

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.

III. Used tools

This laboratory will use:

- An opensource implementation of a 5G Core: Free5GC [free5Gcore]
- A gNB and UE opensource implementation: UERANSIM [ueransim]
- A VirtualBox VM with both components already installed in the laboratory PCs
- Wireshark also installed in the laboratory PCs

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

IV. Network diagram

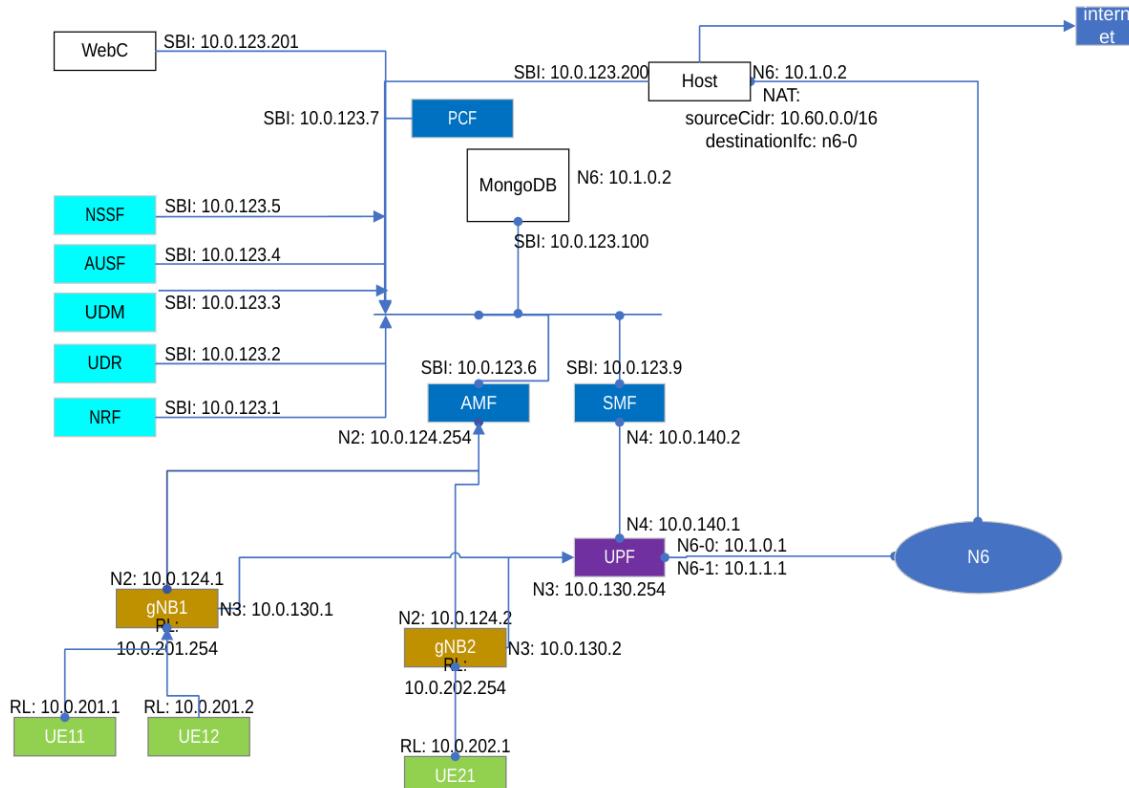


Figure 1: Network diagram

Notes:

1. 5G Core components are represented by dark and light blue boxes, gNBs by brown boxes and UEs by green boxes
2. With UERANSIM, the 5G-NR radio interface ('Radio Link') is emulated over UDP between the UEs (11, 12 and 21, green boxes) and the gNBs (gNB1 and gNB2) 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) (User Plane): Usado para o tráfego de usuário no 5G Core.
 - d. 10.0.140.0/24: N4 interfaces (control plane) Session Management Interface): Usado para a comunicação entre o (AMF e o Session Management Function (SMF) no 5G Core.
 - 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', emulated UEs can reach the Internet 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 and you can better interpret the messages exchange (see that file contents in Annex F at the end).
5. The shown MongoDB in the diagram component serves as persistent data repository for the other components while the network is running.

V. Procedure

During the laboratory execution (VirtualBox Virtual Machine), the 5G network components will be started and stop following this order:

1. 5G Network: 5G Core (2) → gNB1 (3.3) → gNB2 (3.5) →
2. UEs creation: → UE provisioning at the 5G Core (4.2) →
3. UEs start: → UE11 (4.4) → UE12 (5.5) → UE21 (5.7) →
4. Stop the system: → UEs, gNBs and 5G Core (8.1 and 8.3)

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 (shall be started with ‘sudo’) to capture traffic packets exchanged between any two NF, on their own interfaces (you will next figure when selecting capturing interfaces after 5G Core components have been started).

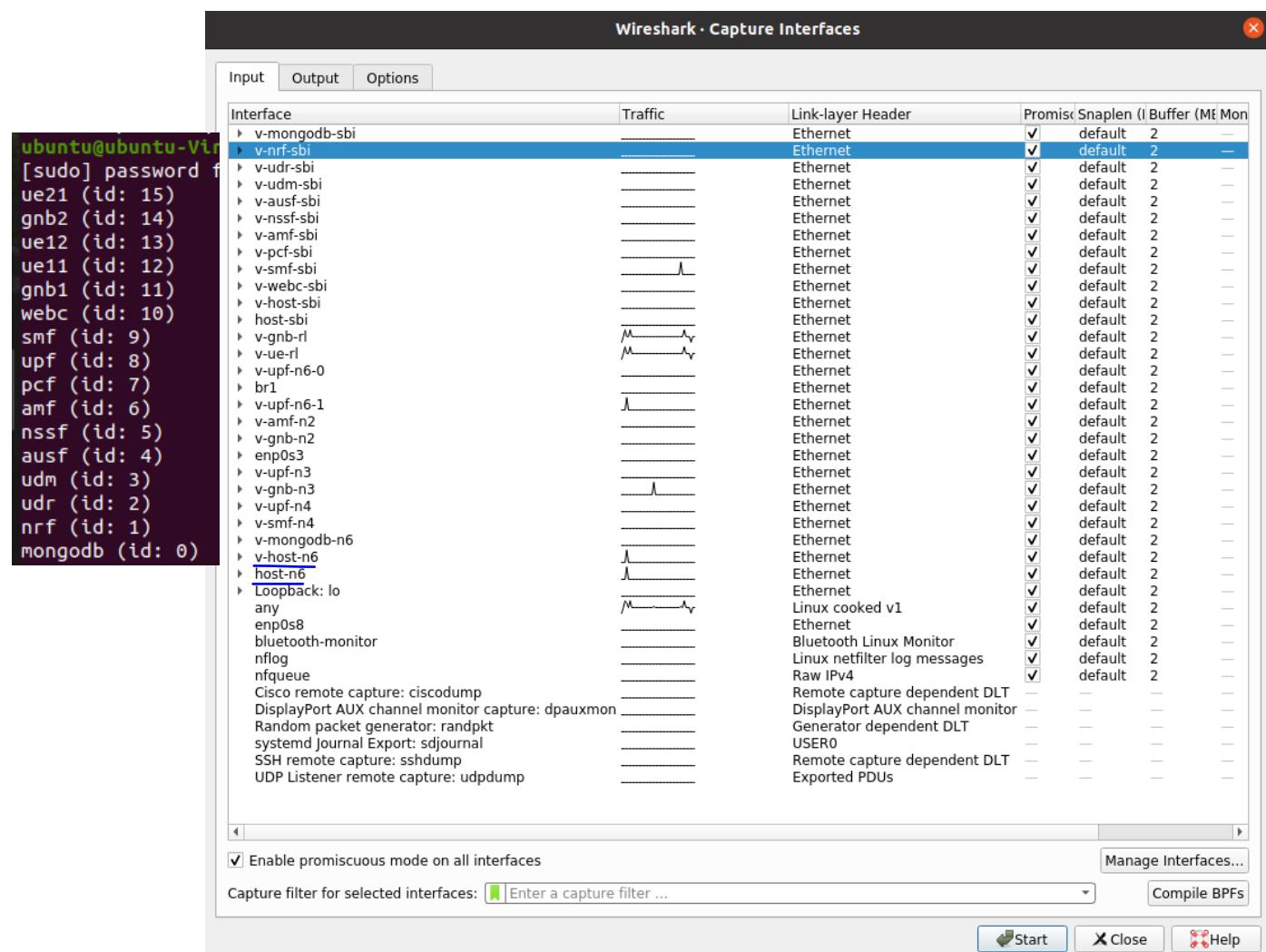


Figure 2: Logical interfaces as seen in Wireshark, after 5G Core start

AMF - Access and Mobility Management Function
 SMF - Session Management Function
 UPF - User Plane Function

A gNB (gNodeB) is a node in a cellular network that provides connectivity between user equipment (UE) and the evolved packet core (EPC).

1. 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`

b. **Parameters:**

- i. MCC and 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)`
 - 001 and ?

- ii. NR Cell Identities and TACs
 - `0x00000010, ? and?`

gNB1
`nci: '0x000000010' # NR Cell Identity (36-bit)`
`idLength: 32 # NR gNB ID length in bits [22...32]`
`tac: 1 # Tracking Area Code`

gNB2
`nci: '0x000000001' # NR Cell Identity (36-bit)`
`idLength: 32 # NR gNB ID length in bits [22...32]`
`tac: 2 # Tracking Area Code`

gNB1: slices:
 - sst: 0x1
 sd: 0x010203
 - sst: 0x2
 sd: 0x112233

gNB2: slices:
 - sst: 0x1
 sd: 0x010203

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

`Slice1: 0x1 and ?; Slice2: ? and 0x112233`

iv. Supported DNN

internet

`ue11: sessions:`
 - type: 'IPv4'
 slice:
 sst: 0x1

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

`imsi-001010000000011, imsi-001010000000012 and imsi-001010000000021`

`free5gc-ue11.yaml`
`supi: 'imsi-001010000000011'`

`free5gc-ue12.yaml`
`supi: 'imsi-001010000000012'`

`free5gc-ue21.yaml`
`supi: 'imsi-001010000000021'`

Subscription Permanent Identifier (SUPI):

5G Subscription Permanent Identifier is a globally unique identifier that is assigned to each subscriber in the 5G system, which is provisioned in the UDM/UDR. A SUPI is usually a string of 15 decimal digits.

2. 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***

(check its contents; it enables routing, creates nat and forwarding rules, stops firewalling, and calls another script that creates the ns and links)

- 4) Check created namespaces and connecting links

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

16 namespaces, numbered 0 to 15 one for each of the components shown in the figure above, except the 'Host'

Ver a figura de cima, nas interfaces de entrada

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

33 links are listed, numbered 0 to 57

Ver resultado em print 2 paginas abaixo

- 5) Start a Wireshark 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')

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

- 6) 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.

Order: **NRF, UDR, UDM, AUSF, NSSF, AMF, PCF, UPF, SMF, WebUI**

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

To register against the NRF (remember the NRF role)

Related the **order appear with the existing inter dependencies**.

- 7) 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?**

Protocols sao tds.

5G são os que têm N4, para ver:

Protocols: ARP, HTTP2, ICMPv6, MDNS, PFCP, TCP

ip.addr >= 10.0.140.0 and ip.addr <= 10.0.140.255

5G protocols: PFCP and, (indirectly, HTTP2) ?

- 8) Identify the dialogs for the 5G protocols (suggestion: apply a display filter to those protocols and check the involved entities)

HTTP2: NRF and all the others, as origin or destination

PFCP: SMF and UPF

No.	Time	Source	Destination	Protocol	Length	Info
69	19.06.042023	69.5.968482935	69.5.968482935	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1220	215.507899888	MF5-SBI	UDR-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1221	215.507899888	UDR-SBI	MF5-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1222	215.507899888	MF5-SBI	NSSF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1223	215.507899888	NSSF-SBI	AMF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1224	215.507899888	AMF-SBI	PCF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1225	215.507899888	PCF-SBI	AUSF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1226	215.507899888	AUSF-SBI	SMF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
1227	215.507899888	SMF-SBI	UPF-SBI	ARP	42	10.8.123.1 1s at 8e:f7:5e:7e:61:40
79	5.06848622	MongoDB	NRF-SBI	ARP	42	10.8.123.180 1s at 7a:ac:de:75:33:r3
299	19.06.042023	MongoDB	AMF-SBI	ARP	42	10.8.123.180 1s at 7a:ac:de:75:33:r3
461	19.484251981	MongoDB	PCF-SBI	ARP	42	10.8.123.180 1s at 7a:ac:de:75:33:r3
1228	215.507899888	AMF-SBI	PCF-SBI	ARP	42	10.8.123.180 1s at 7a:ac:de:75:33:r3
204	7.627378973	UDR-SBI	MongoDB-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1229	215.507899888	UDR-SBI	AMF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1230	215.507899888	AMF-SBI	NRF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1231	215.507899888	NRF-SBI	AMF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1232	215.507899888	AMF-SBI	PCF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1233	215.507899888	PCF-SBI	AUSF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1234	215.507899888	AUSF-SBI	SMF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1235	215.507899888	SMF-SBI	UPF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1236	215.507899888	UPF-SBI	NRF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1237	215.507899888	NRF-SBI	AMF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1238	215.507899888	AMF-SBI	PCF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1239	215.507899888	PCF-SBI	AUSF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1240	215.507899888	AUSF-SBI	SMF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
1241	215.507899888	SMF-SBI	UPF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0
514	23.758153838	UPF-SBI	NRF-SBI	ARP	42	10.8.123.18 1s at ce:c0:83:99:40:c0

- 8) Via the 'Host' emulated UEs can reach the Internet. A **hosts** file has been added to Wireshark.



