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.

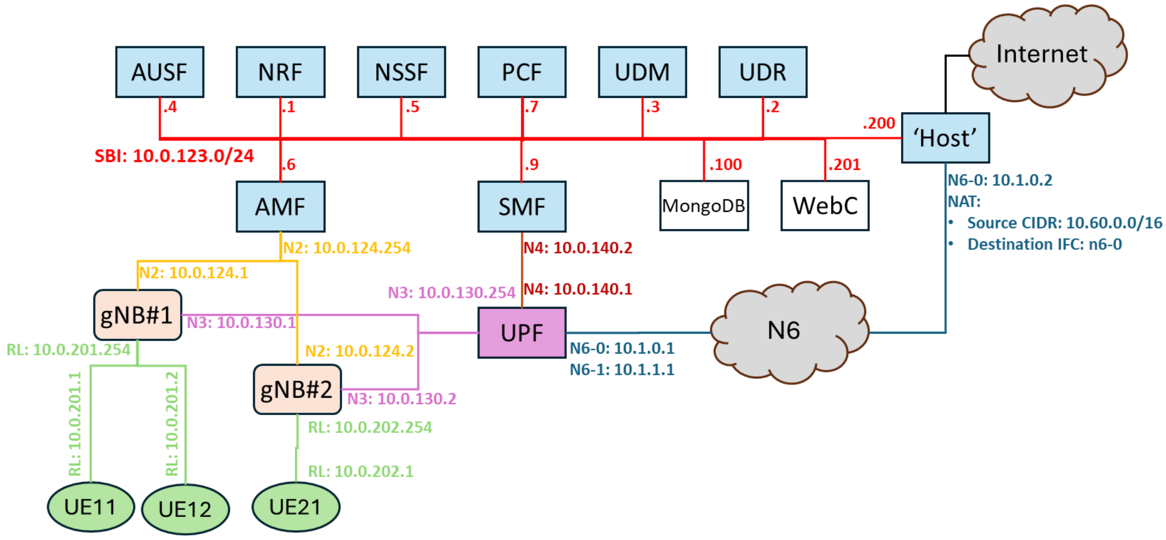


Figure 1: Network diagram

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).

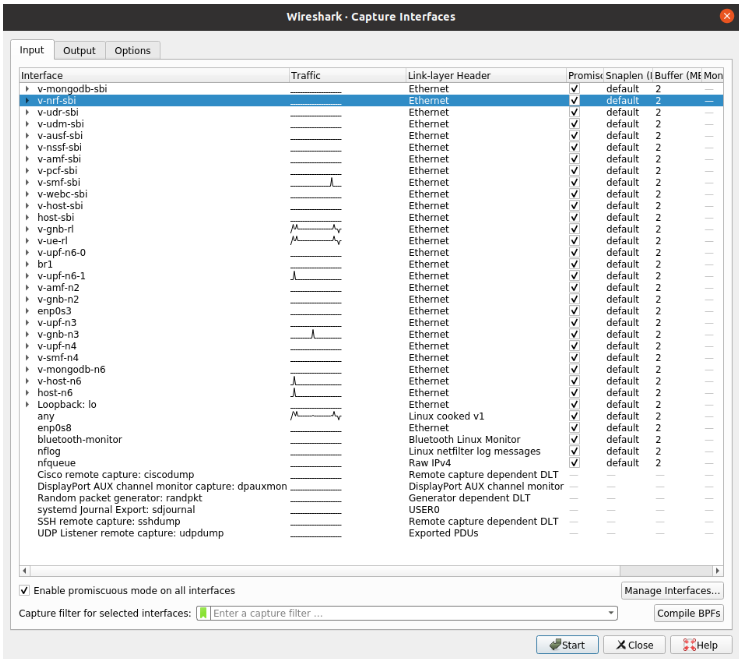


Figure 2: Logical interfaces as seen in Wireshark, after 5G Core start

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.

