Mosaic5G
A community Led Consortium

Leveraging an Ecosystem of 5G services
Outline: 1st part

- Tutorial Requirements
- System Requirements
- Simple Deployment Scenarios
- OAI software
- What is Snap
- OAI-CN and OAI-RAN Snaps
- Testing

Snap version 1.3 and 1.4
Tutorial Requirements

- Basic Knowledge in
  - LTE, SDN, NFV, MEC
  - Linux, networking, Ubuntu tools (e.g. apt)
  - Wireshark
Typical System requirements

**FlexRAN:** 1 machine
- 64-bits, 8GB RAM
- 2 cores Intel Gen. 3/4/5/6 i3, i5, i7
- 1 Ethernet ports (1Gb)

**LL-MEC:** 1 machine
- 64-bits, 8GB RAM
- 2 cores Intel Gen. 3/4/5/6 i3, i5, i7
- 1 Ethernet ports (1Gb)

**OAI-RAN:** 1-2 machines
- Ubuntu 16.04, Kernel > 4.10.x
- 64-bits, 8-16GB RAM
- 4 cores Intel Gen. 3/4/5/6 i5, i7 or 2/3/4 Xeon
- 1-2 Ethernet ports (1Gb S1, Opt: 1/10Gb for Fronthaul)
- SDR interface: 1 USB3 or 1 10Gb

**OAI-CN:** 1-3 machines
- Ubuntu 16.04, Kernel > 4.10.x
- 64-bits, 8-16GB RAM
- 2-4 cores Intel Gen. 3/4/5/6 i5, i7 or 2/3/4 Xeon
- 1-2 Ethernet ports (1Gb S1, 1Gb SGI)

**OVS-GTP:** 1 machine
- Ubuntu 16.04, kernel 4.9.x – 4.11.x
- 64-bits, 8GB RAM
- 2 cores Intel Gen. 3/4/5/6 i3, i5, i7
- Min 2 Ethernet ports (1Gb)
Required Hardware for this tutorial

- 1-2 Ubuntu 16.04 PC with
  - 2-4 cores each
  - 8-16 GB RAM
  - 1-2 Ethernet Ports
  - 1-2 USB3

- For RAN (Few SDRs will be provided by Eurecom during the tutorial):
  - 1 USRP B210 or 205
  - 1 Smartphone

- For LL-MEC, update the kernel for one of the PC as follows:
  - kernel 4.9.x
Required Hardware for OAI-RAN and OAI-UE

- **SDR:**
  - USRP X3xx, N3xx, or B2xx families (recommended to start):
  - LimeSDR
    - [https://www.crowdsupply.com/lime-micro/limesdr](https://www.crowdsupply.com/lime-micro/limesdr)
  - BladerF
Required Hardware for OAI-RAN and OAI-UE

- Duplexer (Filters with and without PA (i.e. passive or active): http://www.expressmimo2.com/
  - https://open-cells.com/index.php/opencellsband7duplexer/
- Smartphone: Samsung Galaxy S6, Nexus6, or Google Pixel 2. You may buy different chipsets, and UE categories with external antenna ports
- SIM cards:
- Optional:
  - Synchronization modules: either GPS (e.g. GPS module in USRP) or 1588v2 master and client module
  - Fronthaul Switch.

Snap version 1.3 and 1.4
3/18/2019
Additional links

- HW/SW requirements M5G [Restricted Access]
  - [https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/hw-sw-reqs](https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/hw-sw-reqs)

- Tutorials M5G [Restricted Access]
  - [https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/tutorials](https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/tutorials)

- Howto
  - [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirSoftwareSupport](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirSoftwareSupport)

- Tutorials
  - [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage)

- T Tracer
  - [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/T](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/T)
LTE Architecture
UE Attach Procedure in LTE (1/4)

