imeiSv: '4370816125816151' # List of gNB IP addresses for Radio Link Simulation gnbSearchList: - 192.168.0.6 ….. # Initial PDU sessions to be established sessions: - type: 'IPv4' apn: 'internet2' slice: sst: 1 sd: 000002 # Configured NSSAI for this UE by HPLMN configured-nssai: - sst: 1 sd: 000002 # Default Configured NSSAI for this UE default-nssai: - sst: 1 sd: 000002	# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits) supi: 'imsi-901701234567892' # Mobile Country Code value of HPLMN mcc: '901' # Mobile Network Code value of HPLMN (2 or 3 digits) mnc: '70' # Permanent subscription key key: '11223344556677889 9aabbccddeeff02' # Operator code (OP or OPC) of the UE op: '0102030405060708090a0b0c0d0e0f02' # This value specifies the OP type and it can be either 'OP' or 'OPC' opType: 'OPC' # Authentication Management Field (AMF) value amf: '8000' # IMEI number of the device. It is used if no SUPI is provided imei: '356938035643802' # IMEISV number of the device. It is used if no SUPI and IMEI is provided imeiSv: '4370816125816151' # List of gNB IP addresses for Radio Link Simulation gnbSearchList: - 192.168.0.6 …… # Initial PDU sessions to be established sessions: - type: 'IPv4' apn: 'voip' slice: sst: 2 sd: 000001 # Configured NSSAI for this UE by HPLMN configured-nssai: - sst: 2 sd: 000001 # Default Configured NSSAI for this UE default-nssai: - sst: 2 sd: 000001	# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits) supi: 'imsi-901701234567893' # Mobile Country Code value of HPLMN mcc: '901' # Mobile Network Code value of HPLMN (2 or 3 digits) mnc: '70' # Permanent subscription key key: '11223344556677889 9aabbccddeeff03' # Operator code (OP or OPC) of the UE op: '0102030405060708090a0b0c0d0e0f03' # This value specifies the OP type and it can be either 'OP' or 'OPC' opType: 'OPC' # Authentication Management Field (AMF) value amf: '8000' # IMEI number of the device. It is used if no SUPI is provided imei: '356938035643803' # IMEISV number of the device. It is used if no SUPI and IMEI is provided imeiSv: '4370816125816151' # List of gNB IP addresses for Radio Link Simulation gnbSearchList: - 192.168.0.6 ….. # Initial PDU sessions to be established sessions: - type: 'IPv4' apn: 'voip2' slice: sst: 2 sd: 000002 # Configured NSSAI for this UE by HPLMN configured-nssai: - sst: 2 sd: 000002 # Default Configured NSSAI for this UE default-nssai: - sst: 2 sd: 000002

Once the configuration changes are made, we need to connect it to a Data Network with service functionalities. Below explains about the service functionalities implemented

Service Functionality in Data Network

Streaming Platform, Owncast 2022:
Owncast is a free and open-source live video and web chat server for use with existing popular broadcasting software. For the given network architecture, The OwnCast streaming server has been implemented on a private Ip subnet 10.0.2.35:8080 in the User Plane 1. The users who are connected with APN/DNN with ‘internet’ and ‘internet2’ should be able to connect to the owncast client and be able to stream the data from the owncast server.

File Sharing Platform, NextCloud 2022:
Nextcloud is a suite of client-server software for creating and using file hosting services. Nextcloud provides functionally similar to Dropbox, Office 365 or Google Drive when used with integrated office suite solutions Collabora Online or OnlyOffice. For the given network architecture, The Nextcloud file hosting server has been implemented on a private Ip subnet 10.0.2.25:2000 in the User Plane 1. The users who are connected with APN/DNN with ‘internet’ and ‘internet2’ should be able to connect to the Nextcloud client and be able to register to the Nextcloud file sharing services.

Results and Discussion

1. Open5GS with srsRAN

Run Open5GS :

After configuration of the components of 5GC in Open5GS, to receive the changes in the machine, we need to restart the 5GC services as mentioned below.

```
$ sudo systemctl restart open5gs-smfd
$ sudo systemctl restart open5gs-amfd
$ sudo systemctl restart open5gs-upfd
$ sudo systemctl restart open5gs-nrfd
$ sudo systemctl restart open5gs-scpd
$ sudo systemctl restart open5gs-ausfd
$ sudo systemctl restart open5gs-udmd
$ sudo systemctl restart open5gs-pcfd
$ sudo systemctl restart open5gs-nssfd
$ sudo systemctl restart open5gs-bsfd
$ sudo systemctl restart open5gs-udrd
$ sudo systemctl restart open5gs-webui
```

After restarting the 5GC components, we have depicted the working of the architecture through Wireshark traces captured at Service Based Interface and below figure (figure no) is the wireshark trace and Message Sequence Chart (MSC) shows the successful PFCP (Packet Forwarding Control Protocol) association request and responses messages between SMF and UPF.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.4	127.0.0.7	PFCP	74	PFCP Association Setup Request
2	0.000415682	127.0.0.7	127.0.0.4	PFCP	82	PFCP Association Setup Response
3	2.361341475	127.0.0.7	127.0.0.4	PFCP	77	PFCP Association Setup Request
4	2.361663433	127.0.0.4	127.0.0.7	PFCP	79	PFCP Association Setup Response
5	11.005790034	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Request
6	11.006024927	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Response
7	11.006108182	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Request
8	11.006293709	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Response
9	22.008978702	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Request
10	22.009393255	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Response
11	22.009493497	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Request
12	22.009789873	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Response
13	33.011988276	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Request
14	33.012195316	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Response
15	33.012270339	127.0.0.7	127.0.0.4	PFCP	60	PFCP Heartbeat Request
16	33.012338609	127.0.0.4	127.0.0.7	PFCP	60	PFCP Heartbeat Response

