Experience Deploying a 5G C-RAN Virtualized Experimental Setup using OpenAirInterface

Oumayma NEJI
IBISC Laboratory and Higher Institute of Computer Sciences of Tunis, ISI, Tunisia
oumayma.neji@gmail.com

Nada CHENDEB
Lebanese University
nchendeb@ul.edu.lb

Olfa CHABBOUH
MEDIATRON Laboratory Higher School of Communication of Tunis SupCom, Tunisia
olfa.chabbouh@supcom.tn

Nazim AGOULMINE
IBISC Laboratory Paris Saclay, University of Evry-Val-d'Essonne, France
Nazim.Agoulmine@univ-evry.fr

Sonia BEN REJEB
MEDIATRON Laboratory Higher School of Communication of Tunis SupCom, Tunisia
sonia.benrejeb@supcom.tn

***Abstract*—The 5th generation of Mobile Networks is planned to be launched in 2020 or before. Although many progress have already been done, many challenges are still facing this technology. The OpenAirInterface Software Alliance (OSA) is a non-profit consortium that insures open source software/hardware development for the core network and both access network and user equipment of 3rd Generation Partnership Project (3GPP) cellular networks and more particularly 5G cellular stack. The goal of this paper is to describe our experience virtualizing the components of the OpenAirInterface (OAI) software in order to facilitate the fast deployment of an emulated 5G network. In particular, we are interested in the concept of the mobile Cloud which allows to offload part of the UE processing directly on Cloud Radio Access Network (C-RAN). We will present here our advances, results and difficulties.**

Keywords C-RAN, OpenAirInterface, virtualization.

1. INTRODUCTION

Mobile networks are rapidly growing which leads to a costly deployment and maintenance of RAN (Radio Access Network) equipment and software. Virtualization technology has been introduced in the RAN to allow the centralization of the main functions of the RAN equipment called eNodeB (eNB) (i.e. the Long-Term Evolution base station). This new type of RAN is called C-RAN for Cloud Radio Access Network which is the lead path to the 5th generation of mobile networks (5G).
As presented in the figure 1, the concept of C-RAN aims to split the functions of the eNB into two parts: the first part known as a Remote Radio Head (RRH), is devoted to physical radio frequency access while the second part, known as Base Band Unit (BBU), is devoted to the digital and the base band signal processing.
The RRH is deployed in the territory (on the roofs), while the BBU is virtualized, centralized and remotely connected to the RRH using a Common Public Radio Interface (CPRI).

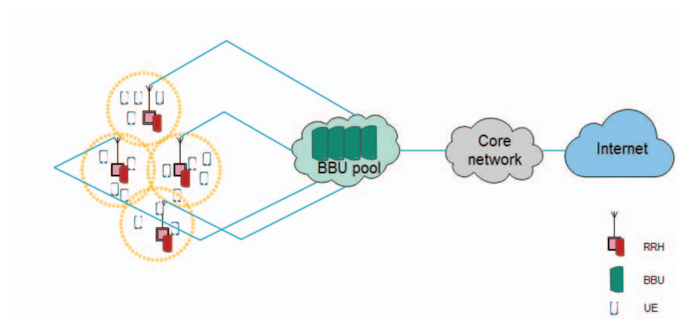

Figure 1. C-RAN architecture

The basic principle of C-RAN is to pool the BBUs of several base stations in a centralized cloud computing infrastructure, called BBU pool, enabling improved network performance, energy efficiency, flexibility and reduced operating costs. Based on the system load, the BBUs are dynamically allocated to RRHs.
Before deploying the network, the operator needs to evaluate the expected performances of this one, which requires a strict, exact and detailed evaluation and a real-world validation. Although networks simulation software have progressed, they haven't yet been able to reproduce the environment correctly due to its complexity and

978-1-5090-5007-9/17/$31.00 ©2017 IEEE

unpredictability.

The OAI-Sim for 5G software is both an emulator and a simulator of such an environment and represents an excellent opportunity for academia (and also industry) for testing existing protocol as well as experimenting new ones.

In this work, our first objective is to deploy this software in the lab and configure a virtualized C-RAN to evaluate its performances in different realistic scenarios. The second objective is to virtualize the different components of the OAI architecture in order to permit a flexible deployment of any configuration in a cloud computing infrastructure.

The remainder of this paper is organized as follows. The section II gives an overview of the OAI emulation platform, its structure as well as the virtualization of its components. Section III, explains the deployment and the configuration of our OAI test-bed. In Section IV, the performed tests and proposed scenarios are explained. Finally, conclusion and perspectives for this work are given in section V.

