Mobile Cellular Communications (5G)

I. Objectives

The objectives of this laboratory are:

- Identify configuration parameters required for the different components
- Understand the main procedures in a mobile cellular (5G) at the control and data planes by running opensource implementations of the main components

II. Duration

This laboratory should last 2h30, divided in 2 classes.

III. Used tools

This laboratory will use:

a) An opensource implementation of a 5G Core: Free5GC [free5GC] [free5gcwiki]

b) A gNB and UE opensource implementation: UERANSIM [ueransim]

c) A VirtualBox VM with both components already installed in the laboratory PCs

d) Wireshark also installed in the laboratory PCs

The VM is also available via SSH at port 2222 for user ‘***ubuntu***’ (e.g. ‘***ssh -p 222 ubuntu@localhost***’, from the hosting machine); password is ‘***ubuntu***’ for users ‘***ubuntu***’ and ***root***.

IV. Network diagram

1. 5G Core components (provided by Free5GC) are represented by light blue and purple boxes (UPF, dataplane), gNBs by brown boxes and UEs by green boxes
2. With UERANSIM, the 5G-NR radio interface ('Radio Link’, RL) is emulated over UDP between the UEs (11, 12 and 21, green boxes) and the gNBs (gNB1 and gNB2, orange boxes) they are connected to.
3. IP addresses:
   a. 10.0.123.0/24: SBI; Core components, Web Console and DB (control plane)
   b. 10.0.124.0/24: N2 interfaces (control plane)
   c. 10.0.130.0/24: N3 interfaces (data plane)
   d. 10.0.140.0/24: N4 interfaces (control plane)
   e. 10.0.20[1|2].0/24: radio interfaces emulation
   f. 10.1.[1|2].0/24: N6 DNNs (data plane)
4. Via the ‘Host’ entity, emulated UEs can reach the Internet.
5. A ***hosts*** file has been added to Wireshark (***/root/.config/wireshark***) for IP addresses resolution so that Wireshark presents components’ names instead of IP addresses allowing you to better interpret the messages exchange (see that file contents in Annex F at the end).
6. The shown MongoDB in the diagram component serves as persistent data repository for the other components while the network is running.

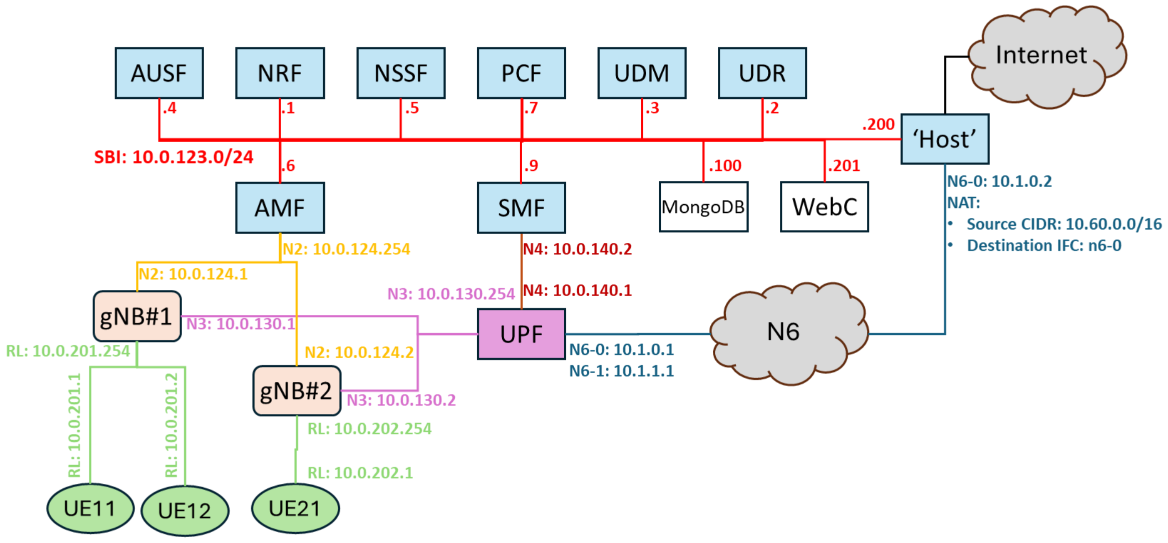


Figure 1: Network diagram

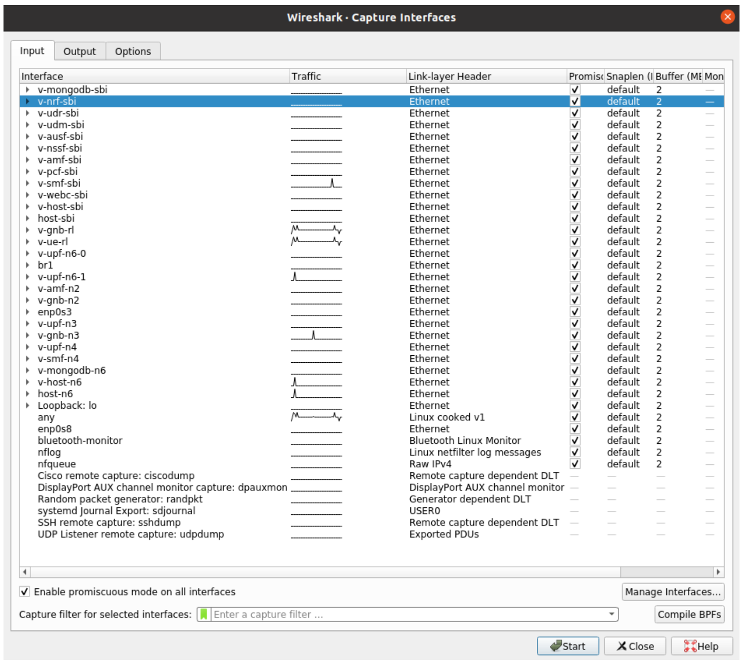
V. Procedures

1. Introduction

During the laboratory execution based on the provided VirtualBox Virtual Machine, the 5G network components will be started and stopped, following this order:

1. **5G Network:** **(a) 5G Core (3.7) → (b) gNB1 (4.3) → (c) gNB2 (4.5) →**
2. **UEs creation:** **→ (d) UE provisioning at the 5G Core (5.2) →**
3. **UEs start:** **→ (e) UE11 (5.4) → (f) UE12 (6.5) → (g) UE21 (6.7) →**
4. **Stop the system:** **→ (h) UEs, gNBs and 5G Core (9.1 and 9.3)**

Linux *Namespaces* are used to have each of the nine 5GC Network Functions (AMF, AUSF, NRF, NSSF, PCF, SMF, UDM, UDR, UPF) running inside its own namespace [konrad]. This allows the usage of Wireshark (**<u>shall be started with 'sudo'</u>**) to capture traffic packets exchanged between any two NFs, on their own interfaces (you will get the list of interfaces present in the following screen capture, when selecting capturing interfaces in Wireshark, after 5G Core components have been started).



Interface	Traffic	Link-layer Header	Promisc	Snaplen (I	Buffer (MB	Mon
v-mongodb-sbi		Ethernet	✓	default	2	—
v-nrf-sbi		Ethernet	✓	default	2	—
v-udr-sbi		Ethernet	✓	default	2	—
v-udm-sbi		Ethernet	✓	default	2	—
v-ausf-sbi		Ethernet	✓	default	2	—
v-nssf-sbi		Ethernet	✓	default	2	—
v-amf-sbi		Ethernet	✓	default	2	—
v-pcf-sbi		Ethernet	✓	default	2	—
v-smf-sbi		Ethernet	✓	default	2	—
v-webc-sbi		Ethernet	✓	default	2	—
v-host-sbi		Ethernet	✓	default	2	—
host-sbi		Ethernet	✓	default	2	—
v-gnb-rl		Ethernet	✓	default	2	—
v-ue-rl		Ethernet	✓	default	2	—
v-upf-n6-0		Ethernet	✓	default	2	—
br1		Ethernet	✓	default	2	—
v-upf-n6-1		Ethernet	✓	default	2	—
v-amf-n2		Ethernet	✓	default	2	—
v-gnb-n2		Ethernet	✓	default	2	—
enp0s3		Ethernet	✓	default	2	—
v-upf-n3		Ethernet	✓	default	2	—
v-gnb-n3		Ethernet	✓	default	2	—
v-upf-n4		Ethernet	✓	default	2	—
v-smf-n4		Ethernet	✓	default	2	—
v-mongodb-n6		Ethernet	✓	default	2	—
v-host-n6		Ethernet	✓	default	2	—
host-n6		Ethernet	✓	default	2	—
Loopback: lo		Ethernet	✓	default	2	—
any		Linux cooked v1	✓	default	2	—
enp0s8		Ethernet	✓	default	2	—
bluetooth-monitor		Bluetooth Linux Monitor	✓	default	2	—
nflog		Linux netfilter log messages	✓	default	2	—
nfqueue		Raw IPv4	✓	default	2	—
Cisco remote capture: ciscodump		Remote capture dependent DLT	—	—	—	—
DisplayPort AUX channel monitor capture: dpauxmon		DisplayPort AUX channel monitor	—	—	—	—
Random packet generator: randpkt		Generator dependent DLT	—	—	—	—
systemd Journal Export: sdjournal		USER0	—	—	—	—
SSH remote capture: sshdump		Remote capture dependent DLT	—	—	—	—
UDP Listener remote capture: udpdump		Exported PDUs	—	—	—	—

[✓] Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

Figure 2: Logical interfaces as seen in Wireshark, after 5G Core start

2. Configurations analysis

1) Analyse the yaml configuration files in the list below (1.1.a), located in folder *~/5GLab/netns5g/config* (you may open them with the File Manager) and search for the listed configuration parameters in 1.1.b.