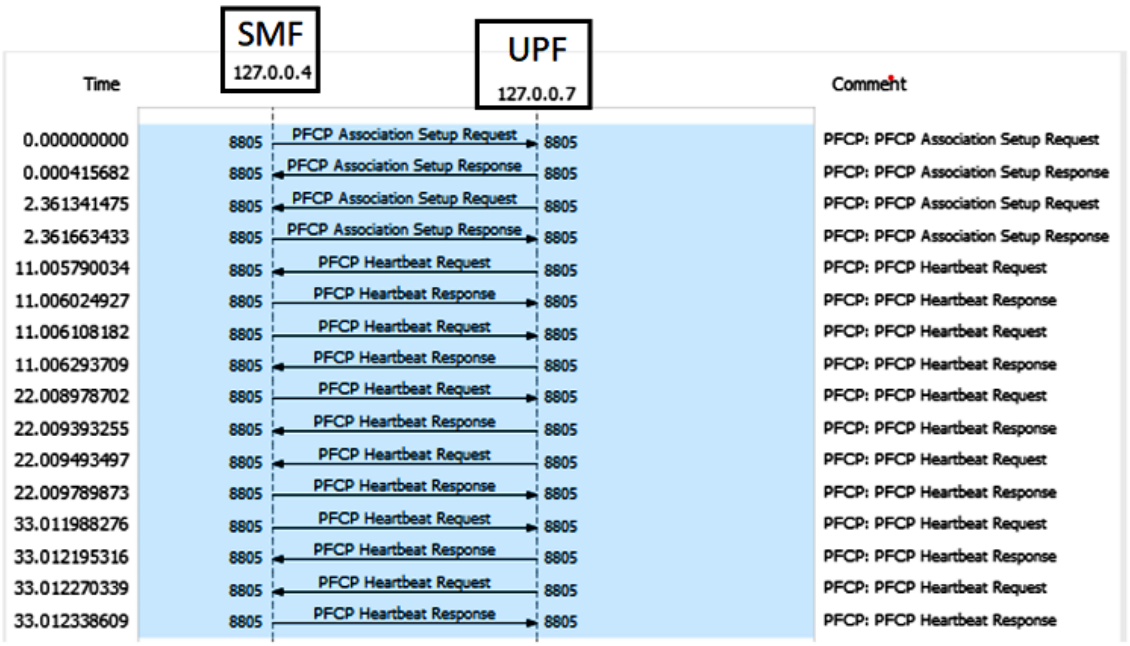

Figure 5. WS traces and MSC between SMF and UPF

Run gNB & UE :

Once the 5GC is up and running, now we can run the gNB & UE with the following commands in the srsRAN directory respectively.
To run enb - “sudo srsenb”.
To ensure successful connection between 5GC and enb, we need to receive “NG connection successful” in the same terminal.
To run ue - “sudo srsue”.
Similarly, the successful initialization of UE can be verified by receiving the “PDU Session Establishment successful. “ in the same terminal

The below figure represents the Wireshark traces and message sequence chart (MSC) which is generated after successful start of srsenb, between enb/gnb and AMF. The NGAP traces has the NG set up request and response to initialize a gNB and PDU association request and response for initilizing the UE.

	Time	Source	Destination	Protocol	Length	Info
108	5.187535306	127.0.1.1	127.0.0.2	NGAP	120	NGSetupRequest
110	5.202697868	127.0.0.2	127.0.1.1	NGAP	120	NGSetupResponse
3183	24.457346638	127.0.1.1	127.0.0.2	NGAP/NAS-5GS	148	InitialUEMessage, Registration request
3309	24.489879103	127.0.0.2	127.0.1.1	NGAP/NAS-5GS	148	SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
3452	24.547862305	127.0.1.1	127.0.0.2	NGAP/NAS-5GS	144	SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Authentication response
3509	24.559612136	127.0.0.2	127.0.1.1	NGAP/NAS-5GS	124	SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Security mode command
3668	24.607080039	127.0.1.1	127.0.0.2	NGAP/NAS-5GS	184	SACK (Ack=2, Arwnd=106496) , UplinkNASTransport
3842	24.640998116	127.0.0.2	127.0.1.1	NGAP/NAS-5GS	232	SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest
4300	24.883423553	127.0.1.1	127.0.0.2	NGAP	84	InitialContextSetupResponse
5077	25.085643256	127.0.1.1	127.0.0.2	NGAP/NAS-5GS	220	UplinkNASTransport, UplinkNASTransport
5081	25.087993801	127.0.0.2	127.0.1.1	NGAP/NAS-5GS	144	SACK (Ack=6, Arwnd=106496) , DownlinkNASTransport
5257	25.133630965	127.0.0.2	127.0.1.1	NGAP/NAS-5GS	224	PDUSessionResourceSetupRequest
5273	25.139994642	127.0.1.1	127.0.0.2	NGAP	104	PDUSessionResourceSetupResponse
5720	25.346052445	127.0.1.1	127.0.0.2	NGAP/NAS-5GS	120	UplinkNASTransport

