Information Technology Course
Module Software Engineering
by Armin Lehman

Open5GS & OpenAirInterface UE / RAN, simple UPF by focous on accessing to File_sharing_Platform, Streaming and Sip_Call_Server Sample Configuration

Mahdieh Pirmoradina
mahdieh.pirmoradian@stud.fra-uas.de

Nusrat Jahan Sumi
nusrat.sumi@stud.fra-uas.de

***Abstract*—The network is designed to provide high-speed data connectivity to users through the 5G wireless standard. The network architecture comprises of a 5G core network, which includes a single User Plane Function (UPF) that serves as a central data traffic hub for all users. The Open-Air Interface provides an open-source implementation of the 5G core network, enabling flexibility in terms of customization and deployment. For the network's implementation, we utilized a virtual box. We have three virtual boxes that simulate the core network, OAI Ran (which is both a UE and an open-air interface), and the user plane function. This network required connectivity to file sharing, a Sip call server, and a streaming server. Overall, the network implementation based on 5G and Open-Air Interface with a single UPF provides a robust and flexible platform for delivering high-speed data connectivity and enabling innovative use cases.**

Keywords— 5gs, open5gs, 5G-RAN, OAI-RAN, OpenAirInterface, Control Plane, SMF, NRF, PCF, UPF, sip, voip, fiel sharing, Service Management Function, Network Repository Function, Policy Control Function, user plane function.

I. INTRODUCTION

The 5G wireless standard has revolutionized the telecommunications industry, providing higher data speeds, lower latency, and improved network reliability. With the increased demand for high-speed connectivity and emerging use cases such as autonomous vehicles and remote healthcare, network operators are looking to deploy 5G networks that can deliver these services efficiently and cost-effectively.

One way to achieve this is through the use of Open-Air Interface, an open-source implementation of the 5G core network that enables customization and flexibility in deployment. Open Air Interface can be integrated with existing hardware and software, making it an attractive option for network operators looking to upgrade their infrastructure.

In this context, the implementation of a network based on 5G and Open-Air Interface with a single User Plane Function (UPF) is a significant step towards building a robust and scalable network architecture. The UPF serves as the central data traffic hub for all users, processing, and routing data packets between the user equipment (UE) and the external network.

The network can support various applications such as ultra-reliable and low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC). The single UPF can ensure quality of service (QoS) management, packet inspection, and policy enforcement, providing a seamless and secure data flow for mission-critical applications.

In summary, the implementation of a network based on 5G and Open-Air Interface with a single UPF can enable network operators to deliver high-speed connectivity, support emerging use cases, and achieve operational efficiency. This can have a significant impact on various industries, from healthcare to transportation, and pave the way for a new era of connectivity and innovation.

II. Introduction to 5GS Network Components Functionality

In a 5G network, several core network components work together to provide end-to-end connectivity and service delivery to users. Here is a high-level overview of how these components work together:

- User Equipment (UE): The UE is the user's device, such as a smartphone or tablet, that connects to the 5G network.
- Radio Access Network (RAN): The RAN consists of base stations that provide wireless connectivity between the UE and the 5G core network.
- 5G Core Network: The 5G core network is a set of network functions that provide key services such as authentication, session management, policy control, and charging.
- Access and Mobility Management Function (AMF): The AMF is responsible for managing user authentication, authorization, and mobility within the network.
- Session Management Function (SMF): The SMF is responsible for establishing and managing data sessions between the user's device and the network.
- Policy Control Function (PCF): The PCF manages policy rules that control how network resources are allocated and used by different network services and users.
- Network Repository Function (NRF): The NRF maintains a registry of available network functions and their associated capabilities, which other network functions can query to discover available services.
- User Plane Function (UPF): The UPF is responsible for routing data packets between the user's device and the network, and for enforcing policy rules related to QoS and network resources.

These components work together to provide end-to-end connectivity and service delivery to users. For example, when a user initiates a data session, the UE communicates with the RAN to establish a wireless connection to the 5G core network. The AMF authenticates the user and authorizes their access to the network, while the SMF establishes and manages the data session. The PCF enforces policy rules related to QoS and network resources, while the UPF routes data packets between the user's device and the network. The NRF helps other network functions discover available services and make informed service selection decisions. Overall, these components work together to provide a flexible, service-oriented network architecture that can support a wide range of use cases and service requirements.

III. REQUIREMENTS

Here is the Requirements for our Network.

Machine ID	Function	IP/MAC Address enp0s3	IP/MAC Address enp0s4	RAM	CPU	HDD	Operating System
Machine 01	UE/RAN Simulation	192.168.0.120 52:54:00:12:34:51	192.168.0.125 52:54:00:12:34:61	8GB	2	40GB	Ubuntu 18.04
Machine 02	C-Plane	192.168.0.121 52:54:00:12:34:52	192.168.0.126 52:54:00:12:34:62	8GB	2	40GB	Ubuntu 18.04
Machine 03	UPF	192.168.0.122 52:54:00:12:34:53	192.168.0.127 52:54:00:12:34:63	8GB	2	40GB	Ubuntu 18.04

The system ’s main IP address is 192.168.0.0/24, indicating that all machines within the system are interconnected via this designated network range.

IV. STRUCTURE

The simulated environment that was created in this project is as follows.

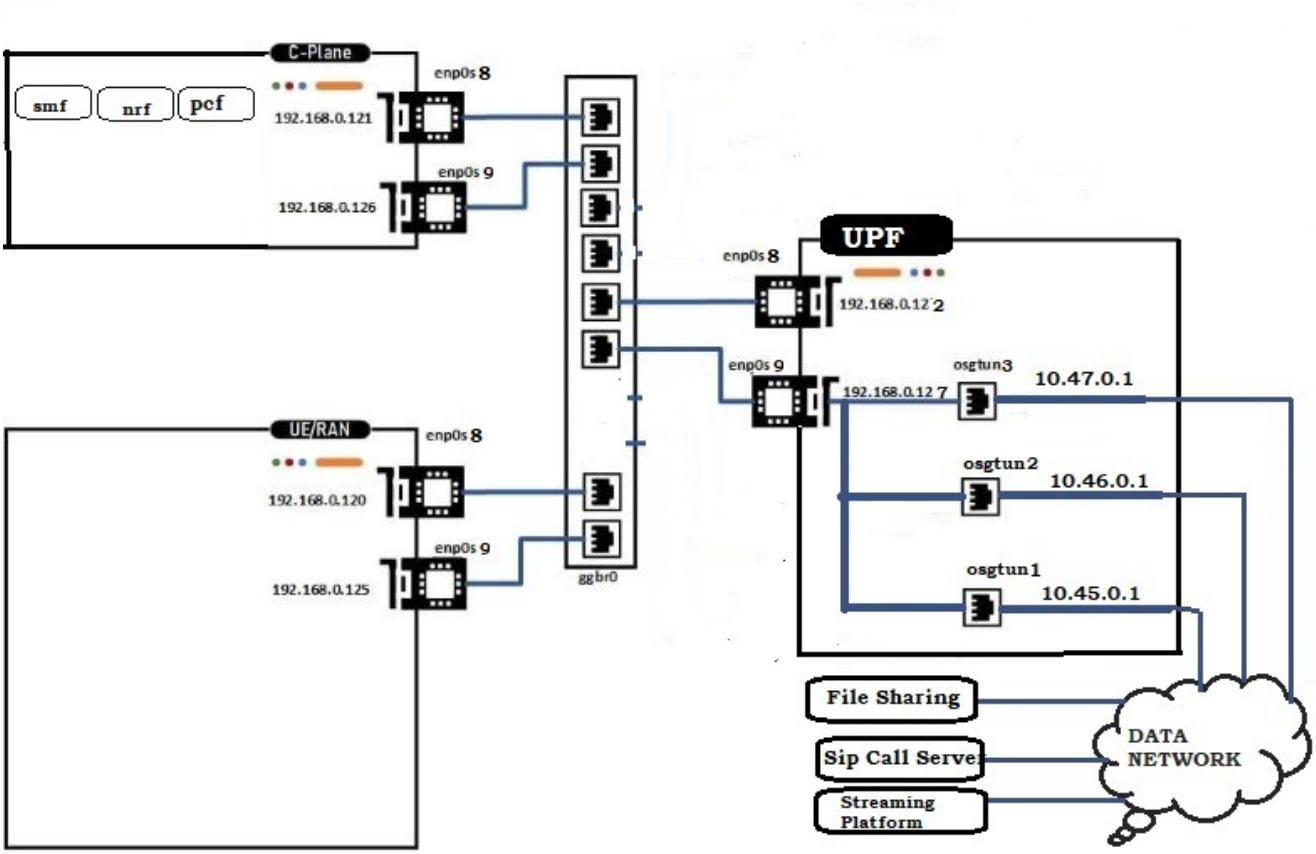


Figure 1 illustrates the overall setting used for the virtual machines to setup the system.

