Technical Report.
5G Framed Routing
Adding management of sub-networks on WebGUI

Leonardo Cordeiro Gonçalves - 2020228071
Gonçalo Tavares Bastos - 2020228071
Supervisores Altice: Engs. Francisco Fontes e Miguel Freitas
Supervisor UC: Prof.Maria Medeiros
Sponsor: Altice Labs

October 2022 - June 2023

Abstract

Framed routing is a new functionality of Open5GS an opensource 5G Core implementation. It doesn't been validated or tested yet. Management/provision operations are still incomplete on the core. Monitoring such networks can be a challenge and we aim to do that by setting up an emulated environment using Virtual Machines.

Contents

1 Introduction

In certain scenarios, cellular mobile devices need to work as access points, sharing their connectivity with others. However, the use of Network Address Translation (NAT) in these cases imposes limitations on the devices, as they can only be identified by a single IP address assigned by the 5G core. This hampers reachability and bi-directional connectivity in advanced use cases.

To address this limitation, our project focuses on implementing the Framed Route functionality in 5G networks. By enabling 5G to behave as a router, we can connect multiple computer networks and facilitate communication between them. This means that one or more Local Area Network (LAN) segments can be served, with traffic from connected devices being forwarded bypassing NAT

The key advantage of framed routing is its ability to simplify data processing and forwarding for network devices. By incorporating this functionality natively into 5G, we can deploy additional IP networks behind User Equipment (UE). This allows access to specific IPv4 addresses or IPv6 prefixes through the established 5G PDU Session, even from external networks like the Internet.

By integrating framed routing into 5G networks, we aim to overcome the limitations imposed by NAT and enable advanced use cases that require multi-network connectivity. This will enhance reachability, and bi-directional connectivity, and expand the possibilities for leveraging 5G technology in various domains.

2 Project Definition

The main goal of the project is to validate and test the Framed Route feature in Open5GS and add the management functionality of subnets in the WebGUI API.

The solution must enable the assignment and the management of IPv4/v6 prefixes for each UE through the WebGUI and this change has to be reflected in the user database (MongoDB) and so on to the 5G Core network functions in charge of the PDU session establishment procedure (SMF, UDM and UPF), in the case of UPF it must also be able to receive the framed route TLVs from the SMF (through the PFCP protocol).

Packets directed to any address associated with the specified prefixes and received at the UPF N6 interface will be routed to the appropriate UE through the corresponding GTP tunnel, without involving any NAT operations.

The development project will employ the UERANSIM simulator to replicate the RAN and UE's and Open5gs to replicate the core and the UPF.

2.1 Objectives

1. Validate framed routing functionality in a state-of-the art open-source 5G core implementation (Open5GS), by setting up an emulated environment on Virtual Box

2. Validation should be done end-to-end, i.e. from an emulated client behind an emulated 5G UE to the internet/N6 (both IPv4 and IPv6 prefixes)

3. Investigate 3GPP standardized northbound APIs (e.g. CAPIF) to validate whether the Open5Gs UE provision API is aligned with current standards (and possible framed route "touch points")

4. Improve/develop the management web-interface of the 5G core by adding support for static framed route provisioning (currently only available in cli/db tools)

5. Check and improve possible validations for overlapping routes

6. Investigate 3GPP standards for possible options for framed route monitoring/charging (e.g. PFCP Session Report requests/response)

3 Project Implementation

3.1 The Evolution of Networks

The first generation of mobile networks, 1G, only allowed voice calls and was based on analog technology. 2G networks introduced digital voice calls and text messaging, and 3G networks brought faster data speeds and mobile internet access.

With the advent of 4G networks, mobile data speeds increased significantly, enabling the widespread adoption of video streaming and other data-intensive applications. 4G also enabled the development of the Internet of Things (IoT) and machine-to-machine communications.

Now 5G networks are taking network evolution to the next level. With lightning-fast speeds and low latency, 5G has the potential to transform industries such as healthcare, transportation and manufacturing. It will also enable new technologies such as autonomous vehicles, augmented reality and virtual reality.

Overall, the evolution of networks has been driven by the demand for faster, more reliable and more efficient communications. With each generation, new technologies and capabilities have emerged and the possibilities for the future of networking are endless.

3.2 Framed Route

In framed routing, data packets encapsulated in frames are routed through the network. The frames are created on the device sending the data and then routed through the network, with each network function responsible for processing and forwarding the frames to their final destination. The Session Management Function (SMF) is responsible for establishing and maintaining connections between devices on the network and plays a critical role in determining the optimal path for data packets moving through the network.

PDU session management, on the other hand, involves setting up and managing data sessions between devices on the network. A PDU session is a logical connection between a device and the 5G core network and is used to transfer data packets between the two. The SMF is also responsible for managing PDU sessions on the network. It establishes new sessions, manages existing sessions and terminates sessions when they are no longer needed.

Framed routing and PDU session management are critical components of the 5G network architecture that ensure data is transmitted efficiently and reliably between devices in the network. The SMF plays a central role in both components. It helps establish and maintain connections, determine the optimal path for data packets and manage PDU sessions across the network. By leveraging these capabilities, 5G networks are able to provide high-speed, low-latency connections to a wide range of devices and applications, making it an indispensable technology for the modern digital world.

3.3 5G main building blocks

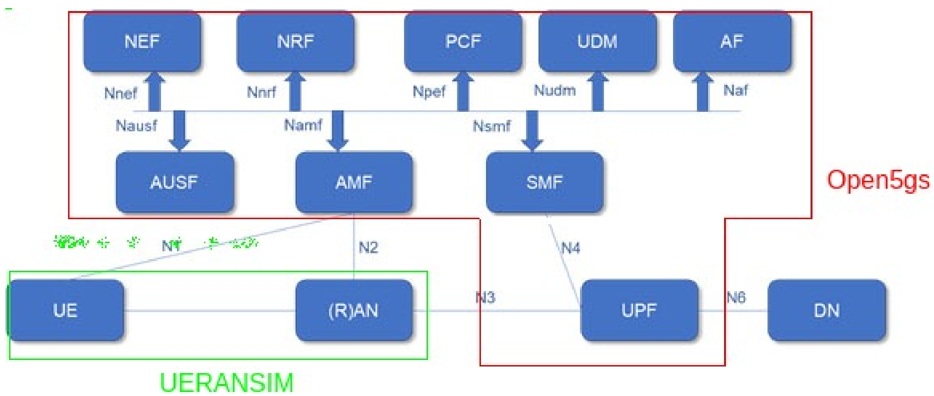


Figure 1: 5G Core Network

At the heart of 5G is the **5G core network**, which is responsible for managing all network functions and services. The 5G core consists of several key blocks, each of which plays a critical role in enabling the advanced features and capabilities of 5G, we can split them into two entities, they are Control entities and User plane entities.

Open5GSis an open-source 5G core network implementation. It is a complete implementation of the 5G core network that is compliant with 3GPP standards and provides all the necessary functionality to deploy and manage a 5G network.