1. S1AP: S1 Setup (eNB ID, TAC)
2. RRC: MIB, SIB 1, SIB 2
3. RA-RNTI: Preamble
4. RAR, TC-RNTI
5. RRC: Connection Req, Random or S-TMSI, Cause
6. RRC: Connection Setup, Contention Resolution, SRB1 parameters
7. RRC Connection Setup Complete + NAS
8. S1AP: Initial UE message (eNB-UE-S1AP-ID) + NAS
9. NAS: EMM attach request, ESM PDN connectivity Request
UE Attach Procedure in LTE (2/4)
UE Attach Procedure in LTE (3/4)
UE Attach Procedure in LTE (4/4)
UE Dedicated Bearer Establishment in LTE (1/1)
UE Detach Procedure in LTE (1/1)

Diagram showing the process:

1. **Release**
   - S1AP: S1 E-RAB Release Command

2. **RRC Connection**
   - RRC: Connection Reconfiguration + NAS
   - NAS: Deactivate EPS Bearer Context Request
   - RRC Connection Reconfiguration Complete
   - S1AP: S1 E-RAB Release Response

3. **EPS Bearer Deactivation**
   - RRC: UL Info Transfer + NAS
   - S1AP: UL NAS Transport + NAS
   - NAS: Deactivate EPS Bearer Context Accept
   - S11: Release Bearer
## Attach with OAI

<table>
<thead>
<tr>
<th>Message of interest</th>
<th>Protocol</th>
<th>Message Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>#185</td>
<td>S1AP/NAS</td>
<td>InitialUEmessage</td>
</tr>
<tr>
<td>#186</td>
<td>S1AP/NAS</td>
<td>Identity Request (IMSI)</td>
</tr>
<tr>
<td>#187</td>
<td>S1AP/NAS</td>
<td>Identity Response (IMSI)</td>
</tr>
<tr>
<td>#188</td>
<td>DIAMETER</td>
<td>Authenticationinfo Request</td>
</tr>
<tr>
<td>#197</td>
<td>DIAMETER</td>
<td>Answer</td>
</tr>
<tr>
<td>#198</td>
<td>S1AP/NAS</td>
<td>Authentication Request</td>
</tr>
<tr>
<td>#201</td>
<td>S1AP/NAS</td>
<td>Authentication Response</td>
</tr>
<tr>
<td>#202</td>
<td>S1AP/NAS</td>
<td>Security mode command</td>
</tr>
<tr>
<td>#203</td>
<td>S1AP/NAS</td>
<td>Security mode complete</td>
</tr>
<tr>
<td>#204</td>
<td>DIAMETER</td>
<td>UpdateLocation Request</td>
</tr>
<tr>
<td>#216</td>
<td>DIAMETER</td>
<td>UpdateLocation Answer</td>
</tr>
<tr>
<td>#218</td>
<td>GTPv2-c</td>
<td>Create Session Request</td>
</tr>
<tr>
<td>#219</td>
<td>GTPv2-c</td>
<td>Create Session Response</td>
</tr>
<tr>
<td>#220</td>
<td>S1AP/NAS</td>
<td>Initial Context Setup Req</td>
</tr>
<tr>
<td>#221</td>
<td>S1AP</td>
<td>UE Capability Info Indication</td>
</tr>
<tr>
<td>#224</td>
<td>S1AP/NAS</td>
<td>Initial Context Setup Resp</td>
</tr>
<tr>
<td>#225</td>
<td>GTP-c</td>
<td>Modify Bearer Request</td>
</tr>
<tr>
<td>#226</td>
<td>GTP-c</td>
<td>Modify Bearer Response</td>
</tr>
</tbody>
</table>
OAI Protocol Stack

Snap version 1.3 and 1.4
Deployment Scenario OAI-RAN and OAI-CN

- OAI-CN
- Commercial/3rd party EPC
- OAI-RAN (eNB, CRAN)
- Commercial/3rd party eNB
- OAI UE
- COTS UE
Ubuntu Snaps