Note that in our C-Plane machine Its asked to config SMF, NRF and PCF.

This networked is configues to have access to Sip-Call-Server, Streaming and File-Sharing Service.

I. DESCRIPTION OF TAP DEVICES

Tap devices are often used in virtualization technologies such as containers and virtual machines to imitate a network interface that so many other virtual devices can access.

A tap device is a virtual network interface that allows data to be diverted from one network device to another.

It is also known as a virtual tap interface or TAP interface.

In virtualization environments, tap devices are used to connect virtual machines or containers to a virtual network, enabling communication between virtual devices and the outside world.

The exact number and configuration of tap devices used in Open5GS will depend on the specific use case and network topology being simulated.

here according to the project description, we configured 2 tap devices for each virtual machine.

for example, each upf has 3 Tap devices because it has 3 interfaces N3, N4 and N6 respectively for connecting to OAI-RAN, 5GS-core-Network and Internet.

or UE-RAN has 3 tap devices because it has 3 interfaces named N1 for connecting UE from UE-RAN to the 5GS-Core-Network, N2 for connecting OAI-RAN from UE-RAN to the 5GS-Core-Network and N3 for connecting UE-RAN to the UPF.

or C-Plane virtual machine has 3 tap devices because it has 3 interfaces named N1 for connecting to UE from UE-RAN, N2 for connecting to OAI-RAN from UE-RAN and N3 for connecting UE-RAN to the UPF.

II. INITIALIZING OF TAP DEVICES

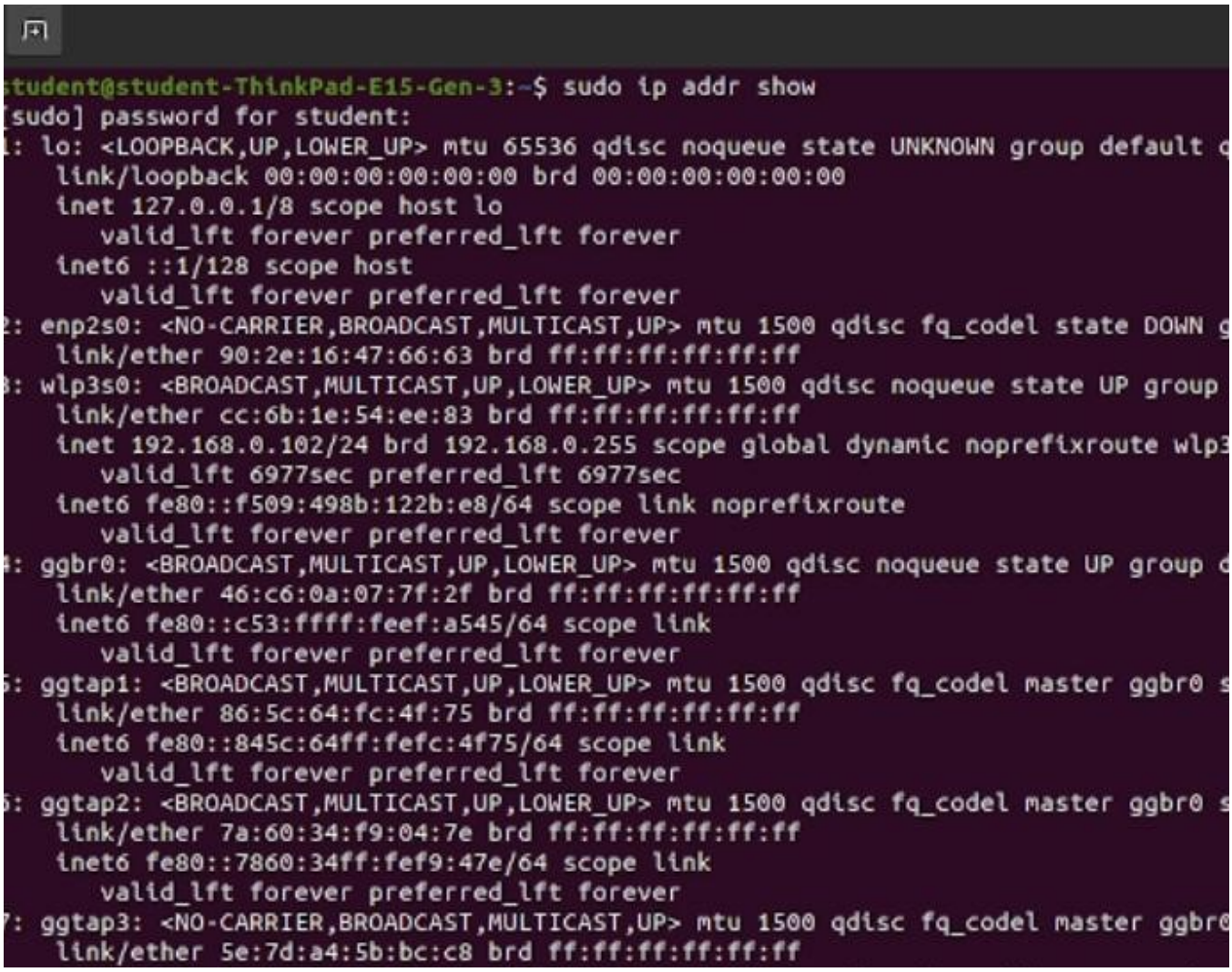
Here we have used Tap devices for data transmission between one network device to another. In addition, to have access from this 5G network elements to the Internet.

To create a tap device, we use the following script, which iterates for 6 times and create a new tap device in each iteration. We dedicate two tap devices for each virtual machine.

```
sudo brctl addbr ggbr0
sudo ip link set ggbr0 up
for i in 1 2 3 4 5 6
do
    sudo ip tuntap add dev ggtap$i mode tap user student
    sudo ip link set dev ggtap$i up
    sudo brctl addif ggbr0 ggtap$i
done
```

We have used this code on our own main system which is an Ubuntu system.

Here is the result of executing the above code.



Note that each time after termination of your system you should run this code again because after system termination these settings will disappear.

- **Code description**

  - sudo brctl addbr ggbr0

    With this command, we will create a virtual bridge named "ggbr0".
    This command is used to create a virtual bridge named "ggbr0" using the brctl tool on a Linux system.
    The brctl tool is used to manage and manipulate the Linux bridge, which is a layer 2 virtual device that can be used to connect multiple network interfaces together, creating a single virtual network interface.
    The addbr option is used to create a new bridge and the ggbr0 is the name of the bridge.

  - sudo ip tuntap add dev ggtap${i} mode tap u

    This is a command in Linux to create a TUN/TAP device.
    The options used in this command are:
    dev ggtap${i} specifies the name of the TUN/TAP device to be created. In this case, the device name is "ggtap${i}".
    mode tap specifies the type of device to create, either TUN (network TUNnel) or TAP (Ethernet TAP).
    In this case, TAP mode is selected.

u specifies that the device should be created with user ownership, meaning that it can be used by a non-root user.
This command requires root privileges, which is why sudo is used to run it with elevated permissions.