At its core, Open5GS is designed to provide a highly scalable, flexible and efficient network architecture that can support a wide range of use cases and applications. It is built on a modular architecture consisting of several functional components, including the Access and Mobility Management Function (AMF), the Session Management Function (SMF), the User Plane Function (UPF) and the Policy Control Function (PCF), among others.

- AMF is responsible for managing the access and mobility of 5G devices within the network. It provides functions such as network registration, authentication and security management, as well as mobility management that allows devices to move seamlessly between different parts of the network.

- SMF is responsible for managing data sessions between 5G devices and the network. It provides functions such as data session creation, modification and termination, as well as policy and charge management.
- PCF is responsible for managing and enforcing policies related to network resource allocation, QoS, and user authentication and authorisation.
- UPF is responsible for managing user data and providing the required connectivity between the 5G devices and the network. It provides functions such as packet filtering, routing and forwarding, and Quality of Service (QoS) management, which ensures that user data is transmitted with the appropriate priority and reliability. It is directly connected to the RAN and the DN and essentially establishes tunnels through which data are exchanged between hosts in the DN and the UE.

UERANSIM is an open-source 5G UE (User Equipment) simulator that allows developers to test their 5G network applications and services in a simulated environment. It provides a realistic emulation of the 5G network architecture, including the core network, radio access network (RAN) and user equipment (UE).

This is done by creating a virtual 5G network that simulates the behaviour of a real-world network, complete with all the necessary components such as base stations, access points and user equipment.

Some of the critical features of UERANSIM is its ability to simulate different network scenarios, such as varying network load, network coverage and network capacity and even support for multiple 5G network protocols, including the 5G New Radio (NR) protocol and the 5G Core Network (CN) protocol. This allows us to test our application across the entire 5G network architecture, from the radio access network to the core network.

3.4 Sub-network managing and WebUI

WebUI Open5GS WebUI is a web-based user interface for the administration and configuration of the Open5GS core network components.

The Open5GS WebUI provides an easy-to-use graphical interface for configuring and monitoring various components of the Open5GS core network, including Subscriber Data Management (SDM), Session Management (SMF), User Plane Function (UPF) and Network Exposure Function (NEF).

Via the WebUI, network administrators can manage subscribers via IMSI, KEY and OPC. With these parameters, we can set up the appropriate PDU sessions, ensuring access to DN for each subscriber via the appropriate interfaces.

We can also configure network policies and monitor network traffic. The WebUI simplifies the management and maintenance of the Open5GS core network.

3.5 3GPP Standarts Overview

This section provides an overview of the framed routing support in 3GPP standards for 5G networks. Framed Routing is a feature that enables the support of an IP network behind a User Equipment (UE) in the context of PDU Sessions, also this section outlines the key concepts, mechanisms, and interfaces involved in framed routing, as defined by 3GPP specifications.

Framed Routing is a feature designed to facilitate connectivity to an IP network encompassing a range of IPv4 addresses or IPv6 prefixes over a single PDU Session in 5G networks. Here we are going to examine the framed routing support within the 3GPP standards and its significance for enterprise connectivity scenarios.

3.5.1 Overview of Framed Routing

Framed Routing in 3GPP standards is specifically defined for PDU Sessions of the IP type, including IPv4, IPv6, and IPv4v6. It allows for the establishment of connectivity to an IP network by associating multiple Framed Routes with a PDU Session. Each Framed Route represents a range of IPv4 addresses or IPv6 prefixes.

3.5.2 Flow of Information and Components

The flow of information in framed routing involves several components within the 5G network architecture. The Session Management Function (SMF) provides the Framed Route information to the User Plane Function (UPF) as part of the Packet Forwarding Action (PSA) in the Packet Detection Rule (PDR) related to the network side (N6) of the UPF.

The SMF can receive the Framed Route information from two sources: the DN-AAA server and the Unified Data Management (UDM). The DN-AAA server provides this information during the authentication/authorization process, while the UDM sends Session Management Subscription data associated with the Data Network Name (DNN) and Single Network Slice Selection Assistance Information (S-NSSAI).

The IP address allocated to the UE during PDU Session establishment may belong to one of the Framed Routes associated with the PDU Session. However, it is also possible for the IP address to be dynamically allocated outside of the defined Framed Routes.

3.5.3 Charging and Reporting

If Policy and Charging Control (PCC) is applied to the PDU Session, the SMF reports the Framed Route information corresponding to the PDU Session to the Policy Control Function (PCF). Additionally, the PCF may report the Framed Route information to the Binding Support Function (BSF) to support session binding.

3.5.4 Updates and Dynamic Allocation

In case the UDM or DN-AAA updates the Framed Route information while the PDU Session is active, the SMF releases the PDU Session and may include an indication in the release request for the UE to re-establish the PDU Session with the updated information.

3.5.5 Conclusion

Framed Routing support within 3GPP standards for 5G networks enables connectivity to an IP network encompassing a range of IPv4 addresses or IPv6 prefixes over a single PDU Session. This feature, commonly used in enterprise connectivity scenarios, involves the association of Framed Routes with PDU Sessions and the flow of information between network components. The reporting of Framed Route information for charging purposes and the dynamic allocation updates further enhance the flexibility and efficiency of framed routing in 5G networks.

4 Methodology

Here we set the goal, define the strategy to achieve it, present the action plan and enumerate the tasks and even the risks. In the appendix, you will find the risk table, the Gant chart and the list of materials for more details.

4.1 Objectives and overview of the selected strategy

The main goal of the project is to validate and test the Framed Route feature in Open5GS and add the management functionality of subnets in the WebGUI API.

The solution must enable the assignment and the management of IPv4/v6 prefixes for each UE through the WebGUI and this change has to be reflected in the user database (MongoDB) and so on to the 5G Core network functions in charge of the PDU session establishment procedure (SMF, UDM and UPF), in the case of UPF it must also be able to receive the framed route TLVs from the SMF (through the PFCP protocol).

Packets directed to any address associated with the specified prefixes and received at the UPF N6 interface will be routed to the appropriate UE through the corresponding GTP tunnel, without involving any NAT operations.

The development project will employ the UERANSIM simulator to replicate the RAN and UE's and Open5gs to replicate the core and the UPF.

4.2 Plan of action

1. Installation and configuration of the Virtual Machines.
2. Validate the framed routing functionality in a state-of-the-art open source 5G core implementation (Open5GS) by setting up an emulated environment (VirtualBox in our case).

   Note: Validation should be done end to end, i.e. from an emulated client behind an emulated 5G UE to the internet/N6 (both IPv4 and IPv6 prefixes).
3. Investigate 3GPP standardised Northbound APIs (e.g. CAPIF) to validate whether the Open5Gs UE Provision API complies with current standards (and possible framed route "touchpoints").
4. Improve/develop the management web interface of the 5G core by adding support for static framed route provisioning (currently only available in db tools).
5. Check and improve possible validations for overlapping routes.

6. Investigate 3GPP standards for possible options for framed route monitoring/charging (e.g. PFCP Session Report requests/response).