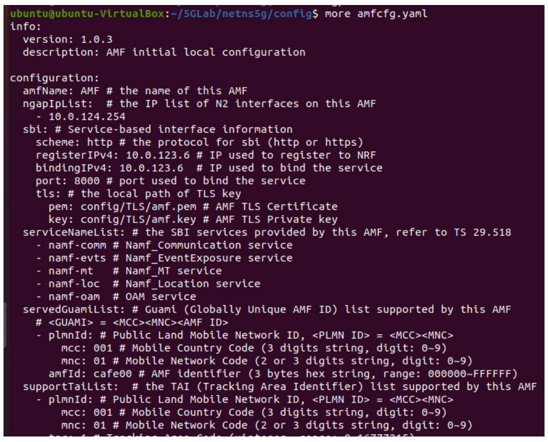
   a. **Files**:

      i. 5G Core: **amfcfg**.yaml, **smfcfg**.yaml, **upfcfg**.yaml

      ii. 5G RAN: free5gc-**gnb1**.yaml, free5gc-**gnb2**.yaml

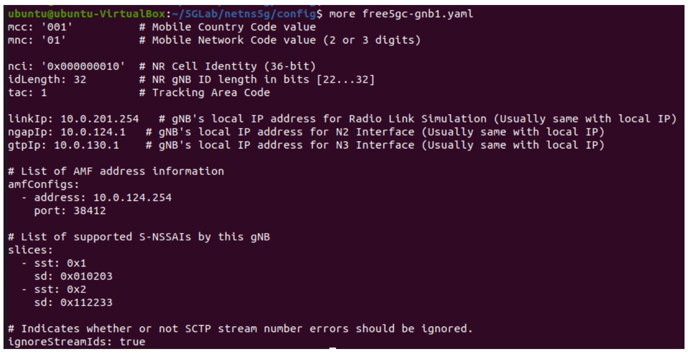
      iii. 5G UEs:

      - free5gc-**ue11**.yaml
      - free5gc-**ue12**-**sl1**.yaml, free5gc-**ue12**-**sl2**.yaml
      - free5gc-**ue21**.yaml



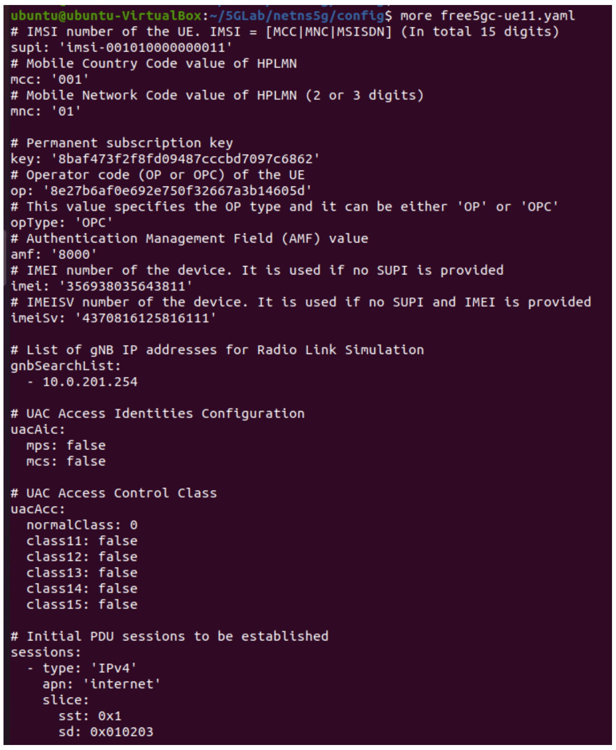
```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g/config$ more amfcfg.yaml
info:
  version: 1.0.3
  description: AMF initial local configuration

configuration:
  amfName: AMF # the name of this AMF
  ngapIpList:  # the IP list of N2 interfaces on this AMF
    - 10.0.124.254
  sbi: # Service-based interface information
    scheme: http # the protocol for sbi (http or https)
    registerIPv4: 10.0.123.6 # IP used to register to NRF
    bindingIPv4: 10.0.123.6  # IP used to bind the service
    port: 8000 # port used to bind the service
    tls: # the local path of TLS key
      pem: config/TLS/amf.pem # AMF TLS Certificate
      key: config/TLS/amf.key # AMF TLS Private key
  serviceNameList: # the SBI services provided by this AMF, refer to TS 29.518
    - namf-comm # Namf_Communication service
    - namf-evts # Namf_EventExposure service
    - namf-mt   # Namf_MT service
    - namf-loc  # Namf_Location service
    - namf-oam  # OAM service
  servedGuamiList: # Guami (Globally Unique AMF ID) list supported by this AMF
    # <GUAMI> = <MCC><MNC><AMF ID>
    - plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC><MNC>
        mcc: 001 # Mobile Country Code (3 digits string, digit: 0~9)
        mnc: 01 # Mobile Network Code (2 or 3 digits string, digit: 0~9)
      amfId: cafe00 # AMF identifier (3 bytes hex string, range: 000000~FFFFFF)
  supportTaiList:  # the TAI (Tracking Area Identifier) list supported by this AMF
    - plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC><MNC>
        mcc: 001 # Mobile Country Code (3 digits string, digit: 0~9)
        mnc: 01 # Mobile Network Code (2 or 3 digits string, digit: 0~9)
```

Figure 3: Example of 5G Core entity (AMF) configuration file (partial)



```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g/config$ more free5gc-gnb1.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: 10.0.201.254   # gNB's local IP address for Radio Link Simulation (Usually same with local IP)
ngapIp: 10.0.124.1   # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 10.0.130.1    # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
  - address: 10.0.124.254
    port: 38412

# List of supported S-NSSAIs by this gNB
slices:
  - sst: 0x1
    sd: 0x010203
  - sst: 0x2
    sd: 0x112233

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

Figure 4 Example of 5G RAN (gNB) configuration file (partial)



```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g/config$ more free5gc-ue11.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-001010000000011'
# Mobile Country Code value of HPLMN
mcc: '001'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '01'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# 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: '356938035643811'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816111'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 10.0.201.254

# UAC Access Identities Configuration
uacAic:
  mps: false
  mcs: false

# UAC Access Control Class
uacAcc:
  normalClass: 0
  class11: false
  class12: false
  class13: false
  class14: false
  class15: false

# Initial PDU sessions to be established
sessions:
  - type: 'IPv4'
    apn: 'internet'
    slice:
      sst: 0x1
      sd: 0x010203
```

Figure 5: Example of 5G User Equipment configuration file (partial)

b. **EXERCISE: In those files, search for and identify the following parameters**:

i. MCC: ___________ and MNC: ___________

ii. NR Cell Identities: ___________ and TACs: ___________

iii. Supported slices at gNB1 and gNB2 (SST+SD): ________________ and ________________

iv. Supported DNN: ________________

v. List of SUPIs (UE11, UE12 and UE21): ________________________________

3. 5GC start

1) Open a terminal window
2) Change to directory (~/5GLab/netns5g) containing the scripts needed to setup and run the 5G environment
3) Initialize environment (create the namespaces and the virtual interfaces)

   ~/5GLab/netns5G$ ***sudo ./5Gsetup.sh***

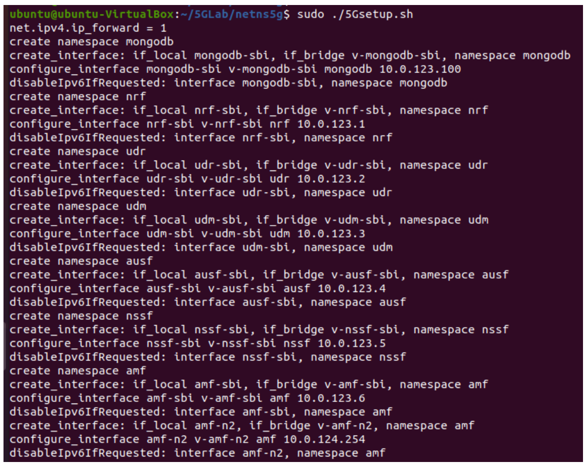
4) Check created namespaces and connecting links

   ~/5GLab/netns5G$ ***sudo ip netns*** – lists created namespaces

   ~/5GLab/netns5G$ ***sudo ip link*** – lists created links

5) Start a Wireshark

   ~/5GLab/netns5G$ ***sudo wireshark &***



```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$ sudo ./5Gsetup.sh
net.ipv4.ip_forward = 1
create namespace mongodb
create_interface: if_local mongodb-sbi, if_bridge v-mongodb-sbi, namespace mongodb
configure_interface mongodb-sbi v-mongodb-sbi mongodb 10.0.123.100
disableIpv6IfRequested: interface mongodb-sbi, namespace mongodb
create namespace nrf
create_interface: if_local nrf-sbi, if_bridge v-nrf-sbi, namespace nrf
configure_interface nrf-sbi v-nrf-sbi nrf 10.0.123.1
disableIpv6IfRequested: interface nrf-sbi, namespace nrf
create namespace udr
create_interface: if_local udr-sbi, if_bridge v-udr-sbi, namespace udr
configure_interface udr-sbi v-udr-sbi udr 10.0.123.2
disableIpv6IfRequested: interface udr-sbi, namespace udr
create namespace udm
create_interface: if_local udm-sbi, if_bridge v-udm-sbi, namespace udm
configure_interface udm-sbi v-udm-sbi udm 10.0.123.3
disableIpv6IfRequested: interface udm-sbi, namespace udm
create namespace ausf
create_interface: if_local ausf-sbi, if_bridge v-ausf-sbi, namespace ausf
configure_interface ausf-sbi v-ausf-sbi ausf 10.0.123.4
disableIpv6IfRequested: interface ausf-sbi, namespace ausf
create namespace nssf
create_interface: if_local nssf-sbi, if_bridge v-nssf-sbi, namespace nssf
configure_interface nssf-sbi v-nssf-sbi nssf 10.0.123.5
disableIpv6IfRequested: interface nssf-sbi, namespace nssf
create namespace amf
create_interface: if_local amf-sbi, if_bridge v-amf-sbi, namespace amf
configure_interface amf-sbi v-amf-sbi amf 10.0.123.6
disableIpv6IfRequested: interface amf-sbi, namespace amf
create_interface: if_local amf-n2, if_bridge v-amf-n2, namespace amf
configure_interface amf-n2 v-amf-n2 amf 10.0.124.254
disableIpv6IfRequested: interface amf-n2, namespace amf
```

Figure 6: start of namespaces (not complete)

6) Start the capture in the interface 'br1' (this will capture all the traffic; you can start other Wireshark instances at specific interfaces, e.g. 'v-amf-sbi')
7) Start the 5G Core (free5gc)

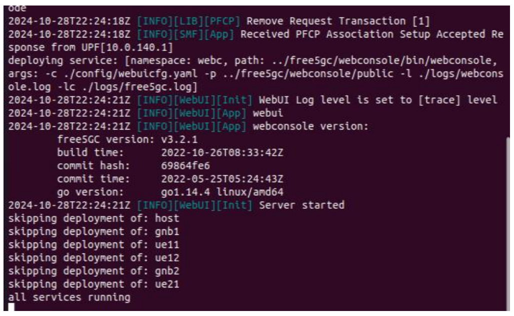
   ~/5GLab/netns5G$ ***sudo ./5Gstart.sh***

   At this point 5G Core Network Functions have started, each in its own namespace.

   Observe the script output and identify the order by which 5G Core components have been started.

   Observe the successive interactions with NRF; what is that for?

   Relate the order they appear with the existing inter dependencies.



