Part 5. Set up Aerial CUDA-Accelerated RAN Layer 1 with OAI gNB and CN5G (For Debug Only)

Table of contents

This section describes how to set up the Aerial private 5G network, which consists of the following:

- Aerial CUDA-Accelerated RAN Layer 1
- Remaining components of OAI gNB
- OAI Core Network
- User Equipment (UE)
- Edge Server Applications (for example, iPerf)

ARC-OTA Software Release Manifest

Component	Version
Aerial CUDA-Accelerated RAN (Layer 1)	23-4
OAI gNB	2024.w15
OAI CN	2024.w15
Sterling Skywave Service Management	v0.4.0

> **Note**
>
> Only Layer 1 from Aerial CUDA-Accelerated RAN will be used by ARC-OTA in this release.

Setup Aerial CUDA-Accelerated RAN Layer 1

In the installation guide for cuBB, find the Aerial CUDA-Accelerated RAN Layer 1 section and follow the instructions (see the ARC-OTA Software Release Manifest for the cuBB document version and link). Installing ARC-OTA using SDK Manager.

> **Note**
>
> You can also refer to the tutorial videos for installation steps.

Running the cuBB Docker Container

```
export GPU_FLAG="--gpus all" export cuBB_SDK=/opt/nvidia/cuBB #Name of your docker container export AERIAL_CUBB_CONTAINER=cuBB_$USER #Docker image downloaded from NGC export AERIAL_CUBB_IMAGE=gitlab-master.nvidia.com:5005/gputelecom/container/cubb:Aerial-cuBB-container-ubuntu22.04-23.04.0-Rel-23-4.256-x86_64 sudo usermod -aG docker $USER docker run --detach --privileged \ -it $GPU_FLAG --name $AERIAL_CUBB_CONTAINER \ --hostname c_aerial_$USER \ --add-host c_aerial_$USER:127.0.0.1 \ --network host \ --shm-size=4096m \ -e cuBB_SDK=$cuBB_SDK \ -w $cuBB_SDK \ -v $(echo ~):$(echo ~) \ -v /dev/hugepages:/dev/hugepages \ -v /usr/src:/usr/src \ -v /lib/modules:/lib/modules \ -v ~/share:/opt/cuBB/share \ --userns=host \ --ipc=host \ -v /var/log/aerial:/var/log/aerial \ $AERIAL_CUBB_IMAGE docker exec -it $AERIAL_CUBB_CONTAINER bash
```

For installation instructions, see the Aerial cuBB Installation Guide, in the link above.

Since the cuBB 23-4 release, the necessary testvectors for running OTA are already included. For running the Aerial E2E test with ru-emulator and test mac, follow the Aerial CUDA-Accelerated RAN documentation for generating the required testvectors.

Setup OAI gNB

Clone the gNB Source Code

Clone the OpenAirInterface5G repository.

```
git clone --branch 2024.w15 https://gitlab.eurecom.fr/oai/openairinterface5g.git ~/openairinterface5g cd openairinterface5g
```

gNB Configuration File

Update the configuration of OAI L2. The configuration is located here.

L1 configuration used is included the latest aerial release image. For Aerial devit please use

> cuPHY-CP/cuphycontroller/config/cuphycontroller_P5G_FXN.yaml

and for Dell R750 please use

> cuPHY-CP/cuphycontroller/config/cuphycontroller_P5G_FXN_R750.yaml

Setup OAI CN5G

Do the iptables setup below after every system reboot, or make this setup permanent in your Ubuntu system configuration.

> On CN5G server, configure it to allow the traffic coming in by adding this rule to iptables: # On CN5G, upon startup: sudo sysctl net.ipv4.conf.all.forwarding=1 sudo iptables -P FORWARD ACCEPT

Install the core network by following the Gitlab steps for setting up OAI CN5G.

To run the correct configuration for ARC-OTA, replace section 2.2 and 2.3 OAI CN5G configuration files with the following:

> # Get openairinterface5g source code git clone --branch 2024.w15 https://gitlab.eurecom.fr/oai/openairinterface5g.git ~/openairinterface5g cd ~/openairinterface5g cp -rT ~/openairinterface5g/doc/tutorial_resources/oai-cn5g ~/oai-cn5g

The user configurable configuration files are:

- ~/oai-cn5g/database/oai_db.sql

Configuring OAI gNB and CN5G

For the purpose of understanding which address is what in the example configuration setting and commands below, we will assume the gNB and CN5G servers have these interface names and IP addresses.