Snap version 1.3 and 1.4
Snaps are containerized software packages that bundle their dependencies,
  - Claimed to work on all major Linux systems without modification.
An Ubuntu software deployment and package management system
The packages called 'snaps' and the tool for using them 'snapd',
Snapcraft is a tool for developers to package their programs in the Snap format for
Auto-build from GitHub
 Reeves:
  - https://snapcraft.io/
  - https://tutorials.ubuntu.com/tutorial/basic-snap-usage#0
  - https://tutorials.ubuntu.com/tutorial/advanced-snap-usage#0
Ubuntu Snap Variables

- **SNAP**: Directory where the snap is mounted. This is where all the files in your snap are visible in the filesystem. All of the data in the snap is **read-only** and cannot be changed.
  - **Typical value**: `/snap/$SNAP_NAME/$SNAP_REVISION`

- **SNAP_DATA**: Directory for system data of a snap. This directory is owned and writable by root and is meant to be used by background applications (daemons, services).
  - **Value**: `/var/snap/$SNAP_NAME/$SNAP_REVISION`

- **SNAP_USER_DATA**: Directory for user data. This directory is backed up and restored across snap refresh and snap revert operations.
  - **Value**: `/home/$USER/snap/$SNAP_NAME/$SNAP_REVISION`

Snap version 1.3 and 1.4
Default config files are stored in $SNAP\_DATA (e.g. /var/snap/oai-cn/current/hss_fd.conf) and a user backup config file is created in $SNAP\_USER\_DATA

A config file can be set by x-conf-set command

- oai-cn.hss-conf-set /tmp/myconfig.conf

For each initialization and conf-set command, the config files are overwritten
Each Snap has

- version: the version of the software being packaged, as assigned by the developers.
- Each Snap has a revision in each channel (e.g. edge, beta, stable)
  - channel: defines which releases of snap associated with a version are pushed
    - https://docs.snapcraft.io/channels/551
  - revision: the sequence number assigned by the store when the snap file was uploaded

Example

<table>
<thead>
<tr>
<th>Channel</th>
<th>Version (Revision)</th>
<th>size</th>
<th>Confinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>stable:</td>
<td>1.0 (2)</td>
<td>34MB</td>
<td></td>
</tr>
<tr>
<td>candidate:</td>
<td>1.0 (2)</td>
<td>34MB</td>
<td></td>
</tr>
<tr>
<td>beta:</td>
<td>1.3 (26)</td>
<td>32MB</td>
<td>devmode</td>
</tr>
<tr>
<td>edge:</td>
<td>1.3 (26)</td>
<td>32MB</td>
<td>devmode</td>
</tr>
<tr>
<td>installed:</td>
<td>1.3 (23)</td>
<td>32MB</td>
<td></td>
</tr>
</tbody>
</table>
Ubuntu Snap useful commands

- `$snap login your@email.lol`
- `$snap find query`
- `$sudo snap install $SNAP_NAME`
- `$sudo snap remove $SNAP_NAME`
- `$snap list`
- `$snap info $SNAP_NAME`
- `$sudo snap refresh $SNAP_NAME`

If the snap implements the service

- `$snap services $SNAP_NAME`
- `$snap start $SNAP_NAME.ServiceName`
- `$snap stop $SNAP_NAME.ServiceName`
- `$snap restart $SNAP_NAME.ServiceName`
All the M5G snaps are released under **edge** and **beta** channel

M5G snap version is incremented **when a new feature is released**
- In future, the version numbering will contain both major and minor number indicating both OAI and M5G version numbers

M5G revisions are incremented **based on a bug fixes and enhancement of existing features**

A Snap may include multiple applications
- Example: OAI-CN snap include hss, mme, and spgw apps

M5G snap releases can be found at [restricted access]
- https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/releases
Clone the mosaic5g meta repository