2. OAI EMULATION PLATFORM

The OAI LTE emulation platform is a set of software provided as open source by the OpenAirInterface Software Alliance (created by EURECOM). *"The software is written in C and runs on the Linux operating system. OAI is the first open-source and real-time software implementing the LTE (Long Term Evolution) network with the complete protocol stack of 3GPP standard for the Evolved Universal Terrestrial Radio Access (E-UTRAN) and Evolved Packet Core (EPC)"*. [1]

2.1. Platform Software Structure

The source code of OAI platform is distributed in the form of three different folders as follows :

Openair1 folder includes the physical layer and the definition of its parameters. The principal function of the included sofwtares is to do the base band signal treatments. With the software included in this folder, an OAI user can emulate a real wireless system.

Openair2 folder contains all the software that permits to implement the layer2 stack (Medium Access Control (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP) and the Radio Resource Control layer (RRC)). Openair2 folder permits to implement all medium access protocols.

Openair3 folder contains all the modules related to the layer3 and use Internet Protocol (IP) [2].

2.2. Virtualization

To deploy an LTE cellular network, OAI offers two modules: (1) The Core Network (CN) part and (2) the Access Network (AN) one. The EPC software is called openair-cn and contains the Mobility Management Entity (MME), the Home Subscriber Server (HSS), the Serving Gateway (S-GW) and the Packet Data Network Gateway (P-GW). The Access Network software is known as openairinterface5G and represents the eNB part [3].

Different deployment configurations can be settled up using OAI. For example, it is possible to deploy all the components in the same machine (i.e. UE, eNB, and all the EPC components) . However, this configuration is not recommended due to the potential conflicts that may raise between packages and the kernel. In addition, this configuration requires a very powerful processing machine (at least 8 cores and 16 GB RAM). Another more recommended configuration is to run each component in a different machine but this configuration requires a lot of equipment [4]. We have followed an in between solution that consists on deploying the eNB and the EPC on different interconnected hosts. The protocol stack of the LTE network implemented in OAI is presented in the figure 2 [1]

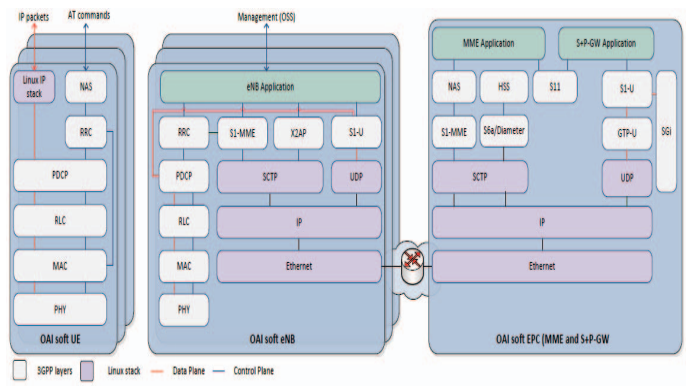

Figure 2. OpenAirInterface LTE software stack [1]

2.3. Deployed Softwares

"As previously introduced, the LTE emulation platform provided by OAI implements the Physical, MAC, RLC, PDCP, RRC and provides an IPv4/IPv6 network device interface under Linux for the User Equipment (UE) and the eNB. For the core network side, OAI includes a full implementation of MME, HSS, S-GW and P-GW. In fact, it implements the authentication and the attach of the UE, the radio bearer establishment, the Non-access stratum (NAS) integrity and encryption, etc". [1], [2]

"Besides, with OAI LTE emulation platform, there is no need to an external S-GW or P-GW to have an access to the IP network. As shown in the figure 2, the xt_GTPUSP kernel module should be installed in the P-GW to manage the GTP User Plan (GTP-U) header and this is to have an OAI eNB application compliant with 3GPP." [1], [5] The GTP-U protocol is used to transport user data in the core network and between the RAN and the core network.

OAI components can be tested collectively or individually with commercial LTE components. The different following configurations can be set up to achieve different combination of OAI platform components with commercial LTE ones:

- *"**OAI** UE __ **OAI** eNB + **OAI** EPC*
- ***OAI** UE __ **OAI** eNB + Commercial EPC*
- ***OAI** UE __ Commercial eNB + **OAI** EPC*
- ***OAI** UE __ Commercial eNB + Commercial EPC*
- *Commercial UE __ Commercial eNB + **OAI** EPC*
- *Commercial UE __ **OAI** eNB + Commercial EPC*
- *Commercial UE __ **OAI** eNB + **OAI** EPC"* [1], [6]

2.4. Hardware

OAI permits to do experiments with real UE. For that, EURECOM has developed an experimental radio card (ExpressMIMO2) with a continuous Radio Frequency (RF) coverage from 250 MHz to 3.8 GHz and channels up to 20MHz of bandwidth. It is also possible to use a commercial card called USRP B210 software radio card which covers a RF spectrum from 70 MHz to 6 GHz with a Full duplex, MIMO (2 Tx and 2 Rx) operation with up to 56 MHz of real-time bandwidth [7], [8].

