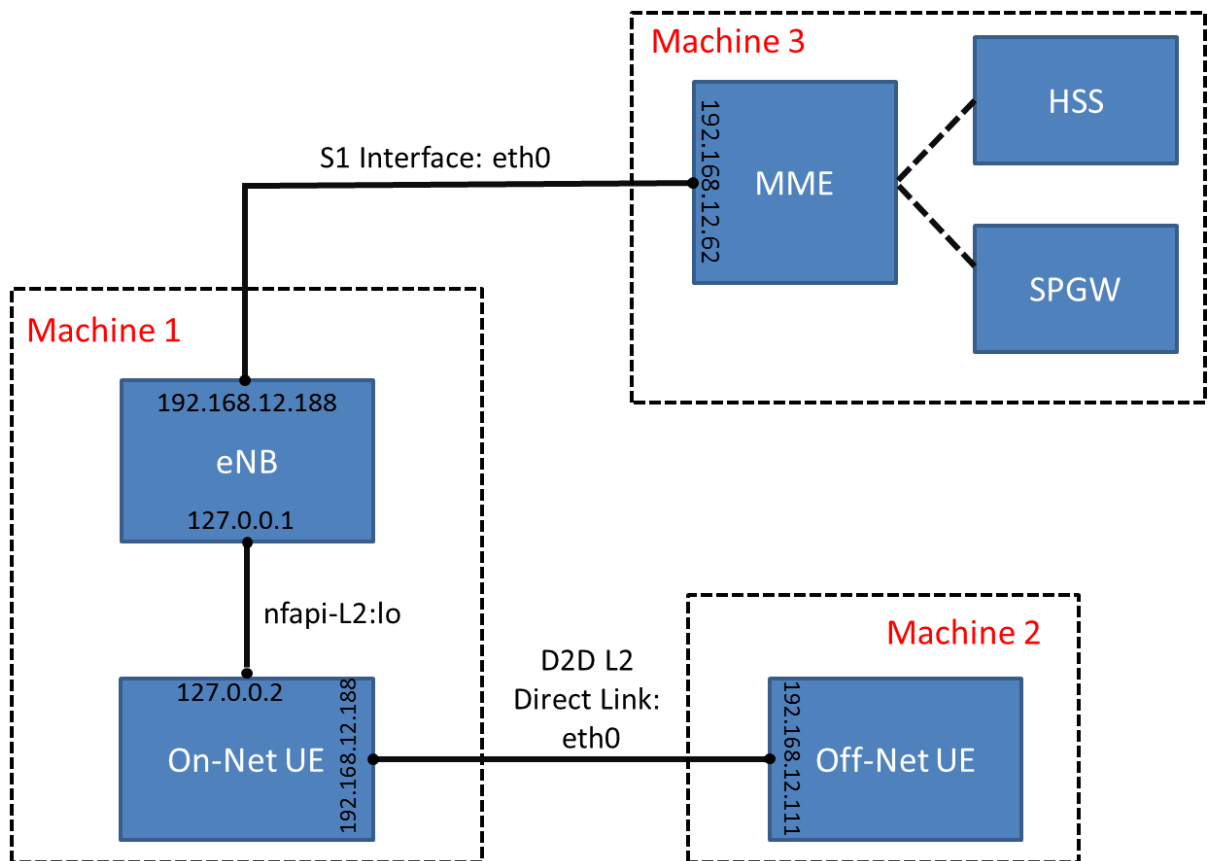


Instructions for launching the D2D On-Net emulator

The following guide describes the necessary steps to configure, build and launch the nfapi-L2-emulator in coordination with the D2D L2 emulator. For the D2D-specific emulation configuration you should still refer to the d2d-emulator-setup.txt document.

1. Testbed Setup



2. Launch EPC blocks

Provided that you have installed the individual EPC blocks (HSS, MME, SPGW), you can launch them through the respective commands in machine 3 from *openair-cn/scripts*:

- `./run_hss`

When launching the hss, you should get some parameters that will be needed for the UE authentication configuration (section 3). Particularly, you will need the **K** and **OPC** fields (see following example) and also make sure that the **IMSI** of your on-network UE exists in the database:

IMSI: 208930000000001 Key:

11.11.11.11.11.11.11.11.11.11.11.11.11.11.

OPc: f4.7f.37.e4.71.9a.9c.37.9e.34.47.f0.89.b1.f1.0a.