```
ode
2024-10-28T22:24:18Z [INFO][LIB][PFCP] Remove Request Transaction [1]
2024-10-28T22:24:18Z [INFO][SMF][App] Received PFCP Association Setup Accepted Re
sponse from UPF[10.0.140.1]
deploying service: [namespace: webc, path: ../free5gc/webconsole/bin/webconsole,
args: -c ./config/webuicfg.yaml -p ../free5gc/webconsole/public -l ./logs/webcons
ole.log -lc ./logs/free5gc.log]
2024-10-28T22:24:21Z [INFO][WebUI][Init] WebUI Log level is set to [trace] level
2024-10-28T22:24:21Z [INFO][WebUI][App] webui
2024-10-28T22:24:21Z [INFO][WebUI][App] webconsole version:
        free5GC version: v3.2.1
        build time:      2022-10-26T08:33:42Z
        commit hash:     69864fe6
        commit time:     2022-05-25T05:24:43Z
        go version:      go1.14.4 linux/amd64
2024-10-28T22:24:21Z [INFO][WebUI][Init] Server started
skipping deployment of: host
skipping deployment of: gnb1
skipping deployment of: ue11
skipping deployment of: ue12
skipping deployment of: gnb2
skipping deployment of: ue21
all services running
```

Figure 7: start of 5G Core (last messages)

8) Stop the capture and identify the involved protocols (to facilitate it, order the capture by the 'Protocol' column by pressing the respective column top); which of those are specific 5G Core protocols?
9) Identify the dialogs for the 5G protocols (suggestion: apply a display filter to those protocols and check the involved entities)

4. gNBs start

1) Open a (new) terminal window/tab

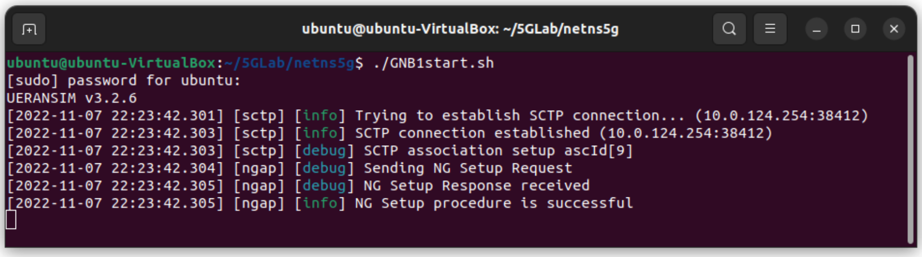
2) Start Wireshark instance and start capturing in interface br1 (do not stop Wireshark until the end of this section, in step 4.7)

   $ **sudo wireshark**

   Capture -> Options -> select 'br1'

3) From the same directory, start the first gNB (gNB1)

   ~/5GLab/netns5G$ **./GNB1start.sh**



```
ubuntu@ubuntu-VirtualBox: ~/5GLab/netns5g
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$ ./GNB1start.sh
[sudo] password for ubuntu:
UERANSIM v3.2.6
[2022-11-07 22:23:42.301] [sctp] [info] Trying to establish SCTP connection... (10.0.124.254:38412)
[2022-11-07 22:23:42.303] [sctp] [info] SCTP connection established (10.0.124.254:38412)
[2022-11-07 22:23:42.303] [sctp] [debug] SCTP association setup ascId[9]
[2022-11-07 22:23:42.304] [ngap] [debug] Sending NG Setup Request
[2022-11-07 22:23:42.305] [ngap] [debug] NG Setup Response received
[2022-11-07 22:23:42.305] [ngap] [info] NG Setup procedure is successful
```

Figure 8: GNB start log

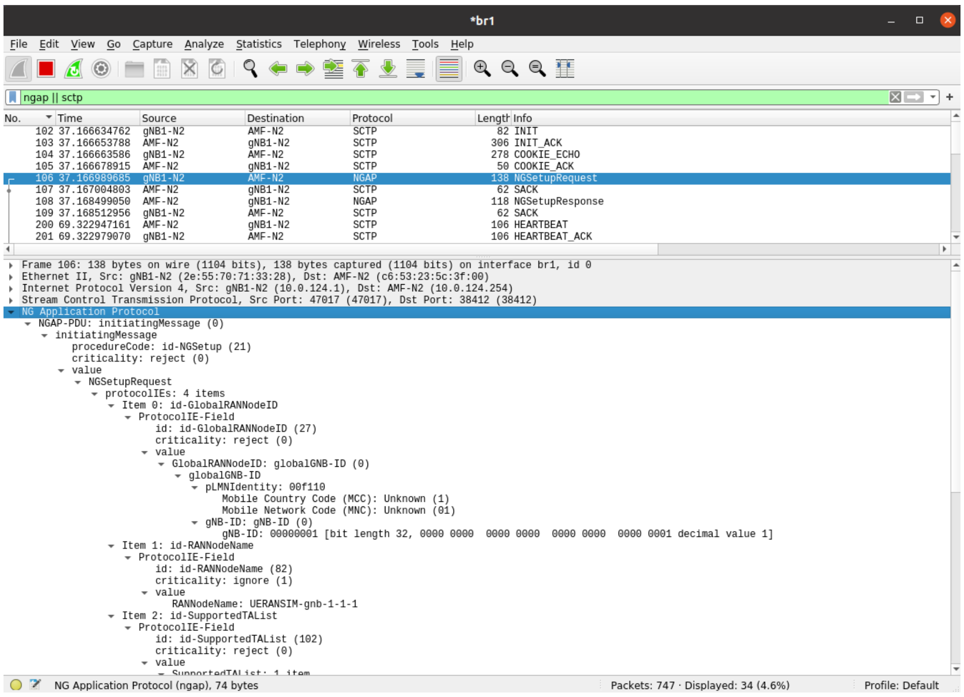
4) In the live Wireshark capture, observe/note the following:

   a. Repeat the identification of the involved protocols and the specific 5G ones

   b. The SCTP connection setup and later the exchanged heartbeats (suggestion: filter the displayed packets by identified 5G protocols)

   c. Identify the involved entities

   d. Detail to the maximum extent, in the Packet Details window, the *NGsetupRequest* and *NGsetupResponse* messages (with mouse right button in '*Packet Details*', select '*Expand Subtrees*'); Confirm observed values with the ones obtained from the configuration files analysis


*br1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ngap || sctp

No.   Time          Source    Destination  Protocol  Length Info
  102 37.166634762  gNB1-N2   AMF-N2       SCTP          82 INIT
  103 37.166653788  AMF-N2    gNB1-N2      SCTP         306 INIT_ACK
  104 37.166663586  gNB1-N2   AMF-N2       SCTP         278 COOKIE_ECHO
  105 37.166678915  AMF-N2    gNB1-N2      SCTP          50 COOKIE_ACK
  106 37.166989685  gNB1-N2   AMF-N2       NGAP         138 NGSetupRequest
  107 37.167004803  AMF-N2    gNB1-N2      SCTP          62 SACK
  108 37.168499050  AMF-N2    gNB1-N2      NGAP         118 NGSetupResponse
  109 37.168512956  gNB1-N2   AMF-N2       SCTP          62 SACK
  200 69.322947161  AMF-N2    gNB1-N2      SCTP         106 HEARTBEAT
  201 69.322979070  gNB1-N2   AMF-N2       SCTP         106 HEARTBEAT_ACK

Frame 106: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface br1, id 0
Ethernet II, Src: gNB1-N2 (2e:55:70:71:33:28), Dst: AMF-N2 (c6:53:23:5c:3f:00)
Internet Protocol Version 4, Src: gNB1-N2 (10.0.124.1), Dst: AMF-N2 (10.0.124.254)
Stream Control Transmission Protocol, Src Port: 47017 (47017), Dst Port: 38412 (38412)
NG Application Protocol
  NGAP-PDU: initiatingMessage (0)
    initiatingMessage
      procedureCode: id-NGSetup (21)
      criticality: reject (0)
      value
        NGSetupRequest
          protocolIEs: 4 items
            Item 0: id-GlobalRANNodeID
              ProtocolIE-Field
                id: id-GlobalRANNodeID (27)
                criticality: reject (0)
                value
                  GlobalRANNodeID: globalGNB-ID (0)
                    globalGNB-ID
                      pLMNIdentity: 00f110
                        Mobile Country Code (MCC): Unknown (1)
                        Mobile Network Code (MNC): Unknown (01)
                      gNB-ID: gNB-ID (0)
                        gNB-ID: 00000001 [bit length 32, 0000 0000  0000 0000  0000 0000  0000 0001 decimal value 1]
            Item 1: id-RANNodeName
              ProtocolIE-Field
                id: id-RANNodeName (82)
                criticality: ignore (1)
                value
                  RANNodeName: UERANSIM-gnb-1-1-1
            Item 2: id-SupportedTAList
              ProtocolIE-Field
                id: id-SupportedTAList (102)
                criticality: reject (0)
                value
                  SupportedTAList: 1 item

NG Application Protocol (ngap), 74 bytes     Packets: 747 · Displayed: 34 (4.6%)     Profile: Default
```

Figure 9: Wireshark capture of gNB start

5) Start the second gNB (gNB2)

```
~/5GLab/netns5G$ ./GNB2start.sh
```

6) In the live Wireshark compare the new *NGsetupRequest* and *NGsetupResponse* messages with previous ones (gBN1) (suggestion: apply a display filter for the NGAP protocol only and order the capture by the 'Info' column and then move the two pairs of captured packets)

7) Observe the logs in the screen (Core, gNB1 and gNB2) and logfiles in: ~/5GLab/netns5g/logs (suggestion: use the 'Files' application to see and open the most recent files, the ones generated until now, executing this 5G Lab)

5. UE creation, registration and default PDU creation

1) Open the Free5GC Web Console from the web browser:
   a. ***http://10.0.123.201:5000***
   b. credentials: '***admin***'/'***free5gc***'

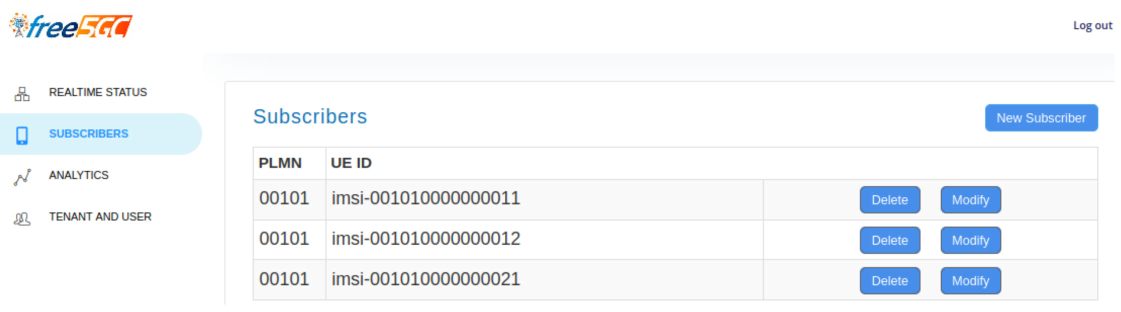
2) Create the 3 UEs from the table below ('***New Subscriber***'; see screen capture in Annex D):

	UE11	**UE12**	**UE21**
PLMN ID (MCC/MNC)	001 01	001 01	001 01
SUPI (IMSI)	001 01 0000 0000 **11**	001 01 0000 0000 **12**	001 01 0000 0000 **21**
SST/SD	1/010203 (sl1)	1/010203 (sl1) 2/112233 (sl2)	1/010203 (sl1)
DNN	internet	internet	internet
UL/DL AMBR	10/20 Mbps	100/200 Mbps	1/2 Mbps
5QI	9	9	9
Note	Will connect to **gNB1**	Will connect to **gNB1**	Will connect to **gNB2**

Notes:

1) Only change the parameters shown in the table and if required
   a. **do not change**: Authentication method, K*, Operator Code Type, Operator Code Value*, and SQN*)
   b. you may search and interpret the other parameters.

