

# Open-Source Prototyping of 5G Wireless Systems for UGV/UAV

OPERATION MANUAL

**Team:** 36

**Client:** Iowa State University

**Advisor:** Hongwei Zhang

**Team Members:** Nathan Whitcome, Ibro Tusic, Andrew Eschweiler, Samuel Stanek, William Byers, Nicholas Lorenz

**Email:** [sdmay20-36@iastate.edu](mailto:sdmay20-36@iastate.edu)

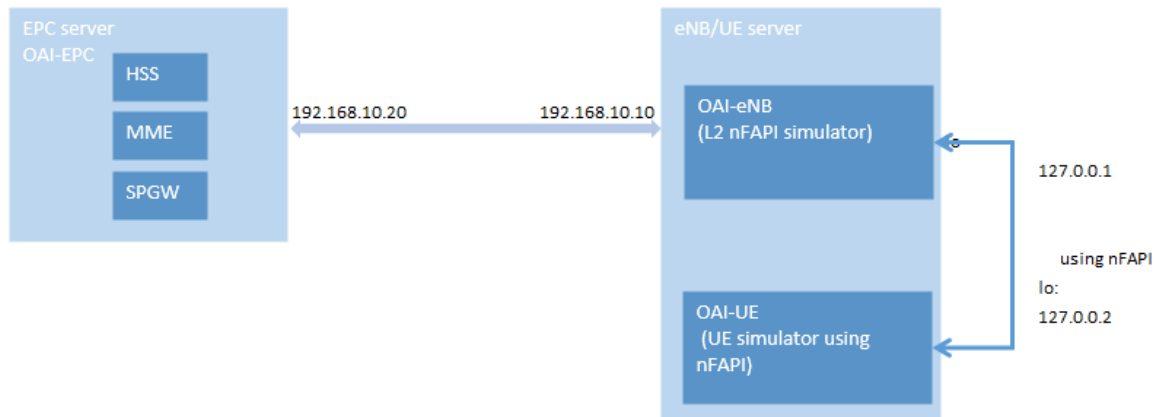
**Website:** <http://sdmay20-36.sd.ece.iastate.edu/>

**Revised:** 4/25/2020

## Operating System Setup (UE and ENB)

1. Install Ubuntu v16.04 on a system running an Intel processor
2. Open the terminal and run the following command:
  - a. `sudo apt-get install linux-image-lowlatency linux-headers-lowlatency`
3. Remove all power management features in the BIOS
  - a. Disable hyperthreading
  - b. Disable C-states
  - c. Disable CPU frequency scaling (Intel Speed step)
    - i. CPU Frequency scaling can be verified with the following command:
      1. `watch grep \"cpu MHz\" /proc/cpuinfo`
      2. The CPU frequency should not change
  - d. Add the below lines to `/etc/default/grub`
    - i. `GRUB_CMDLINE_LINUX_DEFAULT=\"quiet intel_pstate=disable\"`
    - ii. `processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll`
    - iii. Then run: `update-grub`
  - e. Add the below line to `/etc/modprobe.d/blacklist.conf`
    - i. `blacklist intel_powerclamp`
    - ii. if the file doesn't exist, create it and add the above line.
4. To verify setup, install `i7z`
  - a. `Sudo apt-get install i7z`
  - b. Run with: `sudo i7z`
  - c. Verify the CPU does not change its frequency by more than 1-2 hertz and the cores are not in any other C-state other than 0. A sample print out is below.





2. Follow the guide in the section for EPC to setup the EPC.
3. Pull the code for the eNB and UE from gitlab

SSH into the machine that will run the eNB and UE

Run the below commands to set up the UE and eNB folders:

```
$ git clone https://gitlab.eurecom.fr/oai/openairinterface5g/ enb_folder
$ cd enb_folder
$ git checkout -f v1.0.0
$ cd ..
$ cp -Rf enb_folder ue_folder
```

4. Configure the USIM information in the ue\_folder created above

```
$ cd ue_folder
# Edit openair3/NAS/TOOLS/ue_eurecom_test_sfr.conf with your preferred editor
```

This part is **VERY IMPORTANT!** An incorrect configuration here will mean that the UE can not communicate with the EPC!