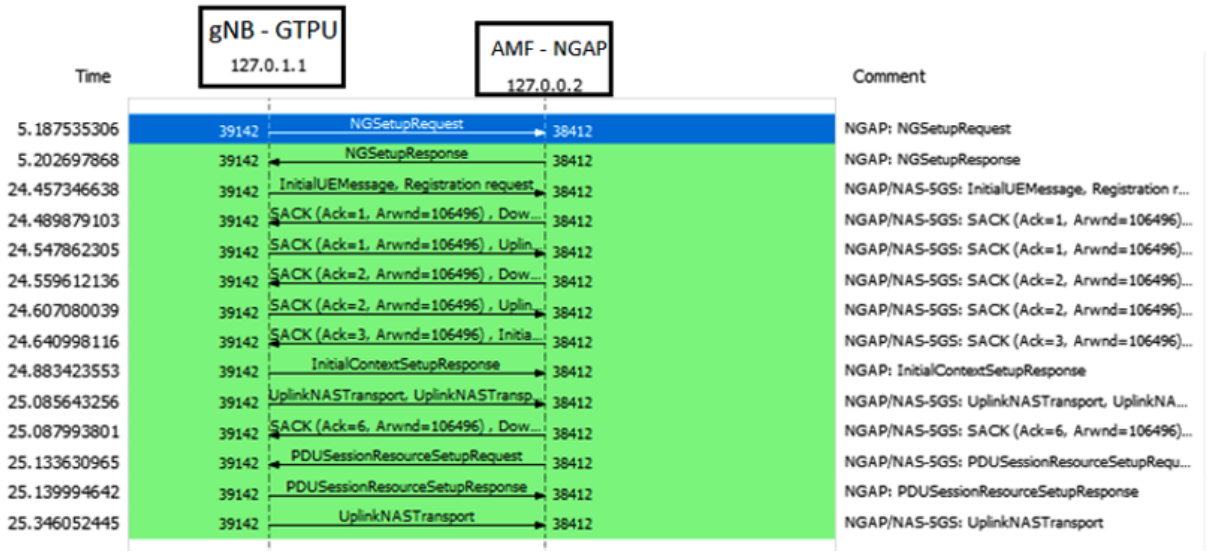

Figure 6. NGAP traces between gNB and AMF

2. Open5GS with UERANSIM

Run Open5GS 5GC in C-Plane (VM1):

After configuration of the components of 5GC in Open5GS, to receive the changes in the machine, we need to restart the 5GC services as mentioned below. In order to support network slicing, the commands have few modifications compared to srsRAN. As multiple SMFs need to be implemented, these services are made to run separately

```
$ sudo systemctl stop open5gs-smfd
$ sudo systemctl stop open5gs-upfd
$ sudo systemctl restart open5gs-amfd
$ sudo systemctl restart open5gs-nrfd
$ sudo systemctl restart open5gs-scpd
$ sudo systemctl restart open5gs-ausfd
$ sudo systemctl restart open5gs-udmd
$ sudo systemctl restart open5gs-pcfd
$ sudo systemctl restart open5gs-nssfd
$ sudo systemctl restart open5gs-bsfd
$ sudo systemctl restart open5gs-udrd
$ sudo systemctl restart open5gs-webui

$ sudo ./open5gs-smfd -c /smf.yaml
$ sudo ./open5gs-smfd -c /smf2.yaml
$ sudo ./open5gs-smfd -c /smf3.yaml
$ sudo ./open5gs-smfd -c /smf4.yaml
```

Run Open5GS 5GC in U-Plane1 (VM3):

Once 5GC C-Plane is up and running, we need to initialize the UPF in U-Plane1. The following commands are used to run and stop the required as mentioned below

```
$ sudo systemctl stop open5gs-smfd
$ sudo systemctl stop open5gs-amfd
$ sudo systemctl stop open5gs-nrfd
$ sudo systemctl stop open5gs-scpd
$ sudo systemctl stop open5gs-ausfd
$ sudo systemctl stop open5gs-udmd
$ sudo systemctl stop open5gs-pcfd
$ sudo systemctl stop open5gs-nssfd
$ sudo systemctl stop open5gs-bsfd
$ sudo systemctl stop open5gs-udrd
$ sudo systemctl stop open5gs-webui

$ sudo systemctl restart open5gs-upfd
```

The connection between SMF1 and UPF1 is associated with PFCP and is shown in the below figure (figure no)Wireshark traces and represented as MSC.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.22	192.168.0.7	PFCP	72	PFCP Association Setup Request
2	0.001115032	192.168.0.7	192.168.0.22	PFCP	80	PFCP Association Setup Response
3	10.999626559	192.168.0.22	192.168.0.7	PFCP	60	PFCP Heartbeat Request
4	10.999921006	192.168.0.7	192.168.0.22	PFCP	58	PFCP Heartbeat Response
5	11.002339731	192.168.0.7	192.168.0.22	PFCP	58	PFCP Heartbeat Request
6	11.003273822	192.168.0.22	192.168.0.7	PFCP	60	PFCP Heartbeat Response

