OAI 5G-NR noS1 mode [installation, execution and testing instructions]

In order to be able to demonstrate IP traffic over 5g-NR while the 5g RAN stack is still under development, we have integrated a special implementation of the **noS1 mode** (previously used for the LTE softmodems as well) in the current version of OAI gNB and nr-UE softmodems. In this mode, there is no real connection of the gNB to any core network over the S1 interface and the exchanges that would normally take place between those two entities during the UE attachment to the network are bypassed/emulated. In the current phase of NR integration, the noS1 mode uses some of the functionality of the LTE RAN stack *(MAC, RLC, PDCP layers)* on top of *NR-PHY layer* implementation. Thus, by performing static preconfiguration for the data plane (i.e., data radio bearer, IP address configuration etc.) we can demonstrate IP downlink traffic flow between the gNB and UE entities.

In the following, we provide some installation, execution and testing instructions for the current noS1 mode.

1. **Configuration for real time operation**

In order to prepare your system's kernel for real time operation, you should follow the steps described in the following guide in order to install a low latency kernel and disable the power management settings and CPU frequency scaling.

https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirKernelMainSetup

2. **Getting the code**

```
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git

git checkout nr-ip-over-lte-v.1.5
```

3. **Building**

*gNB:

```
cd openairinterface5g/cmake_targets

./build_oai --gNB -w USRP -c
```

*nr-UE:

```
cd openairinterface5g/cmake_targets

./build_oai --nrUE -w USRP -c
```

4. Configure and launch the executables

*gNB:

Edit the following highlighted line at the gNB configuration file (*targets/PROJECTS/GENERIC-LTE-EPC/CONF/gnb.band78.tm1.106PRB.usrpn300.conf*) to indicate whether the *internal* clock reference of the USRP, or an *external* clock reference is going to be used.

```
RUs = (

{

local_rf       = "yes"

nb_tx          = 1

nb_rx          = 1

att_tx         = 0

att_rx         = 0;

bands          = [7];

max_pdschReferenceSignalPower = -27;

max_rxgain                    = 114;

eNB_instances  = [0];

sdr_addrs=
"addr=192.168.10.2,second_addr=192.168.20.2,mgmt_addr=192.168.10.2,clock_source=internal,time_source=internal";

}

);
```

Launch the gNB executable from *cmake_targets/ran_build/build* as follows:

```
sudo ./nr-softmodem -O ../../../targets/PROJECTS/GENERIC-LTE
EPC/CONF/gnb.band78.tm1.106PRB.usrpn300.conf --noS1 --nokrnmod 1
```

*nrUE:

Launch the nr-UE executable as follows depending on whether you use internal or external clock

```
#Internal clock

sudo ./nr-uesoftmodem --numerology 1 -r 106 -C 3510000000 --usrp-args
"addr=192.168.30.2,second_addr=192.168.50.2,clock_source=internal,time_source=int
ernal" --ue-fo-compensation --noS1 --nokrnmod 1

#External clock

sudo ./nr-uesoftmodem --numerology 1 -r 106 -C 3510000000 --usrp-args
"addr=192.168.30.2,second_addr=192.168.50.2,clock_source=external,time_source=ext
ernal" --ue-timing-correction-disable --noS1 --nokrnmod 1
```

5. **Test with downlink IP traffic**

*Ping

- From gNB using the dedicated oai tun IP interface:

```
ping -I oaitun_enb1 10.0.1.2 -i 0.2
```

-Validate reception at the nr-UE using wireshark and monitor oaitun_ue1 interface

*iperf

Server (nrUE): `iperf -s -i 1 -u -B 10.0.1.2`

Client (gNB): `iperf -c 10.0.1.2 -u -b 1M --bind 10.0.1.1 -t 100`

6. **Launch the executables with the T-tracer**

In order to launch the *textlog* tracer and export some timestamped traces, you first go to *common/utils/T/tracer*.

```
#Compile by running

make

#Launch the tracer (the groups of enabled logs can be selected either from the
#command line or from the event-selector GUI that pops-up). E.g:
```

```
./textlog -d ../T_messages.txt  -on LEGACY -off LEGACY_GROUP_TRACE -off
LEGACY_GROUP_DEBUG -on ENB_PHY_DLSCH_UE_DCI -on ENB_PHY_ULSCH_UE_DCI |tee oai.log

#If you want to deactivate the GUI menu add the –no-gui option to the above
#command.
```

A complete list of the different log groups can be found in *T_messages.txt* file

Then you should add the option `--T_stdout 0` to the execution command of any of the two softmodems (gNB and/or UE) where you want to activate the tracer.

For more information on the T-tracer, you can consult the following:

https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/T