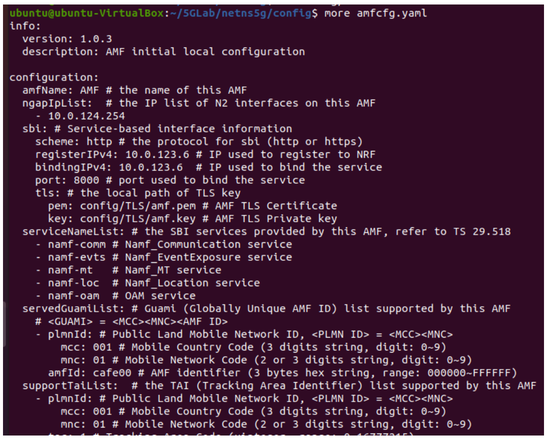
   a. **Files**:

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

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

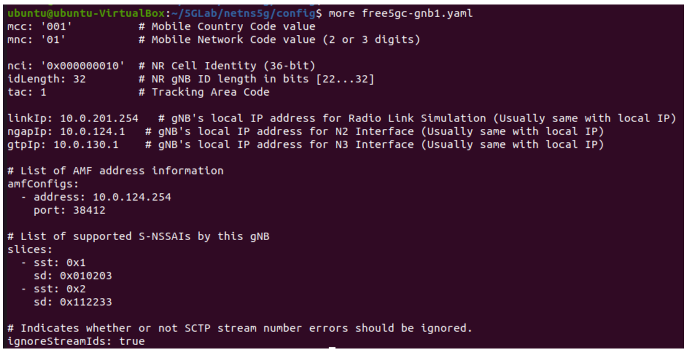
      iii. 5G UEs:

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



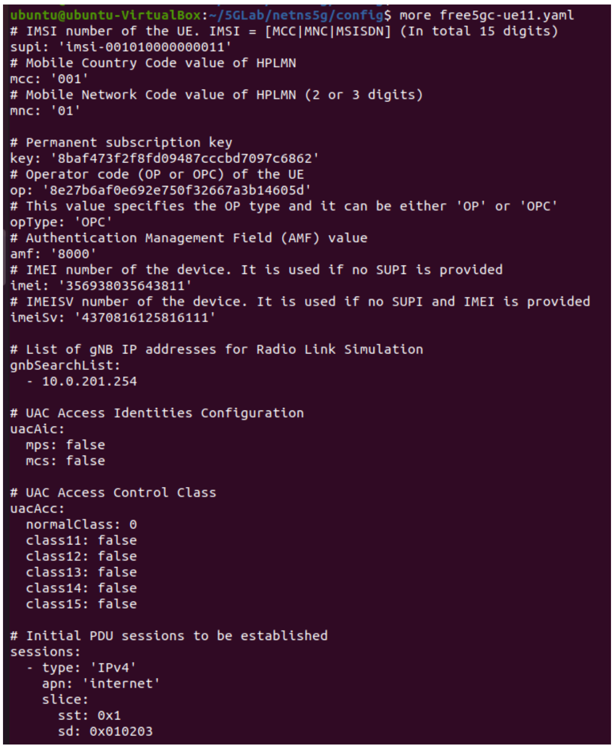
```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g/config$ more amfcfg.yaml
info:
  version: 1.0.3
  description: AMF initial local configuration

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 &***

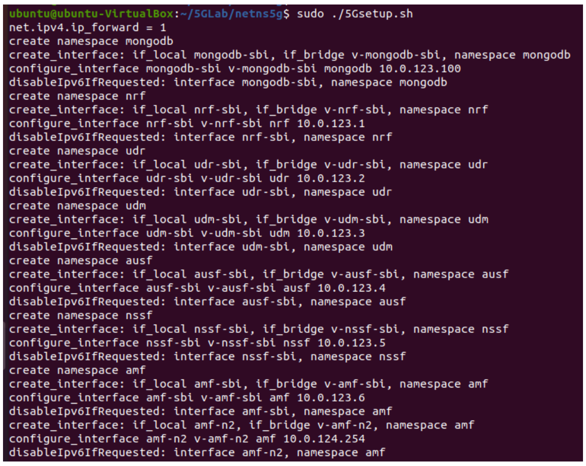


Figure 6: start of namespaces (not complete)

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.