2) In the Free5GC "New Subscriber" form, delete the second appearing S-NSSAI (*Single Network Slice Selection Assistance Information*) and the second DNN ('internet2')



Subscribers

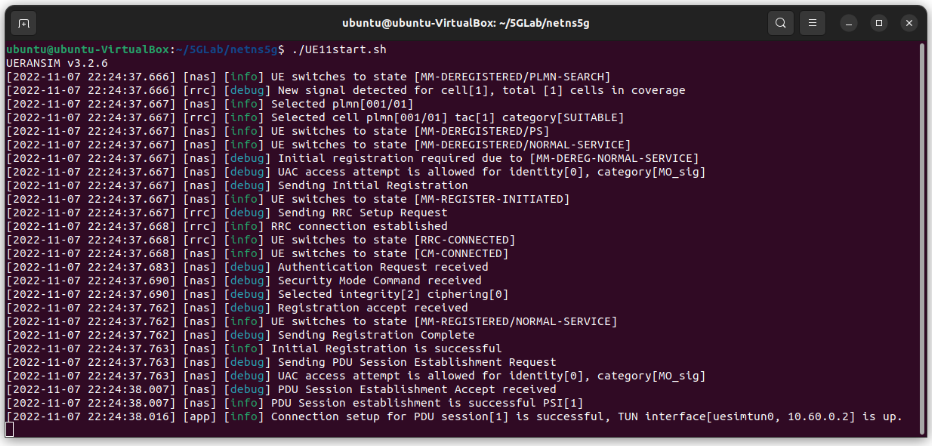
PLMN	UE ID		
00101	imsi-001010000000011	Delete	Modify
00101	imsi-001010000000012	Delete	Modify
00101	imsi-001010000000021	Delete	Modify

Figure 10: free5GC WebGUI after creation of the 3 subscribers

3) Restart the Wireshark capture, keeping the capture in the same interface ('br1')

4) Start the first UE (UE11)

   ~/5GLab/netns5G$ **./UE11start.sh**



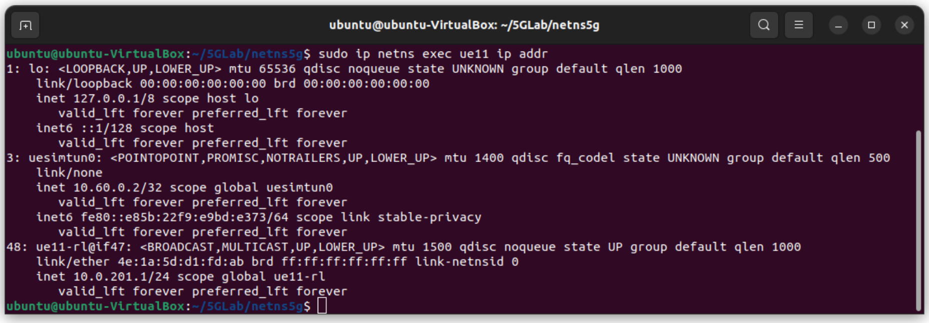
```
ubuntu@ubuntu-VirtualBox: ~/5GLab/netns5g
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$ ./UE11start.sh
UERANSIM v3.2.6
[2022-11-07 22:24:37.666] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2022-11-07 22:24:37.666] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2022-11-07 22:24:37.667] [nas] [info] Selected plmn[001/01]
[2022-11-07 22:24:37.667] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
[2022-11-07 22:24:37.667] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2022-11-07 22:24:37.667] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2022-11-07 22:24:37.667] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2022-11-07 22:24:37.667] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2022-11-07 22:24:37.667] [nas] [debug] Sending Initial Registration
[2022-11-07 22:24:37.667] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2022-11-07 22:24:37.667] [rrc] [debug] Sending RRC Setup Request
[2022-11-07 22:24:37.668] [rrc] [info] RRC connection established
[2022-11-07 22:24:37.668] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2022-11-07 22:24:37.668] [nas] [info] UE switches to state [CM-CONNECTED]
[2022-11-07 22:24:37.683] [nas] [debug] Authentication Request received
[2022-11-07 22:24:37.690] [nas] [debug] Security Mode Command received
[2022-11-07 22:24:37.690] [nas] [debug] Selected integrity[2] ciphering[0]
[2022-11-07 22:24:37.762] [nas] [debug] Registration accept received
[2022-11-07 22:24:37.762] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2022-11-07 22:24:37.762] [nas] [debug] Sending Registration Complete
[2022-11-07 22:24:37.763] [nas] [info] Initial Registration is successful
[2022-11-07 22:24:37.763] [nas] [debug] Sending PDU Session Establishment Request
[2022-11-07 22:24:37.763] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2022-11-07 22:24:38.007] [nas] [debug] PDU Session Establishment Accept received
[2022-11-07 22:24:38.007] [nas] [info] PDU Session establishment is successful PSI[1]
[2022-11-07 22:24:38.016] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.60.0.2] is up.
```

Figure 11: log of UE11 start

5) Observe the states the UE went through, during the process; observe the other messages and its sequence

6) Observe the creation of the new TUN interface ('uesimtun0'); in a new terminal window, you can check the creation of this interface in namespace 'ue11' and note its IP address (10.60.0.2, in the example)

   ~/5GLab/netns5G$ **sudo ip netns exec ue11 ip addr**



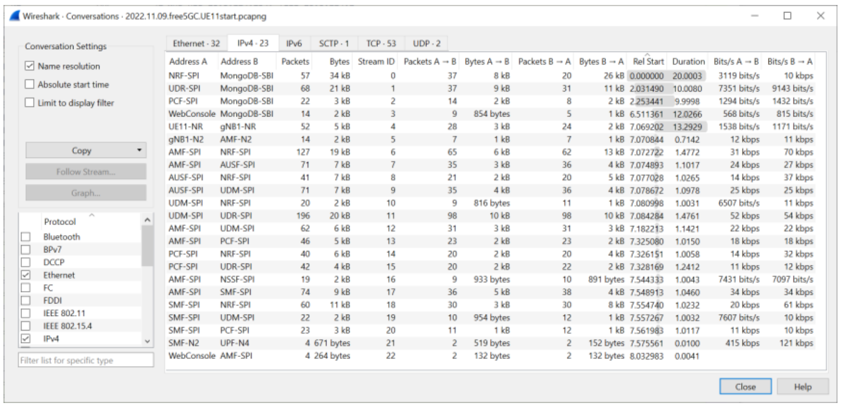
```
ubuntu@ubuntu-VirtualBox: ~/5GLab/netns5g
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$ sudo ip netns exec ue11 ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.60.0.2/32 scope global uesimtun0
       valid_lft forever preferred_lft forever
    inet6 fe80::e85b:22f9:e9bd:e373/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
48: ue11-rl@if47: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 4e:1a:5d:d1:fd:ab brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.201.1/24 scope global ue11-rl
       valid_lft forever preferred_lft forever
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$
```

Figure 12: check UE11 assigned IP addresses

This interface will be used to exchange the traffic via the 5G network.

- Order the Wireshark displayed capture by 'Protocol' (press the respective column name) and list the relevant protocols

7) Go to 'Statistics' and select 'Conversations' in Wireshark, order by 'IPv4' and enable 'Name resolution'; order by 'Rel Start' and observe the 5G dialogs; you may also order 'Address A' and 'Address B'



Conversation Settings: [✓] Name resolution; [ ] Absolute start time; [ ] Limit to display filter

Protocol: [ ] Bluetooth; [ ] BPv7; [ ] DCCP; [✓] Ethernet; [ ] FC; [ ] FDDI; [ ] IEEE 802.11; [ ] IEEE 802.15.4; [✓] IPv4

Address A	Address B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
NRF-SPI	MongoDB-SBI	57	34 kB	0	37	8 kB	20	26 kB	0.000000	20.0003	3119 bits/s	10 kbps
UDR-SPI	MongoDB-SBI	68	21 kB	1	37	9 kB	31	11 kB	2.031490	10.0080	7351 bits/s	9143 bits/s
PCF-SPI	MongoDB-SBI	22	3 kB	2	14	2 kB	8	2 kB	2.253441	9.9998	1294 bits/s	1432 bits/s
WebConsole	MongoDB-SBI	14	2 kB	3	9	854 bytes	5	1 kB	6.511361	12.0266	568 bits/s	815 bits/s
UE11-NR	gNB1-NR	52	5 kB	4	28	3 kB	24	2 kB	7.069202	13.2929	1538 bits/s	1171 bits/s
gNB1-N2	AMF-N2	14	2 kB	5	7	1 kB	7	1 kB	7.070844	0.7142	12 kbps	11 kbps
AMF-SPI	NRF-SPI	127	19 kB	6	65	6 kB	62	13 kB	7.072722	1.4772	31 kbps	70 kbps
AMF-SPI	AUSF-SPI	71	7 kB	7	35	3 kB	36	4 kB	7.074893	1.1017	24 kbps	27 kbps
AUSF-SPI	NRF-SPI	41	7 kB	8	21	2 kB	20	5 kB	7.077028	1.0265	14 kbps	37 kbps
AUSF-SPI	UDM-SPI	71	7 kB	9	35	4 kB	36	4 kB	7.078672	1.0978	25 kbps	25 kbps
UDM-SPI	NRF-SPI	20	2 kB	10	9	816 bytes	11	1 kB	7.080998	1.0031	6507 bits/s	11 kbps
UDM-SPI	UDR-SPI	196	20 kB	11	98	10 kB	98	10 kB	7.084284	1.4761	52 kbps	54 kbps
AMF-SPI	UDM-SPI	62	6 kB	12	31	3 kB	31	3 kB	7.182213	1.1421	22 kbps	22 kbps
AMF-SPI	PCF-SPI	46	5 kB	13	23	2 kB	23	2 kB	7.325080	1.0150	18 kbps	18 kbps
PCF-SPI	NRF-SPI	40	6 kB	14	20	2 kB	20	4 kB	7.326151	1.0058	14 kbps	32 kbps
PCF-SPI	UDR-SPI	42	4 kB	15	20	2 kB	22	2 kB	7.328169	1.2412	11 kbps	12 kbps
AMF-SPI	NSSF-SPI	19	2 kB	16	9	933 bytes	10	891 bytes	7.544333	1.0043	7431 bits/s	7097 bits/s
AMF-SPI	SMF-SPI	74	9 kB	17	36	5 kB	38	4 kB	7.548913	1.0460	34 kbps	34 kbps
SMF-SPI	NRF-SPI	60	11 kB	18	30	3 kB	30	8 kB	7.554740	1.0232	20 kbps	61 kbps
SMF-SPI	UDM-SPI	22	2 kB	19	10	954 bytes	12	1 kB	7.557267	1.0032	7607 bits/s	10 kbps
SMF-SPI	PCF-SPI	23	3 kB	20	11	1 kB	12	1 kB	7.561983	1.0117	11 kbps	10 kbps
SMF-N2	UPF-N4	4	671 bytes	21	2	519 bytes	2	152 bytes	7.575561	0.0100	415 kbps	121 kbps
WebConsole	AMF-SPI	4	264 bytes	22	2	132 bytes	2	132 bytes	8.032983	0.0041		