1. $> lsb_release -a
   Distributor ID: Ubuntu
   Description: Ubuntu 16.04.x LTS
   Release: 16.04
   Codename: xenial
2. $> sudo apt install git vim gitk
3. $> git clone https://gitlab.eurecom.fr/mosaic5g/mosaic5g.git
4. $> cd mosaic5g
5. $mosaic5g > ./build_m5g -help
   https://gitlab.eurecom.fr/mosaic5g/mosaic5g/blob/master/README.md
Build OAI-CN from Snap

- `$mosaic5g > ./build_m5g -C # snap OAI-CN`
  - Ref: [https://snapcraft.io/oai-cn](https://snapcraft.io/oai-cn)
  - `$mosaic5g > oai-cn.help # check the help

- Install manually
  - `sudo snap install oai-cn -channel=edge --devmode`

- Considered deployment scenario:
  - **EPC : ALL-IN-ONE Machine**
  - **Other options:**
    - MME+HSS: 1 machine
    - SPGW: 1 machine

- **Ref:**
  - [https://gitlab.eurecom.fr/mosaic5g/mosaic5g.wikis/tutorials/oai-cn](https://gitlab.eurecom.fr/mosaic5g/mosaic5g.wikis/tutorials/oai-cn)

Snap version 1.3 and 1.4

3/18/2019
RUN oai-cn.hss

- `sudo apt install mysql-server mysql-client`
  - Default Username: root
  - Default Password: linux
  - The actual username and password for oai-hss are taken from the hss.conf file

- Get the current hss config file
  - `sudo oai-cn.hss-conf-get`

- Configure hss_fd.conf located under the same dir
  - `sudo vim /var/snap/oai-cn/current/hss_fd.conf`
  - Change the identity to the name of your machine: `identity="[machine].openair4G.eur"`
  - Optionally: change the REALM (i.e. `openair5G.eur`) to the realm of your choice
    - Update also needed in hss_fd.conf and acl.conf

- Double check the hss FQDN in /etc/hosts, and make sure that it corresponds to “machine.realm”

- Check the OP key in: `sudo vim /var/snap/oai-cn/current/hss.conf` and update if needed

- Create new certificates with the correct machine identity
  - `sudo oai-cn.hss-init`

- Run HSS:
  - `sudo oai-cn.hss` # run manually
  - `sudo oai-cn.hss-start` # run as a service

- To check if the service is running
  - `oai-cn.hss-status` or `oai-cn.hss-journal`
OAI-CN.HSS Manage Users

- Add a user to hss DB
  - `sudo oai-cn.hss-add-user IMSI [USER_KEY][SQN][MMEIDENTITY][OPC]`
    - USER_KEY and OPC shall be in Hexadecimal while the others in decimal
    - Example: `oai-cn.hss-add-user 208950000000020 6b6a5e 20 7 8e44a8b`
  - Operator key is taken from hss.conf, default user key, sqn, and mmeid will be taken if not provided
  - Have a look at: [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/SIMInfo](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/SIMInfo)

- Dump all users from hss db:
  - `oai-cn.hss-dump-users`

- Dump particular users or a user from hss db:
  - `oai-cn.hss-dump-users | grep 20895`
  - `oai-cn.hss-dump-users | grep 208950000000001`

- Reset the DB to its initial values
  - `sudo oai-cn.hss-reset-db`
  - `sudo oai-cn.hss-init`
OAI-CN.HSS Manage MMEs

- Add a MME identified by its FQDN (default is the local FQDN)
  - `oai-cn.hss-add-mme MME_FQDN`
  - `oai-cn.hss-add-mme mme.openair4G.eur`
  - Note that
    - MME_FQDN is composed of machinename.realm, e.g. `ubuntu.openair4G.eur`
    - MME_FQDN is mapped to a routable IP address in `/etc/resolvle.conf`
- Dump the current authenticated mme hosts
  - `oai-cn.hss-dump-mmeid`
Check if OAI-HSS is running