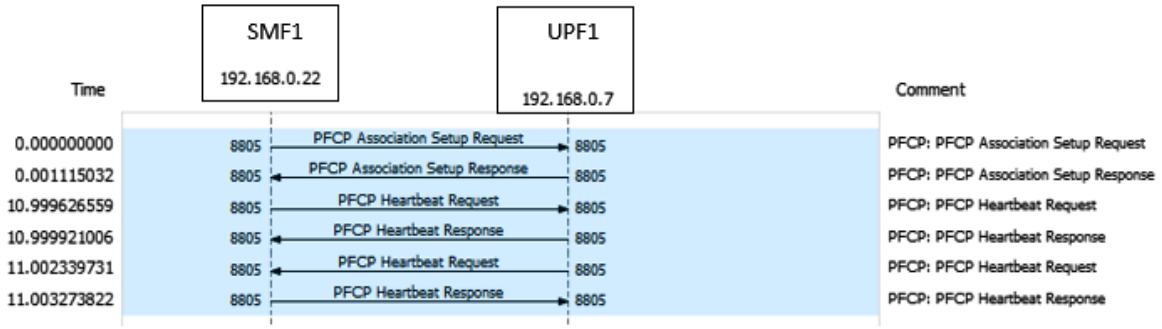

Figure 7. WS traces and MSC of PFCP association b/w SMF1 and UPF1

Run Open5GS 5GC in U-Plane2 (VM4):

Similarly, initialization of the UPF in U-Plane2. The following commands can be performed by running and stopping the required as mentioned below

```
$ sudo systemctl stop open5gs-smfd
$ sudo systemctl stop open5gs-amfd
$ sudo systemctl stop open5gs-nrfd
$ sudo systemctl stop open5gs-scpd
$ sudo systemctl stop open5gs-ausfd
$ sudo systemctl stop open5gs-udmd
$ sudo systemctl stop open5gs-pcfd
$ sudo systemctl stop open5gs-nssfd
$ sudo systemctl stop open5gs-bsfd
$ sudo systemctl stop open5gs-udrd
$ sudo systemctl stop open5gs-webui

$ sudo systemctl restart open5gs-upfd
```

The connection between SMF3 and UPF2 is associated with PFCP and is shown in the below figure (figure no)Wireshark traces and represented as MSC.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.23	192.168.0.9	PFCP	72	PFCP Association Setup Request
2	0.000246638	192.168.0.9	192.168.0.23	PFCP	80	PFCP Association Setup Response
3	11.004288162	192.168.0.9	192.168.0.23	PFCP	58	PFCP Heartbeat Request
4	11.005428196	192.168.0.23	192.168.0.9	PFCP	60	PFCP Heartbeat Response
5	11.005428369	192.168.0.23	192.168.0.9	PFCP	60	PFCP Heartbeat Request
6	11.005756527	192.168.0.9	192.168.0.23	PFCP	58	PFCP Heartbeat Response

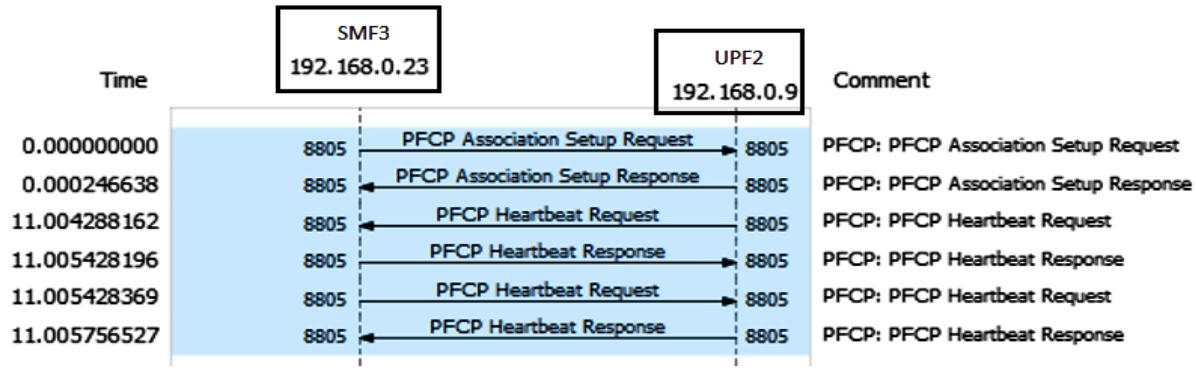

Figure 8. WS traces and MSC of PFCP association b/w SMF3 and UPF2

Run UERAN (gNB & UEs) in U-Plane (VM2) :

Once the 5GC is up and running, initialization of the gNB & UE can be performed by the following commands in the UERANSIM/build directory respectively.
To run gnb - “sudo ./nr-gnb -c ../config/open5gs-gnb.yaml”.
To ensure successful connection between 5GC and gnb, we need to receive “NG connection successful” in the same terminal.
To run the first ue, UE0 - “sudo ./nr-ue -c ../config/open5gs-ue0.yaml”.
The successful initialization of UE can be verified by receiving the “PDU Session Establishment successful” with TUNnel interface[uesimtun0,10.45.0.2] in the same terminal.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.6	192.168.0.4	NGAP	152	NGSetupRequest
2	0.002531604	192.168.0.4	192.168.0.6	NGAP	164	NGSetupResponse
3	15.201457920	192.168.0.6	192.168.0.4	NGAP/NAS-5GS	140	InitialUEMessage, Registration request
4	15.217376153	192.168.0.4	192.168.0.6	NGAP/NAS-5GS	148	SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
5	15.221056535	192.168.0.6	192.168.0.4	NGAP/NAS-5GS	148	SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Authentication response
6	15.228767539	192.168.0.4	192.168.0.6	NGAP/NAS-5GS	128	SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Security mode command
7	15.230365912	192.168.0.6	192.168.0.4	NGAP/NAS-5GS	192	SACK (Ack=2, Arwnd=106496) , UplinkNASTransport
8	15.246570901	192.168.0.4	192.168.0.6	NGAP/NAS-5GS	236	SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest
9	15.247471504	192.168.0.6	192.168.0.4	NGAP	100	SACK (Ack=3, Arwnd=106496) , InitialContextSetupResponse
10	15.452245111	192.168.0.6	192.168.0.4	NGAP/NAS-5GS	240	UplinkNASTransport, UplinkNASTransport
11	15.453362742	192.168.0.4	192.168.0.6	NGAP/NAS-5GS	144	SACK (Ack=6, Arwnd=106496) , DownlinkNASTransport
12	15.469470593	192.168.0.4	192.168.0.6	NGAP/NAS-5GS	248	PDUSessionResourceSetupRequest
13	15.473160804	192.168.0.6	192.168.0.4	NGAP	104	PDUSessionResourceSetupResponse