HTTP2: NRF and all the others, as origin or destination

PFCP: SMF and UPF

Ferramentas Pratical Work SNMP x

3) Da parte 2

- 10.0.123.0/24: SBI; Core components, Web Console and DB (control plane)
- 10.0.124.0/24: N2 interfaces (control plane)
- 10.0.130.0/24: N3 interfaces (data plane)
- 10.0.140.0/24: N4 interfaces (control plane)
- 10.0.20[1|2].0/24: radio interfaces emulation
- 10.1.[1|2].0/24: N6 DNNs (data plane)

4) Via the 'Host' emulated UEs can reach the Internet. A **hosts** file has been added to 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
create_interface: if_local upf-sbi, if_bridge v-upf-sbi, namespace upf
configure_interface upf-sbi v-upf-sbi upf 10.1.0.1
disableIPv6IfRequested: interface upf-sbi, namespace upf
create_interface: if_local smf-sbi, if_bridge v-smf-sbi, namespace smf
configure_interface smf-sbi v-smf-sbi smf 10.0.123.9
disableIPv6IfRequested: interface smf-sbi, namespace smf
create_interface: if_local host-sbi, if_bridge v-host-sbi, namespace host
configure_interface host-sbi v-host-sbi host 10.0.123.200
```


4. UE creation, registration and default PDU creation

- 1) Open the Free5GC Web Console from the web browser:

- <http://10.0.123.201:5000>
- credentials: '**admin**'/'**free5gc**'

- 2) Create the 3 UEs from the table below ('**New Subscriber**', 

	UE11	UE12	UE21
PLMN ID (MCC/MNC)	00101	00101	00101
SUPI (IMSI)	001010000000011	001010000000012	001010000000021
SST/SD	1/010203 (sl1)	1/010203 (sl1) 2/112233 (sl2)	1/010203 (sl1)
data network name DNN	internet	internet	internet
UpLink/DownLink 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:

- Only change the parameters shown in the table and if required
 - do not change: Authentication method, K*, Operator Code Type, Operator Code Value*, and SQN*)
 - you may search and interpret the other parameters.
- In the Free5GC "New Subscriber" form, delete the second appearing S-NSSAI (*Single Network Slice Selection Assistance Information*) and the second DNN ('internet2')
- Restart Wireshark, keeping the capture in the same interface ('br1')
- Start the first UE (UE11)

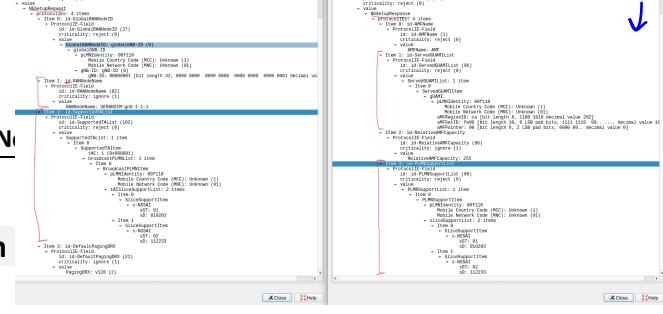
~5GLab/netns5G\$./UE11start.yaml

```
ubuntu@ubuntu-VirtualBox:~/5GLab/netns5g$ ./UE11start.sh
UERAN[IN 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] [rcc] [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-DREG-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] [info] Sending Initial Registration
[2022-11-07 22:24:37.667] [rcc] [info] UE switches to state [MM-REGISTER-INITIATED]
[2022-11-07 22:24:37.667] [rcc] [debug] Sending RRC Setup Request
[2022-11-07 22:24:37.668] [rcc] [info] RRC connection established
[2022-11-07 22:24:37.668] [rcc] [info] UE switches to state [RRC-CONNECTED]
[2022-11-07 22:24:37.669] [rcc] [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 switched 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[uesintun0, 10.60.0.2] is up.
```

- Observe the states the UE went through, during the process; observe the other messages and its sequence
Para ver tds os UEs: ip.addr == 10.0.201.1
- Observe the creation of the new TUN interface ('uesintun0'); 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$ 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
  inetc 0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1400 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inetc 10.60.0.1/32 scope global ue11
      valid_lft forever preferred_lft forever
    inetc fe80::9de4:a23b:4d2c:287f/64 scope link stable-privacy
      valid_lft forever preferred_lft forever
4: ue11: <POINTPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
  link/ether 86:f8:0d:be:b9:61 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.60.0.1/24 scope global ue11
      valid_lft forever preferred_lft forever
```



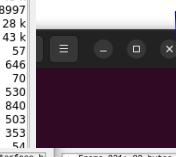
Subscribers	
PLMN	UE ID
00101	imsi-001010000000011
00101	imsi-001010000000012
00101	imsi-001010000000021

[Delete](#) [Modify](#)