Edit the USIM info in the file under bullet 4 above and ensure that the following configurations on the UE and the HSS database on the EPC match completely!

PLMN + MSIN and IMSI of users table needs to match!

OPC of the file above and OPC of the user table in EPC must match!

USIM\_API\_K of this file and the key of users table of HSS database must match!

Below is a sample configuration. To add more UEs, just include another top level UE with another configuration in JSON format.

```

#in openair3/NAS/TOOLS/ue_eurecom_test_sfr.conf
UE0:
{
  USER: {
    IMEI="356113022094149";
    MANUFACTURER="EURECOM";
    MODEL="LTE Android PC";
    PIN="0000";
  };

  SIM: {
    MSIN="0000000001"; // <-- Modify here
    USIM_API_K="8baf473f2f8fd09487cccbd7097c6862";
    OPC="e734f8734007d6c5ce7a0508809e7e9c";
    MSISDN="33611123456";
  };
  ...
};
//YOU DON'T NEED THIS, JUST AN EXAMPLE FOR MULTI-UE
UE1: // <- Edit here
{
  USER: {
    IMEI="356113022094149";
    MANUFACTURER="EURECOM";
    MODEL="LTE Android PC";
    PIN="0000";
  };

  SIM: {
    MSIN="0000000002"; // <-- Modify here
    USIM_API_K="8baf473f2f8fd09487cccbd7097c6862";
    OPC="e734f8734007d6c5ce7a0508809e7e9c";
    MSISDN="33611123456";
  };
  ...
};

```

5. The nFAPI configuration files, we need to configure the loopback interfaces to enable communication between the components.

Enter the enb\_folder we created above.

Edit: ci-scripts/conf\_files/rcc.band7.tmi.nfapi.conf

Verify the nFAPI interface on the 2<sup>nd</sup> loopback interface:

```

MACRLCS = (
  {
    num_cc = 1;
    local_s_if_name = "lo:"; // <-- HERE
    remote_s_address = "127.0.0.1"; // <-- HERE
    local_s_address = "127.0.0.2"; // <-- HERE
    local_s_portc = 50001;
    remote_s_portc = 50000;
    local_s_portd = 50011;
    remote_s_portd = 50010;
    tr_s_preference = "nfapi";
    tr_n_preference = "local_RRC";
  }
);

```

## Set the S1 interface:

```
////////// MME parameters:
mme_ip_address      = ( { ipv4      = "CI_MME_IP_ADDR"; // replace with 192.168.10.20
                        ipv6      = "192:168:30::17";
                        active    = "yes";
                        preference = "ipv4";
                      }
                    );

NETWORK_INTERFACES :
{
  ENB_INTERFACE_NAME_FOR_S1_MME      = "ens3";           // replace with the proper interface name
  ENB_IPV4_ADDRESS_FOR_S1_MME       = "CI_ENB_IP_ADDR"; // replace with 192.168.10.10
  ENB_INTERFACE_NAME_FOR_S1U        = "ens3";           // replace with the proper interface name
  ENB_IPV4_ADDRESS_FOR_S1U          = "CI_ENB_IP_ADDR"; // replace with 192.168.10.10
  ENB_PORT_FOR_S1U                  = 2152; # Spec 2152
  ENB_IPV4_ADDRESS_FOR_X2C          = "CI_ENB_IP_ADDR"; // replace with 192.168.10.10
  ENB_PORT_FOR_X2C                  = 36422; # Spec 36422
}
```