Figure 13: interactions analysis for UE11 start

8) Apply a Display Filter to see just NGAP and PFCP protocols (“***ngap or pfcp***”)

    a. Identify the involved 5G control functions (IP addresses are already translated to the functional entity interface, according to the diagram above); identify the dialogs UE-AMF, AMF-SMF, SMF-UPF and their sequence

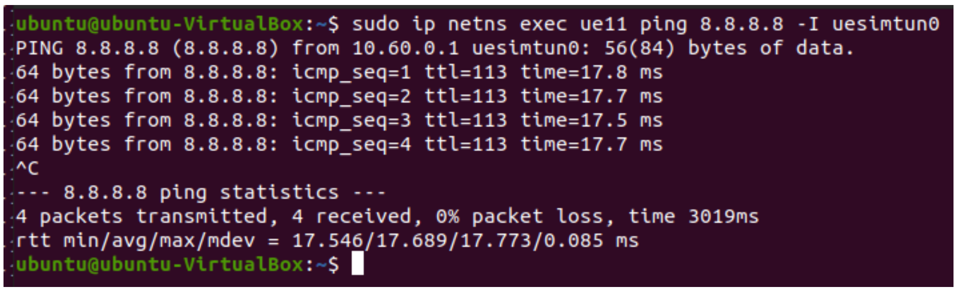
    b. Observe the sequence of exchanged messages, looking into their details in the Packet Details window (see, for instance, the ‘*PFCP Session Establishment Request*’ and compare with message ‘*PFCP Session Modification Request*’)

    c. You may filter the display of messages by protocol and pair of entities, filtering the protocol and their IP addresses (e.g. for HTTP2 between AMF and AUSF: “***ip.addr==10.0.123.4 and ip.addr==10.0.123.6) and http2***”)

6. Connectivity

1) Start a Wireshark capture in the interface ‘upf-n3’ and another capture in the interface ‘upf-n6-0'
2) Apply a Display Filter to see protocols GTP and ICMP
3) In a terminal window, start a ping to 8.8.8.8 from UE11

    ~/5GLab/netns5G$ **sudo ip netns exec ue11 ping 8.8.8.8 -I uesimtun0**



```
ubuntu@ubuntu-VirtualBox:~$ sudo ip netns exec ue11 ping 8.8.8.8 -I uesimtun0
PING 8.8.8.8 (8.8.8.8) from 10.60.0.1 uesimtun0: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=17.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=17.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=17.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=17.7 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3019ms
rtt min/avg/max/mdev = 17.546/17.689/17.773/0.085 ms
ubuntu@ubuntu-VirtualBox:~$
```

Figure 14: UE11 working ping to an external test IP (8.8.8.8 as example)

4) Analyse, in the Wireshark Packet Details, the GTP encapsulation
    - Observe the *Tunnel Endpoint IDentifier* (TEID) in both directions of the communication
5) In a new Terminal Window/Tab, Start UE12

    ~/5GLab/netns5G$ **./UE12start-sl1.sh**

    (check the contents of file *./config/free5gc-ue12-sl1.yaml*)
6) Make a ping from UE11 to UE12

    ~/5GLab/netns5G$ **sudo ip netns exec ue11 ping <U12 IP addr> -I uesimtun0**

    - Analyse the observed GTP packets
7) In a new Terminal Window/Tab, Start UE21

    ~/5GLab/netns5G$ **./UE12start-sl1.sh**

    (check the contents of file *./config/free5gc-ue21.yaml*)
8) Make a ping from UE12 to UE21 and observe the exchanged packets at the UPF

7. QoS (<u>optional</u>; not for the evaluation Quiz)

1) Open a new terminal window
2) Start an iperf3 server at the DNN

    $ ***iperf3 –s***

3) Check the TUN interface name and assigned IP address

   $ **sudo ip netns exec ue11 ip addr**

4) Start an iperf3 client at UE11 towards the server instance and register the achieved bandwidth in the UL and DL directions

   $ **sudo ip netns exec ue11 iperf3 -c 10.1.0.2 -B *<ue11 IP address>*** *-- uplink*

   $ **sudo ip netns exec ue11 iperf3 -c 10.1.0.2 -R -B *<ue11 IP address>*** *-- downlink*

5) Repeat previous measurements with the other two UEs (UE12 and UE21) and compare the results

8. Slicing (optional; not for the evaluation Quiz)

1) Stop UE12 (Ctrl-C)
2) Restart UE12, now in the second slice (2/112233) with a new configuration file and check the results

   ~/5GLab/netns5G$ **./UE12start-sl2.sh**

   (check the contents of file *./config/free5gc-ue12-sl2.yaml*)

3) Observe the newly assigned IP address; what are the changes?
4) Make a ping from UE11 to UE12, now in different slices and observe the exchanged packets at the UPF; Is there connectivity?
   a. Check routing at the UPF namespace

      $ **sudo ip netns exec upf ip route**

   b. Add a new route in the UPF namespace

      $ **sudo ip netns exec upf ip route add 10.61.0.0./24 dev upfgtp**

5) Repeat the ping above.

9. Stop and reset the environment

1) Stop the UEs, gNB nodes (Ctrl-C), and the 5G Core
2) Wait for final processes to close (this takes some seconds, ending with “NRF terminated”)
3) Delete the namespaces

   ~/5GLab/netns5G$ **sudo ./5Gcleanup.sh**

Annexes

A. 5G System architecture

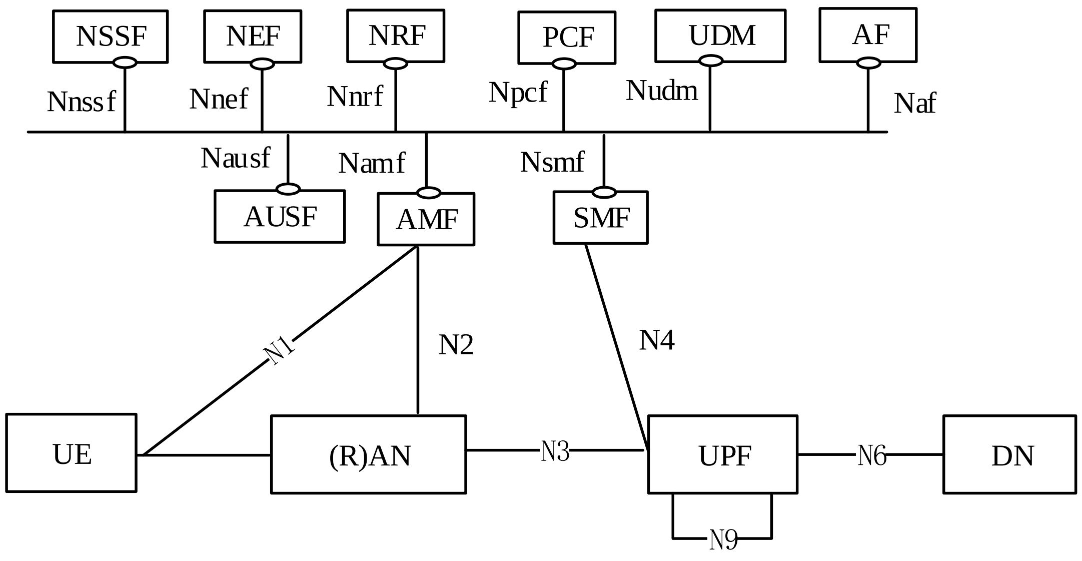


Figure 15: 5G Core, RAN and UE reference system architecture [3gpp]

B. Example procedure

Release 16 87 3GPP TS 23.502 V16.4.0 (2020-03)