- sudo ip link set dev ggtap${i} up

  This is a command in Linux to activate a network interface.

  The options used in this command are:

  dev ggtap${i} specifies the name of the network interface to be activated. In this case, the interface name is "ggtap${i}".

  up specifies the state of the interface, either "up" to activate it or "down" to deactivate it. In this case, the interface is set to "up" to activate it.

  This command requires root privileges, which is why sudo is used to run it with elevated permissions.
- Once the interface is activated, it becomes visible to the system and can be used for network communication.

- sudo brctl addif ggbr0 ggtap${i}

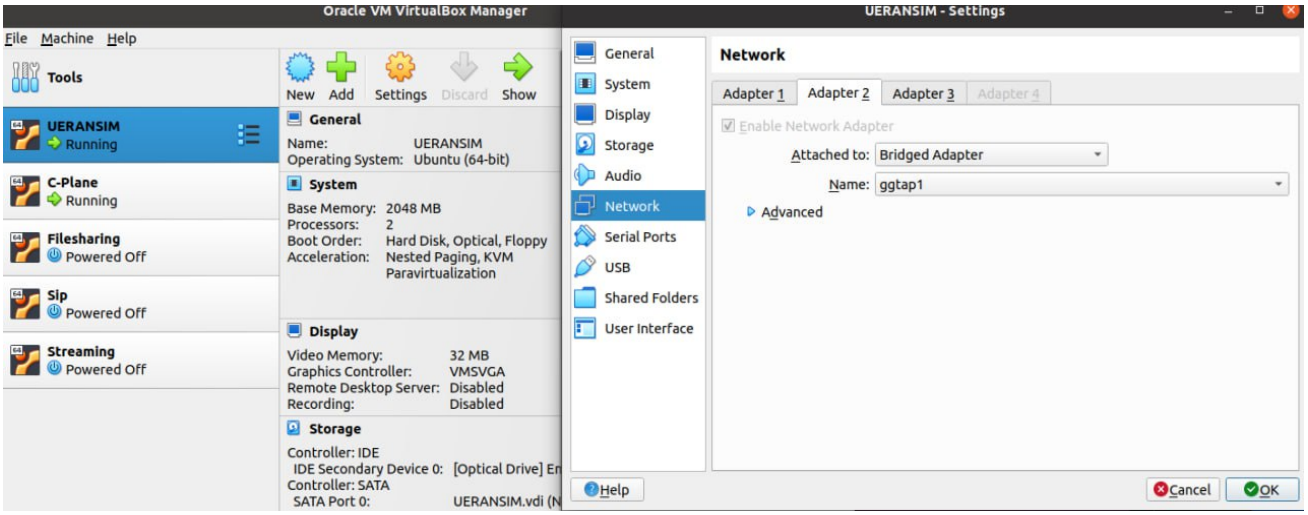
  This is a command in Linux to add a network interface to a bridge device.
  The options used in this command are:
  ggbr0 specifies the name of the bridge device. A bridge device is a software-defined network switch that allows multiple network interfaces to be connected together.
  ggtap${i} specifies the name of the network interface to be added to the bridge.
  This command requires root privileges, which is why sudo is used to run it with elevated permissions.
  After the network interface is added to the bridge, it becomes part of the bridge and will participate in the network communication through the bridge.

As we run the script using the `ip` command, we can see the tap devices are created and added to our bridge properly.

```
$ sudo ip addr show | egrep ggtap*
135: ggtap1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
136: ggtap2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
137: ggtap3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
138: ggtap4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
139: ggtap5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
141: ggtap6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ggbr0 state UP group default qlen 1000
```

III. SETTING THE INTERFACES FOR EACH VIRTUAL MACHINE

Here is the setting process for UERANSIM as an example.



IV. CONFIGURING VIRTUAL MACHINES BY CODE

```
#!/usr/bin/env bash

set -ex


machineName=$1
tap1=$machineName
tap2=$((machineName+5))

STTY_SETTINGS="$( stty -g )"

stty intr ^]
stty susp ^]

cnt=$(($cnt+1))
# Parameters.
id=ubuntu-18.04.6-desktop-amd64
disk_img="base.img"
disk_img_snapshot="machine-0${machineName}.snapshot.qcow2"

if [[ $machineName == help ]]; then
        echo "virtual-manager.sh is a script for running virtual machines"
        echo "usage virtual-manager.sh image [tap|proxy] [index]"
fi


qemu-system-x86_64 \
        -initrd initrd.img-5.4.0-84-generic \
        -kernel vmlinuz-5.4.0-84-generic \
        -nographic -monitor none -serial stdio \
        -append 'root=/dev/vda1 console=ttyS0' \
        -machine q35 -cpu host \
        -drive "file=${disk_img_snapshot},format=qcow2,if=virtio" \
        -enable-kvm \
        -netdev tap,ifname=ggtap${tap1},script=no,downscript=no,vhost=on,id=nInt\
        -device virtio-net-pci,mac=52:54:00:12:34:5${tap1},netdev=nInt \
        -netdev tap,ifname=ggtap${tap2},script=no,downscript=no,vhost=on,id=mInt\
        -device virtio-net-pci,mac=52:54:00:12:34:6${tap1},netdev=mInt \
        -net user,hostfwd=tcp::1002${machineName}-:22 \
        -net nic \
        -m 8G \
        -smp 2 \
        ;

stty "$STTY_SETTINGS"
```

V. CONFIGURING NETWORK SETTING OF VIRTUAL MACHINES

We set the ip addresses of the two interfaces our VMS have by adding the following files to /etc/network/ directory.

```
# /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
source /etc/network/interfaces.d/*
```

```
# /etc/network/interfaces.d/enp0s8
auto enp0s8
iface enp0s3 inet static
address 192.168.0.120
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

```
# /etc/network/interfaces.d/enp0s9
auto enp0s9
iface enp0s4 inet static
address 192.168.0.125
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

After network configuration of all the virtual machines, we were able to ping each virtual machine from the other one. The following code block shows that ping messages traverse each virtual machine interface to its tap device and from the tap device via bridge to the other tap device and to the virtual machine of destination.

```
$ ping 192.168.0.121
PING 192.168.0.121 (192.168.0.121) 56(84) bytes of data.
64 bytes from 192.168.0.121: icmp_seq=1 ttl=64 time=0.523 ms
64 bytes from 192.168.0.121: icmp_seq=2 ttl=64 time=0.692 ms
```

```
$ sudo tcpdump -i ggtap1
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ggtap1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:16:11.623712 IP 192.168.0.120 > 192.168.0.121: ICMP echo request, id 8327, seq 5, length 64
21:16:11.624145 IP 192.168.0.121 > 192.168.0.120: ICMP echo reply, id 8327, seq 5, length 64
```

```
$ sudo tcpdump -i ggtap2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ggtap2, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:16:21.779241 ARP, Request who-has 192.168.0.120 tell 192.168.0.121, length 28
21:16:21.779680 ARP, Reply 192.168.0.120 is-at 52:54:00:12:34:51 (oui Unknown), length 28
21:16:21.863620 IP 192.168.0.120 > 192.168.0.121: ICMP echo request, id 8327, seq 15, length 64
21:16:21.863924 IP 192.168.0.121 > 192.168.0.120: ICMP echo reply, id 8327, seq 15, length 64
```

VI. PREPAIRING VIRTUAL MACHINES BY INSTALLING OPEN5GS AND OPENAIRINTERFACE ON THEM

- On 2 of Virtual Machines (C-Plane and U-Plane) we install Open5GS
- on UERAN machine we will install Open-Air-Interface

To check if the Open5GS is installed and its functions are working:

sudo service open5gs-amfd status

or

systemctl status open5gs*

VII. CONFIGURING U-PLANE

UPF (User Plane Function) is a key component in the architecture of 5G networks. It is responsible for forwarding user data (the "user plane") between the 5G Radio Access Network (RAN) and the core network. The UPF is the main point of interaction between the RAN and the core network, and it performs functions such as packet filtering, routing, and traffic management.
The UPF is designed to handle the large amounts of data traffic generated by 5G devices and networks, and to ensure that this data is delivered quickly, reliably, and securely. The UPF is a critical component in enabling the high speeds and low latency that are key features of 5G networks.
By enetring upf.yaml and going to line 173, we have changed the ip address to the ip address that we considered for upf of Sip U-Plane. Which is the ip address of enp0s9 which we considered as 192.168.0.127.