RijndaelKeySchedule: K 11111111111111111111111111111111

Compute opc:

K: 11111111111111111111111111111111

In: 11111111111111111111111111111111

Rinj: E56E26F5608B8D268F2556E198A0E01B

Out: F47F37E4719A9C379E3447F089B1F10A

Query: UPDATE `users` SET

`OPc`=UNHEX('f47f37e4719a9c379e3447f089b1f10a') WHERE

`users`.`imsi`='2089300000000997'

IMSI 2089300000000997 Updated OPc

f47f37e4719a9c379e3447f089b1f10a ->

f47f37e4719a9c379e3447f089b1f10a

- `./run_mme`
- `./run_spgw`

targets/PROJECTS/GENERIC-LTE-EPC/CONF/rcc.band7.tm1.50PRB.nfapi.conf,
as shown in the example:

```
////////// MME parameters:
mme_ip_address      = ( { ipv4      = "192.168.12.62";
                        ipv6      = "192:168:30::17";
                        active    = "yes";
                        preference = "ipv4";
                        }
                      );

NETWORK_INTERFACES :
{

    ENB_INTERFACE_NAME_FOR_S1_MME = "enp0s31f6";
    ENB_IPV4_ADDRESS_FOR_S1_MME   = "192.168.12.188/24";
    ENB_INTERFACE_NAME_FOR_S1U    = "enp0s31f6";
    ENB_IPV4_ADDRESS_FOR_S1U      = "192.168.12.188/24";
    ENB_PORT_FOR_S1U              = 2152; # Spec 2152
};
```

- You also need to add a new loopback interface address:

```
sudo ifconfig lo: 127.0.0.2 netmask 255.0.0.0 up
```

- Launch the eNB executable from *cmake_targets/lte_build_oai/build*:

```
sudo ./lte-softmodem -O PATH_OF:rcc.band7.tm1.50PRB.nfapi.conf
```

- Launch the connected UE executable from
cmake_targets/lte_build_oai/build:

```
sudo ./lte-uesoftmodem -U -O PATH_OF:oaiL1.nfapi.usrpb210.conf --L2-emul
3 --D2D-emul 1 --emul-iface enp0s31f6
```

4. Remote UE

- Launch the off-net UE executable from *cmake_targets/lte_build_oai/build*:

```
sudo ./lte-uesoftmodem -U --D2D-emul 1 --emul-iface enp0s31f6
```

5. Basic Relay scenario

In the following, we provide the instructions in order to be able to relay traffic from/to the remote UE through the relay UE, based on the network configuration shown in section 1. The traffic relaying achieved at this point is not 3gpp-aligned (e.g., no dedicated PDN connection for the remote UE is established; no RemoteUEReport message transmission to the EPC is performed at the Relay UE) and the EPC is not aware of the remote UE. Instead, the EPC (PGW) sees the remote's UE traffic as relay UE traffic, based on applying NAT masking at the relay UE.

- Make sure you load the `ue_ip` kernel module after you build the latest code:

```
cd LTE-D2D/cmake_targets/lte_build_oai/build
sudo insmod ../../targets/bin/ue_ip.ko
```

- If the module already exists you should remove it first
`sudo rmmod ue_ip`

And then re-load the latest one.

- Launch the on-net UE, eNB and EPC so that the UE acquires a configured `oip1` interface for the UL/DL scope.

- Verify that the DRB is successfully established by pinging from the relay UE to an external destination address (e.g. 8.8.8.8)

- Then, run the **LTE-D2D/relay_UE_routing.sh** script at the relay UE. Some of the applied commands may be overlapping with the ones described in `d2d-emulator-setup.txt` guide, or your application. In that case, you can remove the duplicated commands. As some of the routing rules are destination address specific (destination address: 8.8.8.8 in our scripts) modify accordingly if you want to exchange traffic with another destination.

- Run the **LTE-D2D/remote_UE_routing.sh** script at the remote UE. Some of the applied commands may be overlapping with the ones described in `d2d-emulator-setup.txt` guide, or your application. In that case, you can remove the duplicated commands. As some of the routing rules are destination address-specific (destination address: 8.8.8.8 in our scripts) modify accordingly if you want to exchange traffic with another destination.

- Start the OAI remote UE.

- Verify successful data traffic relaying by pinging from the remote UE (oip0) to external destination (e.g., 8.8.8.8). Use Wireshark to verify that the traffic is successfully relayed:

```
ping -I oip0 8.8.8.8
```