Ethernet · 27 IPv4 · 24 IPv6 · 2 TCP · 55 UD

Address A	Address B	Packets	Bytes	packets A → B	Bytes A → B	packets B → A	Bytes B → A	Ret Stat	Duration	Bits/s A → B	Bits/s B → A
ubuntu-VirtualBox.local	mdns.mcast.net	1	87	1	87	0	0	189.925717	0.0000	—	—
ubuntu-VirtualBox.local	mdns.mcast.net	1	87	1	87	0	0	189.763908	0.0000	—	—
UPF-N4	SMF-N4	6	792	3	215	3	577	12.090356	0.3593	4786	1
NSFF-SBI	AMF-SBI	19	1824	9	825	10	999	12.066116	1.0382	6356	7
AMF-SBI	PCF-SBI	23	2559	11	1225	12	1334	11.698845	1.0432	9393	1
NRF-SBI	AUSF-SBI	24	11 k	12	10 k	12	1045	11.316909	1.0540	78 k	7
UDM-SBI	AUSF-SBI	66	6185	32	2843	34	3342	11.326885	1.0683	21 k	2
AUSF-SBI	AMF-SBI	46	4526	23	2400	23	2126	11.306162	1.0887	17 k	1
UDM-SBI	SMF-SBI	23	2696	12	1677	11	1019	12.124324	1.1011	12 k	7
PCF-SBI	SMF-SBI	70	29 k	36	26 k	34	2982	12.112184	1.12432	168 k	1
PCF-SBI	SMF-SBI	43	4495	21	2110	22	2385	12.088802	1.12593	13 k	1
UDM-SBI	AMF-SBI	108	10 k	55	4853	53	5399	11.416299	1.12850	30 k	3
AMF-SBI	SMF-SBI	119	13 k	58	6306	61	7390	12.086976	1.13679	36 k	4
UDR-SBI	PCF-SBI	40	3595	19	1741	21	1854	11.701158	1.14685	8448	8
NRF-SBI	AMF-SBI	146	59 k	74	52 k	72	6364	11.291994	1.18122	233 k	2
UDR-SBI	UDM-SBI	216	21 k	107	11 k	109	10 k	11.338279	1.18867	47 k	4
gNB1-N2	AMF-N2	290	31 k	144	15 k	146	15 k	8.942041	210.60399	57	—
UE11-NR	gNB1-NR	4,719	320 k	2,148	144 k	2,571	175 k	11.256889	2174.2784	533	—
AMF-SBI	WebConsole	580	38 k	290	19 k	290	19 k	1.773118	2174.9800	70	—
UDR-SBI	MongoDB-SBI	1,656	247 k	1,051	102 k	605	144 k	2.202573	2180.0181	376	—
NRF-SBI	MongoDB-SBI	1,648	330 k	1,046	101 k	602	229 k	0.137967	2180.0223	371	—
PCF-SBI	MongoDB-SBI	1,617	233 k	1,028	96 k	589	137 k	3.360486	2180.0331	354	—
MongoDB-SBI	WebConsole	1,620	233 k	591	137 k	1,029	96 k	0.000000	2180.3336	503	—
gNB2-N2	AMF-N2	282	29 k	141	14 k	141	1 k	0.000000	2181.1156	54	—

7931
25 k
15 k **Star Communications (5G)**



esta img esta igual em cim

- 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

PFCEP **SCTP** **SSL, SSLv2, TCP** **UDP**

- 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 pfcpc or sctp*")

 - 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
 - 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*')
 - 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*")

5. Connectivity

- 1) Start a Wireshark capture in the interface 'upf-n3' and another capture in the interface 'upf-n6-C'
 - 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



- 5) In a new Terminal Window/Tab, Start UE12
~/5GLab/netns5G\$./UE12start-sl1.yaml nao dá se correr ao mesmo tempo com o outro EU
(check the contents of file ./config/free5gc-ue12-sl1.yaml)
 - 6) Make a ping from UE11 to UE12

Para abrir o web console e editar os EU's tenho que ter sp tudo a correr senao n abre a pagina, n esquecer (é obvio) msm os nGMS

- 7) In a new Terminal Window/Tab, Start UE21
 ~ /5GLab/netns5G\$./UE21start-sl1.yaml
(check the contents of file ./config/free...

- 8) Make a ping from UE12 to UE21 and observe the exchanged packets at the UPF.

Digitized by srujanika@gmail.com

```

> Frame 1: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface v-upf-n3
> Ethernet II, Src: 86:85:81:da:cd:35 (86:85:81:da:cd:35), Dst: ae:23:3c:07:ac:36 (ae:23:3c:07:ac:36)
> Internet Protocol Version 4, Src: 10.0.130.1 (10.0.130.1), Dst: 10.0.130.254 (10.0.130.254)
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
  > Flags: 0x34
    Message Type: T-PDU (0xff)
    Length: 92
    TEID: 0x00000001 (1)
    Next extension header type: PDU Session container (0x85)
    < Extension header
      Extension Header Length: 1
      > PDU Session Container
        Next extension header type: No more extension headers (0x00)
    > Internet Protocol Version 4, Src: 10.60.0.1 (10.60.0.1), Dst: dns.google (8.8.8.8)
    > Internet Control Message Protocol

```

```

> Frame 2: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface v-upf-n3, id 0
> Ethernet II, Src: ae:23:3c:07:ac:36 (ae:23:3c:07:ac:36), Dst: 86:85:81:da:cd:35 (86:85:81:da:cd:35)
> Internet Protocol Version 4, Src: 10.0.130.254 (10.0.130.254), Dst: 10.0.130.1 (10.0.130.1)
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
  > Flags: 0x34
    Message Type: T-PDU (0xff)
    Length: 92
    TEID: 0x00000001 (1)
    Next extension header type: PDU Session container (0x85)
    < Extension header
      Extension Header Length: 1
      > PDU Session Container
        Next extension header type: No more extension headers (0x00)
    > Internet Protocol Version 4, Src: dns.google (8.8.8.8), Dst: 10.60.0.1 (10.60.0.1)
    > Internet Control Message Protocol

```

6. QoS

- 1) Open a new terminal window
- 2) Start an iperf3 server at the DNN
 $\$ \text{iperf3 -s}$
- ✓ 3) Check the TUN interface name and assigned IP address
 $\$ \text{sudo ip netns exec ue11 ip addr}$
- 4) Start an iperf3 client at UE11 towards the server instance and register the achieved bandwidth in the UL and DL directions -B, --bind <host> bind to the interface associated with the address <host>
 $\$ \text{sudo ip netns exec ue11 iperf3 -c 10.1.0.2 -B <ue11 IP address> -- uplink}$
 $\$ \text{sudo ip netns exec ue11 iperf3 -c 10.1.0.2 -R -B <ue11 IP address> -- downlink}$
 $-\text{R}, --\text{reverse}$ run in reverse mode (server sends, client receives)
- 5) Repeat previous measurements with the other two UEs (UE12 and UE21) and compare the results
 

```

Server listening on 5201
Accepted connection from 10.1.0.1, port 36551
[ 5] local 10.1.0.2 port 5201 connected to 10.1.0.1 port 56503
[ ID] Interval Transfer Bitrate
[ 5] 0.00-1.00 sec 2.23 MBytes 18.7 Mbites/sec
[ 5] 1.00-2.00 sec 1023 KBytes 8.39 Mbites/sec
[ 5] 2.00-3.00 sec 1.22 MBytes 10.3 Mbites/sec
[ 5] 3.00-4.00 sec 1010 KBytes 8.26 Mbites/sec
[ 5] 4.00-5.00 sec 1.33 MBytes 11.2 Mbites/sec
[ 5] 5.00-6.00 sec 1.07 MBytes 8.39 Mbites/sec
[ 5] 6.00-7.00 sec 1.26 MBytes 10.8 Mbites/sec
[ 5] 7.00-8.00 sec 1.03 MBytes 8.65 Mbites/sec
[ 5] 8.00-9.00 sec 1.12 MBytes 9.39 Mbites/sec
[ 5] 9.00-10.00 sec 7.90 KBytes 1.26 Mbites/sec
[ 5] 10.00-10.05 sec 12.4 MBytes 10.3 Mbites/sec
[ ID] Interval Transfer Bitrate
[ 5] 0.00-10.05 sec 12.4 MBytes 10.3 Mbites/sec