- configuring upf

cd Projects/open5gs/install/etc/open5gs
vim upf.yaml

This is a configuration file written in YAML format. It contains information about network settings for a certain application or system.

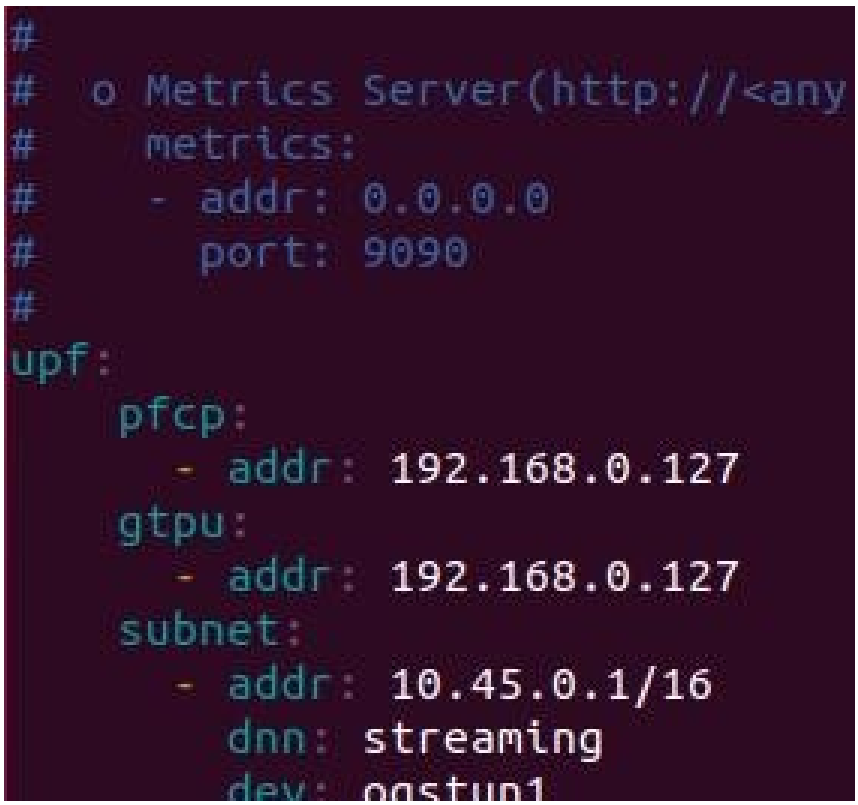
Here's what each line means:
It is a configuration file written in YAML format. It contains information about network settings for a certain application or system. (Here Sip_Call_Server, FileSharing and Streaming)

1. "upf" refers to a user plane function. This configuration is for a single UPF.
2. The UPF has a PFCP (Packet Forwarding Control Protocol) configuration. The PFCP configuration has a single address (192.168.0.127).
3. The UPF also has a GTP-U (GPRS Tunnelling Protocol - User Plane) configuration. The GTP-U configuration also has a single address (192.168.0.127).
4. The UPF has a subnet configuration with public addresses of (10.45.0.1/16, 10.46.0.1/16, 10.47.0.1/16). This address is the network address for a subnet with a subnet mask of 255.255.0.0. The subnet is associated with a Data Network Name (DNN) of filesharing for File_Sharing, Voip for Sip_Call_Server and Streaming for Streaming use, and is accessible through the devices osgtun3, osgtun2 and osgtun1 respectively. The subnet configuration also specifies a DNN. The subnet configuration also specifies the device (osgtun2 for example) through which the subnet is accessible.
5. The UPF has a metrics configuration with a single address (127.0.0.7) and a port (9090). This configuration could be used for monitoring or other management purposes.

Hints:

- SIP clients usually use port number 5060 for non-encrypted signaling traffic(UDP) to connect to SIP servers and other SIP endpoints.

Here the configured part shows:

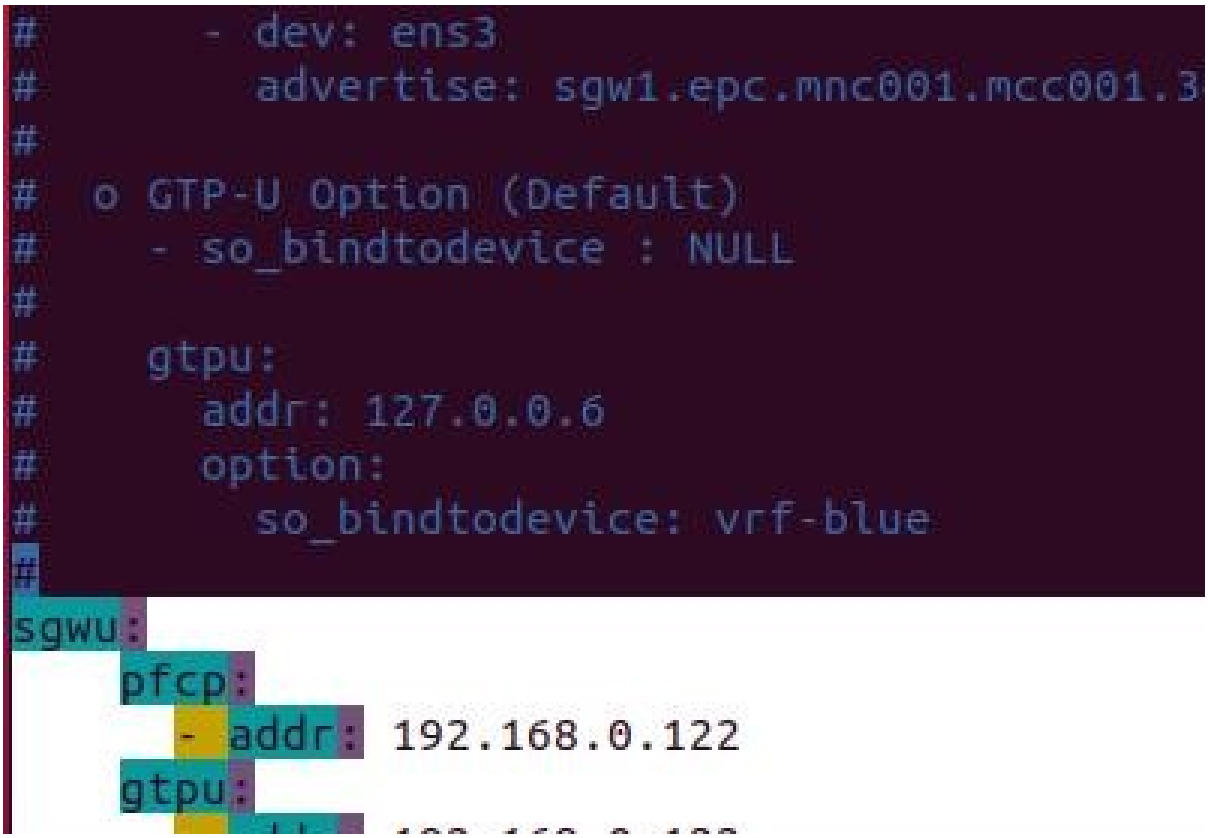


- configuring sgwu

cd Projects/open5gs/install/etc/open5gs
vim sgwu.yaml

By enetring sgwu.yaml and going to line 98, we have changed the ip address to the ip address that we considered for sgwu of U-Plane. Which is the ip address of enp0s8 which we considered as 192.168.0.122.