Snap version 1.3 and 1.4

3/18/2019
Optional: install phpmyadmin and check the hss DB

- `$sudo apt install phpmyadmin`
- `$Firefox http://localhost/phpmyadmin`
  - User: root, pwd: linux
- Issue if phpmyadmin not found:
  - `$sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf`
  - `$sudo a2enconf phpmyadmin.conf`
  - `$service apache2 reload`
  - `$sudo systemctl restart apache2`
OAI-CN.HSS available Commands

- oai-cn.hss-init
- oai-cn.hss
- oai-cn.hss-journal
- oai-cn.hss-reset-db
- oai-cn.hss-restart
- oai-cn.hss-start
- oai-cn.hss-status
- oai-cn.hss-stop
- oai-cn.hss-add-mme
- oai-cn.hss-remove-mme
- oai-cn.hss-add-user
- oai-cn.hss-remove-user
- oai-cn.hss-conf-get
- oai-cn.hss-conf-list
- oai-cn.hss-conf-set
- oai-cn.hss-conf-show
- oai-cn.hss-dump-mmeid
- oai-cn.hss-dump-users
Run oai-cn.mme

- Configure mme
  - sudo oai-cn.mme-conf-get
  - update the mme.conf file: mainly the “HSS_HOSTNAME”, “NETWORK_INTERFACES”, and “S-GW” sections
    - sudo vim /var/snap/oai-cn/current/mme.conf
    - Optionally: change the REALM (i.e. openair5G.eur) to the realm of your choice
  - Update the mme_fd.conf: Change the identity to the name of your machine: identity="[machine].openair4G.eur"
    - sudo vim /var/snap/oai-cn/current/mme_fd.conf
    - Optionally: change the REALM (i.e. openair5G.eur) to the realm of your choice

- Double check the MME FQDN in /etc/hosts, and make sure that it corresponds to "machine.realm"
  - Note that the realm should be the same as the one used for HSS

- Create new certificates for your machine
  - sudo oai-cn.mme-init

- Run MME:
  - sudo oai-cn.mme # run manually
  - sudo oai-cn.mme-start # run as a service

- Check the status of MME
  - oai-cn.mme-status or oai-cn.mme-journal
Check if OAI-MME is running

Snap version 1.3 and 1.4

3/18/2019
OAI-CN.MME available commands

- **mme-start:** start the oai-cn.mmed daemon
- **mme-stop:** stop the oai-cn.mmed daemon
- **mme-restart:** restart the oai-cn.mmed daemon
- **mme-status:** get the oai-cn.mmed status
- **mme-journal:** get the oai-cn.mmed logs

- **mme:** run MME
- **mme-init:** initialize MME
- **mme-conf-set:** set the MME configuration file
- **mme-conf-get:** get the current MME configuration file (mme.conf)
- **mme-conf-show:** show the path to the MME configuration file (mme.conf)
- **mme-conf-list:** list all the SPGW configuration files (mme.conf, mme_fd)

Snap version 1.3 and 1.4

3/18/2019
Run oai-cn.spgw

- **Configure SPGW**
  - Get the current path to `spgw.conf`:
    - `sudo oai-cn.spgw-conf-get`
  - Update the `spgw.conf` file: mainly the “NETWORK_INTERFACES” in S-GW (S1 and S11) section and P-GW (SGi interface through which you access Internet, interface name, and DNS) section
    - `sudo vim /var/snap/oai-cn/current/spgw.conf`

- **Run SP-GW**
  - `sudo oai-cn.spgw` # run manually
  - `sudo oai-cn.spgw-start` # run as a service

- **Check the status**
  - `oai-cn.spgw-status` or `oai-cn.spgw-journal`
Check if OAI-SPGW is running

Snap version 1.3 and 1.4

Copyright (c) 2019-2021 Anit Chawre
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THE SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
**OAI-CN.SPGW available commands**

- **spgw-start:** start the oai-cn.spgwdaemon
- **spgw-stop:** stop the oai-cn.spgwdaemon
- **spgw-restart:** restart the oai-cn.spgwdaemon
- **spgw-status:** get the oai-cn.spgwdaemon status
- **spgw-journal:** get the oai-cn.spgwdaemon logs