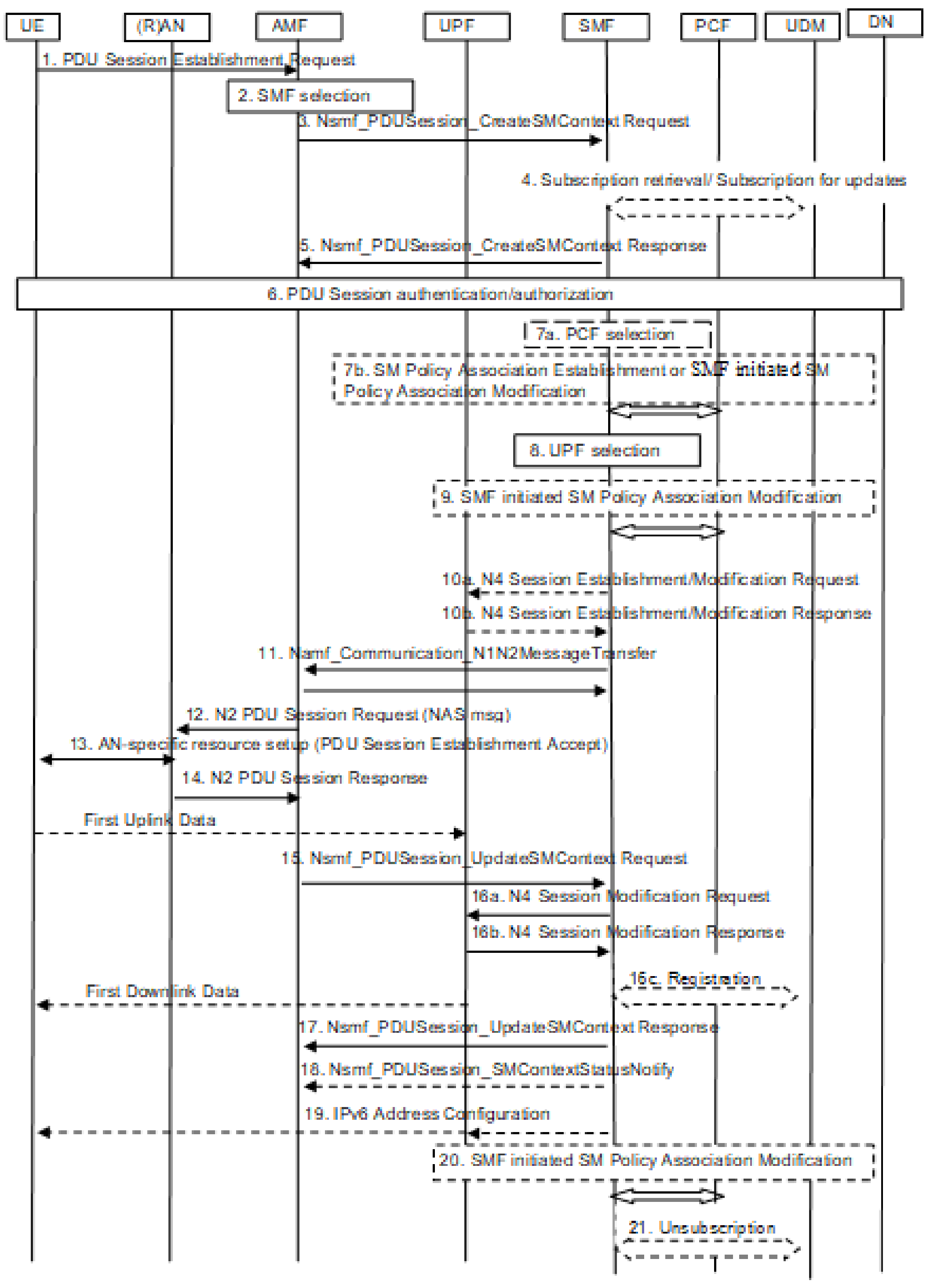
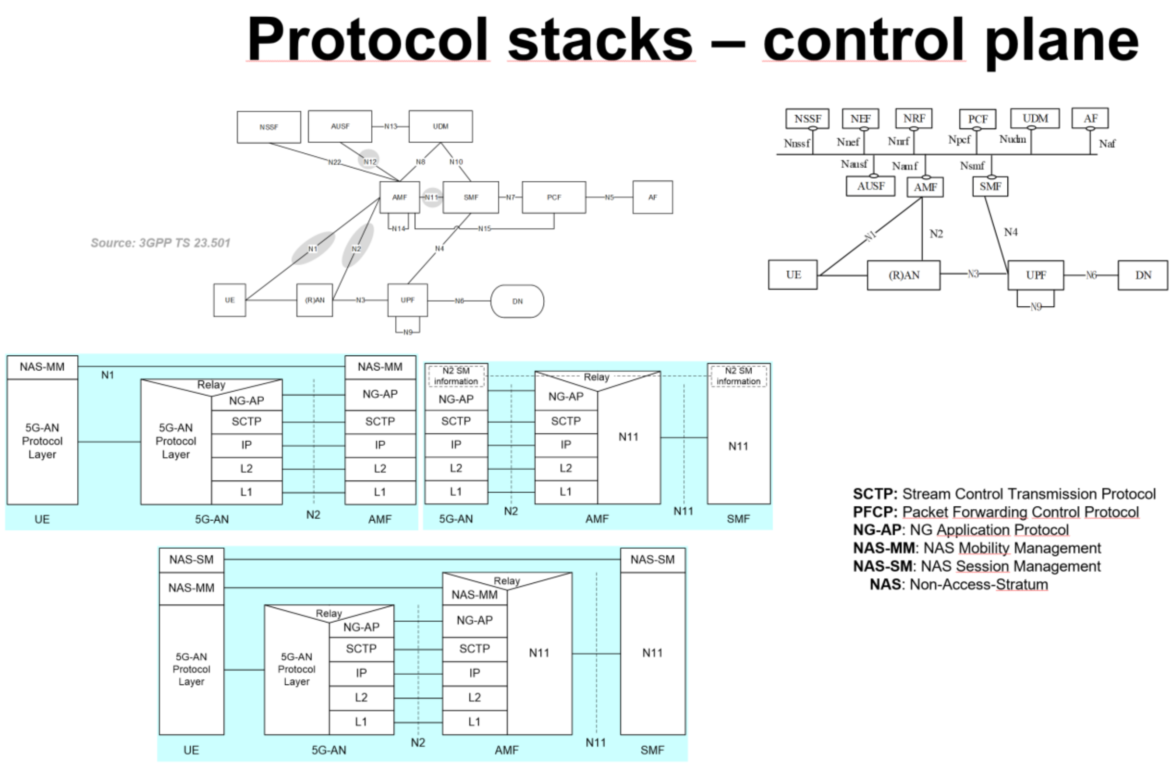

UE
(R)AN
AMF
UPF
SMF
PCF
UDM
DN
1. PDU Session Establishment Request
2. SMF selection
3. Nsmf_PDUSession_CreateSMContext Request
4. Subscription retrieval/ Subscription for updates
5. Nsmf_PDUSession_CreateSMContext Response
6. PDU Session authentication/authorization
7a. PCF selection
7b. SM Policy Association Establishment or SMF initiated SM Policy Association Modification
8. UPF selection
9. SMF initiated SM Policy Association Modification
10a. N4 Session Establishment/Modification Request
10b. N4 Session Establishment/Modification Response
11. Namf_Communication_N1N2MessageTransfer
12. N2 PDU Session Request (NAS msg)
13. AN-specific resource setup (PDU Session Establishment Accept)
14. N2 PDU Session Response
First Uplink Data
15. Nsmf_PDUSession_UpdateSMContext Request
16a. N4 Session Modification Request
16b. N4 Session Modification Response
16c. Registration
First Downlink Data
17. Nsmf_PDUSession_UpdateSMContext Response
18. Nsmf_PDUSession_SMContextStatusNotify
19. IPv6 Address Configuration
20. SMF initiated SM Policy Association Modification
21. Unsubscription

Figure 4.3.2.2.1-1: UE-requested PDU Session Establishment for non-roaming and roaming with local breakout

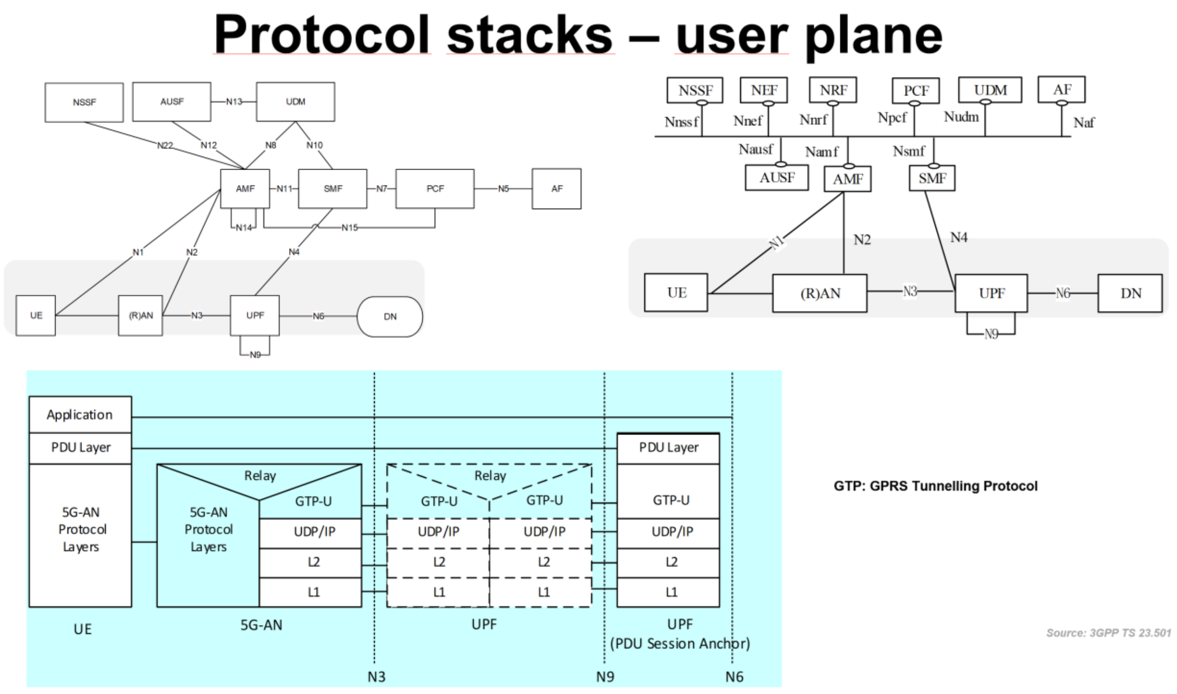
C. 5G Protocol stacks



Protocol stacks – control plane

SCTP: Stream Control Transmission Protocol
PFCP: Packet Forwarding Control Protocol
NG-AP: NG Application Protocol
NAS-MM: NAS Mobility Management
NAS-SM: NAS Session Management
NAS: Non-Access-Stratum