Here the configured part shows

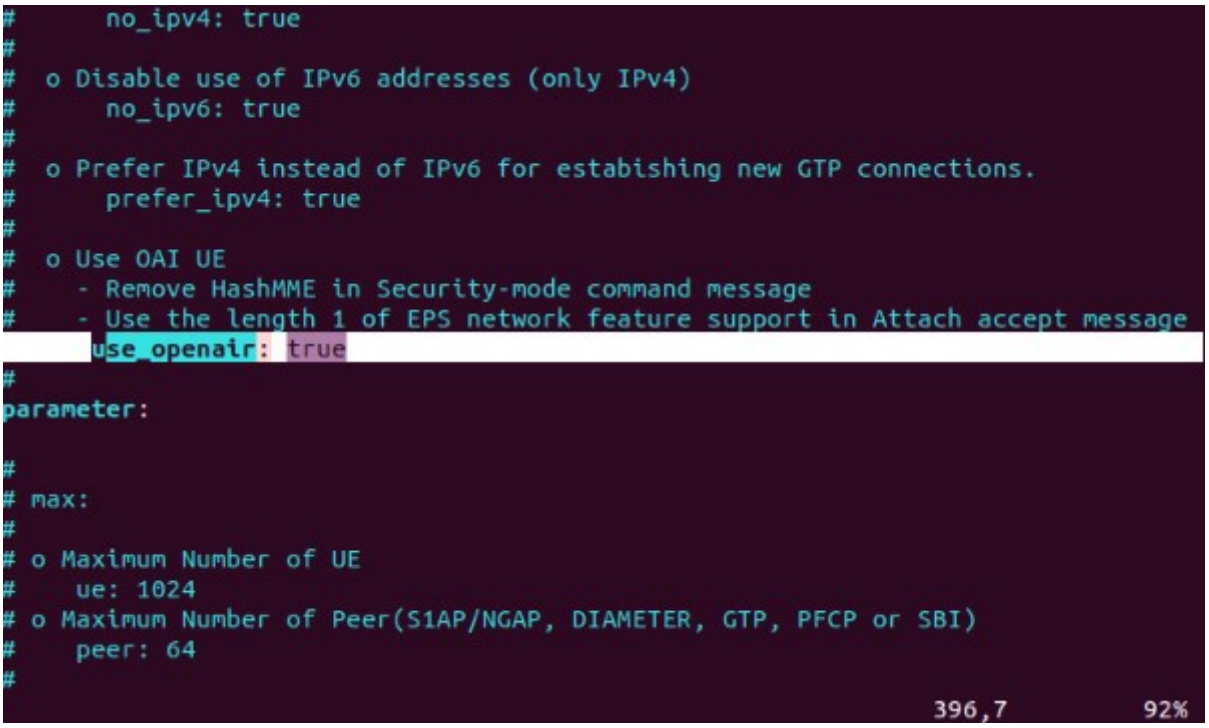


VIII. CONFIGURING C-PLANE

- configuring mme

```
cd Projects/open5gs/install/etc/open5gs
vim mme.yaml
```

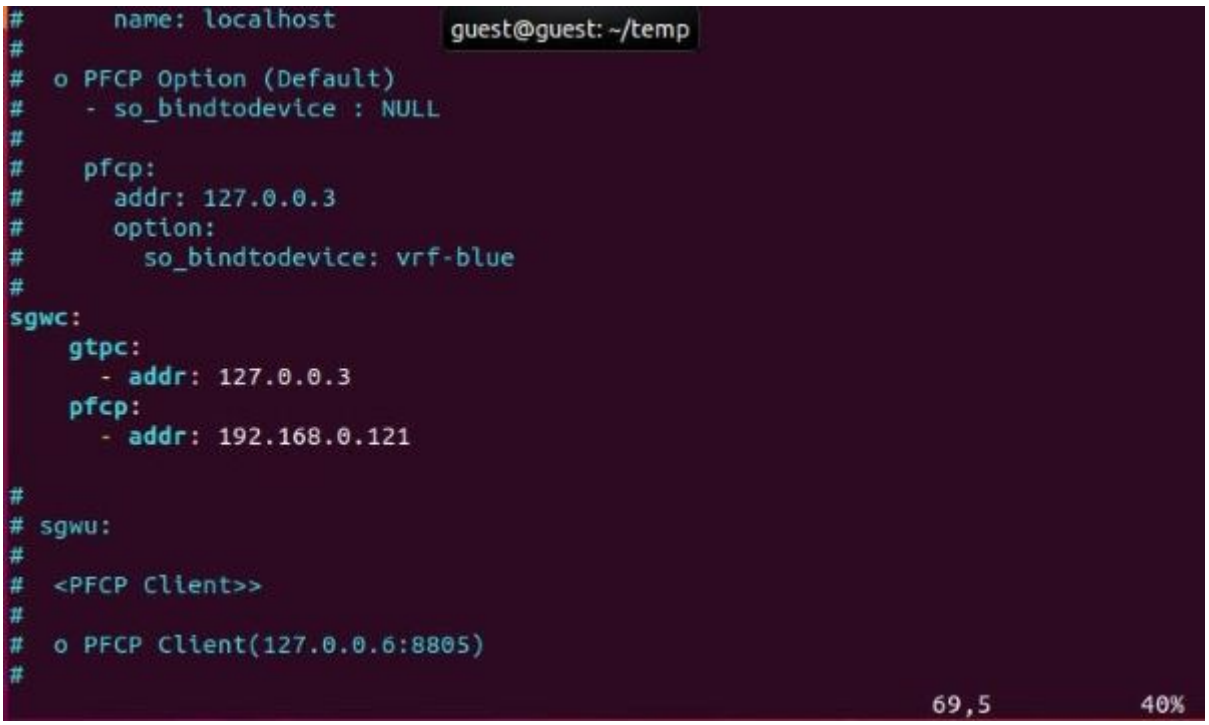
Here the configured part shows

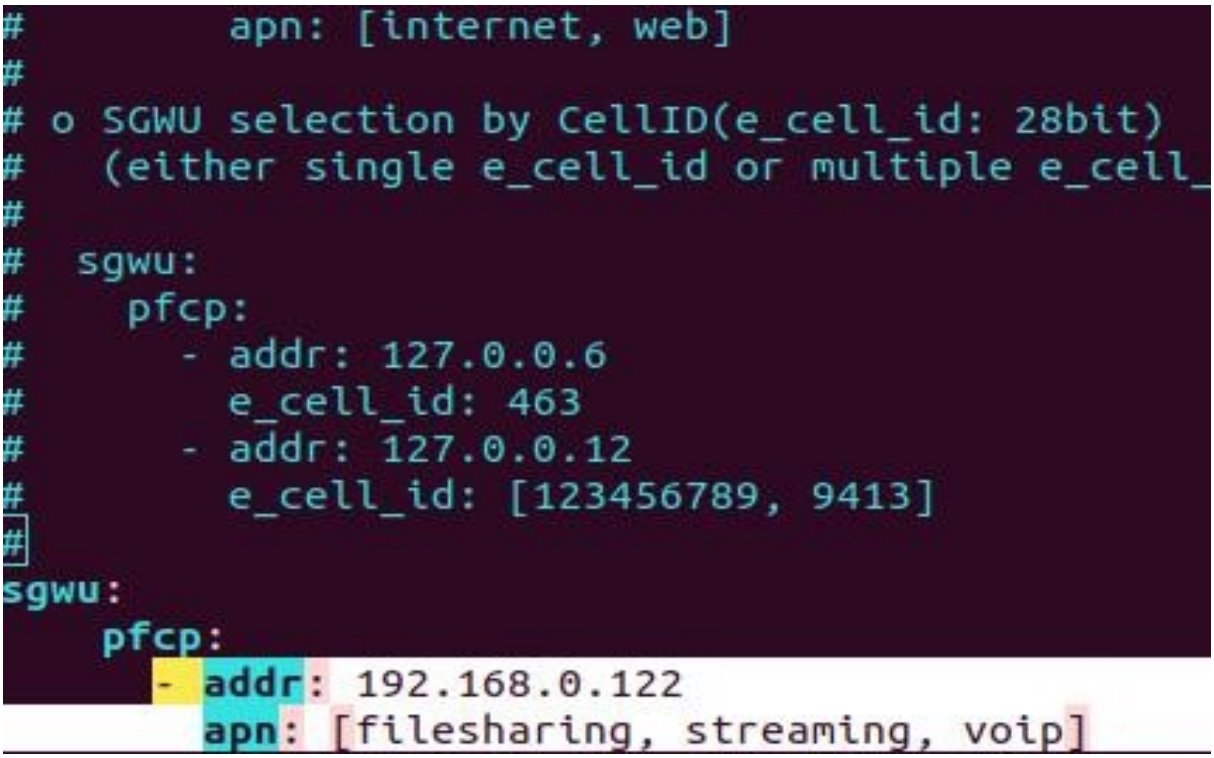


- configuring sgwc

```
cd Projects/open5gs/install/etc/open5gs
vim sgwc.yaml
```