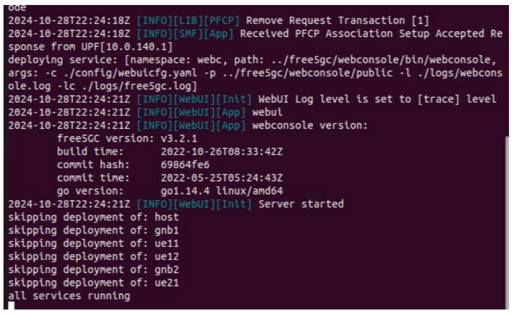


Figure 7: start of 5G Core (last messages)

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**

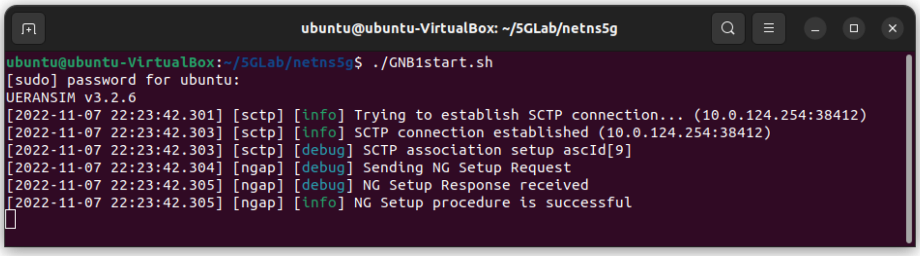


Figure 8: GNB start log

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

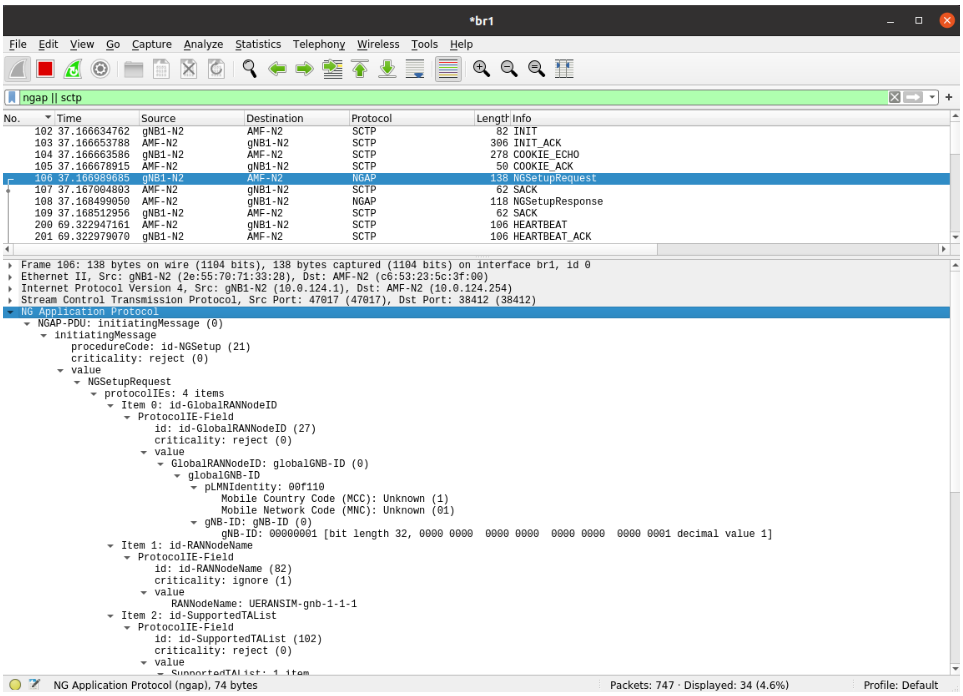


Figure 9: Wireshark capture of gNB start

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')