- **spgw:** run SPGW
- **spgw-init:** initialize SPGW
- **spgw-conf-set:** set the SPGW configuration file
- **spgw-conf-get:** get the current SPGW configuration file (spgw.conf)
- **spgw-conf-show:** show the path to the SPGW configuration file (spgw.conf)
- **spgw-conf-list:** list all the SPGW configuration files (spgw.conf)
Additional OAI-CN Commands

- `oai-cn.status-all:` status of hssd, mmed, and spgwd daemons
- `oai-cn.start-all:` start the hssd, mmed, and spgwd daemons
- `oai-cn.stop-all:` stop the hssd, mmed, and spgwd daemons
- `oai-cn.test-all:` test the operation of hssd, mmed, and spgwd daemons
OAI-RAN Snap

Snap version 1.3 and 1.4
Build OAI-RAN from Snap

- $mosaic5g > ./build_m5g -R # SNAP OAI-RAN
  - Ref: https://snapcraft.io/oai-ran
- $mosaic5g > oai-ran.help # check the help

- Supported RF:
  - USRP SDFR family

- Ref: https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/tutorials/oai-ran

Snap version 1.3 and 1.4
3/18/2019
1. Monolithic RAN/BS commands
   - oai-ran.enb*

2. Disaggregated RAN/BS commands
   - oai-ran.cudu*

3. Monolithic or Disaggregated RRU
   - oai-ran.rru*: Works with option 2
RUN oai-ran.enb

- Get the path to the enb config file
  - oai-ran.enb-conf-get # update the configuration file
  - update the conf file: mainly the "MCC and MNC", "mme_ip_address", and "NETWORK_INTERFACES" sections
    - sudo vim /var/snap/oai-cn/current/enb.band7.tm1.50PRB.usrp210.conf
  - Optional: Update the mme.conf: Change the MNC and MCC accordingly
    - sudo vim /var/snap/oai-cn/current/mme.conf

- oai-ran.enb-conf-show # show the current configuration file

- Run monolithic eNB
  - sudo oai-ran.enb # run manually
  - sudo oai-ran.enb-start # run as a service

- Check the status
  - oai-ran.enb-status or oai-ran.enb-journal
Check if OAI-RAN is running

Snap version 1.3 and 1.4

3/18/2019
RUN oai-ran.cudu

- Get the path to the cudu config file
  - `oai-ran.cudu-conf-get`  # update the configuration file
  - update the conf file: “MCC and MNC”, “mme_ip_address”, “NETWORK_INTERFACES”, and RU sections
    - `sudo vim /var/snap/oai-cn/current/rcc.band7.tm1.50PRB.usrpb210.conf`
  - Optional: Update the mme.conf: Change the MNC and MCC accordingly
    - `sudo vim /var/snap/oai-cn/current/mme.conf`
  - `oai-ran.cudu-conf-show`  # show the current configuration file

- Run disaggregated BS
  - `sudo oai-ran.cudu`  # run manually
  - `sudo oai-ran.cudu-start`  # run as a service

- Check the status
  - `oai-ran.cudu-status` or `oai-ran.cudu-journal`

Snap version 1.3 and 1.4
RUN oai-ran.rru

- Get the path to the rru config file
  - `oai-ran.rru-conf-get`  # update the configuration file
- Update the conf file: mainly the RRU sections
  - `sudo vim /var/snap/oai-cn/current/rru.conf`
  - Note that the supported bands must be among the one used by CUDU
    - e.g. Bands = [7, 13]
- `oai-ran.rru-conf-show`  # show the current configuration file
- Run Disaggregated BS
  - `sudo oai-ran.rru`  # run manually
  - `sudo oai-ran.rru-start`  # run as a service
- Check the status
  - `oai-ran.rru-status` or `oai-ran.rru-journal`
Example of local CUDU and RRU config with functional split IF4.5