Here the configured part shows

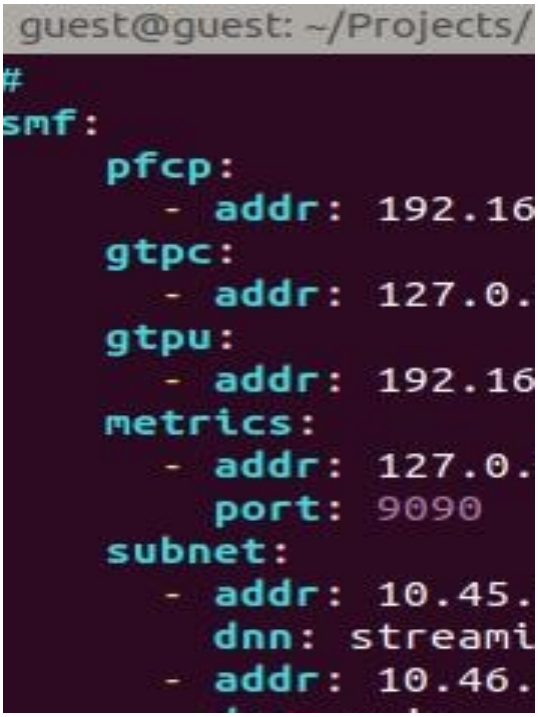


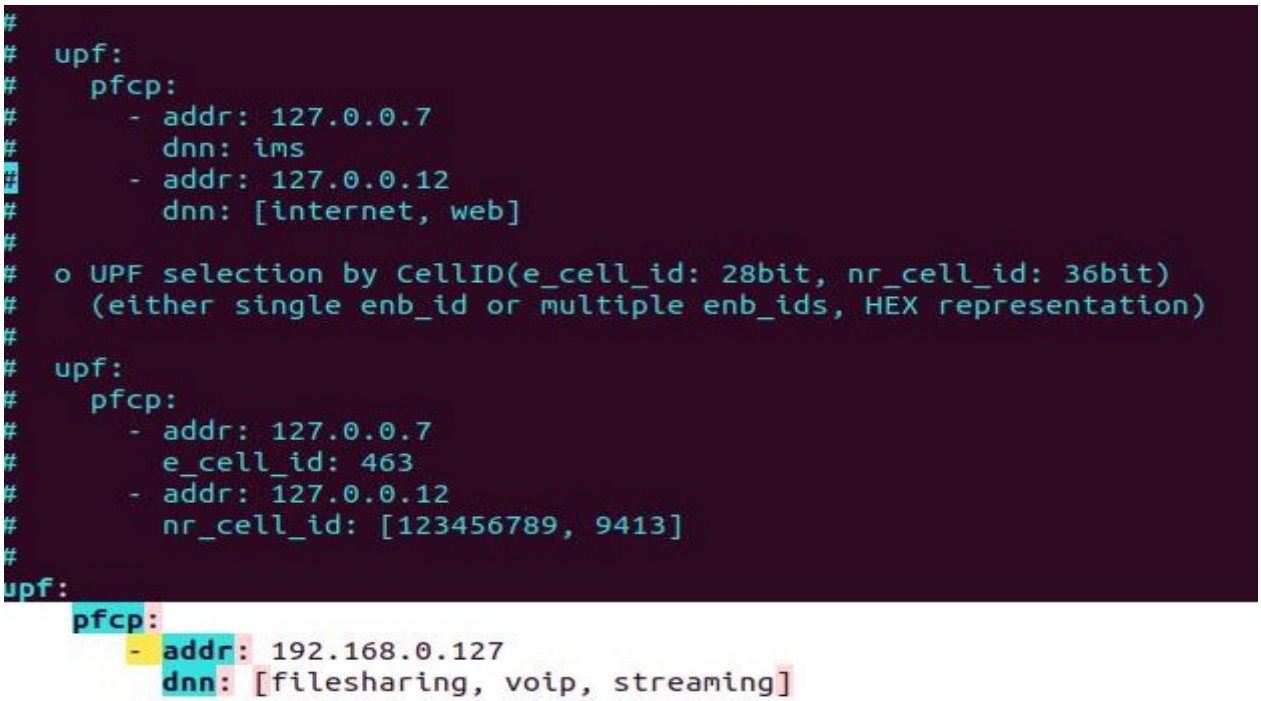


- Configuring smf

```
cd Projects/open5gs/install/etc/open5gs
vim smf.yaml
```

Here the configured part shows



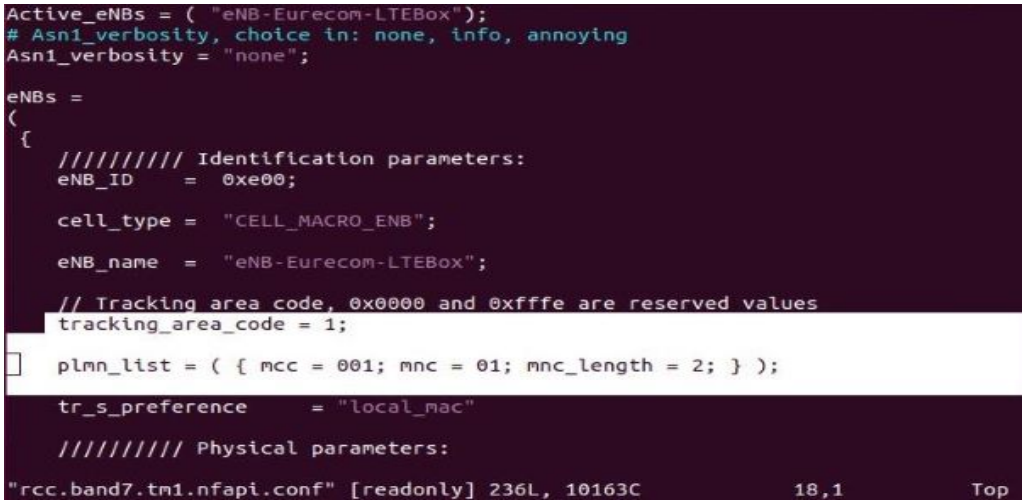


IX. CONFIGURING UE-RAN

- **Changes in configuration files of RAN:**
  - "rcc.band7.tm1.nfapi.conf".

```
cd Projects/enb/ci-scripts-conf_files
vim rcc.band7.tm1.nfapi.conf
```

Here the configured part shows:



- **Changes in configuration files of UE**
  - "ue.nfapi.conf"

```
cd Projects/ue/ci-scripts/conf_files
vim ue.nfapi.conf
```

Here the configured part shows

```
  phy_log_verbosity                     ="medium";
  mac_log_level                         ="info";
  mac_log_verbosity                     ="medium";
  rlc_log_level                         ="info";
  rlc_log_verbosity                     ="medium";
  pdcp_log_level                        ="info";
  pdcp_log_verbosity                    ="medium";
  rrc_log_level                         ="info";
  rrc_log_verbosity                     ="full";
};


L1s = (
        {
        num_cc = 1;
        tr_n_preference = "nfapi";
        local_n_if_name  = "lo";
        remote_n_address = "127.0.0.1";
        local_n_address  = "127.0.0.2";
        local_n_portc    = 50000;
        remote_n_portc   = 50001;
        local_n_portd    = 50010;
        remote_n_portd   = 50011;
                                                        24,1          28%
```