```

receiver

7. Slicing

- 1) Stop UE12 (Ctrl-C)
- 2) Restart UE12, now in the second slice (2/112233) with a new configuration file and check the results
 $\sim/5GLab/netns5G\$./UE12start-sl2.yaml$
 $(\text{check the contents of file } ./config/free5gc-ue12-sl2.yaml)$
- 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
 $\$ \text{sudo ip netns exec upf ip route}$
 - b. Add a new route in the UPF namespace
 $\$ \text{sudo ip netns exec upf ip route add 10.61.0.0./24 dev upfgtp}$
- 5) Repeat the ping above.

```
2023-11-12T03:28:30Z [INFO][LwM2M][CM][AM-UE_NGAP_ID:10][SMP]:[MSI-00101000000001] CreateS1ContextRequest Error: undefined response type
```

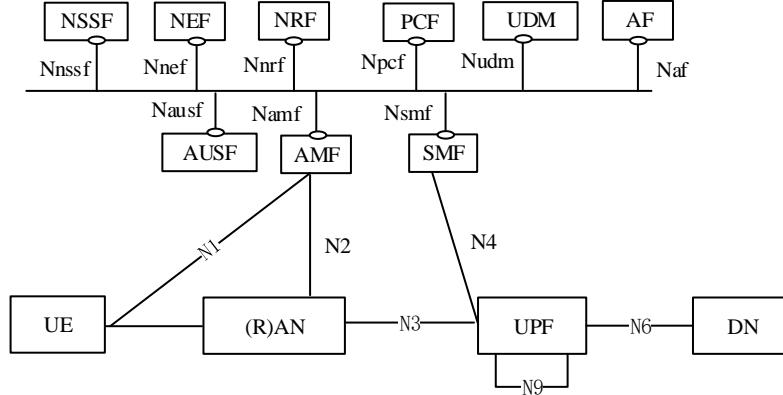
```
./UE12start-sl2.sh
Nao consegui correr este, deve ser no subscriber que fizer mal
```

8. Stop and reset the environment

- 1) Stop the 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
 $\sim/5GLab/netns5G\$ \text{sudo ./5Gcleanup.sh}$

Anexes

A. 5G System architecture



4 b)
parte 2

```

5: v-mongodb-sbi@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 7a:03:c6:b7:1c:3e brd ff:ff:ff:ff:ff:ff link-netns mongodb
7: v-nrf-sbi@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue mas
link/ether a2:b0:ce:ed:3b:f0 brd ff:ff:ff:ff:ff:ff link-netns nrf
9: v-udr-sbi@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ma
link/ether 26:54:9d:7b:a1:55 brd ff:ff:ff:ff:ff:ff link-netns udr
11: v-udm-sbi@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue m
link/ether 52:b1:c0:14:0f:e4 brd ff:ff:ff:ff:ff:ff link-netns udm
13: v-ausf-sbi@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 76:26:f8:c7:5f:3d brd ff:ff:ff:ff:ff:ff link-netns ausf
15: v-nssf-sbi@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 52:8d:32:34:99:4b brd ff:ff:ff:ff:ff:ff link-netns nssf
17: v-amf-sbi@if18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue m
link/ether 9e:50:ce:bd:5e:6c brd ff:ff:ff:ff:ff:ff link-netns amf
19: v-amf-n2@if20: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ma
link/ether ba:b2:11:7c:c5:f0 brd ff:ff:ff:ff:ff:ff link-netns amf
21: v-pcf-sbi@if22: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue m
link/ether 12:a3:f2:59:7d:62 brd ff:ff:ff:ff:ff:ff link-netns pcf
23: v-upf-n3@if24: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ma
link/ether 0a:19:c9:31:a2:3c brd ff:ff:ff:ff:ff:ff link-netns upf
25: v-upf-n4@if26: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ma
link/ether 26:b4:d0:07:f7 brd ff:ff:ff:ff:ff:ff link-netns upf
27: v-upf-n6_0@if28: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 32:ef:26:af:16:3b brd ff:ff:ff:ff:ff:ff link-netns upf
29: v-upf-n6_1@if30: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 0e:85:be:18:d9:d9 brd ff:ff:ff:ff:ff:ff link-netns upf
31: v-smf-sbi@if32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue m
link/ether 62:89:0e:bc:70:5d brd ff:ff:ff:ff:ff:ff link-netns smf
33: v-smf-n4@if34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ma
link/ether 36:c7:ae:ec:54:c1 brd ff:ff:ff:ff:ff:ff link-netns smf
35: v-webc-sbi@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether ea:28:02:eb:f5:d9 brd ff:ff:ff:ff:ff:ff link-netns webc
37: v-host-sbi@host-sbi: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqu
link/ether 3a:bd:6e:92:99:10 brd ff:ff:ff:ff:ff:ff
38: host-sbi@v-host-sbi: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqu
link/ether ae:1c:cc:45:d1:2d brd ff:ff:ff:ff:ff:ff
39: v-host-n6@host-n6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noque
link/ether 3e:57:d5:9c:8c:82 brd ff:ff:ff:ff:ff:ff
40: host-n6@v-host-n6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noque
link/ether 06:17:d7:d5:35:0f brd ff:ff:ff:ff:ff:ff
41: v-gnb1-r1@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 86:7f:6c:58:42:e3 brd ff:ff:ff:ff:ff:ff link-netns gnb1
43: v-gnb1-n2@if44: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether f6:d8:c8:9f:76:32 brd ff:ff:ff:ff:ff:ff link-netns gnb1
45: v-gnb1-n3@if46: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 32:c1:51:c3:b8:b1 brd ff:ff:ff:ff:ff:ff link-netns gnb1
47: v-ue11-r1@if48: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 2a:30:aa:b1:59:47 brd ff:ff:ff:ff:ff:ff link-netns ue11
49: v-ue12-r1@if50: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether a6:6a:1f:6f:10:63 brd ff:ff:ff:ff:ff:ff link-netns ue12
51: v-gnb2-r1@if52: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 26:d3:70:7b:ca:63 brd ff:ff:ff:ff:ff:ff link-netns gnb2
53: v-gnb2-n2@if54: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 1e:10:47:c3:b0:0b brd ff:ff:ff:ff:ff:ff link-netns gnb2
55: v-gnb2-n3@if56: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether 92:b8:12:4d:08:1c brd ff:ff:ff:ff:ff:ff link-netns gnb2
57: v-ue21-r1@if58: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
link/ether ba:75:f7:c8:7e:59 brd ff:ff:ff:ff:ff:ff link-netns ue21

