FlexRAN: A Mosaic5G Projects

JOX
(Orchestration & Management)

Store
(Data & Service & App)

FlexRAN
(RAN Platform)

OAI-RAN & OAI-CN
(Infrastructure)

JSlice controller
JCloud controller
JOX plugins
Juju VNFM

Open Data APIs
Monitoring Apps
Control Apps
Analytics Apps
SDK

Virtualization & Slicing
RAN Runtime

Virtualization & Slicing
Edge/CN Controller

LL-MEC
(Edge/CN Platform)

RAN-CU
(Edge Node)

Cloud

Store

LL-MEC

FlexRAN

OpenAirInterface

(C) Mosaic5G
Outline

1. Installation/configuration first, OAI-RAN, OAI-CN, FlexRAN Snaps
2. Absolute basics of FlexRAN (What?)
3. A simple use case of FlexRAN (How?) and show other use-case

- The Tutorial and training materials can be found at
  - http://mosaic-5g.io/resources/flexran_training_bupt_2019.zip
- We follow the tutorials in the Mosaic5G Wiki (restricted access)
- Reference: http://mosaic-5g.io/flexran/
Ubuntu 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
- REFS:
  - https://snapcraft.io/
  - https://tutorials.ubuntu.com/tutorial/basic-snap-usage#0
  - https://tutorials.ubuntu.com/tutorial/advanced-snap-usage#0
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

<table>
<thead>
<tr>
<th>Channels</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>
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](https://gitlab.eurecom.fr/mosaic5g/mosaic5g/wikis/releases)
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-in-one Network Setup
Installation of OAI-CN

- Video-Tutorial: Youtube

- Install OAI-CN as a snap:
  
sudo snap install oai-cn --channel=edge --devmode

- Check the installation:
  
sudo oai-cn.help
Install MySQL and login. If you cannot login, [do this](#).

* Initialize the HSS: `sudo oai-cn.hss-init`
* Get the configuration file: `sudo oai-cn.hss-conf-get`
* In `hss_fd.conf`, change `Identity` to match "<hostname>.openair4G.eur" (use hostname)
* In `hss.conf`, ensure the right MySQL username and password. Set `OPERATOR_key` to "1111..."
* Create certificates: `sudo oai-cn.hss-init`
* Run HSS: `sudo oai-cn.hss`
* The last line should read “Initializing S6a layer: DONE”
Initialize the MME: `sudo oai-cn.mme-init`

Locate configuration files in directory: `sudo oai-cn.mme-conf-get`

In `mme.conf`
- Correct hostname in `HSS_HOSTNAME`
- Edit `GUMMEI_LIST` and `TAI_LIST`
- `NETWORK_INTERFACES`:
  - `MME_IPV4_ADDRESS_FOR_S1_MME` to 127.0.1.10/24
  - `MME_IPV4_ADDRESS_FOR_S11_MME` to 127.0.11.1/8
- `S-GW`:
  - `SGW_IPV4_ADDRESS_FOR_S11` to 127.0.11.2/8

In `mme_fd.conf`:
- Identity needs to match hostname, ConnectPeer maybe too

Start the MME: `sudo oai-cn.mme`

Last line: Peer `<hostname>.openair4G.eur` is now connected...
Initialize the SPGW: `sudo oai-cn.spgw-init`

In `spgw.conf`:
- `SGW_IPV4_ADDRESS_FOR_S11` to `127.0.11.2/8`
- `SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP` to `127.0.1.10/24`
- `PGW_INTERFACE_NAME_FOR_SGI`: the interface to the Internet
- `DEFAULT_DNS_IPV4_ADDRESS`: your DNS

Start the SPGW: `sudo oai-cn.spgw`

Last line: Initializing SPGW-APP task interface: DONE
Installation of OAI-RAN

- Video-Tutorial: [Youtube](https://www.youtube.com)

- Install OAI-RAN as a snap:
  ```
  sudo snap install oai-ran --channel=edge --devmode
  ```