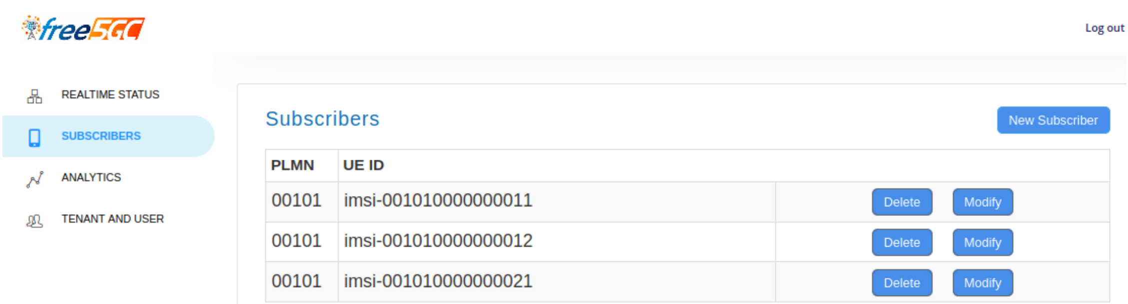


Figure 10: free5GC WebGUI after creation of the 3 subscribers

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

4) Start the first UE (UE11)

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

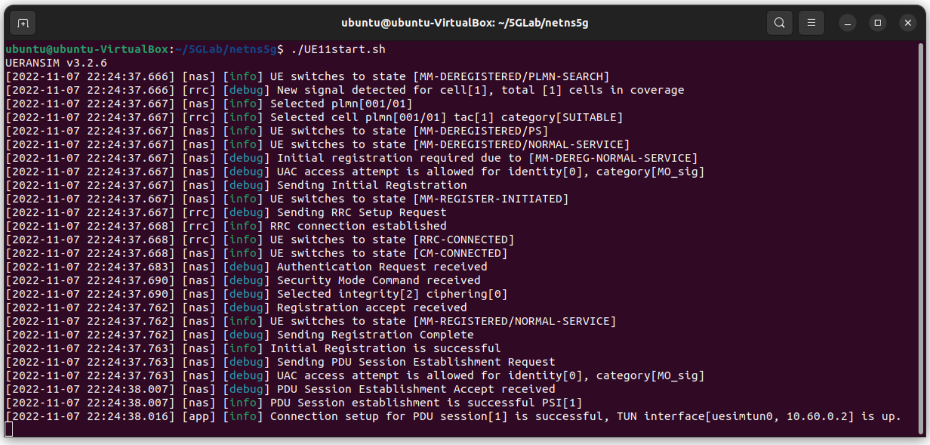


Figure 11: log of UE11 start

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**

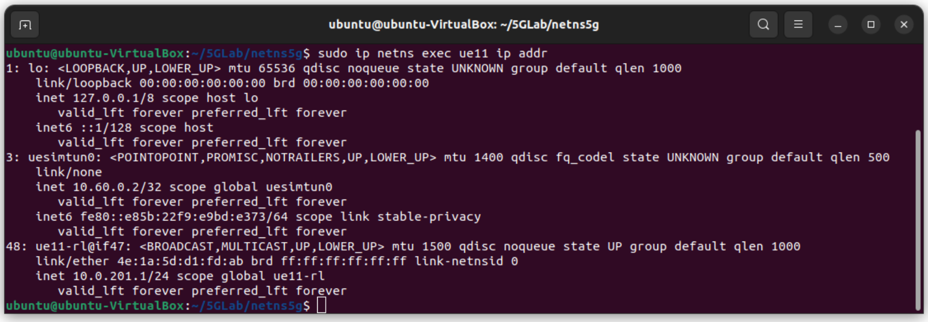