X. PACKET TRACES

- Uplane and Coreplane

As it is shown we have Heartbeat between coreplane and uplane which is enb0s8 to enb0s8 of the others and enb0s9 to enb0s9 of the others.

cplane.uplane.1.pcap

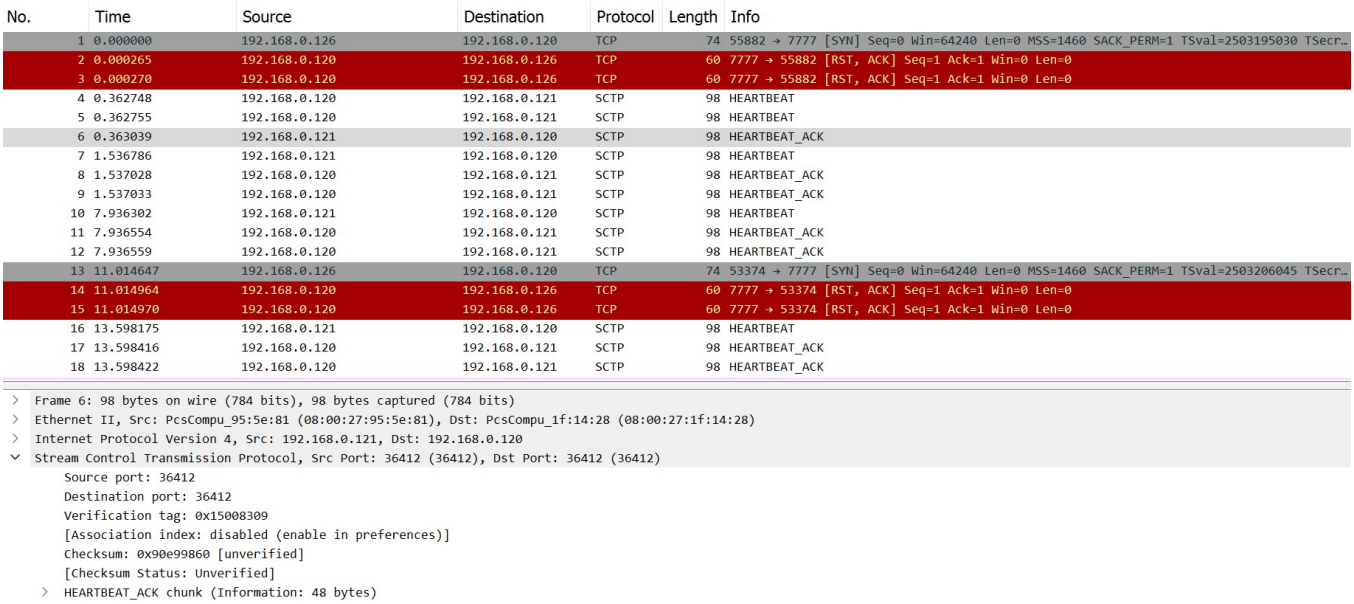
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.127	192.168.0.126	PFCP	60	PFCP Heartbeat Request
2	0.000007	192.168.0.127	192.168.0.126	PFCP	60	PFCP Heartbeat Request
3	0.000476	192.168.0.126	192.168.0.127	PFCP	60	PFCP Heartbeat Response
4	0.116346	PcsCompu_95:5e:81	PcsCompu_f5:a2:15	ARP	60	Who has 192.168.0.127? Tell 192.168.0.126
5	0.116579	PcsCompu_f5:a2:15	PcsCompu_95:5e:81	ARP	60	192.168.0.127 is at 08:00:27:f5:a2:15
6	0.116586	PcsCompu_f5:a2:15	PcsCompu_95:5e:81	ARP	60	192.168.0.127 is at 08:00:27:f5:a2:15
7	0.829606	192.168.0.121	192.168.0.122	PFCP	60	PFCP Heartbeat Request
8	0.830027	192.168.0.122	192.168.0.121	PFCP	60	PFCP Heartbeat Response
9	0.830035	192.168.0.122	192.168.0.121	PFCP	60	PFCP Heartbeat Response
10	0.924216	192.168.0.122	192.168.0.121	PFCP	60	PFCP Heartbeat Request
11	0.924223	192.168.0.122	192.168.0.121	PFCP	60	PFCP Heartbeat Request
12	0.924651	192.168.0.121	192.168.0.122	PFCP	60	PFCP Heartbeat Response
13	5.089913	PcsCompu_f5:a2:15	PcsCompu_95:5e:81	ARP	60	Who has 192.168.0.126? Tell 192.168.0.122
14	5.089920	PcsCompu_f5:a2:15	PcsCompu_95:5e:81	ARP	60	Who has 192.168.0.126? Tell 192.168.0.122
15	5.090247	PcsCompu_95:5e:81	PcsCompu_f5:a2:15	ARP	60	192.168.0.126 is at 08:00:27:95:5e:81
16	5.857974	PcsCompu_f5:a2:15	PcsCompu_95:5e:81	ARP	60	Who has 192.168.0.121? Tell 192.168.0.122

```
> Frame 7: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: PcsCompu_95:5e:81 (08:00:27:95:5e:81), Dst: PcsCompu_f5:a2:15 (08:00:27:f5:a2:15)
> Internet Protocol Version 4, Src: 192.168.0.121, Dst: 192.168.0.122
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805
v Packet Forwarding Control Protocol
    > Flags: 0x20
      Message Type: PFCP Heartbeat Request (1)
      Length: 12
      Sequence Number: 44
      Spare: 0
    > Recovery Time Stamp : Feb 28, 2023 20:24:42.000000000 UTC
      [Response In: 8]
```

- Enodeb to Core-lane



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.126	192.168.0.120	TCP	74	55882 → 7777 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2503195030 TSecr…
2	0.000265	192.168.0.120	192.168.0.126	TCP	60	7777 → 55882 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	0.000270	192.168.0.120	192.168.0.126	TCP	60	7777 → 55882 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	0.362748	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT
5	0.362755	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT
6	0.363039	192.168.0.121	192.168.0.120	SCTP	98	HEARTBEAT_ACK
7	1.536786	192.168.0.121	192.168.0.120	SCTP	98	HEARTBEAT
8	1.537028	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK
9	1.537033	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK
10	7.936302	192.168.0.121	192.168.0.120	SCTP	98	HEARTBEAT
11	7.936554	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK
12	7.936559	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK
13	11.014647	192.168.0.126	192.168.0.120	TCP	74	53374 → 7777 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2503206045 TSecr…
14	11.014964	192.168.0.120	192.168.0.126	TCP	60	7777 → 53374 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	11.014970	192.168.0.120	192.168.0.126	TCP	60	7777 → 53374 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
16	13.598175	192.168.0.121	192.168.0.120	SCTP	98	HEARTBEAT
17	13.598416	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK
18	13.598422	192.168.0.120	192.168.0.121	SCTP	98	HEARTBEAT_ACK

```
> Frame 6: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: PcsCompu_95:5e:81 (08:00:27:95:5e:81), Dst: PcsCompu_1f:14:28 (08:00:27:1f:14:28)
> Internet Protocol Version 4, Src: 192.168.0.121, Dst: 192.168.0.120
v Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 36412 (36412)
      Source port: 36412
      Destination port: 36412
      Verification tag: 0x15008309
      [Association index: disabled (enable in preferences)]
      Checksum: 0x90e99860 [unverified]
      [Checksum Status: Unverified]
    > HEARTBEAT_ACK chunk (Information: 48 bytes)
```

As it is shown we have Heartbeat between coreplane and EnodeB.