## 6. Configure the UE

Enter the ue\_folder we created above

Edit ci-scripts/conf\_files/ue.nfapi.conf with your preferred editor

Verify the nFAPI interface setup on the loopback interface

```
L1s = (
  {
    num_cc = 1;
    tr_n_preference = "nfapi";
    local_n_if_name = "lo";           // <- HERE
    remote_n_address = "127.0.0.2"; // <- HERE
    local_n_address = "127.0.0.1"; // <- HERE
    local_n_portc = 50000;
    remote_n_portc = 50001;
    local_n_portd = 50010;
    remote_n_portd = 50011;
  }
);
```

## 7. Configure a second loopback interface to connect the eNB and the UE

Ensure you are on the machine with the ue\_folder and enb\_folder

Run this command: `sudo ifconfig lo: 127.0.0.2 netmask 255.0.0.0 up`

## 8. Build the eNB

```
$ ssh sudousername@machineB
$ cd enb_folder
$ source oaienv
$ cd cmake_targets
# If you test less than 16 UEs, type below command.
$ ./build_oai --eNB -t ETHERNET -c
# If you test more than 16 UEs, type below command and this command also can be used in case of less than 16
$ ./build_oai --eNB -t ETHERNET -c --mu
```

## 9. Build the UE

Open a second terminal:

```
$ ssh sudousername@machineB
$ cd ue_folder
$ source oaienv
$ cd cmake_targets
# If you test less than 16 UEs, type below command.
$ ./build_oai --UE -t ETHERNET -c
# If you test more than 16 UEs, type below command and this command also can be used in case of less than 16 UEs.
$ ./build_oai --UE -x -t ETHERNET -c --musim
```

After finishing building UE(s), some files are generated in ue\_folder/targets/bin/ and these files are necessary in cmake\_targets.

```
$ ssh sudousername@machineB
$ cd ue_folder/targets/bin/
$ cp .u* ../../cmake_targets/
$ cp usim ../../cmake_targets/
$ cp nvram ../../cmake_targets/
```

## 10. Initialize the NAS UE layer

On the same machine at the ue\_folder and enb\_folder, run the following:

In the 2nd terminal:

```
$ ssh sudousername@machineB
$ cd ue_folder/cmake_targets/tools
$ source init_nas_s1 UE
```

## 11. Start the EPC as directed in the EPC instructions.

## 12. Start the eNB

```
$ cd enb_folder/cmake_targets
$ sudo -E ./lte_build_oai/build/lte-softmodem -O ../ci-scripts/conf_files/rcc.band7.tm1.nfapi.conf > enb.log 2>&1
```

### 13. Start the UE

In the ue\_folder, run the following. Change the -num-ues to 1 to just use one UE!

```
$ cd ue_folder/cmake_targets
```

```
# Test 64 UEs, 64 threads in FDD mode
```

```
$ sudo -E ./lte_build_oai/build/lte-uesoftmodem -O ../ci-scripts/conf_files/ue.nfapi.conf -L2-emul 3 --num-ues 64 > ue.log 2>&1
```

```
# Test 64 UEs, 64 threads in TDD mode
```

```
$ sudo -E ./lte_build_oai/build/lte-uesoftmodem -O ../ci-scripts/conf_files/ue.nfapi.conf -L2-emul 3 --num-ues 64 -T 1 > ue.log 2>&1
```

```
# The "-T 1" option means TDD config
```

### 14. Verify the configuration works by testing with ping

SSH into the machine housing the EPC

```
# Ping UE0 IP address based on the EPC pool used: in this example:
```

```
$ ping -c 20 192.168.200.2
```

```
# Ping UE1 IP address based on the EPC pool used: in this example:
```

```
$ ping -c 20 192.168.200.4
```

---

## OAI v1.2.1 EPC Setup (nFAPI)

This is a large amount of work and the setup instructions in the link below do it justice. Follow each step exactly! The UE/eNB machine and the EPC machine should be connected directly to each other through an ethernet connection!

Note the relevant section is 1.3 Installing EPC/HSS on 3rd Ubuntu Machine!! You will mess up your UE/eNB configuration if you try to follow the steps before this.

<https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectOAIENBWithOAIUEWithSInterface>