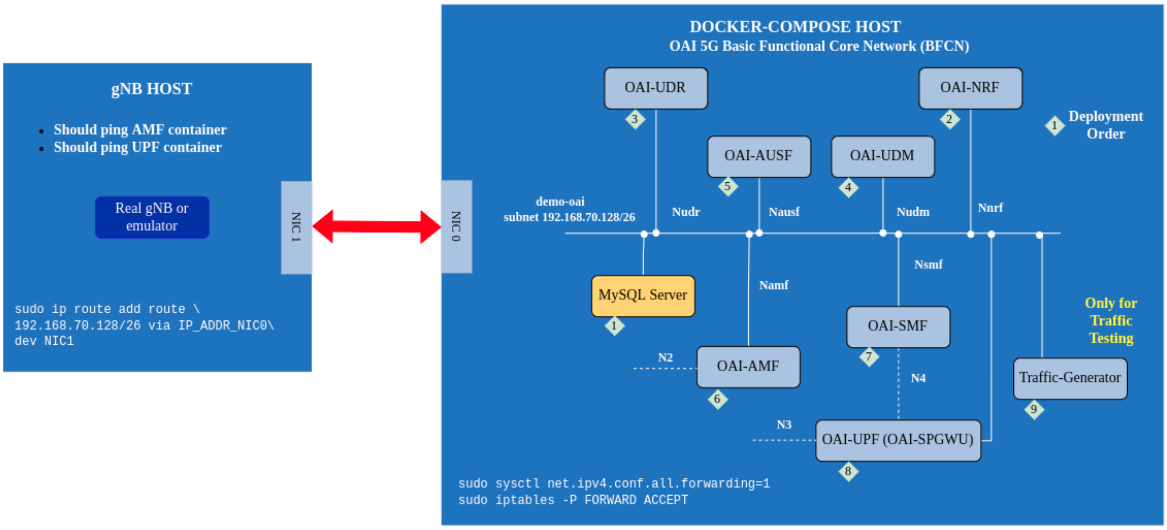
Lab 1: Setup a 5G Core Network based on OpenAirInterface (OAI)

OpenAirInterface (OAI) is a 3GPP-Compliant 4G/5G implementation of Radio Access Network (RAN) and Core Network (CN). In the scope of this lab, we will deploy OAI 5G CN and use a 5G RAN simulator to simulate gNB and User Equipment (UE). All the 5G Network Functions (NFs) and the RAN simulator are based on containers. You will use docker compose to easily run and configure all the containers. You will analyze different NFs and the UE registration procedure to understand the main functions of a 5G CN.



```
sudo ip route add route \
192.168.70.128/26 via IP_ADDR_NIC0\
dev NIC1
```

```
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo iptables -P FORWARD ACCEPT
```

OAI NFs are open source, and the source code can be found on OAI Gitlab or locally in your machines in /home/mobsys/lab1/oai-cn5g-fed/components. All the NF can run as containers connected to a virtual network.
Each folder inside /home/mobsys/lab1/oai-cn5g-fed/components contains the code source of a NF. Inside each folder, you will find:

- The source code inside src/ directory,
- a DockerFile file inside docker/ directory, used to build the container image (the latter will be used to instantiate the container)
- scripts to deploy the containers (health-check and entrypoint).
    - Docker executes the health-check scripts for each container: if the health-check script succeeds (returns 0) then docker reports healthy as status, unhealthy otherwise.
    - The entrypoint script is used to read the environment variables and to update the NF configuration file template. You can login inside the NF's container and check the updated configuration files inside /openair-XXX/etc directory (check docker commands at the end of this document).
- The template of the configuration file that will be updated using the entrypoint script.

Docker-compose is used to manage the containers and the virtual network connecting the containers. All the NFs are described in a docker-compose file which allows starting/stopping all the containers together using docker-compose command. You may check [1] for more details about the syntax used in the docker-compose files

Part 1:

Open a terminal and execute the following command: *watch docker ps*

Open another terminal, type:

cd /home/mobsys/*lab1/oai-cn5g-fed/docker-compose* and run the following command to start the 5G CN Network Functions (NFs): *docker-compose -f docker-compose-basic-nrf.yaml up -d*

wait until the docker ps command reports that each container is healthy (STATUS attribute).