7. Validate the scenario on a real 5G network (Altice Labs campus).

4.3 Framed Route Validation

In order to test Framed Routes, we built a mobile 5GC simulated network, WITH Open5GS and UERANSIM.

The following minimum configuration was set as a condition:

- Two UEs have the same DNN and connect to the same DN.
- Two UEs have different Framed Routes. On the UPF VM, make sure to be able to ping the Framed Routes via the IP address (Tunnel GW/uesimtun0) assigned to each UE.

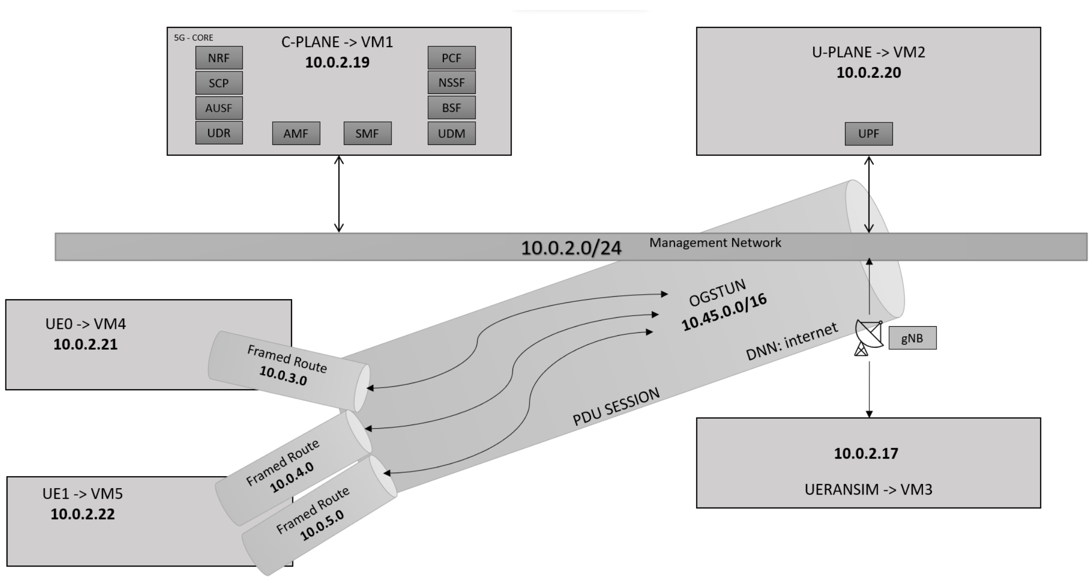


Figure 2: Our Network Illustration

The VM's are as follow:

VM #	SW & Role	IP address	OS	Memory (Min)	HDD (Min)
VM1	Open5GS 5GC C-Plane	10.0.2.19	Ubuntu 20.04	2GB	50GB
VM2	Open5GS 5GC U-Plane	10.0.2.20	Ubuntu 20.04	2GB	50GB
VM3	UERANSIM RAN (gNodeB)	10.0.2.17	Ubuntu 20.04	2GB	50GB
VM4	UERANSIM UE0	10.0.2.21	Ubuntu 20.04	2GB	50GB
VM5	UERANSIM UE1	10.0.2.22	Ubuntu 20.04	2GB	50GB

Figure 3: VM configurations

4.3.1 Core Network Configuration

To perform the validation of Framed Routing we need to change some core configuration files of Open5GS, in order to connect different components of 5G Network.

First of all, we set up C-Plane virtual machine (VM1), which represents the core of 5G Network with the following functions:

- NRF
- SCP
- AMF
- SMF
- AUSF
- UDM
- UDR
- PCF
- NSSF
- BSF

Changes in AMF (Access and Mobility Management Function): the AMF works in conjunction with other core network functions, such as the User Plane Function (UPF) and the Session Management Function (SMF), to ensure efficient communication between user devices and the network. It performs tasks such as authentication, authorisation, and accounting, as well as routing.

```
#
amf:
    sbi:
      - addr: 127.0.0.5
        port: 7777
    ngap:
      - addr: 10.0.2.19
    metrics:
      - addr: 127.0.0.5
        port: 9090
    guami:
      - plmn_id:
          mcc: 001
          mnc: 01
        amf_id:
          region: 2
          set: 1
    tai:
      - plmn_id:
          mcc: 001
          mnc: 01
        tac: 1
    plmn_support:
      - plmn_id:
          mcc: 001
          mnc: 01
        s_nssai:
          - sst: 1
    security:
        integrity_order : [ NIA2, NIA1, NIA0 ]
        ciphering_order : [ NEA0, NEA1, NEA2 ]
    network_name:
        full: Open5GS
    amf_name: open5gs-amf0

#
```

Figure 4: amf.yaml file configuration

NGAP is found on the N2 reference point between gNB and the AMF in order to support both UE and non UE associated services. The NGAP protocol handles the establishment, modification, and termination of PDU sessions by exchanging messages with the network functions involved in the session in order to transfer packets data between the user equipment and the 5G core network.

Changes in SMF (Session Management Function): where we change GTPU(GPRS Tunneling Protocol User Plane) and PFCP(Packet Forwarding Control Protocol) IP's. The SMF uses the GTP-U protocol to manage the data sessions, GTP-U tunnels are used to transport the data between the user equipment and the UPF, the SMF sets up these tunnels and ensures that the necessary resources are allocated to support the data transfer. PFCP is used by the SMF to control and manage the user plane functions, the protocol enables the SMF to communicate with the UPF for session establishment, modification, and termination.

To conclude the core network configuration, we need to set up UPF, that runs on U-PLane(VM2), in order to established connections with C-Plane functions and also with gNodeB on RAN machine (VM3).

```yaml
smf:

    sbi:
      - addr: 127.0.0.4
        port: 7777
    pfcp:
      - addr: 10.0.2.19
    gtpc:
      - addr: 127.0.0.4
    gtpu:
      - addr: 10.0.2.19
    metrics:
      - addr: 127.0.0.4
        port: 9090
    subnet:
      - addr: 10.45.0.1/16
        dnn: internet
    dns:
      - 8.8.8.8
      - 8.8.4.4
    mtu: 1400
    ctf:
      enabled: auto
    freeDiameter: /etc/freeDiameter/smf.conf
#
```

Figure 5: smf.yaml file configuration

```yaml
upf:
    pfcp:
      - addr: 10.0.2.20
    gtpu:
      - addr: 10.0.2.20
    subnet:
      - addr: internet
      - dev: ogstun
    metrics:
      - addr: 127.0.0.7
        port: 9090

```

Figure 6: upf.yaml file configuration

```yaml
upf:
    pfcp:
      - addr: 10.0.2.20
        dnn: internet

```

Figure 7: upf.yaml file configuration

4.3.2 UERANSIM Configuration

- RAN (gNodeB) - Configuration (VM3): The gNB connects to the 5G core network through the NG-RAN (Next Generation Radio Access Network) interface, which enables communication between the gNB and the core network functions.

open5gs-gnb.yaml
/home/fiveg/UERANSIM/config

```yaml
mcc: '001'          # Mobile Country Code value
mnc: '01'           # 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: 10.0.2.17   # gNB's local IP address for Radio Link Simulation (Usually same with local IP)
ngapIp: 10.0.2.17   # gNB's local IP address for N2 Interface (Usually same with local IP)
Help Ip: 10.0.2.17   # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
  - address: 10.0.2.19
    port: 38412

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

# Indicates whether or not SCTP stream number errors should be ignored.
ignoreStreamIds: true
```

Figure 8: UERANSIM - open5gs-gnb.yaml file configuration

The gNB sends signaling messages to the AMF over the NG-C interface to initiate and manage the connection between the UE and the core network. These messages include information such as UE identification, location, and requested services. The AMF uses this information to authenticate the UE, allocate network resources, and establish the necessary network functions, such as the SMF and UPF, to support the requested services.