CN5G Server

> eno1: 10.31.66.x = SSH management port for terminal eno2: 169.254.200.6 = BH connection on SFP switch for gNB-CN5G traffic

gNB Server

> eno1: 10.31.66.x = SSH management port for terminal ens6f0: b8:ce:f6:4e:75:40 = FH MAC address ens6f0.2: 169.254.1.105 = FH IP address ens6f1: 169.254.200.5 = BH connection SFP switch for gNB-CN5G traffic

gNB to set static route

On the gNB server, add this static route for a path to the CN5G server. Apply this route after each server power-on.

> Syntax: sudo ip route add 192.168.70.128/26 via <CN5G IP> dev <gNB interface for CN5G> Example: sudo ip route add 192.168.70.128/26 via 169.254.200.6 dev ens6f1

gNB to set the CN5G server to uses for AMF

Edit gNB configuration file: targets/PROJECTS/GENERIC-NR-5GC/CONF/vnf.sa.band78.fr1.273PRB.Aerial.conf

Below is an example with lab-specific network parameters. Your IP address and interface names may differ.

> GNB_INTERFACE_NAME_FOR_NG_AMF = "ens6f1"; # gNB side interface name of the SFP port toward CN (was eno1) GNB_IPV4_ADDRESS_FOR_NG_AMF =

```
"169.254.200.5"; # gNB side IP address of interface above (was 172.21.16.130)
GNB_INTERFACE_NAME_FOR_NGU = "ens6f1"; # gNB side interface name of the SFP port toward CN (was eno1) GNB_IPV4_ADDRESS_FOR_NGU = "169.254.200.5"; # Same IP as GNB_IPV4_ADDRESS_FOR_NG_AMF above (was 172.21.16.130)
```

Running CN5G

To start CN5G

```
docker-compose up -d
```

To Stop CN5G

```
docker-compose down
```

To monitor CN5G logs while running

```
docker logs oai-amf -f
```

To capture PCAPs

```
docker exec -it oai-amf /bin/bash apt update && apt install tcpdump -y tcpdump -i any -w /tmp/amf.pcap
```

Then copy the pcap out from the container.

```
docker cp oai-amf:/tmp/amf.pcap .
```

Example Screenshot of Starting CN5G

```
aerial@aerial-rf-r630:~/oai-cn5g$ docker compose up -d [+] Building 0.0s (0/0) [+] Running 11/11  Network demo-oai-public-net Created 0.1s  Container oai-nrf Started 0.7s  Container mysql Started 0.7s  Container asterisk-ims Started 0.7s  Container oai-udr Started 0.9s  Container oai-udm Started 1.2s  Container oai-ausf Started 1.5s  Container oai-amf Started 1.7s  Container oai-smf Started 2.0s  Container oai-spgwu-tiny Started 2.3s  Container oai-ext-dn Started 2.6s
```

```
aerial@aerial-rf-r630:~/oai-cn5g$ docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES d5af4f51c393 oaisoftwarealliance/trf-gen-cn5g:latest "/bin/bash -c ' ip r..." About a minute ago Up About a minute (healthy) oai-ext-dn a9b2d18c7f77 oaisoftwarealliance/oai-spgwu-tiny:v1.5.1 "python3 /openair-sp..." About a minute ago Up About a minute (healthy) 2152/udp, 8805/udp oai-spgwu-tiny b61c383f9e60 oaisoftwarealliance/oai-smf:v1.5.1 "python3 /openair-sm..." About a minute ago Up About a minute (healthy) 80/tcp, 8080/tcp, 8805/udp oai-smf 3681b1048c53 oaisoftwarealliance/oai-amf:v1.5.1 "python3 /openair-am..." About a minute ago Up About a minute (healthy) 80/tcp, 9090/tcp, 38412/sctp oai-amf c602f7cb1c67 oaisoftwarealliance/oai-ausf:v1.5.1 "python3 /openair-au..." About a minute ago Up About a minute (healthy) 80/tcp oai-ausf 752acae83ac0 oaisoftwarealliance/oai-udm:v1.5.1 "python3 /openair-ud..." About a minute ago Up About a minute (healthy) 80/tcp oai-udm 4bf281d08229 oaisoftwarealliance/oai-udr:v1.5.1 "python3 /openair-ud..." About a minute ago Up About a minute (healthy) 80/tcp oai-udr 33aa959be775 mysql:8.0 "docker-entrypoint.s..." About a minute ago Up About a minute (healthy) 3306/tcp, 33060/tcp mysql 5d22e4745d00 asterisk-ims:latest "asterisk -fp" About a minute ago Up About a minute (healthy) asterisk-ims 1a93b3ffe305 oaisoftwarealliance/oai-nrf:v1.5.1 "python3 /openair-nr..." About a minute ago Up About a minute (healthy) 80/tcp, 9090/tcp oai-nrf
```