3. OAI DEPLOYMENT

In this work, we have used the oaisim-rru branch of the openairinterface5g to extract the modules to deploy the RRH/UE and the BBU. Meanwhile, we have used the openair-cn branch to extract the core network module. We have virtualized the different modules on different VMs (Virtual Machines) using the Oracle Virtual Box.

Unfortunately, due to budget restriction, we were not able to provide the RF card (ExpressMIMO2 or USRP B210). Therefore, we decided to use OAI in a fully simulated mode of the UE, i.e. using oaisim-rru branch. The modules contained in this branch are supposed to permit to do all simulations without a RF card or real UE.

We have created the eNB VM using ubuntu 14.04.3 with a low-latency kernel 3.19, 4 CPUs and a 4GB RAM. The EPC VM was created and configured with ubuntu 14.04.3 with generic kernel 4.7.1, 4 CPUs and a 4GB RAM.

Our network configuration is represented in the figure 3.

As it can be seen in the figure 3, oaisim-rru branch provides an integrated module for the UE and the RRH. Since the RRH and the BBU are on the same host, they are connected via the loopback interface, while the BBU is connected via its Ethernet interface 0 (192.168.25.102) to the EPC.

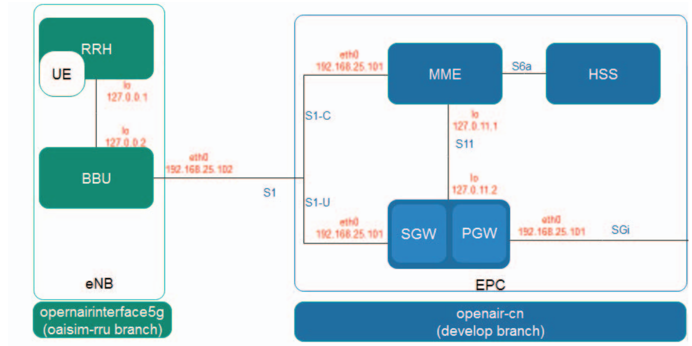

Figure 3. Logical components and interfaces in our setup

The DL frequency used is 2685 Mhz and the UL frequency is 2565 Mhz on the Evolved Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access (E-UTRA) band 7 and using a Frequency Division Duplex (FDD) mode.

The configuration files of the eNB part are the following:

- rru.band7.tm1.if4p5.50PRB.oaisim.conf (for the RRH)
- rcc.band7.tm1.if4p5.50PRB.lo.conf (for the BBU)

For the USIM (Universal Subscriber Identity Module) card parameters and permanent data of the UE, they can be edited in the ue_eurecom_test_sfr.conf file. These information are also stored in the HSS database and are used to verify the identity of the UE when it tries to connect to the MME.

In our setup, we have used the following information to identify the UE:

- **Ki Value**: fec86ba6eb707ed08905757b1bb44b8f
- **Operator key**: dbc4e25644591a59aa700857a2bf095b
- **IMSI**: 208930100001111
- **MSISDN**: 33611123456
- **IMEI**: 356113022094149

When a UE tries to connect to the MME, the HSS verifies the security information by calculating automatically the operator key using the Ki Value of the UE. If the calculated operated key is identical to the one provided by the UE, then the UE is authorized to be attached to the MME otherwise it is rejected.

For the EPC part, the MME is connected to the BBU via its Ethernet interface 0 (192.168.25.101), same for the SPGW(S-GW and P-GW).

The MME, HSS and SPGW are connected to each other via their loopback interface since they are on the same machine.

To access the internet, the P-GW assigns to the UE an IP address from a pool of addresses specified in the spgw.conf configuration file. The UE can then have access to internet through its virtual interface called oip (OpenAirInterface IP).
To run the RRH/UE we have used the following command:
./oaisim -O $OPENAIR_DIR/targets/
PROJECTS/GENERIC-LTE-EPC
/CONF/rru.band7.tm1.if4p5.50PRB.oaisim.conf –xforms

To run the BBU we have used the following command:
./lte-softmodem -O $OPENAIR_DIR/targets/
PROJECTS/GENERIC-LTE-
EPC/CONF/rcc.band7.tm1.if4p5.50PRB.lo.conf

To run the EPC part, we proceed as follows:

- ./run_hss (to run the HSS)
- ./run_mme (to run the MME)
- ./run_spgw (to run the SPGW)

4. TESTS AND SCENARIOS

4.1. Initial Tests Results

The figure 4 shows that the initial security activation is complete. First, the eNB sends to the UE a Security-ModeCommand message on the Downlink Control Channel (DCCH). The UE decodes the received message from the RRH, configures the security mode for PDCP and then sends a SecurityModeComplete message to the RRH on the DCCH. Finally the RRH receives it and configures the PDCP. Once the initial security activated, the UE is then attached to the MME, which is represented by the information shown in the figure 5 and the BBU is successfully associated to the MME as shown in the figure 6.