- UE0 and UE1 Configuration(VM4 and VM5): where we need to set up gNBSearchList to connect the user equipment with gNB(VM3).

open5gs-ue0.yaml
/home/fiveg/UERANSIM/config

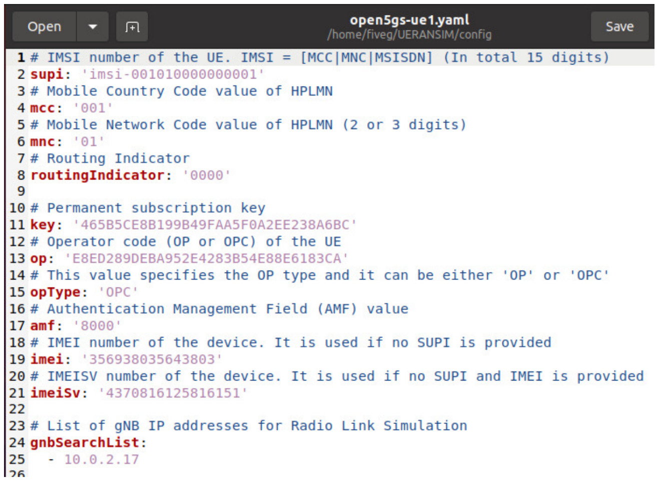
```yaml
mcc: '001'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '01'
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '465B5CE8B199B49FAA5F0A2EE238A6BC'
# Operator code (OP or OPC) of the UE
op: 'E8ED289DEBA952E4283B54E88E6183CA'
# 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:
  - 10.0.2.17

```

Figure 9: UERANSIM - open5gs-ue0.yaml file configuration



open5gs-ue1.yaml
/home/fiveg/UERANSIM/config

```yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-001010000000001'
# Mobile Country Code value of HPLMN
mcc: '001'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '01'
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '465B5CE8B199B49FAA5F0A2EE238A6BC'
# Operator code (OP or OPC) of the UE
op: 'E8ED289DEBA952E4283B54E88E6183CA'
# 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:
  - 10.0.2.17
```

Figure 10: UERANSIM - open5gs-ue1.yaml file configuration

4.3.3 WebGUI Subscriptions

One of the main points of our project is to add and manage users on open5GS WebGUI. WenGUI is a web-based user interface for the administration and configuration of the Open5GS core network components. By using WebGUI network administrators can manage subscribers via IMSI, KEY and OPC. With these parameters, we can set up the appropriate PDU sessions, ensuring access to DN for each subscriber via the appropriate interfaces. So we need to set the IMSI correspondence to our users. We have two ways to do this: the first one is directly on WebGUI interface, and the other one is by mogoDB Compass application that allows to change directly the subscriber's information on the database. We prefer to add the subscriber's information directly on mongoDB Compass.

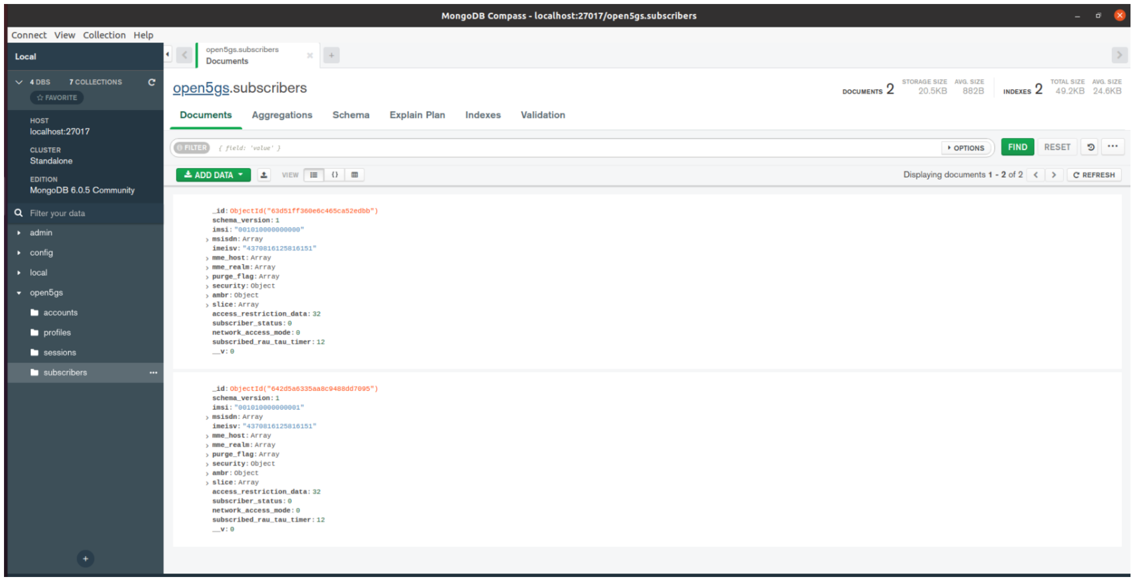


Figure 11: MongoDB Compass Interface

We add the following on subscribers' JSON file with the essential parameters to be added to the Framed Routes to users subscribers.

Once we add the subscribers file on mongoDB Compass, the users will automatically appear on the WebGUI interface as we can see in the image below:

```
"slice": [
  {
    "sst": 1,
    "default_indicator": true,
    "session": [
      {
        "name": "internet",
        "type": 3,
        "qos": {
          "index": 9,
          "arp": {
            "priority_level": 8,
            "pre_emption_capability": 1,
            "pre_emption_vulnerability": 1
          }
        },
        "ambr": {
          "downlink": {
            "value": 1,
            "unit": 3
          },
          "uplink": {
            "value": 1,
            "unit": 3
          }
        },
        "_id": {
          "$oid": "642d5a6335aa8c9488dd7097"
        },

        "ipv4_framed_routes": ["10.0.4.0/24", "10.0.5.0/24"],

        "pcc_rule": []
      }
    ],
    "_id": {
      "$oid": "642d5a6335aa8c9488dd7096"
    }
  }
],
```