```

B. Example procedure

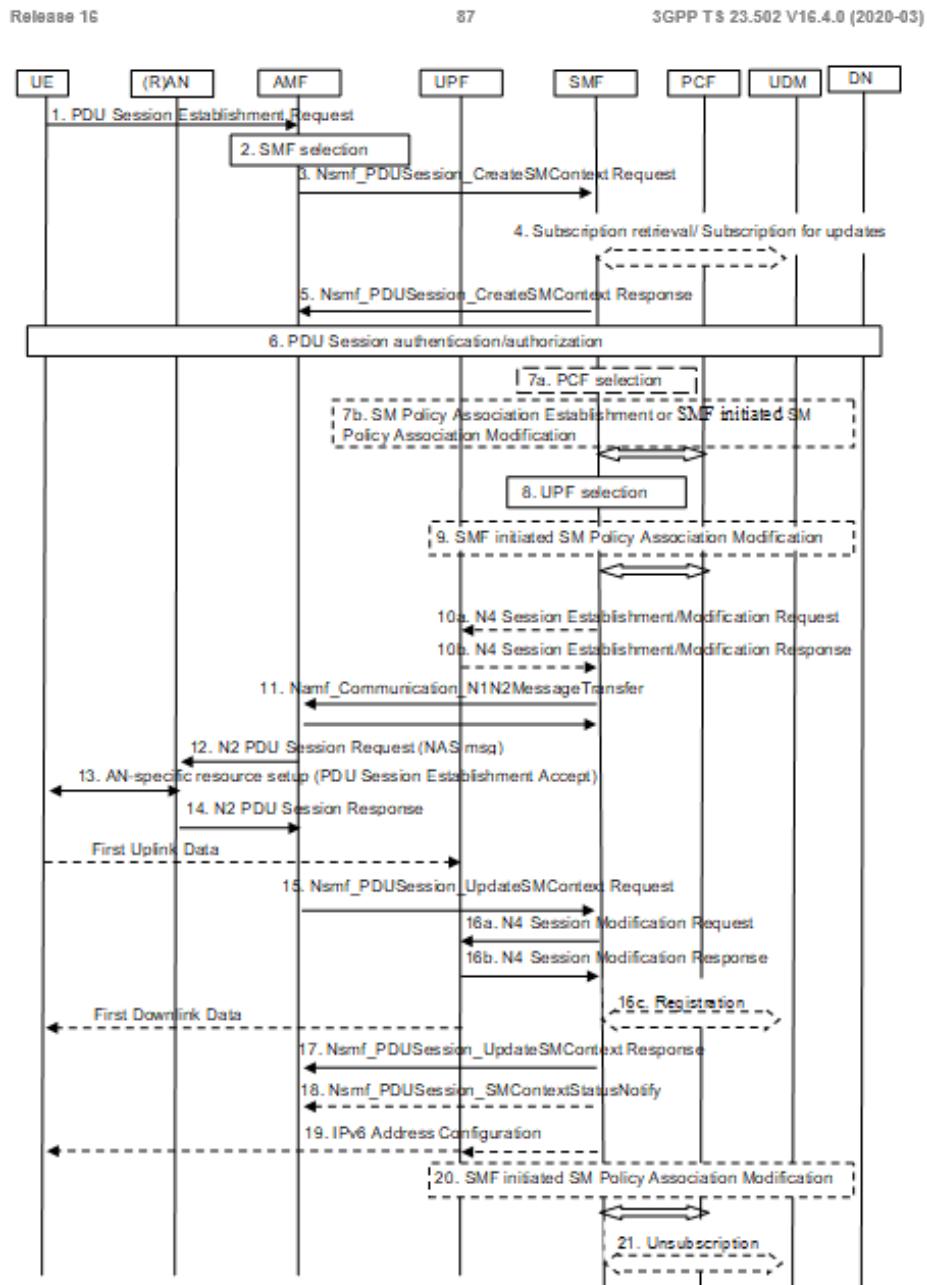
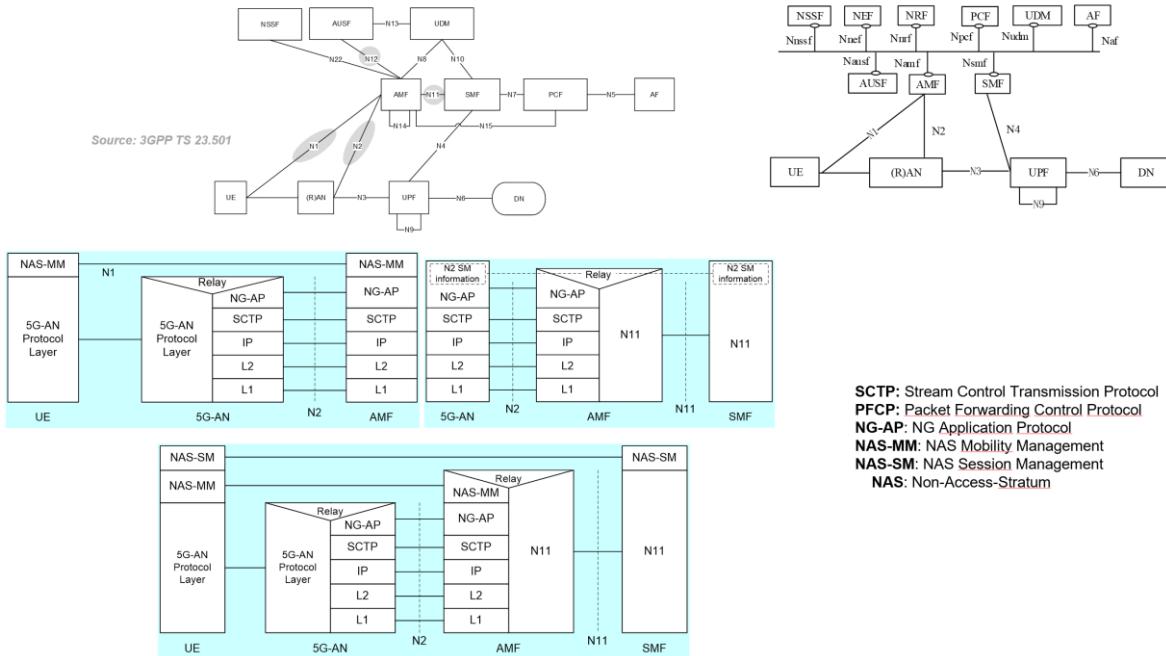