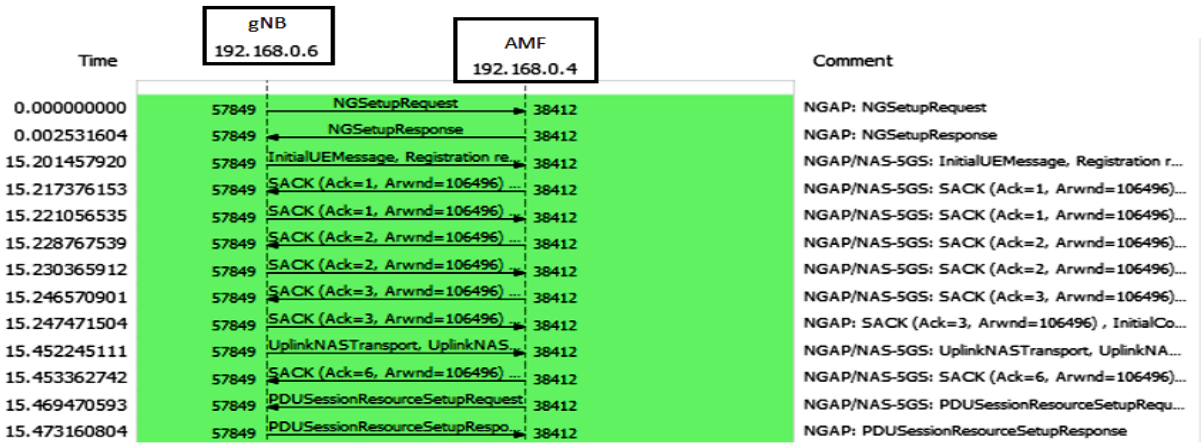

Figure 9. NGAP traces between gNB and AMF

Similarly, other UEs (UE1, UE2, UE3) can be realized.

The above figure represents the Wireshark traces and message sequence chart (MSC) which is generated after successful start of UERANSIM gNB, between gnb and AMF. The NGAP traces has the NG set up request and response to initialize a gNB and PDU association request and response for initializing the UE0.

Accessing the Data Network

Ping GOOGLE server:
In order to test the UE that has been realised to our 5G Core, A ping request has been made to the domain www.google.com where a reply has received from the respective server. The Figure "XX" shows the ICMP packets exchange between the User Equipment with Ip: 10.45.0.2 and Google Server hosted in 142.250.185.78 in Wireshark along with the respective Message Sequence Chart.