Figure 12: Subscribers file, UE1 example

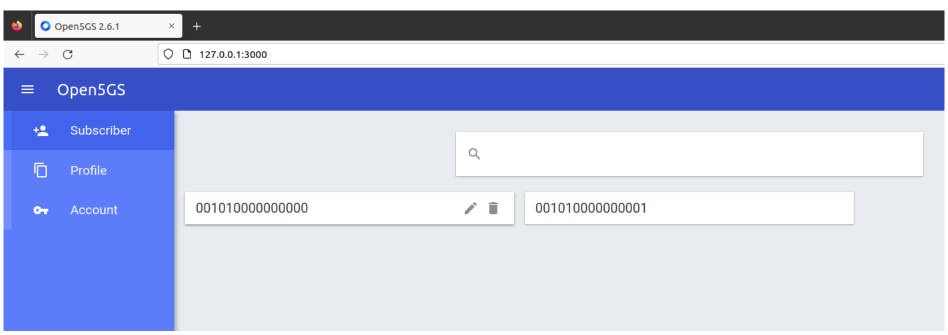


Figure 13: WebGUI Interface

4.3.4 Validation

After all these changes have been made, we are able to proceed to the validation of the Frame Routing.
We need to start and initialise the virtual machines in the following order:

1. Start C-Plane (VM1)

2. Start U-Plane (VM2)

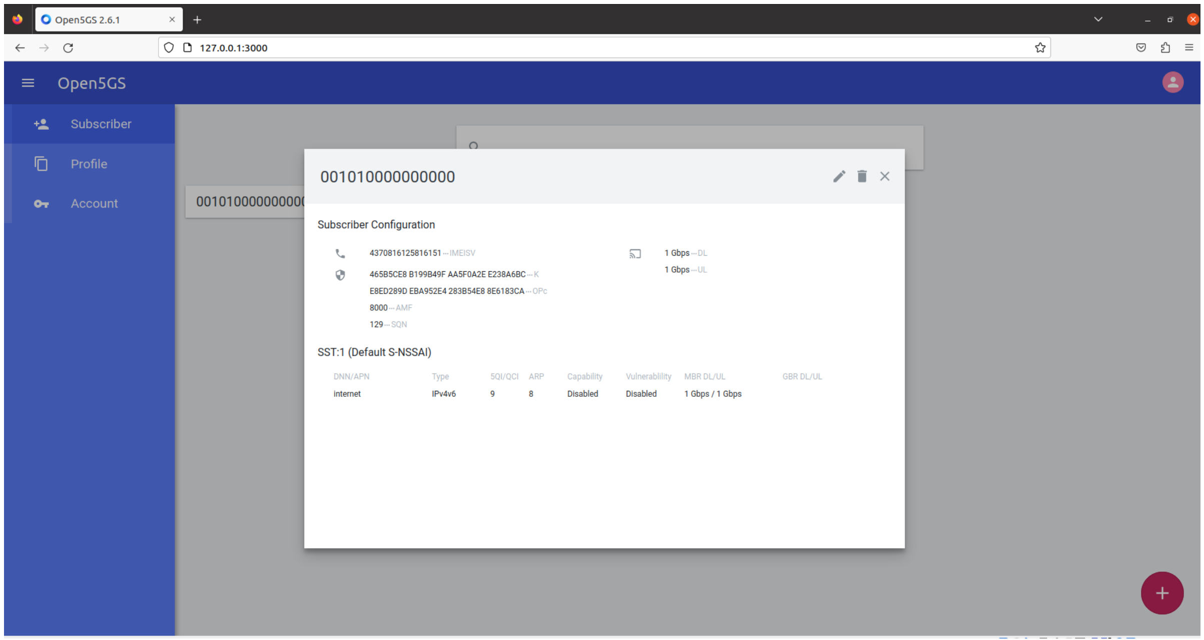


Figure 14: WebGUI Interface - UE subscriber information

3. Start gNB (VM3)

4. Start UE0 (VM4)

5. Start UE1 (VM5)

First of all, we need to start open5gs daemons/services, for that we could use the following procedure:

- systemctl restart open5gs-mmed;
- systemctl restart open5gs-sgwcd;
- systemctl restart open5gs-smfd;
- systemctl restart open5gs-amfd
- systemctl restart open5gs-sgwud
- systemctl restart open5gs-hssd
- systemctl restart open5gs-pcrfd
- systemctl restart open5gs-nrfd
- systemctl restart open5gs-scpd
- systemctl restart open5gs-ausfd
- systemctl restart open5gs-udmd

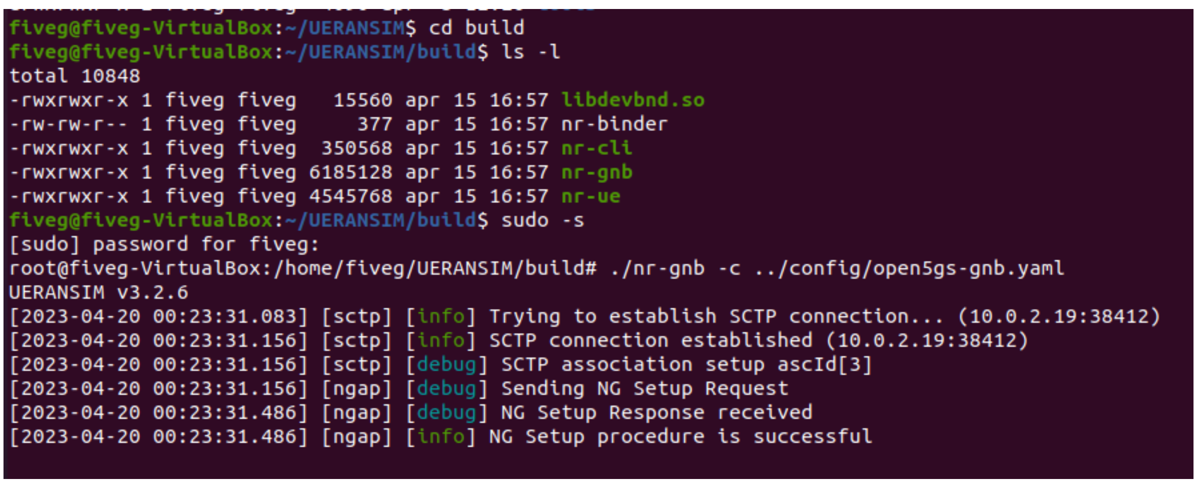
- systemctl restart open5gs-pcfd
- systemctl restart open5gs-nssfd
- systemctl restart open5gs-bsfd
- systemctl restart open5gs-udrd
- systemctl restart open5gs-webui

We should check if all daemons are running using *systemctl status open5gs-**.