Figure 12: check UE11 assigned IP addresses

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 - 24' and enable 'Name resolution'; observe the 5G dialogs, ordering by 'Address A' and 'Address B'

8) Apply a Display Filter to see just NGAP, SCTP and PFCP protocols ("***ngap or pfcp or sctp***")

   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**

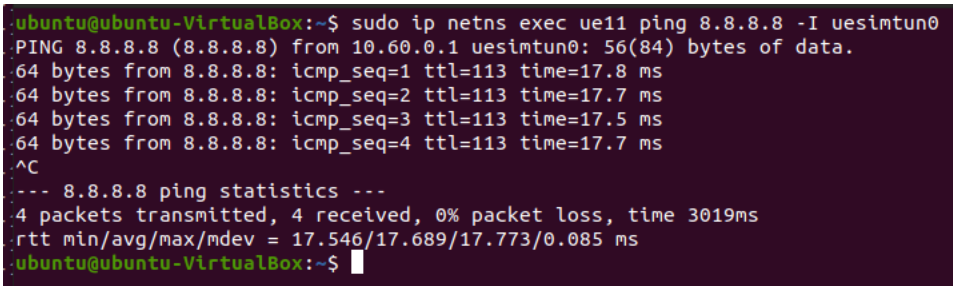


Figure 13: UE11 working ping to an external test IP (8.8.8.8 as example)

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

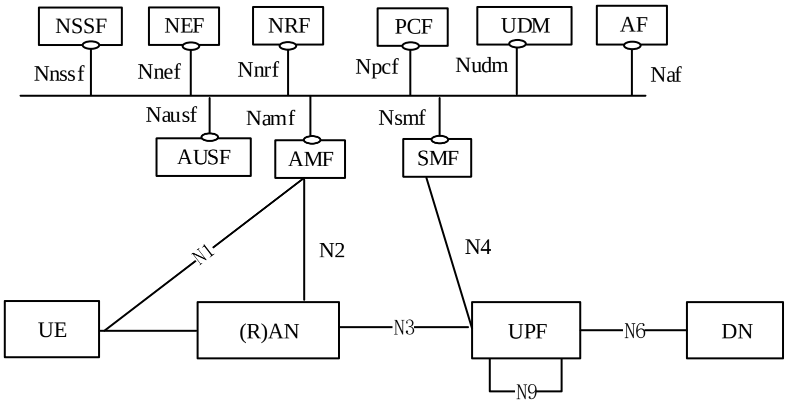


Figure 14: 5G Core, RAN and UE reference system architecture [3gpp]

B. Example procedure

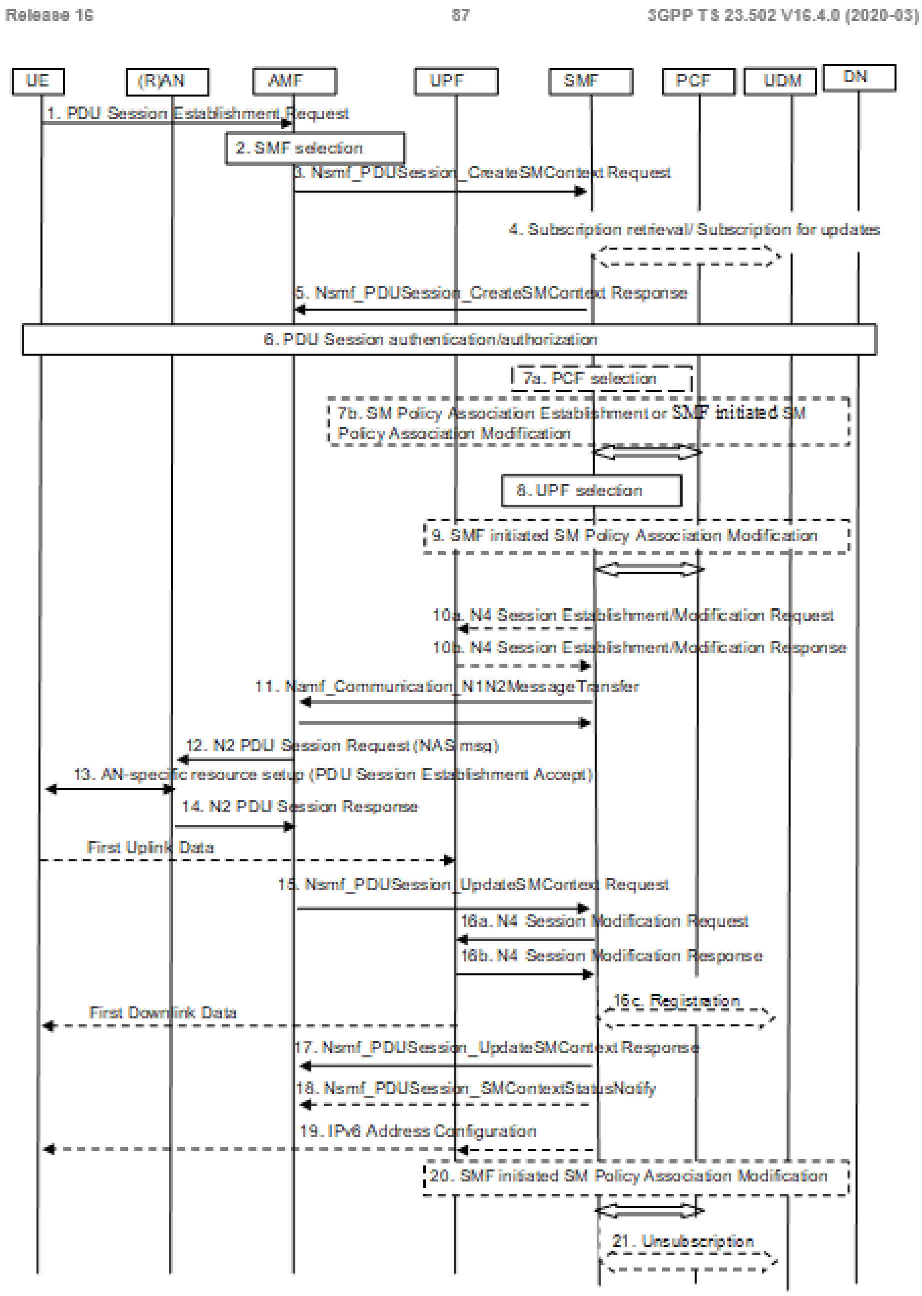


Figure 4.3.2.2.1-1: UE-requested PDU Session Establishment for non-roaming and roaming with local breakout