CUDU

RUs = (
{
    local_if_name = "lo";
    remote_address = "127.0.1.1";
    local_address = "127.0.2.2";
    local_portc = 50000;
    remote_portc = 50000;
    local_portd = 50001;
    remote_portd = 50001;
    local_rf = "yes"
    tr_preference = "udp_if4p5";
    nb_tx = 1;
    nb_rx = 1;
    max_pdschReferenceSignalPower = -27;
    max_rxgain = 125;
    bands = [7,13];
}
);

RRU

RUs = (
{
    local_if_name = "lo";
    remote_address = "127.0.1.1";
    local_address = "127.0.2.2";
    local_portc = 50000;
    remote_portc = 50000;
    local_portd = 50001;
    remote_portd = 50001;
    local_rf = "no"
    tr_preference = "udp_if4p5"
    nb_tx = 1;
    nb_rx = 1;
    att_tx = 0;
    att_rx = 0;
    eNB_instances = [0];
}
)
### Additional OAI-RAN Commands (1/2)

**Usage:**
```
sudo oai-ran.[x] args, where
app[x] is enb, cudu, rru, and uesim
```

**Note:**
root privilege required

**Commands:**
- `oai-ran.info` get the info for this snap
- `oai-ran.status-all` status of all the services of this snap
- `oai-ran.stop-all` stop all the services of this snap
- `oai-ran.test` test snap services
- `oai-ran.init` initialize oai-ran (after the 1st installation)
Additional OAI-RAN Commands (2/2)

- `oai-ran.[x]-journal`  get the oai-ran.[x] daemon logs
- `oai-ran.[x]-restart` restart the oai-ran.[x] daemon
- `oai-ran.[x]-start` start the oai-ran.[x] daemon
- `oai-ran.[x]-status` status the oai-ran.[x] daemon
- `oai-ran.[x]-stop` stop the oai-ran.[x] daemon

- `oai-ran.[x]` run the App [x] manually
- `oai-ran.[x]-conf-get` get the current oai-ran.[X] configuration file
- `oai-ran.[x]-conf-list` list the oai-ran.[x] configuration files
- `oai-ran.[x]-conf-set` set the oai-ran.[x] configuration file
- `oai-ran.[x]-conf-show` show the path to the oai-ran.[x] configuration file
- `oai-ran.[x]-debug` debug the oai-ran.[x] with gdb, only valid for enb and cudu
Build OAI-UE from Snap

- `$mosaic5g > ./build_m5g -U`  # SNAP OAI-UE
  - Ref: [https://snapcraft.io/oai-ue](https://snapcraft.io/oai-ue)
- `$mosaic5g > oai-ue.help`  # check the help

- Two operation mode:
  - RF
  - Emulated/Simulated UE with embedded RRU functions
  - Ref: TBD
OAI-UE.UE and OAI-UE.UESIM

- Get the path to the UE command and config file, and update accordingly
  - oai-ran.ue-cmd-get and oai-ran.ue-conf-get
  - oai-ran.uesim-cmd-get and oai-ran.uesim-conf-get

- Update the command file:
  - Parameters of interests are: 
    -U -C2660000000 -r25 --ue-scan-carrier --ue-txgain 70 --ue-rxgain 80
    - For more details: oai-ue.ue -h or oai-ue.uesim -h

- Setup the OAI UE IP interface
  - sudo oai-ue.oip
  - This will try to insert the OAI UE_IP module, which may fail due to magic version. To proceed, you need to manually compile and insert the module as follows:
    - cd /root/snap/oai-ue/current/ue_ip; make; ./oip add ./ue_ip.ko;
  - This will setup the oip interfaces, which will be later configured by the core network
    - ip link show oipl
  - After the successful attached, check the OAU-UE IP address as follows:
    - ip address show oipl

- Run OAI-UE.UE or OAI-UE.UESIM
  - sudo oai-ue.ue or sudo oai-ue.uesim # run manually
  - sudo oai-ue.ue-start or sudo oai-ue.uesim-start # run as a service