Next, we need to start UPF daemon on U-Plane machine with:

- systemctl restart open5gs-mmed;

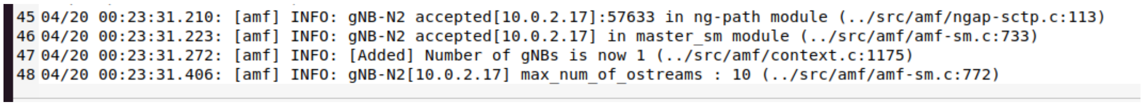
Once the services of C-PLane and U-Plane are running without any error or warning, we are able to start gNB on UERANSIM-VM3, which can be started by the following command: *./nr-gnb -c ../config/open5gs-gnb.yaml*.
Case it started correctly, we are supposed to see the following logs:



```
fiveg@fiveg-VirtualBox:~/UERANSIM$ cd build
fiveg@fiveg-VirtualBox:~/UERANSIM/build$ ls -l
total 10848
-rwxrwxr-x 1 fiveg fiveg   15560 apr 15 16:57 libdevbnd.so
-rw-rw-r-- 1 fiveg fiveg     377 apr 15 16:57 nr-binder
-rwxrwxr-x 1 fiveg fiveg  350568 apr 15 16:57 nr-cli
-rwxrwxr-x 1 fiveg fiveg 6185128 apr 15 16:57 nr-gnb
-rwxrwxr-x 1 fiveg fiveg 4545768 apr 15 16:57 nr-ue
fiveg@fiveg-VirtualBox:~/UERANSIM/build$ sudo -s
[sudo] password for fiveg:
root@fiveg-VirtualBox:/home/fiveg/UERANSIM/build# ./nr-gnb -c ../config/open5gs-gnb.yaml
UERANSIM v3.2.6
[2023-04-20 00:23:31.083] [sctp] [info] Trying to establish SCTP connection... (10.0.2.19:38412)
[2023-04-20 00:23:31.156] [sctp] [info] SCTP connection established (10.0.2.19:38412)
[2023-04-20 00:23:31.156] [sctp] [debug] SCTP association setup ascId[3]
[2023-04-20 00:23:31.156] [ngap] [debug] Sending NG Setup Request
[2023-04-20 00:23:31.486] [ngap] [debug] NG Setup Response received
[2023-04-20 00:23:31.486] [ngap] [info] NG Setup procedure is successful
```

Figure 15: Start logs of gNB

We also can confirm the gNB initialisation on the AMF logs file, on C-Plane(VM1) side.

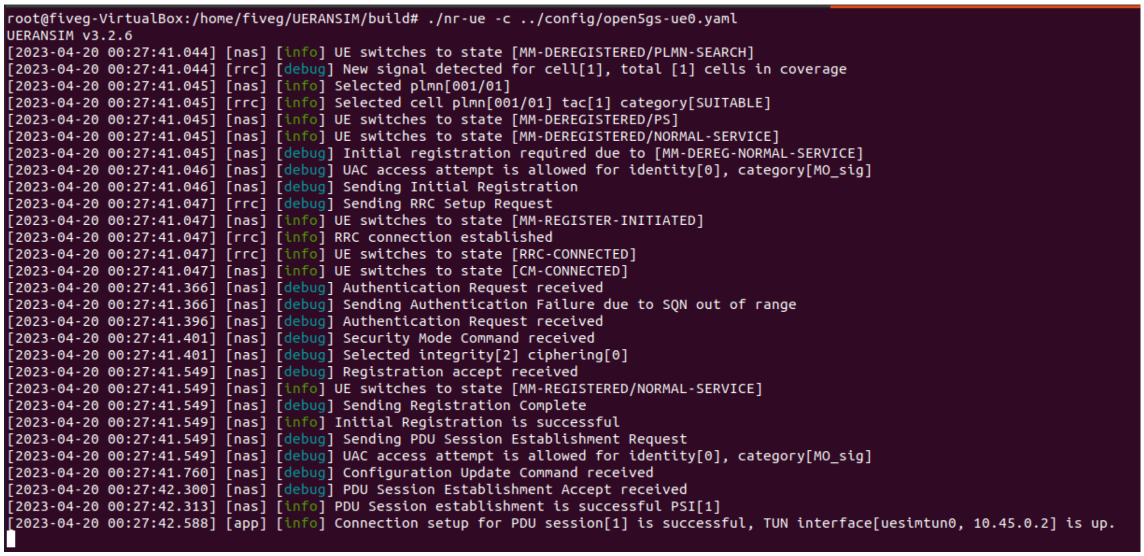


```
45 04/20 00:23:31.210: [amf] INFO: gNB-N2 accepted[10.0.2.17]:57633 in ng-path module (../src/amf/ngap-sctp.c:113)
46 04/20 00:23:31.223: [amf] INFO: gNB-N2 accepted[10.0.2.17] in master_sm module (../src/amf/amf-sm.c:733)
47 04/20 00:23:31.272: [amf] INFO: [Added] Number of gNBs is now 1 (../src/amf/context.c:1175)
48 04/20 00:23:31.406: [amf] INFO: gNB-N2[10.0.2.17] max_num_of_ostreams : 10 (../src/amf/amf-sm.c:772)
```

Figure 16: Start logs of gNB

With the image above we can see that the connection established between gNodeB and AMF was done by the N2 interface, this is essential because before a service can be accessed, the UE must be connected to the network. That‘s why N2 is so important because it handles control-plane signalling.

So now, we are able to connect the first user UE0(VM4). It can be done by the following command *./nr-ue -c ../config/open5gs-ue0.yaml*. This will register the UE with 5GC and establish a PDU session.