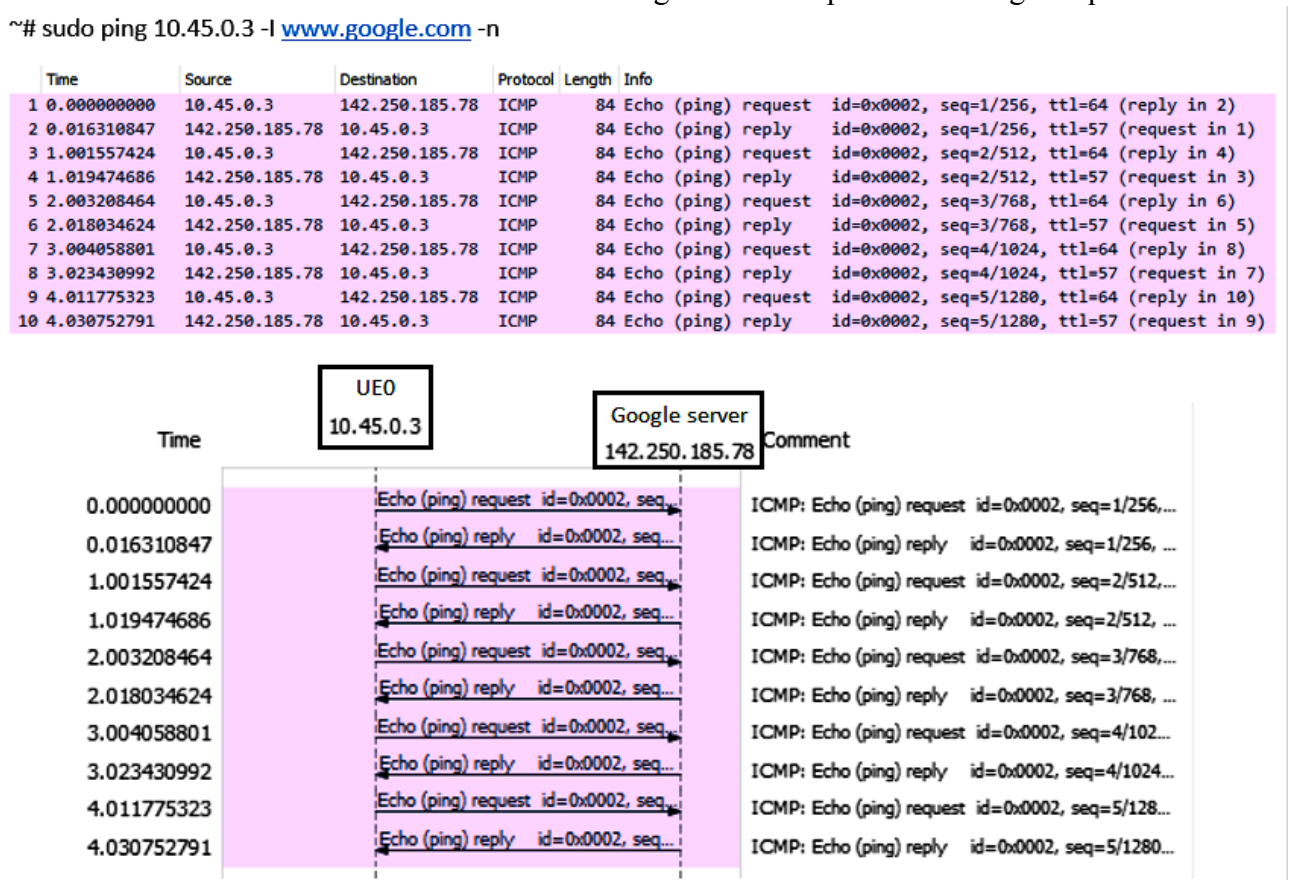

Figure 10. Ping Google server

Streaming Platform, OwnCast:

The OwnCast streaming server has been implemented on a private Ip subnet 10.0.2.35:8080 in the User Plane 1. The users who are connected to the 5G Core via 'internet' and 'internet2' DNN shall be able to access the OwnCast server by with pinging to the server Ip. The Figure "XX" describes the traces and Message Sequence Chart between the User Equipment 10.45.0.2 and owncast server 10.0.2.35:8080, where the UE hits a GET request using CURL via nr-binder to the owncast server and receives its response from the server respectively.
API request: ~# sudo ./nr-binder 10.45.0.2 curl -X GET "10.0.235:8080"

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.45.0.2	10.0.2.35	TCP	60	41823 → 8080 [SYN] Seq=0 Win=65280 Len=0 MSS=1360 SACK_PERM TSval=312288943 TSecr=0 WS=128
2	0.003199604	10.0.2.35	10.45.0.2	TCP	60	8080 → 41823 [SYN, ACK] Seq=0 Ack=1 Win=64704 Len=0 MSS=1360 SACK_PERM TSval=816784373 TSecr=312288943 WS=128
3	0.003261938	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=1 Ack=1 Win=65280 Len=0 TSval=312288946 TSecr=816784373
4	0.003372579	10.45.0.2	10.0.2.35	HTTP	130	GET / HTTP/1.1
5	0.006631450	10.0.2.35	10.45.0.2	TCP	52	8080 → 41823 [ACK] Seq=1 Ack=79 Win=64640 Len=0 TSval=816784377 TSecr=312288946
6	0.017024638	10.0.2.35	10.45.0.2	TCP	1160	8080 → 41823 [PSH, ACK] Seq=1 Ack=79 Win=64640 Len=1108 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
7	0.017035965	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=1109 Win=64256 Len=0 TSval=312288960 TSecr=816784384
8	0.017121237	10.0.2.35	10.45.0.2	TCP	1400	8080 → 41823 [ACK] Seq=1109 Ack=79 Win=64640 Len=1348 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
9	0.017128838	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=2457 Win=63232 Len=0 TSval=312288960 TSecr=816784384
10	0.017917214	10.0.2.35	10.45.0.2	TCP	1400	8080 → 41823 [ACK] Seq=2457 Ack=79 Win=64640 Len=1348 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
11	0.017921919	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=3805 Win=64256 Len=0 TSval=312288961 TSecr=816784384
12	0.018019069	10.0.2.35	10.45.0.2	TCP	1400	8080 → 41823 [ACK] Seq=3805 Ack=79 Win=64640 Len=1348 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
13	0.018023506	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=5153 Win=63232 Len=0 TSval=312288961 TSecr=816784384
14	0.018044735	10.0.2.35	10.45.0.2	TCP	1400	8080 → 41823 [ACK] Seq=5153 Ack=79 Win=64640 Len=1348 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
15	0.018046666	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=6501 Win=62080 Len=0 TSval=312288961 TSecr=816784384
16	0.018066071	10.0.2.35	10.45.0.2	TCP	1400	8080 → 41823 [PSH, ACK] Seq=6501 Ack=79 Win=64640 Len=1348 TSval=816784384 TSecr=312288946 [TCP segment of a reassembled PDU]
17	0.018068991	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=7849 Win=60928 Len=0 TSval=312288961 TSecr=816784384
18	0.018088182	10.0.2.35	10.45.0.2	HTTP	493	HTTP/1.1 200 OK (text/html)
19	0.018089754	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=79 Ack=8290 Win=64256 Len=0 TSval=312288961 TSecr=816784384
20	0.018295235	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [FIN, ACK] Seq=79 Ack=8290 Win=64256 Len=0 TSval=312288961 TSecr=816784384
21	0.021261232	10.0.2.35	10.45.0.2	TCP	52	8080 → 41823 [FIN, ACK] Seq=8290 Ack=80 Win=64640 Len=0 TSval=816784392 TSecr=312288961
22	0.021273384	10.45.0.2	10.0.2.35	TCP	52	41823 → 8080 [ACK] Seq=80 Ack=8291 Win=64256 Len=0 TSval=312288964 TSecr=816784392