- Check the installation:
  ```
  sudo oai-ran.help
  ```
- Get the configuration file: `sudo oai-ran.enb-conf-get`
- Edit `plmn_list`
- Edit `mme_ip_address`
- Edit `NETWORK_INTERFACES`
- Lower `max_rxgain`
- Set `parallel_config` to `PARALLEL_SINGLE_THREAD`
- Disable `FLEXRAN_ENABLED` (no)
- Possibly lower `downlink_frequency`
- Recommended: `N_RB_DL` to 25
Run OAI

- Start Wireshark on any interface, capture filter port 36412
- Start the RAN `sudo oai-ran.enb`
- Verify that S1SetupRequest is followed by S1SetupResponse (without error . . . )
- Connection of a phone, troubleshooting individually
Overview of FlexRAN

Goals:
- Realtime and flexible RAN control
- Separation of CP&UP
- Programmability of the RAN
Installation of FlexRAN

- Install FlexRAN as a snap:
  
  ```
  sudo snap install flexran --channel=edge --devmode
  ```

- Check the installation:
  
  ```
  sudo flexran.help
  ```

- Enable the controller in OAI’s configuration: `FLEXRAN_ENABLED` to “yes”
1. Start the CN
2. Start FlexRAN: sudo flexran
3. Start OAI
4. Connect a phone

FlexRAN output:

[INFO][FLEXRAN_RTC] - Listening on port 2210 for incoming agent connections
[INFO][FLEXRAN_RTC] - Listening on port 9999 for incoming REST connections
[INFO][RIB] - New agent connection established (agent ID 0), sending hello
[WARN][RIB] - Agent 0 with illegal BS ID 0, assigned BS ID 10000
[INFO][RIB] - Agent 0: hello BS 10000, capabilities [LOPHY,HIPHY,LOMAC,HIMAC,RLC,PDCP,SDAP,RRC]
[INFO][RIB] - New BS 10000, creating RIB entry
[INFO][APP] - Sending 100 ms periodical full stats request to BS 10000
[INFO][RIB] - BS 10000: UE RNTI 10445 activated
[INFO][RIB] - BS 10000: UE RNTI 10445 updated
Get running apps (**capabilities**): in a browser, open `localhost:9999/capabilities`

Get current **statistics**: `localhost:9999/stats`

For scripting purposes: we need **jq** and **curl**:
```
sudo apt-get install jq curl
```

Sometime we might need `-XGET` (default) or `-XPOST` for curl.

**jq** . means “format everything from root”

To get the cell configuration of the first eNB on the commandline
```
curl localhost:9999/stats | jq .eNB_config[0].eNB.cellConfig
```

**Apidoc documentation**: [http://mosaic-5g.io/apidocs/flexran/](http://mosaic-5g.io/apidocs/flexran/)
Use case: RAN slicing

- Use a convenience script to show current slice configuration: `watch_slicing_config.sh` (also located in the FlexRAN sources under `tools/`)

- Use the `slice-lifecycle.sh` script to create a slice, associate the first active phone, and destroy the slice. Via the convenience script, OAI and FlexRAN logging output, observe what happens!

- Try to recreate the scripts behavior! The JSON you need to post is printed in FlexRAN’s logging output if you successfully ran the script:
  - Add/modify a slice: `apidoc ApplySliceConfiguration` (short version)
  - Associate a user to a slice: `apidoc ChangeUeSliceAssociation` (short version)
  - Delete a slice: `apidoc DeleteSlice` (short version)
Use case: Network-controlled handover

- Video: Youtube
- Right: drone application in the store
Example Applications

- Dynamic MME management
- Remote scheduling
- RU management
- RAN-aware video optimization
- Positioning and crowd distribution
- RAN Data Mining and Analytics
- Data-driven RAN Control
- QoS-aware RAN Sharing and Slicing
- Block Chained resource counting
- Spectrum Sharing and management
- Mobility management
- Coordinated Scheduling and RRM
- Interference management
- Dynamic function split
Questions

E-mail: contact@mosaic-5g.io
Website: mosaic-5g.io
Twitter: @mosaic5g
Linkedin: mosaic-5g
Mosaic5G-Contact

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