Figure 16: Protocol stacks, control plane

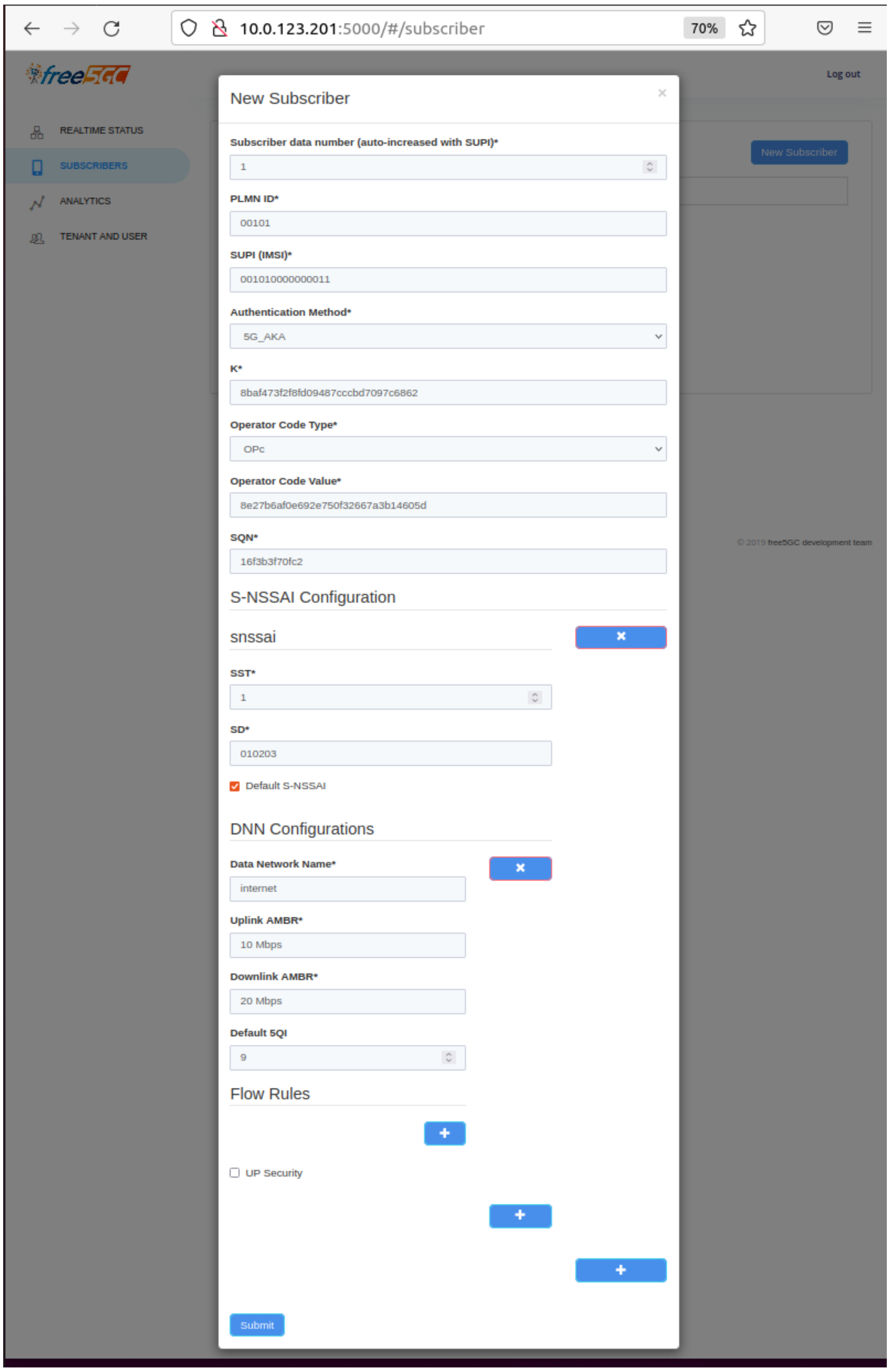


Protocol stacks – user plane

GTP: GPRS Tunnelling Protocol

Figure 17: Protocol stacks, user plane

D. Free5GC New Subscriber creation form (UE11)



New Subscriber

Subscriber data number (auto-increased with SUPI)*	1
PLMN ID*	00101
SUPI (IMSI)*	001010000000011
Authentication Method*	5G_AKA
K*	8baf473f2f8fd09487cccbd7097c6862
Operator Code Type*	OPc
Operator Code Value*	8e27b6af0e692e750f32667a3b14605d
SQN*	16f3b3f70fc2

S-NSSAI Configuration

snssai

SST*	1
SD*	010203

[✓] Default S-NSSAI

DNN Configurations

Data Network Name*	internet
Uplink AMBR*	10 Mbps
Downlink AMBR*	20 Mbps
Default 5QI	9

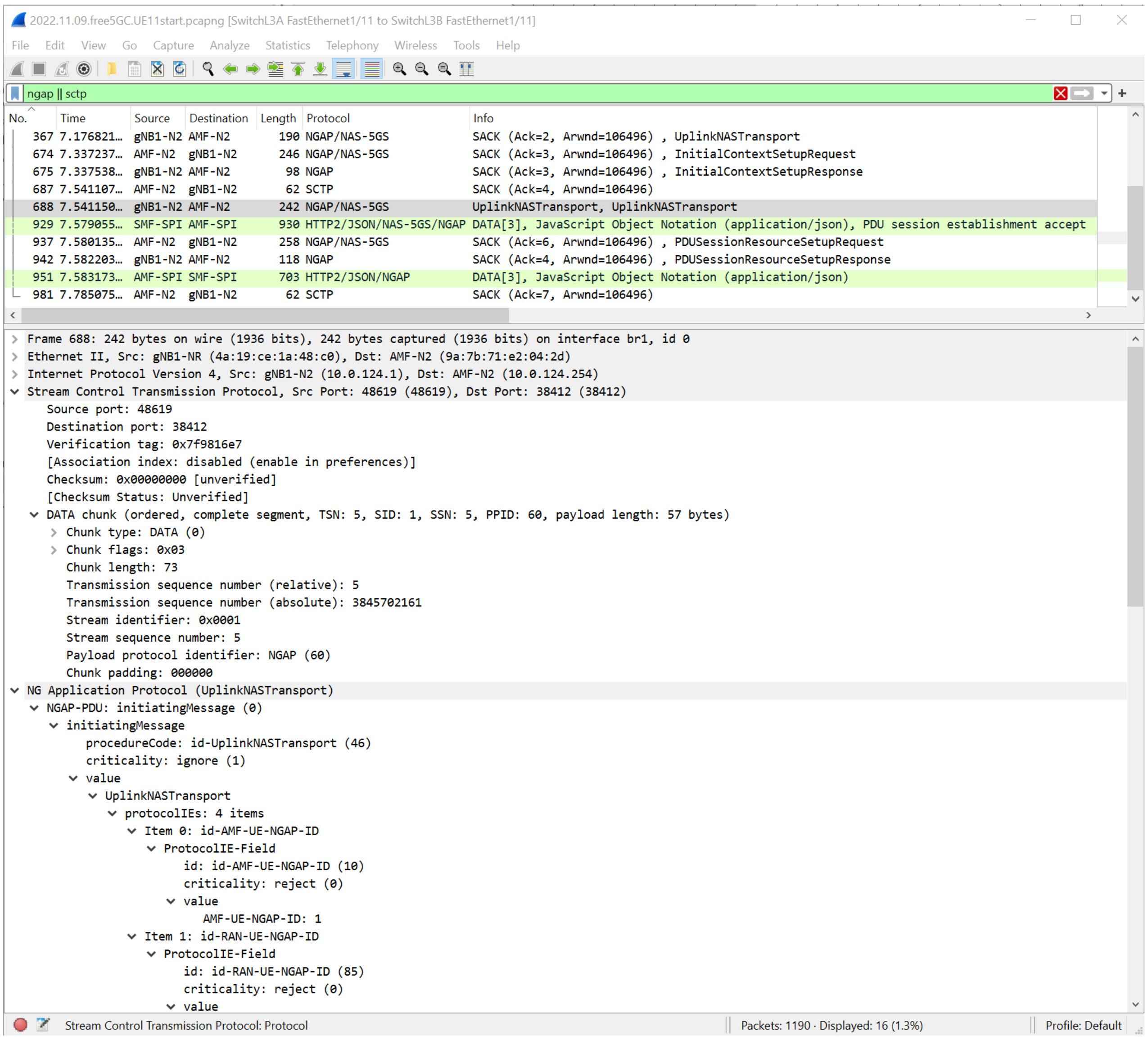
Flow Rules

[ ] UP Security

Submit

Figure 18: Example of a subscriber provisioning at Free5GC (UE11)

E. Example of a capture with Wireshark (with addresses resolution and display filter)



2022.11.09.free5GC.UE11start.pcapng [SwitchL3A FastEthernet1/11 to SwitchL3B FastEthernet1/11]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