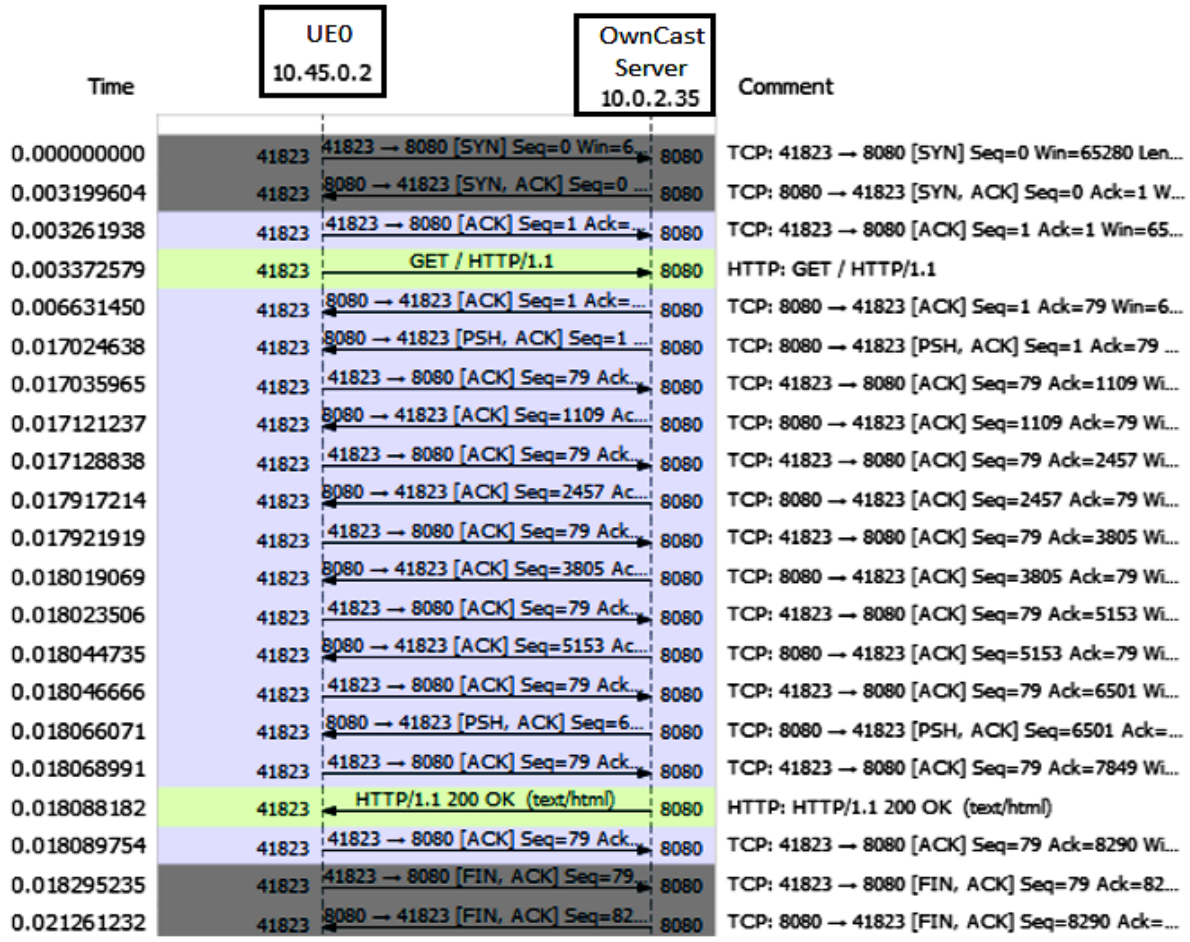

Figure 11. WS traces and MSC on OwnCast

File Sharing Platform, Nextcloud:

The NextCloud File Hosting server has been implemented on a private Ip subnet 10.0.2.25:2000 in the User Plane 1. The users who are connected to the 5G Core via ‘internet’ and ‘internet2’ DNN shall be able to access the NextCloud server by with pinging to the server Ip. The Figure “XX” describes the traces and message sequence chart between the User Equipment 10.45.0.2 and Nextcloud server 10.0.2.25:2000, where the UE hits a GET request using CURL via nr-binder to the server and receives its response from the server respectively.