- Check the status
  - oai-ue.ue-status or oai-ue.ue-journal -e
  - oai-ue.uesim-status or oai-ue.uesim-journal -e
RUN oai-ue USIM info

- Check if the pre-generated UE SIM info is valid
  - oai-ue.usim-print
  - oai-ue.nvram-print

- To update the UE SIM info
  - Update the USER and SIM INFO, in particular IMSI, PLMNS, K, OPC, SQN
    - oai-ue.usim.get
    - sudo vim /var/snap/oai-ue/4/usim_default.conf

- Regenerate USIM and NVRAM files
  - sudo oai-ue.usim-gen

- Print again the USIM and NVRAM info

- Provision the HSS DB with this newly added UE
  - oai-cn.hss-add-user IMSI (refer to hss)
Usage: sudo oai-ue or oai-ue.[x] args, where app [x] is ue or uesim
Note: root privilege required
Options:
oai-ue.[x]: run App [x] manually as a process. Option -h for help.
oai-ue.init: initialize oai-ue (after the 1st installation)
oai-ue.[x]-conf-get: get the current oai-ue.[x] configuration file
oai-ue.[x]-conf-list: list the oai-ue.[x] configuration files
oai-ue.[x]-conf-set: set the oai-ue.[x] configuration file
oai-ue.[x]-conf-show: show the path to the oai-ue.[x] configuration file
oai-ue.ue-cmd-set: set the oai-ue configuration file
oai-ue.ue-cmd-get: get the current oai-ue configuration file
oai-ue.ue-cmd-show: show the path to the current oai-ue configuration file
oai-ue.ue-cmd-list: list the oai-ue configuration files
oai-ue.oip: bring up the oai-ue UE IP address
oai-ue available commands (2/2)

- `oai-ue.[x]-journal`: get the oai-ue.[x] daemon logs
- `oai-ue.[x]-restart`: restart the oai-ue.[x] daemon
- `oai-ue.[x]-start`: start the oai-ue.[x] daemon
- `oai-ue.[x]-status`: status the oai-ue.[x] daemon
- `oai-ue.[x]-stop`: stop the oai-ue.[x] daemon

- `oai-ue.usim-gen`: generate usim data file for single or multi UEs
- `oai-ue.usim-print`: print the usim configure data file
- `oai-ue.nvram-print`: print the nvram data file
- `oai-ue.debug`: debug the oai-ue UE with gdb
- `oai-ue.usim-set`: set the USIM configuration file
- `oai-ue.usim-get`: get the current USIM configuration file
- `oai-ue.usim-show`: show the current USIM configuration file
- `oai-ue.usim-list`: list the USIM configuration files

- `oai-ue.help`: print this help
OAI-TRACER Snap

Snap version 1.3 and 1.4
Build OAI-TRACER from Snap

- $mosaic5g > ./build_m5g -T  # snap OAI-TRACER
  - Ref: [https://snapcraft.io/oai-tracer](https://snapcraft.io/oai-tracer)
- $mosaic5g > oai-tracer.help  # check the help
  - Ref: [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/T](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/T)
RUN oai-tracer

- oai-tracer -ip eNB@IP
- Run oai-ran.enb with the following additional options:
  - --T_stdout 0

- For more options:
  - oai-tracer -h
Simple Testing Example
Prepare for Testing

- Open the Wireshark in the PC running EPC and filter S1AP packets
  - You should see the Attach Complete message
    - See pcap file [here](#)
- Set the APN in the phone to that of HSS
  - Default APN for OAI is: oai.ipv4
- Ping the UE from the EPC:
  - `ping 172.16.0.X`
Connect a UE

- Disable/enable airplane mode
- Check if you are connected
- Check the Internet connectivity
- Install nperf application at UE and perform a test
Mosaic5G-Contact

E-mail: contact@mosaic-5g.io
Website: http://mosaic-5g.io
Twitter: @mosaic5g
Linkedin: mosaic-5g