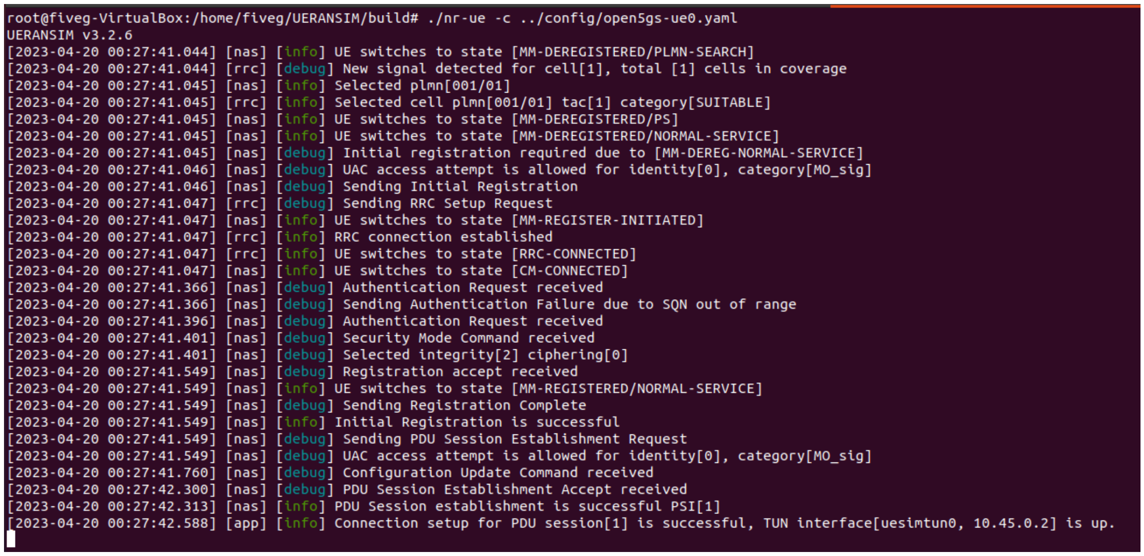
```
root@fiveg-VirtualBox:/home/fiveg/UERANSIM/build# ./nr-ue -c ../config/open5gs-ue0.yaml
UERANSIM v3.2.6
[2023-04-20 00:27:41.044] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2023-04-20 00:27:41.044] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2023-04-20 00:27:41.045] [nas] [info] Selected plmn[001/01]
[2023-04-20 00:27:41.045] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.045] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2023-04-20 00:27:41.046] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.046] [nas] [debug] Sending Initial Registration
[2023-04-20 00:27:41.047] [rrc] [debug] Sending RRC Setup Request
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2023-04-20 00:27:41.047] [rrc] [info] RRC connection established
[2023-04-20 00:27:41.047] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-04-20 00:27:41.366] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.366] [nas] [debug] Sending Authentication Failure due to SQN out of range
[2023-04-20 00:27:41.396] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.401] [nas] [debug] Security Mode Command received
[2023-04-20 00:27:41.401] [nas] [debug] Selected integrity[2] ciphering[0]
[2023-04-20 00:27:41.549] [nas] [debug] Registration accept received
[2023-04-20 00:27:41.549] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.549] [nas] [debug] Sending Registration Complete
[2023-04-20 00:27:41.549] [nas] [info] Initial Registration is successful
[2023-04-20 00:27:41.549] [nas] [debug] Sending PDU Session Establishment Request
[2023-04-20 00:27:41.549] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.760] [nas] [debug] Configuration Update Command received
[2023-04-20 00:27:42.300] [nas] [debug] PDU Session Establishment Accept received
[2023-04-20 00:27:42.313] [nas] [info] PDU Session establishment is successful PSI[1]
[2023-04-20 00:27:42.588] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.45.0.2] is up.
```

Figure 17: Start logs of UE0

At last, we start the UE1 with the following command *./nr-ue -c ../config/open5gs-ue1.yaml*, and if all going well we have completed the starting procedure.

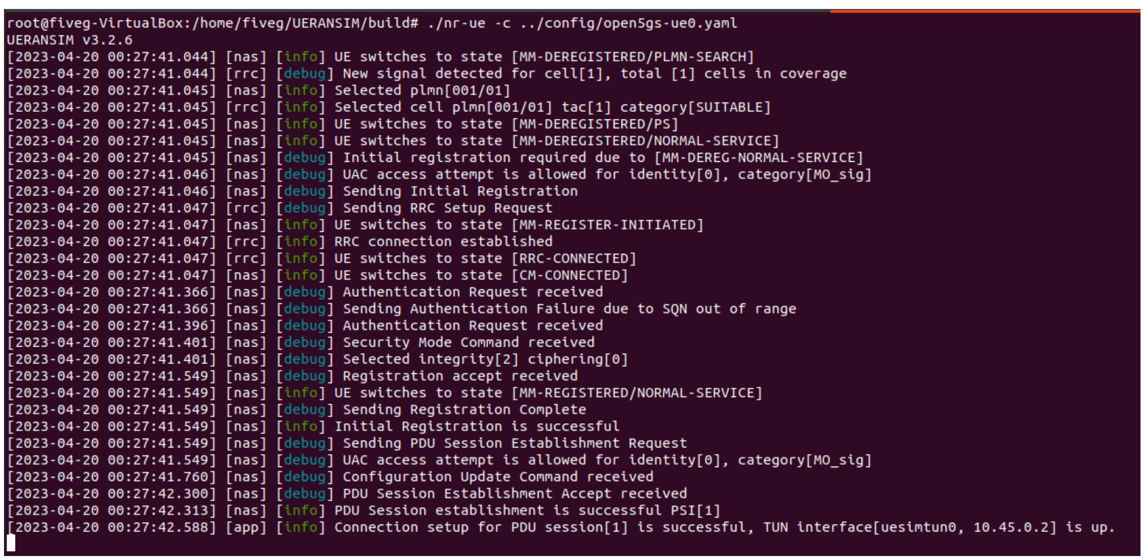
Now we need to do some Network configurations which consist to configure the TUNnel interface (ogstun - 10.45.0.0) and NAPT on U-Plane(VM2), and set the respectively Framed Route IP adrress (ex:10.0.3.0//10.0.4.0//10.0.5.0) and the routing DN(10.45.0.0) to the uesimtun0 interface on UE0(VM4) and UE1(VM5).

So now, we have all configurations done. To confirm the framed route we have done some PING tests.



```
root@fiveg-VirtualBox:/home/fiveg/UERANSIM/build# ./nr-ue -c ../config/open5gs-ue0.yaml
UERANSIM v3.2.6
[2023-04-20 00:27:41.044] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2023-04-20 00:27:41.044] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2023-04-20 00:27:41.045] [nas] [info] Selected plmn[001/01]
[2023-04-20 00:27:41.045] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.045] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2023-04-20 00:27:41.046] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.046] [nas] [debug] Sending Initial Registration
[2023-04-20 00:27:41.047] [rrc] [debug] Sending RRC Setup Request
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2023-04-20 00:27:41.047] [rrc] [info] RRC connection established
[2023-04-20 00:27:41.047] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-04-20 00:27:41.366] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.366] [nas] [debug] Sending Authentication Failure due to SQN out of range
[2023-04-20 00:27:41.396] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.401] [nas] [debug] Security Mode Command received
[2023-04-20 00:27:41.401] [nas] [debug] Selected integrity[2] ciphering[0]
[2023-04-20 00:27:41.549] [nas] [debug] Registration accept received
[2023-04-20 00:27:41.549] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.549] [nas] [debug] Sending Registration Complete
[2023-04-20 00:27:41.549] [nas] [info] Initial Registration is successful
[2023-04-20 00:27:41.549] [nas] [debug] Sending PDU Session Establishment Request
[2023-04-20 00:27:41.549] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.760] [nas] [debug] Configuration Update Command received
[2023-04-20 00:27:42.300] [nas] [debug] PDU Session Establishment Accept received
[2023-04-20 00:27:42.313] [nas] [info] PDU Session establishment is successful PSI[1]
[2023-04-20 00:27:42.588] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.45.0.2] is up.
```