GLOSSARY

Abbreviation	Explaination
Amf	Access and Mobility management function
nrf	Network Repository function
pcf	Policy Control Function
mmed	Mobility Management Entity Design
sgwcd	Serving Gateway Control Plane Daemon
smfd	Service Management Function Deomon
hssd	Home Subscriber Server Deamon

- **Amf**

AMF (Access and Mobility Management Function) is a key component of the 5G core network that is responsible for managing user authentication, authorization, and mobility within the network.
The AMF performs several functions, including:

- Authentication: The AMF authenticates the user's identity and verifies that they are authorized to access the network.
- Authorization: The AMF determines the user's access rights based on their subscription profile, which includes information such as the services they are authorized to use and their quality of service (QoS) requirements.
- Session Management: The AMF manages the user's connection to the network, including establishing and terminating sessions, monitoring user activity, and enforcing policies.
- Mobility Management: The AMF manages the user's mobility within the network, including handovers between different cells or base stations, and tracking the user's location as they move through the network.

Overall, the AMF plays a critical role in ensuring that users can securely and reliably access 5G network services, while also maintaining the quality of service and managing network resources efficiently.

- **Nrf**

In 5G, NRF stands for "Network Repository Function". It is a core network component responsible for storing and managing network function descriptions, which are used by other network functions to discover and access network services.

The NRF performs several functions, including:

- Service discovery: The NRF maintains a registry of available network functions and their associated capabilities, which other network functions can query to discover available services.
- Network function selection: The NRF can provide recommendations on which network function to use for a particular service, based on factors such as network conditions, QoS requirements, and network policies.
- Load balancing: The NRF can distribute requests for network services across multiple network functions to ensure that resources are used efficiently, and that service availability is maintained.
- Network slicing: The NRF can help to manage network slicing, which is a key feature of 5G that allows network resources to be partitioned and allocated to different services or user groups.

Overall, the NRF plays a critical role in enabling the dynamic, service-oriented nature of 5G networks, where network functions can be quickly discovered, accessed, and composed to support a wide range of use cases and service requirements.

- **Pcf**

n 5G, PCF stands for "Policy Control Function". It is a core network component responsible for managing policy rules that control how network resources are allocated and used by different network services and users.
The PCF performs several functions, including:

- Policy rule management: The PCF manages policy rules that define how network resources are allocated and used by different network services and users. These rules can be based on factors such as user identity, QoS requirements, and network conditions.
- Policy decision making: The PCF makes policy decisions based on input from other network functions, such as the SMF and the UPF. These decisions can determine how network resources are allocated and used, and can be based on real-time network conditions and user needs.
- Policy enforcement: The PCF enforces policy rules across the network, ensuring that resources are used efficiently and that the quality of service for different services and users is maintained.
- Charging: The PCF is responsible for tracking network usage and generating usage records for charging purposes.

Overall, the PCF plays a critical role in enabling 5G networks to support a wide range of use cases and service requirements, while also managing network resources efficiently and maintaining a high quality of service for users.

- **Mmed**

MME (Mobility Management Entity) is a key component in 5G networks responsible for managing and controlling the flow of user data in the network. The MME is responsible for several functions, including user authentication, network entry and exit, and handling of user mobility between different cells in the network.
In 5G networks, the MME operates in the control plane and is separate from the UPF (User Plane Function), which is responsible for forwarding user data in the user plane. The MME is responsible for maintaining information about the location of each user and ensuring that user data is correctly forwarded to the appropriate network elements as the user moves between different cells in the network.
The MME also plays a key role in maintaining the security of the network, as it is responsible for authenticating users and encrypting user data to ensure its confidentiality. In addition, the MME is responsible for handling the signaling required for the establishment and release of user connections, and for handling the handover of user connections between different cells in the network.
In summary, the MME is a central control element in 5G networks that plays a critical role in managing and controlling the flow of user data, maintaining network security, and supporting user mobility.

- **Sgwcd**

SGW-C stands for Serving Gateway Control plane in 5G networks. It is a key component of the 5G core network architecture and is responsible for managing and controlling the flow of user data between the mobile network and the external packet data network (PDN). The SGW-C performs several important functions, including user plane traffic forwarding, policy enforcement, and management of user mobility between different cells in the network.
In 5G networks, the SGW-C operates in the control plane and is separate from the Serving Gateway User Plane (SGW-U), which is responsible for forwarding user data in the user plane. The split between the SGW-C and SGW-U allows for greater scalability and flexibility in the network, enabling the network to efficiently manage and control the large volumes of user data that are generated in a 5G network.

- **Smfd**

SMF (Service Management Function) is a key component in 5G networks responsible for managing and controlling the flow of user data in the network. The SMF is responsible for several functions, including network slice selection, policy enforcement, and user plane traffic forwarding.

The SMF performs functions, including:

- Session establishment: The SMF is responsible for establishing data sessions between the user's device and the network, including setting up security associations and negotiating QoS parameters.
- Policy enforcement: The SMF enforces policy rules related to user access, QoS, and network resources. This includes ensuring that the user is authorized to access the requested service and that the network has sufficient resources to support the session.
- Charging: The SMF tracks and records usage data for charging purposes, including the amount of data transmitted and received, and the duration of the session.
- Session termination: The SMF is responsible for terminating data sessions when the user has completed their session or when an event triggers the session to end.

In 5G networks, the SMF is implemented as a network function that operates in the control plane and is separate from the UPF (User Plane Function), which is responsible for forwarding user data in the user plane. The split between the SMF and UPF allows for greater scalability and flexibility in the network, enabling the network to efficiently manage and control the large volumes of user data that are generated in a 5G network.
It works closely with other 5G core network components, such as the AMF and UPF (User Plane Function), to ensure that data sessions are established and managed efficiently and securely.
The SMF also plays a key role in supporting network slicing, which is a key feature of 5G networks. Network slicing allows different types of user traffic to be separated and treated differently, based on their specific requirements. The SMF is responsible for selecting the appropriate network slice for each user session and managing the flow of user data through the appropriate slice.
Overall, the SMF plays a critical role in enabling 5G network services and ensuring a high quality of service for users.

- **Hssd**

HSS stands for Home Subscriber Server in 5G networks. It is a central database that stores information about subscribers and their services, such as their identities, profiles, and security keys. The HSS is a key component of the 5G core network architecture and is used by other network functions, such as the MME (Mobility Management Entity) and the SMF (Service Management Function), to manage and control the flow of user data in the network.
In 5G networks, the HSS plays a critical role in managing user identities and maintaining the security of the network. It stores user-specific information, such as authentication and authorization data, and provides this information to other network functions as needed. The HSS also plays a key role in supporting network slicing, which is a key feature of 5G networks. Network slicing allows different types of user traffic to be separated and treated differently, based on their specific requirements. The HSS is responsible for storing information about the network slices available to each user and providing this information to the SMF, which is responsible for selecting the appropriate network slice for each user session.

CONCLUSION

In summary, network operators may offer high-speed connection, support developing use cases, and achieve operational efficiency by implementing a network based on 5G and Open-Air Interface with a single UPF. This has the potential to have a tremendous influence on industries ranging from healthcare to transportation, paving the way for a new era of connectedness and innovation.

REFERENCES

[1] M. Pirmoradian, N. Sumi " *https://github.com/FRA-UAS/mobcom-project-lte" 2023.*

[2] Open5GS, 2022. Open5GS. [Online] Available at: https://open5gs.org/.

[3] OpenAirInterface, 2022. OpenAirInterface | 5G software alliance for democratising wireless innovation. [Online] Available at: https://openairinterface.org/.

[4] Trick, U., 2021. 5G - An Introduction to the 5th Generation Mobile Networks. De Gruyter STEM ed. Berlin/München/Boston: De Gruyter Oldenbourg

[5] Ishida, S., 2020. *open5gs_epc_oai_sample_config.* [Online]
Available at: https://github.com/s5uishida/open5gs_epc_oai_sample_config/blob/main/README.md
[Accessed 01 Feb 2023].