C. 5G Protocol stacks

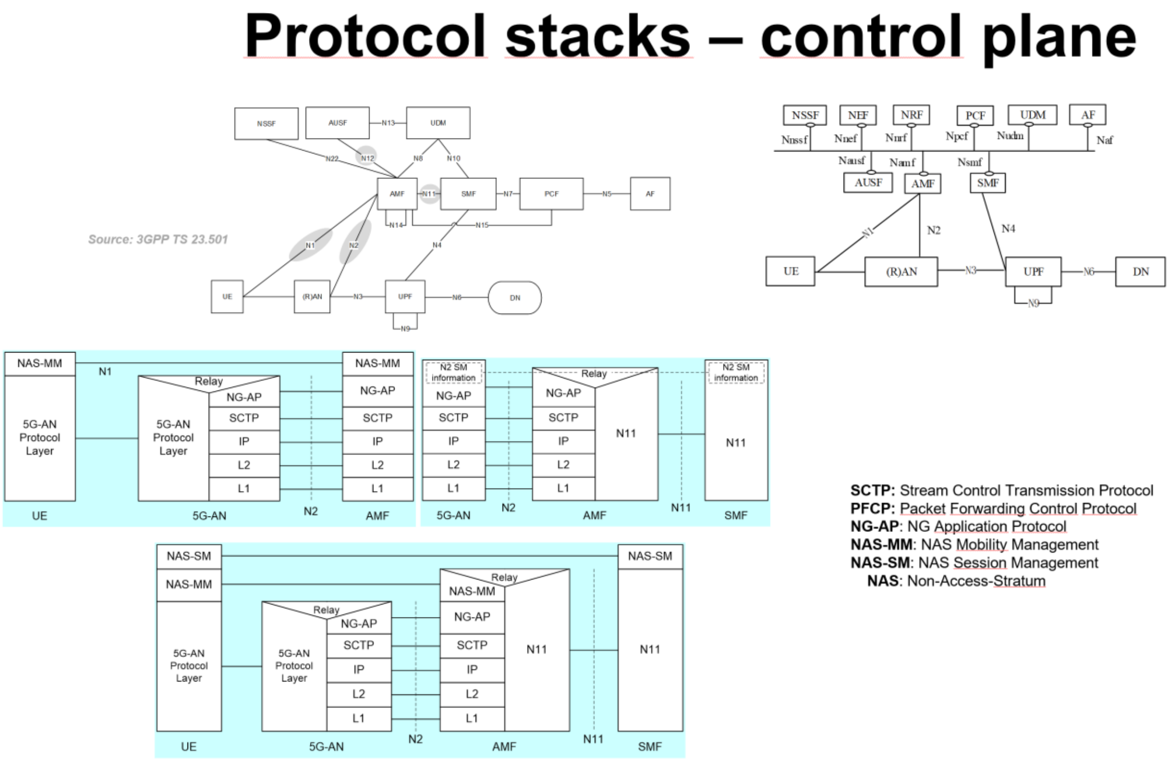


Figure 15: Protocol stacks, control plane

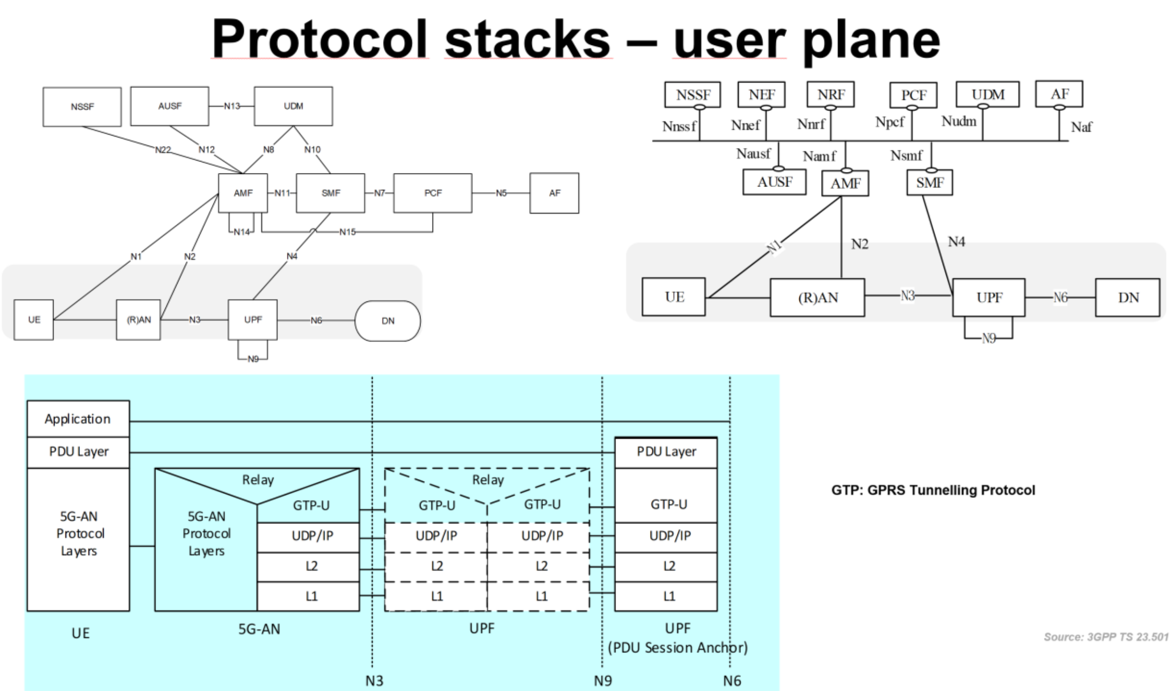


Figure 16: Protocol stacks, user plane

D. Free5GC New Subscriber creation form (UE11)

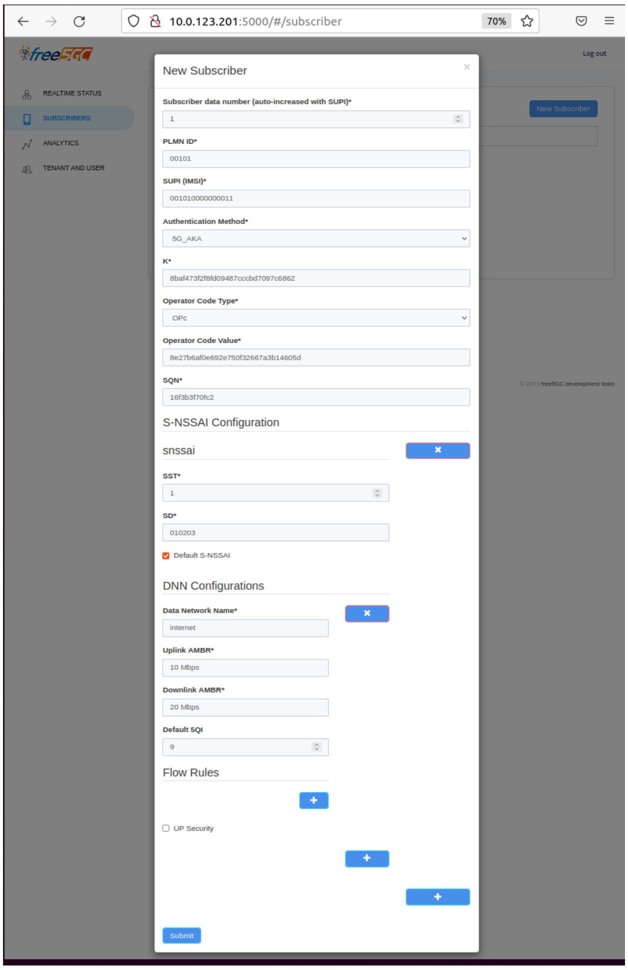


Figure 17: Example of a subscriber provisioning at Free5GC (UE11)

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