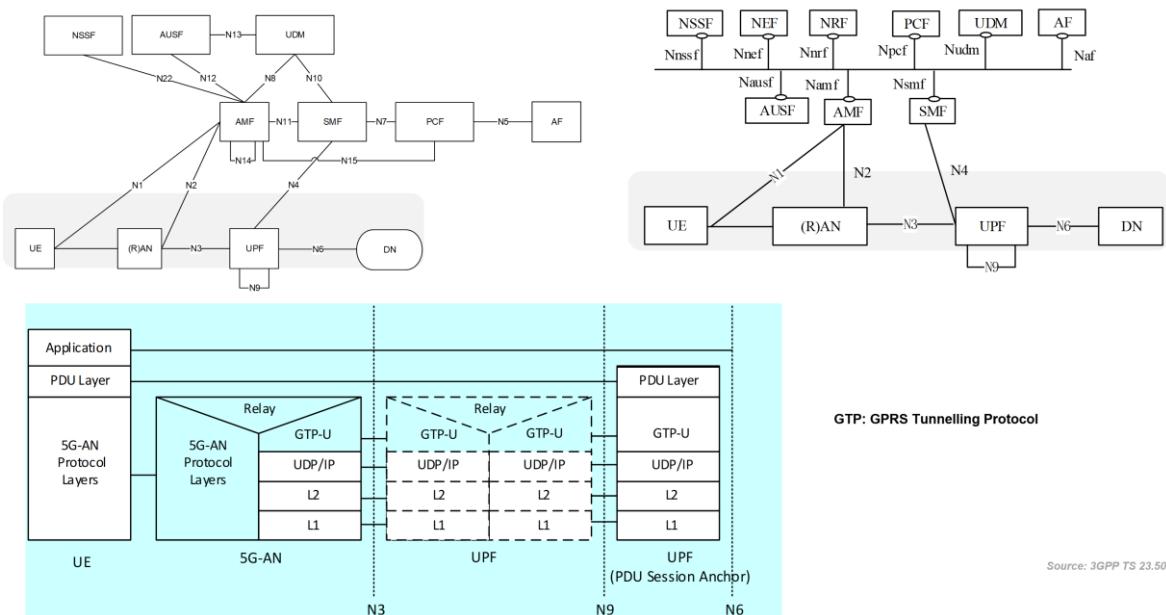
Figure 4.3.2.2.1-1: UE-requested PDU Session Establishment for non-roaming and roaming with local breakout

C. 5G Protocol stacks

Protocol stacks – control plane



Protocol stacks – user plane



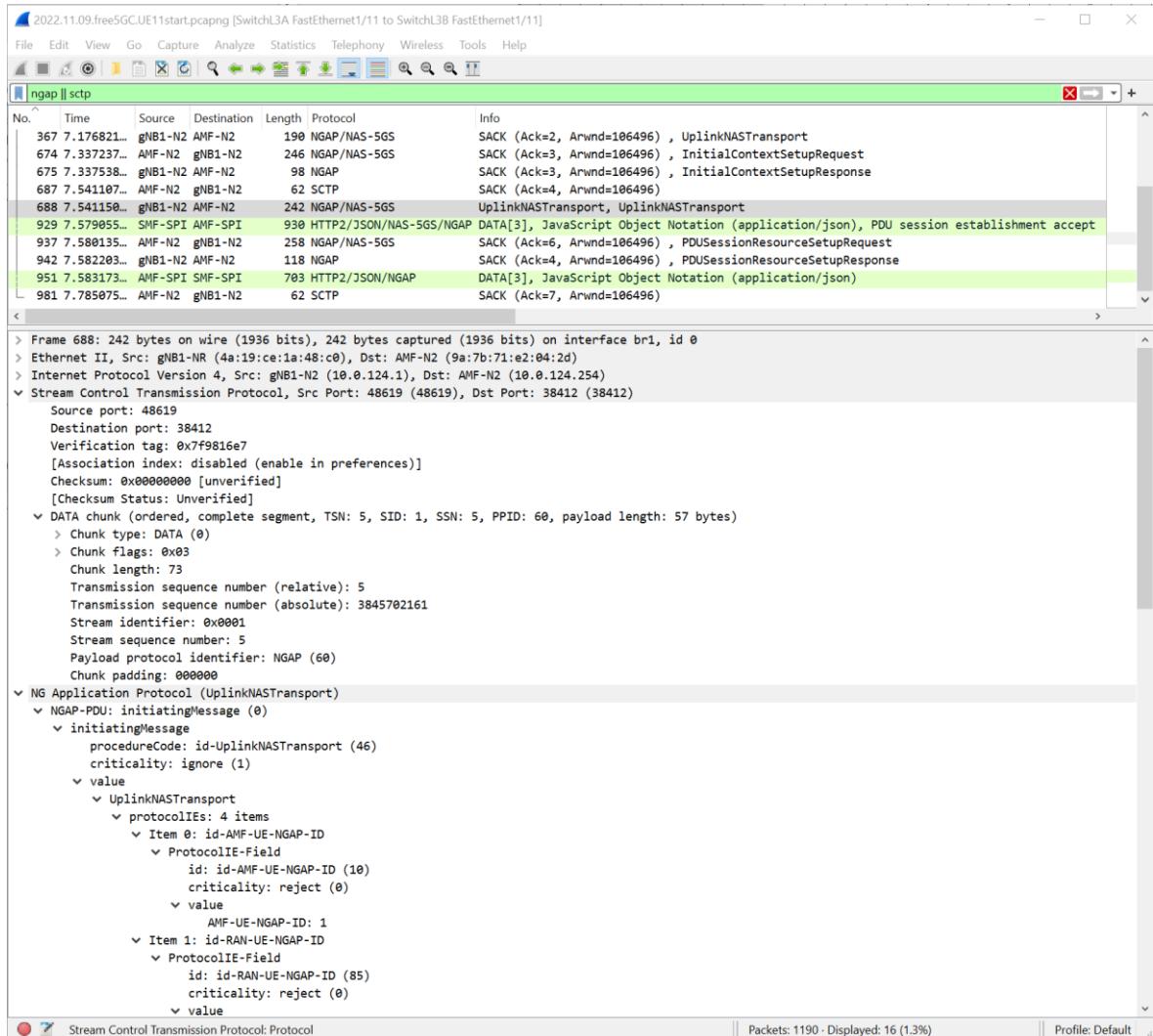
D. Free5GC New Subscriber creation form (UE11)