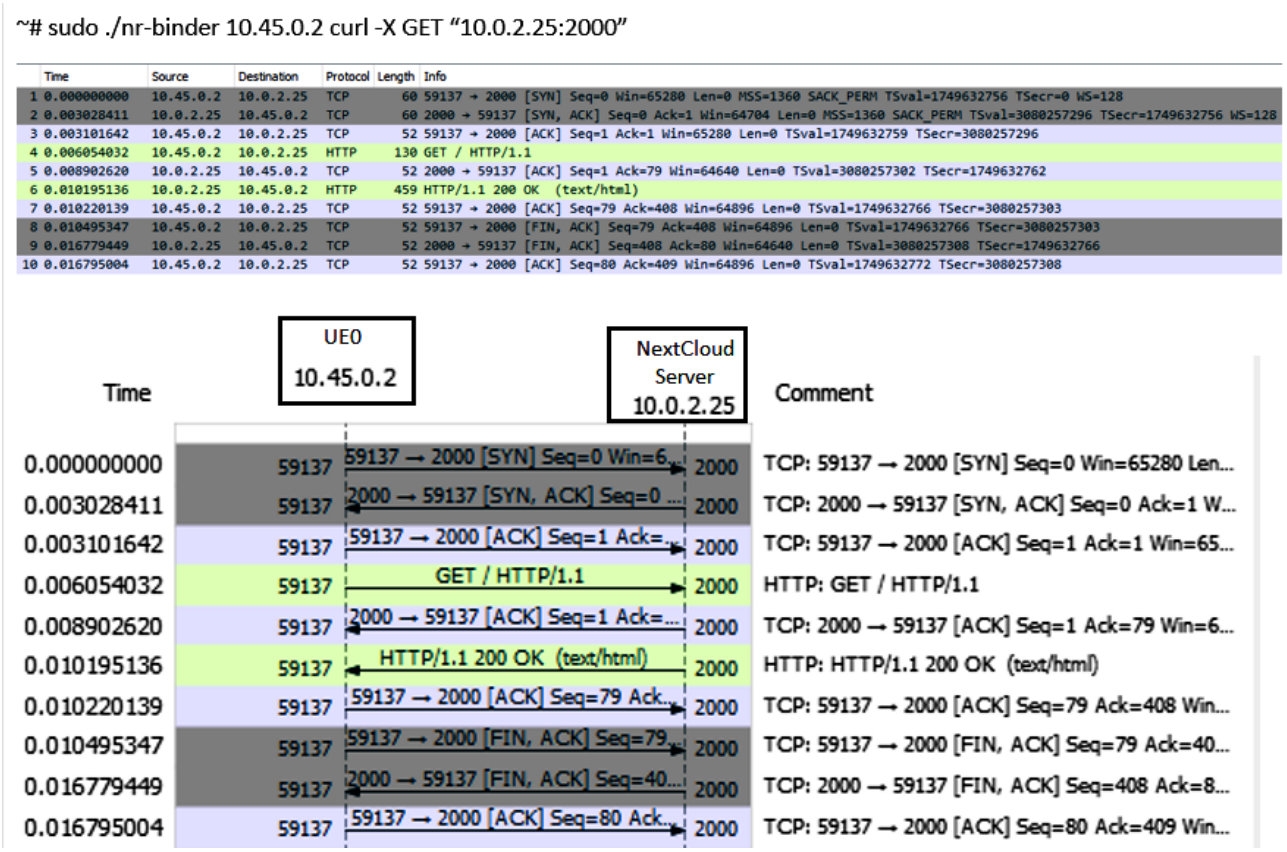

Figure 11. WS traces and MSC on NextCloud

Conclusion

This study described a revolutionary strategy to running the 5G Next Generation Core in a fully virtualized environment and operating the network operations as Cloud-Native in a public cloud environment with service-based programmable interfaces that can scale-up on demand in a totally automated manner. In our configuration, we integrated two open-source tools to test the functioning of the 5G Stand Alone: Open5GS for the 5G Core deployment and SRSRAN/UERANSIM for the state-of-the-art 5G User Equipment and Radio Access Network (gNodeB) simulator.

Our objective is to configure a network architecture using the concept of network slicing with multiple DNN/APNs for which multiple user end devices has to be registered to the 5G Core and should be able to access the data network and further, the streaming platform Own Cast Server, and File sharing platform NeXT Cloud server has configured in order to access the functionalities of the data network for the User Equipment's that are connected to the 5G Core.

The 5G Core network that has been established allows users to access a variety of web applications. Using the current configuration, a lot more services, such as multimedia streaming, has be accessed. So, the 5G Core Network effectively sets the tone for a 5G Private Network. The entire potential of 5G can be unlocked by scaling this up the system further and introducing additional technologies.

References

- Open5GS, 2022. *Open5GS*. [Online] Available at: https://open5gs.org/.
- SRSRAN, 2022. *SRSRAN*. [Online] Available at: https://www.srslte.com/.
- Enterprise 5 : Guide to planning, architecture and benefits [Network Slicing], John Burke, Nemertes Research. [Online] : https://www.techtarget.com/whatis/definition/network-slicing
- 5G : Network slicing, Ericsson. [Online] : https://www.ericsson.com/en/network-slicing
- GitHub, Inc., 2022. *GitHub*. [Online] Available at: https://github.com/.
- GÜNGÖR, A., 2022. *UERANSIM*. [Online] Available at: https://github.com/aligungr/UERANSIM.
- Shigeru Ishida, 2022. *UERANSIM*. [Online] Available at : https://github.com/s5uishida/open5gs_5gc_ueransim_sample_config
- Calee, Github, Inc., 2022. *GitHub*. [Online] : https://github.com/calee0219/awesome-5g#ran
- Kamailio, 2022. *Kamailio SIP Server.* [Online] Available at: https://www.kamailio.org/
- NextCloud, 2022. *NextCloud - Online Collaboration Platform*. [Online] Available at: https://nextcloud.com/.
- Oracle, 2022. *VirtualBox*. [Online] Available at: https://www.virtualbox.org/.
  Owncast, 2022. *Owncast - Free and Open Source Livestreaming*. [Online] Available at: https://owncast.online/
- Stallings, W., 2021. *5G wireless : a comprehensive introduction*. Boston; Columbus; New York; San Francisco; Amsterdam; Cape Town; Dubai; London; Madrid; Milan; Munich; Paris; Montreal; Toronto; Delhi; Mexico City; São Paulo; Sydney; Hong Kong; Seoul; Singapore; Taipei; Tokyo: Addison-Wesley.
- Trick, U., 2021. *5G - An Introduction to the 5th Generation Mobile Networks*. De Gruyter STEM ed. Berlin/München/Boston: De Gruyter Oldenbourg.