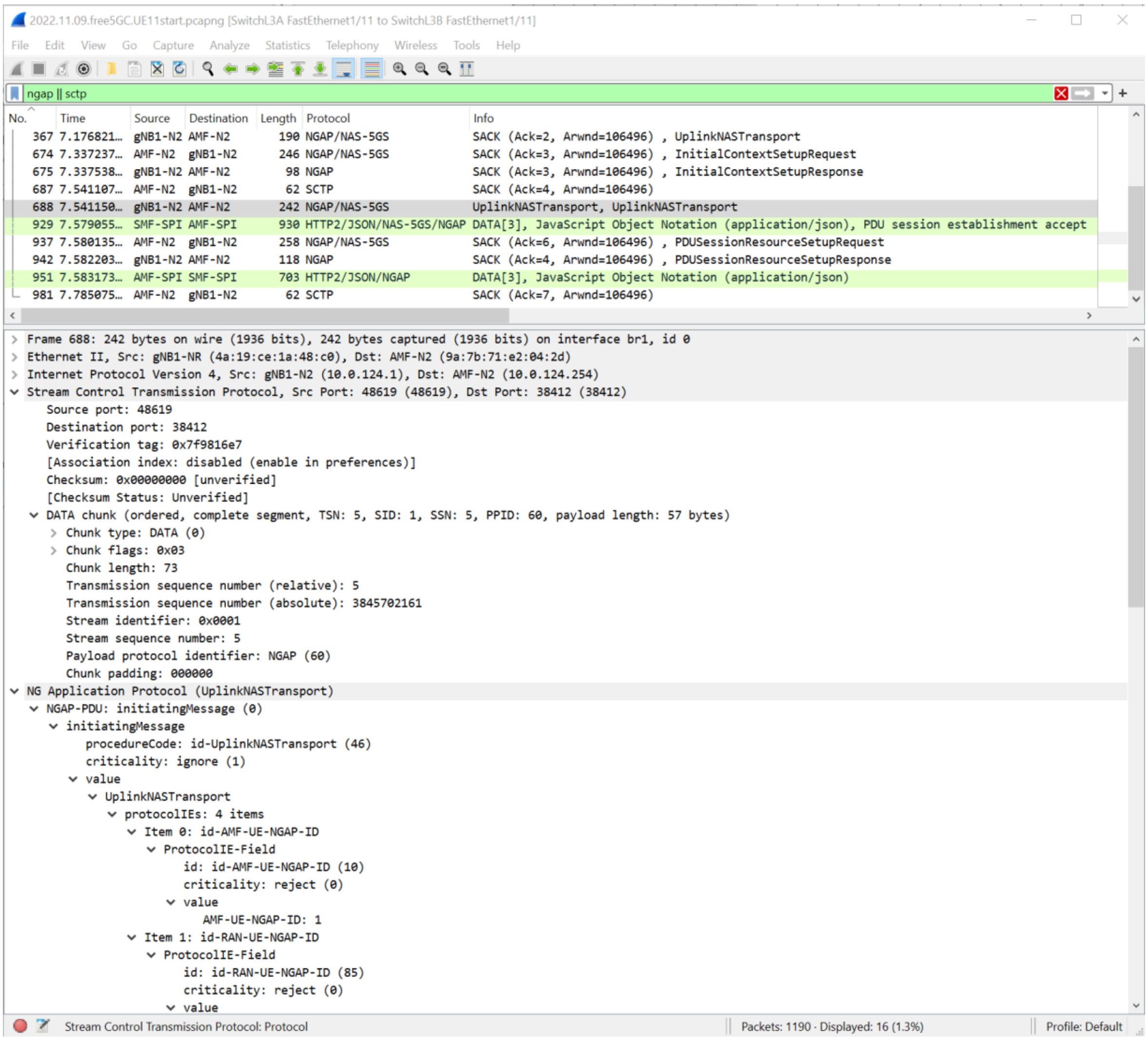


Figure 18: Example of a 5G exchange of messages (gNB – AMF interactions)

```
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: 7e02cad24c2a027e00680100432e0101c2110009010006313101ff090606001406000a…
                          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

[ ] Show packet bytes
Close   Help
```

Figure 19: 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