Figure 18: Start logs of UE0 on amf log file




```
root@fiveg-VirtualBox:/home/fiveg/UERANSIM/build# ./nr-ue -c ../config/open5gs-ue0.yaml
UERANSIM v3.2.6
[2023-04-20 00:27:41.044] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2023-04-20 00:27:41.044] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2023-04-20 00:27:41.045] [nas] [info] Selected plmn[001/01]
[2023-04-20 00:27:41.045] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2023-04-20 00:27:41.045] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.045] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2023-04-20 00:27:41.046] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.046] [nas] [debug] Sending Initial Registration
[2023-04-20 00:27:41.047] [rrc] [debug] Sending RRC Setup Request
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2023-04-20 00:27:41.047] [rrc] [info] RRC connection established
[2023-04-20 00:27:41.047] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2023-04-20 00:27:41.047] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-04-20 00:27:41.366] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.366] [nas] [debug] Sending Authentication Failure due to SQN out of range
[2023-04-20 00:27:41.396] [nas] [debug] Authentication Request received
[2023-04-20 00:27:41.401] [nas] [debug] Security Mode Command received
[2023-04-20 00:27:41.401] [nas] [debug] Selected integrity[2] ciphering[0]
[2023-04-20 00:27:41.549] [nas] [debug] Registration accept received
[2023-04-20 00:27:41.549] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2023-04-20 00:27:41.549] [nas] [debug] Sending Registration Complete
[2023-04-20 00:27:41.549] [nas] [info] Initial Registration is successful
[2023-04-20 00:27:41.549] [nas] [debug] Sending PDU Session Establishment Request
[2023-04-20 00:27:41.549] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-04-20 00:27:41.760] [nas] [debug] Configuration Update Command received
[2023-04-20 00:27:42.300] [nas] [debug] PDU Session Establishment Accept received
[2023-04-20 00:27:42.313] [nas] [info] PDU Session establishment is successful PSI[1]
[2023-04-20 00:27:42.588] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.45.0.2] is up.
```


Figure 19: Start logs of UE0 on upf log file

- Ping Framed Route IP (10.0.3.0) from U-Plane by TUN interface (ogstun)and confirm tcpdump running on UE0(VM4):

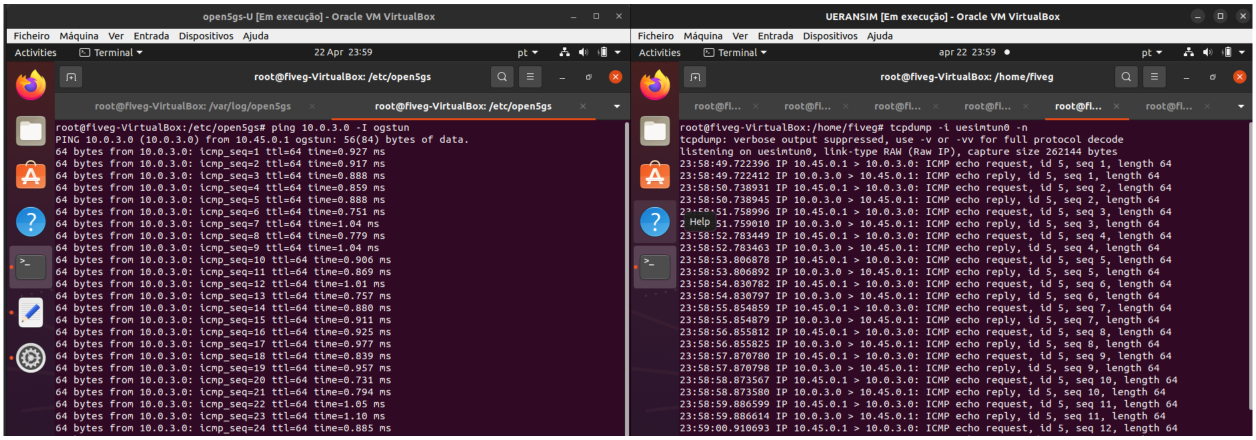


Figure 20: Ping UE0 Framed Route

4.4 Northbound API's

To validate whether the Open5Gs UE Provision API complies with the current 3GPP standardised Northbound APIs, we need to first examine the relevant standards and specifications. Because CAPIF is a regularly used Northbound API, we can begin by analyzing its specifications to discover the important touchpoints with which the Open5Gs UE Provision API must comply.

The UE Provision API must conform with the following important touchpoints, according to the CAPIF specifications:

1. Common data model - The UE Provision API must employ a common data model that allows information to be exchanged between different network operations.

2. Access control - The API must have measures to ensure that only authorised applications and services have access to network resources.

3. Transaction management - The API must have transaction management capabilities to ensure that transactions are correctly completed and that any problems are handled appropriately.

4. Logging - The API must have logging features to allow for the recording of transactions and to assist in diagnosing and troubleshooting any issues that may develop.

5. Function abstraction - The API must include abstraction functions that allow programs to access various network functions and resources without having to understand the underlying systems' details.

To determine whether the Open5Gs UE Provision API meets these requirements, we must study it. The Open5Gs UE Provision API is intended to allow for the provisioning of User Equipment (UE) data in a 5G network. It provides RESTful APIs that let programs create, read, update, and delete UE data.

Based on our findings, the Open5Gs UE Provision API appears to match the CAPIF-specified critical touchpoints. It employs a common data architecture that adheres to 3GPP standards and includes access control, transaction management, and logging systems to ensure transaction security and integrity. It also has function abstraction methods, which allow programs to access various network functions and resources without having to understand the underlying systems.

Concluding, the Open5Gs UE Provision API appears to adhere to the current 3GPP-defined Northbound APIs, including CAPIF. It meets the CAPIF key touchpoints and provides a consistent method of provisioning UE information in a 5G network.

5 Major Difficulties and Corrective Actions

We have faced two major difficulties during the elaboration of the project:

- Firstly, we had a lack of knowledge about the concepts and definitions approached. So we need to focus on studying the concepts, and activities that cost us a lot of time because we never had in touch with these topics.
- Secondly, we faced several errors and bugs during the installation and setting of the simulated environment (open5GS and UERANSIM). The simulated environment has limitations and we need to find out ways to turn around these limitations.

6 Conclusions

Finally, we have reached that this project has been a valuable opportunity to acquire both practical and theoretical knowledge on communications technology area, focusing on 5G. Throughout the project's development, we widely explored the application of Framed Routes in 5G networks to overcome the limitations imposed by NAT.

As we gain ground, we obta in a deeper understanding of the sophistication involved in configuring 5G networks, from the CORE elements functionalities to the management of user information through a database using Open5GS API.

Throughout the project, we consistently make a run at our technical skills, demanding a meticulous expansion of our knowledge on communication network protocols, router configurations, and fundamental communication processes. However, these challenges served as invaluable opportunities for growth, honing our problem-solving abilities.

Regrettably, not all of the project's objectives were successfully achieved, notably the enhancement of sub-network management in the web user interface. Nevertheless, we acknowledge the accomplishments we have made, particularly in validating the effectiveness of the Framed Route.

In conclusion, this project has provided us with substantial value, enriching our comprehension of the intricacies involved in configuring and establishing communication networks. Furthermore, it has cultivated a enthusiastic interest in the implementation of novel communication technologies, exemplified by the Framed Route concept and it countless advantages. We firmly believe that this endeavour has furnished us with vital lessons and knowledge that will guide our professional carreer as electrical engineers.

7 Acknowledgements

We would like to express our gratitude to the University of Coimbra, especially lecturers Maria do Carmo Raposo de Medeiros and Henrique Leonel Gomes, as well as Altice Labs engineers Francisco Fontes and Miguel Freitas.