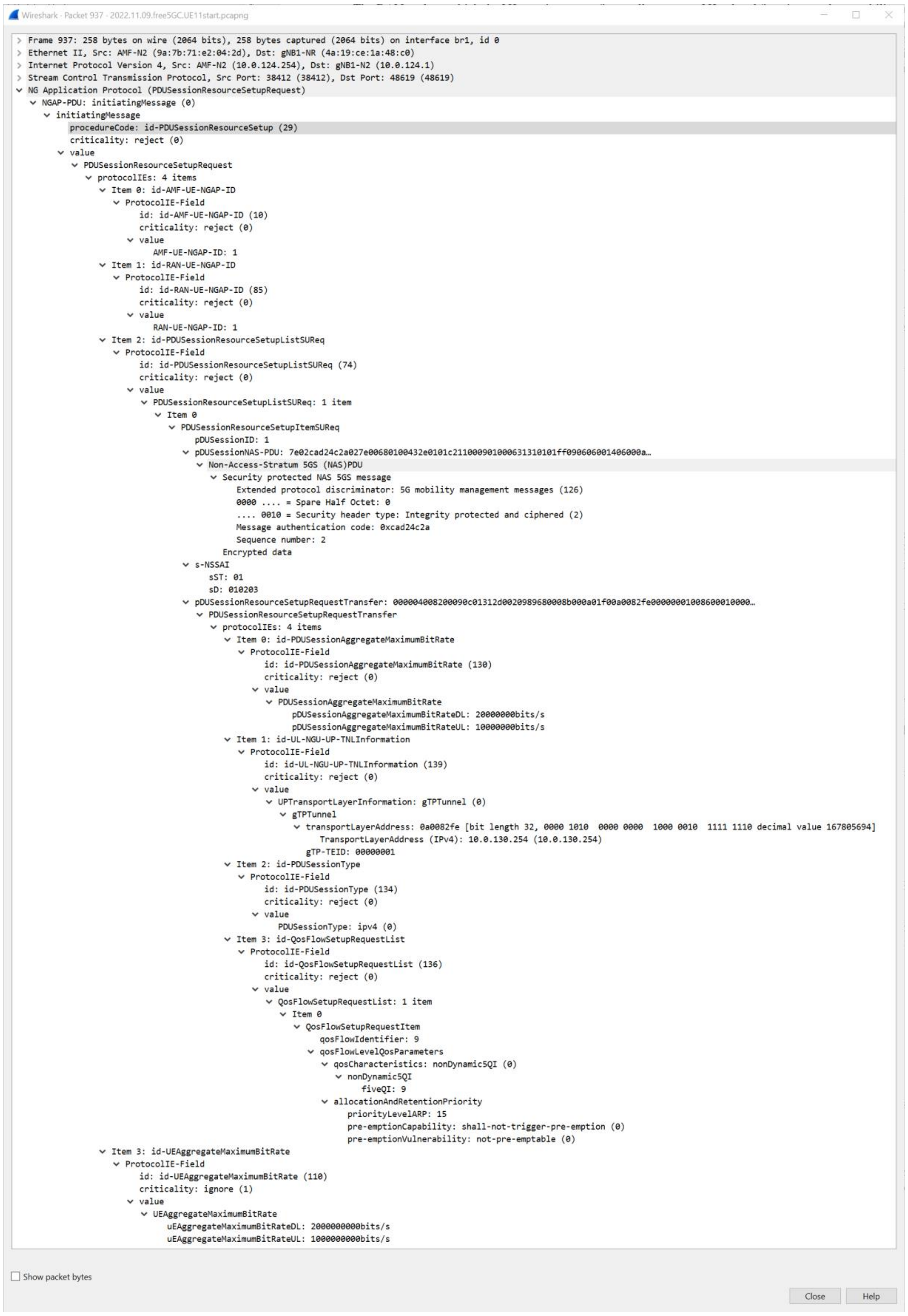
ngap || sctp

```
No.   Time        Source    Destination  Length Protocol                 Info
  367 7.176821…  gNB1-N2   AMF-N2         190 NGAP/NAS-5GS            SACK (Ack=2, Arwnd=106496) , UplinkNASTransport
  674 7.337237…  AMF-N2    gNB1-N2        246 NGAP/NAS-5GS            SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest
  675 7.337538…  gNB1-N2   AMF-N2          98 NGAP                    SACK (Ack=3, Arwnd=106496) , InitialContextSetupResponse
  687 7.541107…  AMF-N2    gNB1-N2         62 SCTP                    SACK (Ack=4, Arwnd=106496)
  688 7.541150…  gNB1-N2   AMF-N2         242 NGAP/NAS-5GS            UplinkNASTransport, UplinkNASTransport
  929 7.579055…  SMF-SPI   AMF-SPI        930 HTTP2/JSON/NAS-5GS/NGAP DATA[3], JavaScript Object Notation (application/json), PDU session establishment accept
  937 7.580135…  AMF-N2    gNB1-N2        258 NGAP/NAS-5GS            SACK (Ack=6, Arwnd=106496) , PDUSessionResourceSetupRequest
  942 7.582203…  gNB1-N2   AMF-N2         118 NGAP                    SACK (Ack=4, Arwnd=106496) , PDUSessionResourceSetupResponse
  951 7.583173…  AMF-SPI   SMF-SPI        703 HTTP2/JSON/NGAP         DATA[3], JavaScript Object Notation (application/json)
  981 7.785075…  AMF-N2    gNB1-N2         62 SCTP                    SACK (Ack=7, Arwnd=106496)
```

```
> Frame 688: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits) on interface br1, id 0
> Ethernet II, Src: gNB1-NR (4a:19:ce:1a:48:c0), Dst: AMF-N2 (9a:7b:71:e2:04:2d)
> Internet Protocol Version 4, Src: gNB1-N2 (10.0.124.1), Dst: AMF-N2 (10.0.124.254)
v Stream Control Transmission Protocol, Src Port: 48619 (48619), Dst Port: 38412 (38412)
    Source port: 48619
    Destination port: 38412
    Verification tag: 0x7f9816e7
    [Association index: disabled (enable in preferences)]
    Checksum: 0x00000000 [unverified]
    [Checksum Status: Unverified]
  v DATA chunk (ordered, complete segment, TSN: 5, SID: 1, SSN: 5, PPID: 60, payload length: 57 bytes)
    > Chunk type: DATA (0)
    > Chunk flags: 0x03
      Chunk length: 73
      Transmission sequence number (relative): 5
      Transmission sequence number (absolute): 3845702161
      Stream identifier: 0x0001
      Stream sequence number: 5
      Payload protocol identifier: NGAP (60)
      Chunk padding: 000000
v NG Application Protocol (UplinkNASTransport)
  v NGAP-PDU: initiatingMessage (0)
    v initiatingMessage
        procedureCode: id-UplinkNASTransport (46)
        criticality: ignore (1)
      v value
        v UplinkNASTransport
          v protocolIEs: 4 items
            v Item 0: id-AMF-UE-NGAP-ID
              v ProtocolIE-Field
                  id: id-AMF-UE-NGAP-ID (10)
                  criticality: reject (0)
                v value
                    AMF-UE-NGAP-ID: 1
            v Item 1: id-RAN-UE-NGAP-ID
              v ProtocolIE-Field
                  id: id-RAN-UE-NGAP-ID (85)
                  criticality: reject (0)
                v value
```

Stream Control Transmission Protocol: Protocol | Packets: 1190 · Displayed: 16 (1.3%) | Profile: Default

Figure 19: Example of a 5G exchange of messages (gNB – AMF interactions)



Wireshark - Packet 937 - 2022.11.09.free5GC.UE11start.pcapng

```
> Frame 937: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits) on interface br1, id 0
> Ethernet II, Src: AMF-N2 (9a:7b:71:e2:04:2d), Dst: gNB1-NR (4a:19:ce:1a:48:c0)
> Internet Protocol Version 4, Src: AMF-N2 (10.0.124.254), Dst: gNB1-N2 (10.0.124.1)
> Stream Control Transmission Protocol, Src Port: 38412 (38412), Dst Port: 48619 (48619)
v NG Application Protocol (PDUSessionResourceSetupRequest)
  v NGAP-PDU: initiatingMessage (0)
    v initiatingMessage
        procedureCode: id-PDUSessionResourceSetup (29)
        criticality: reject (0)
      v value
        v PDUSessionResourceSetupRequest
          v protocolIEs: 4 items
            v Item 0: id-AMF-UE-NGAP-ID
              v ProtocolIE-Field
                  id: id-AMF-UE-NGAP-ID (10)
                  criticality: reject (0)
                v value
                    AMF-UE-NGAP-ID: 1
            v Item 1: id-RAN-UE-NGAP-ID
              v ProtocolIE-Field
                  id: id-RAN-UE-NGAP-ID (85)
                  criticality: reject (0)
                v value
                    RAN-UE-NGAP-ID: 1
            v Item 2: id-PDUSessionResourceSetupListSUReq
              v ProtocolIE-Field
                  id: id-PDUSessionResourceSetupListSUReq (74)
                  criticality: reject (0)
                v value
                  v PDUSessionResourceSetupListSUReq: 1 item
                    v Item 0
                      v PDUSessionResourceSetupItemSUReq
                          pDUSessionID: 1
                        v pDUSessionNAS-PDU: 7e02cad24c2a027e00680100432e0101c211000901000631310101ff090606001406000a…
                          v Non-Access-Stratum 5GS (NAS)PDU
                            v Security protected NAS 5GS message
                                Extended protocol discriminator: 5G mobility management messages (126)
                                0000 .... = Spare Half Octet: 0
                                .... 0010 = Security header type: Integrity protected and ciphered (2)
                                Message authentication code: 0xcad24c2a
                                Sequence number: 2
                              Encrypted data
                        v s-NSSAI
                            sST: 01
                            sD: 010203
                        v pDUSessionResourceSetupRequestTransfer: 000004008200090c01312d0020989680008b000a01f00a0082fe0000000100860001000…
                          v PDUSessionResourceSetupRequestTransfer
                            v protocolIEs: 4 items
                              v Item 0: id-PDUSessionAggregateMaximumBitRate
                                v ProtocolIE-Field
                                    id: id-PDUSessionAggregateMaximumBitRate (130)
                                    criticality: reject (0)
                                  v value
                                    v PDUSessionAggregateMaximumBitRate
                                        pDUSessionAggregateMaximumBitRateDL: 20000000bits/s
                                        pDUSessionAggregateMaximumBitRateUL: 10000000bits/s
                              v Item 1: id-UL-NGU-UP-TNLInformation
                                v ProtocolIE-Field
                                    id: id-UL-NGU-UP-TNLInformation (139)
                                    criticality: reject (0)
                                  v value
                                    v UPTransportLayerInformation: gTPTunnel (0)
                                      v gTPTunnel
                                        v transportLayerAddress: 0a0082fe [bit length 32, 0000 1010  0000 0000  1000 0010  1111 1110 decimal value 167805694]
                                            TransportLayerAddress (IPv4): 10.0.130.254 (10.0.130.254)
                                          gTP-TEID: 00000001
                              v Item 2: id-PDUSessionType
                                v ProtocolIE-Field
                                    id: id-PDUSessionType (134)
                                    criticality: reject (0)
                                  v value
                                      PDUSessionType: ipv4 (0)
                              v Item 3: id-QosFlowSetupRequestList
                                v ProtocolIE-Field
                                    id: id-QosFlowSetupRequestList (136)
                                    criticality: reject (0)
                                  v value
                                    v QosFlowSetupRequestList: 1 item
                                      v Item 0
                                        v QosFlowSetupRequestItem
                                            qosFlowIdentifier: 9
                                          v qosFlowLevelQosParameters
                                            v qosCharacteristics: nonDynamic5QI (0)
                                              v nonDynamic5QI
                                                  fiveQI: 9
                                            v allocationAndRetentionPriority
                                                priorityLevelARP: 15
                                                pre-emptionCapability: shall-not-trigger-pre-emption (0)
                                                pre-emptionVulnerability: not-pre-emptable (0)
            v Item 3: id-UEAggregateMaximumBitRate
              v ProtocolIE-Field
                  id: id-UEAggregateMaximumBitRate (110)
                  criticality: ignore (1)
                v value
                  v UEAggregateMaximumBitRate
                      uEAggregateMaximumBitRateDL: 2000000000bits/s
                      uEAggregateMaximumBitRateUL: 1000000000bits/s
```

Figure 20: Detail of a 5G exchanged message (AMF and gNB via N2 interface)

F. Hosts file

```
#5G Core
10.0.123.1   NRF-SBI
10.0.123.2   UDR-SBI
10.0.123.3   UDM-SBI
10.0.123.4   AUSF-SBI
10.0.123.5   NSSF-SBI
10.0.123.6   AMF-SBI
10.0.123.7   PCF-SBI
10.0.123.9   SMF-SBI
10.0.123.100 MongoDB-SBI
10.0.123.201 WebConsole

10.0.124.254 AMF-N2
10.0.124.1   gNB1-N2
10.0.124.2   gNB2-N2

10.0.140.2   SMF-N4
10.0.140.1   UPF-N4

#5G dataplane
10.1.0.1     UPF-N6
10.1.0.1     Host-N6

#RAN1
10.0.201.1   UE11-NR
10.0.201.2   UE12-NR
10.0.201.254 gNB1-NR

#RAN2
10.0.202.1   UE11-NR
10.0.202.254 gNB1-NR
```

G. Useful links

- **Free5GC:**
  - [free5Gcore] https://www.free5gc.org/
  - [free5gcwiki] https://github.com/free5gc/free5gc/wiki
  - [konrad] https://github.com/konradkar2/netns5g
- **UERANSIM:**
  - [ueransim] https://github.com/aligungr/UERANSIM/wiki
- **3GPP**
  - [3gpp] https://www.3gpp.org