Questions:

a/ Describe the output of watch docker ps (i.e., the columns of the output)

b/ what does the "services" attribute in the docker-compose file mean? describe the main attributes of a service.

c/ Which NF started first (refer to the docker-compose command output)? Why? (PS: mysql is not a NF).

d/ What are the running NFs? explain briefly the role of each NF (PS: the ext-dn is not a NF).

e/ What are the ports exposed by each NF? What is the role of each port? (refer to the SBA architecture of the 5G CN and the docker ps output)

f/ How OAI checks for healthy/unhealthy status of NFs? (check health-check scripts in each NF directory inside components/)

g/ Open Wireshark in privilege mode (sudo wireshark) and start capturing traffic for 30 seconds on the interface demo-oai. What are the two types of heartbeats that you see? precise between which entities are exchanged. What is the role of each heartbeat?

Part 2:

Open wireshark in privilege mode (sudo wireshark) and start capturing traffic on the demo-oai interface.

Now, connect gNB + UE using the following command:

docker-compose -f docker-compose-ue.yaml up -d

Stop the wireshark capture and save the logs which will be needed to answer the questions.

if you want to do the test again: stop the UE : docker-compose -f docker-compose-ue.yaml down, then connect it again using docker-compose -f docker-compose-ue.yaml up -d.

Questions:

a/ What are the 5G protocols involved in this capture? specify for each protocol, its endpoints and the protocol's role?

b/ Explain the messages related to security? Explain the content of each message? What is the relation between AUTN and RES? What is the value of SUCI? Which message includes this information? Can we deduce the IMSI value?

c/ What is the purpose of : (i) PDU session establishment request, (ii) PDU session establishment accept, (iii)PDUSessionResourceSetupRequest, (iv) PDUSessionResourceSetupResponse? Why step (i) and (ii) are not enough and we need step (iii) and (iv)? What is/are the most important parameter(s) in each message (the parameter and its value)? Why? Can a UE have multiple PDU sessions? Why?

d/ What is the IP address of the UE? Which entity assigned this IP address?

e/ What are the differences between PDU session establishment {Request/Accept} and Registration {Request, Accept} procedures?

f/ What is the main content and the purpose of : (i) PFCP Establishment Request/Response and (ii) PFCP Modification Request/Response. What is the Modification needed for?

g/ What is the relation between the procedure in question f) and the procedure in question c)?

h/ In which message(s) are the information related to slices present? What are these parameters? Explain the purpose of having this information in multiple messages and not only in one message.

Bonus Question: Update the SMF health-check script to report healthy only when heartbeats are successfully exchanged between SMF and UPF (you can test it by removing the UPF, the healthcheck should report unhealthy in this case).
Hint: Docker containers emit logs to the stdout and stderr output streams. Because containers are stateless, the logs are stored **on the Docker host in JSON files** by default. Create a healthcheck.sh to read the logs and to check for heartbeats. You will have to add a volume attribute in SMF service inside the docker-compose file to override the file /openair-smf/scripts/healthcheck.sh

Some useful docker commands:

docker-compose -h : list the available options.

docker-compose -f <FILE> up -d : start containers, networks described in <FILE> in detached mode (-d flag).

docker-compose -f <FILE> down: stop the containers, networks described in <FILE>.

docker-compose -f <FILE> up -d –force-recreate: restart the containers described in <FILE>.

docker logs <CONTAINER NAME>: read the logs of <CONTAINER NAME>.

docker exec -it <CONTAINER NAME> bash: login inside the container <CONTAINER NAME>

docker stop <CONTAINER NAME>: stop the container

docker rm <CONTAINER NAME>: remove the container

[1] https://docs.docker.com/compose/compose-file/