The screenshot shows the 'New Subscriber' creation form in the Free5GC web interface. The URL in the browser is 10.0.123.201:5000/#/subscriber. The form is divided into several sections:

- Subscriber data number (auto-increased with SUPI)***: Input field containing '1'.
- PLMN ID***: Input field containing '000101'.
- SIMI (MSISDN)***: Input field containing '2001010000000011'.
- Authentication Method***: Input field containing '5G_AKA'.
- K***: Input field containing '0baa473f2fb8d9e487cccd7097c6862'.
- Operator Code Type***: Input field containing 'Opc'.
- Operator Code Value***: Input field containing '9ec27b65a0e992e750f32667a3b14605d'.
- SQN***: Input field containing '1af5b370fc2'.
- S-NSSAI Configuration**:
 - Snssai**: Input field containing '1'.
 - SD***: Input field containing '010203'.
 - Default S-NSSAI
- DNN Configurations**:
 - Data Network Name***: Input field containing 'Internet'.
 - Uplink AMBR***: Input field containing '10 Mbps'.
 - Downlink AMBR***: Input field containing '20 Mbps'.
 - Default SQL**: Input field containing '9'.
- Flow Rules**: A section with three '+' buttons for adding new flow rules.

At the bottom right of the form is a 'Submit' button.

E. Example of capture with Wireshark, with addresses resolution and display filter



```

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:d), Dst: gNB1-NR (4a:19:ce:1a:48:c8)
> 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)
└ NG Application Protocol (PDUSESSIONRESOURCESETUPREQUEST)
  └ NGAP-PDU: initiatingMessage (0)
    └ initiatingMessage
      procedureCode: id-PDUSessionResourceSetup (29)
      criticality: reject (0)
    └ value
      └ PDUSESSIONRESOURCESETUPREQUEST
        └ protocolIEs: 4 items
          └ Item 0: id-AMF-UE-NGAP-ID
            └ ProtocolIE-Field
              id: id-AMF-UE-NGAP-ID (10)
              criticality: reject (0)
            └ value
              ANF-UE-NGAP-ID: 1
          └ Item 1: id-RAN-UE-NGAP-ID
            └ ProtocolIE-Field
              id: id-RAN-UE-NGAP-ID (85)
              criticality: reject (0)
            └ value
              RAN-UE-NGAP-ID: 1
          └ Item 2: id-PDUSessionResourceSetupListSUReq
            └ ProtocolIE-Field
              id: id-PDUSessionResourceSetupListSUReq (74)
              criticality: reject (0)
            └ value
              └ PDUSessionResourceSetupListSUReq: 1 item
                └ Item 0
                  └ PDUSESSIONRESOURCESETUPITEMSUReq
                    pDUSessionID: 1
                  └ pdUSessionNAS-PDU: 7e02cad24c2a027e00680100432e0101c211000901000631310101ff090606001406000a...
                    └ Non-Access-Stratum SGS (NAS)PDU
                      └ Security protected NAS SGS 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: 0xcd24c2a
                        Sequence number: 2
                        Encrypted data
                    └ s-NSSAI
                      sST: 01
                      sD: 010203
                    └ pdUSessionResourceSetupRequestTransfer: 000004080200090c01312d0020989680008b000a01f00a0082fe00000001008600010000...
                    └ protocolIEs: 4 items
                      └ Item 0: id-PDUSessionAggregateMaximumBitRate
                        └ ProtocolIE-Field
                          id: id-PDUSessionAggregateMaximumBitRate (130)
                          criticality: reject (0)
                        └ value
                          └ PDUSESSIONAGGREGATEMAXIMUMBITRATE
                            pDUSESSIONAggregateMaximumBitRateDL: 20000000bits/s
                            pDUSESSIONAggregateMaximumBitRateUL: 10000000bits/s
                    └ Item 1: id-UL-NGU-UP-TNLInformation
                      └ ProtocolIE-Field
                        id: id-UL-NGU-UP-TNLInformation (139)
                        criticality: reject (0)
                      └ value
                        └ UPTunnelInformation: gTPtunnel (0)
                          └ gTPtunnel
                            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
                    └ Item 2: id-PDUSessionType
                      └ ProtocolIE-Field
                        id: id-PDUSessionType (134)
                        criticality: reject (0)
                      └ value
                        PDUSESSIONTYPE: ipv4 (0)
                    └ Item 3: id-QosflowSetupRequestList
                      └ ProtocolIE-Field
                        id: id-QosflowSetupRequestlist (136)
                        criticality: reject (0)
                      └ value
                        └ QosFlowSetupRequestList: 1 item
                          └ Item 0
                            └ QosFlowSetupRequestItem
                              qosFlowIdentifier: 9
                              qosFlowLevelQosParameters
                                └ qosCharacteristics: nonDynamic5QI (0)
                                  nonDynamic5QI
                                    fiveQI: 9
                                allocationAndRetentionPriority
                                  priorityLevelARP: 15
                                  pre-emptionCapability: shall-not-trigger-pre-emption (0)
                                  pre-emptionVulnerability: not-pre-emptable (0)
                        └ Item 3: id-UEAggregateMaximumBitRate
                          └ ProtocolIE-Field
                            id: id-UEAggregateMaximumBitRate (110)
                            criticality: ignore (1)
                          └ value
                            └ UEAggregateMaximumBitRate
                              ueAggregateMaximumBitRateDL: 200000000bits/s
                              ueAggregateMaximumBitRateUL: 100000000bits/s
  
```

Show packet bytes

[Close](#) [Help](#)

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/>
 - [free5g/wiki] <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>