Figure 4. Security mode command complete

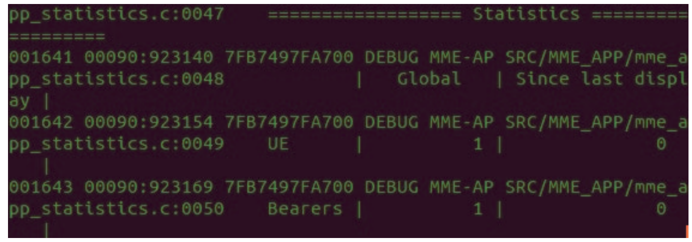

Figure 5. Successful attach of the UE to the MME

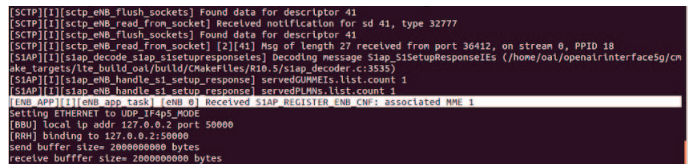

Figure 6. Successful attach of the MME to the BBU

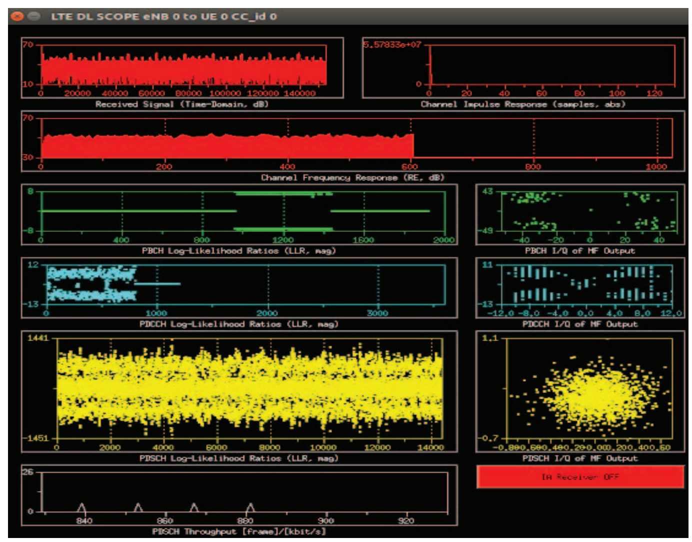

Figure 7. DL scope fromthe eNB (RRH/BBU) to the UE 0

In the figure 7, a snapshot of the the downlink scope of the control information is displayed. It presents the data transmitted on the current subframe and the information about the resources assigned to the UE for the uplink. These control information are carried out by the Physical Downlink Control Channel(PDCCH). More precisely, this channel carries the Downlink Control Information (DCI) that involves resource assignments for the UE.

On the same figure, the user downlink traffic transported by the Physical Downlink Shared Channel (PDSCH) is also presented. This channel carries also the Downlink Shared Channel (DL-SCH) information.

On the figure 8, we can see that the UE managed to access the internet. This is proved by the successful ping to ”google.com” through the oip1 interface of the UE which has been assigned the 172.16.0.2 address by the P-GW.

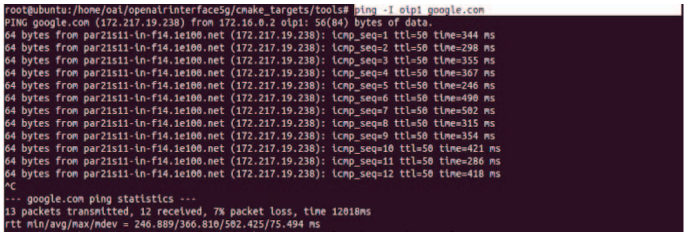

Figure 8. successful ping from the UE to ”google.com” through its oip1 interface

We have also pinged the UE from the P-GW through its gtp0 interface (172.16.0.1), which is shown on the figure 9.

Figure 9. successful ping from the P-GW to the UE through the gtp0 interface

Unfortunately, we were not been able yet to change the behavior of the UE. When the UE is attached to the RRH/BBU, the uplink traffic from the UE to the eNB is not visible in the uplink scope. This may come from the implementation of the UE/RRH in the oasim-rru which is not using the physical RF card and therefore maybe the complete UE physical layer is not implemented in this case.

4.2. Specified Scenarios

In this section, We present a set of scenarios we have specified for the experiments (that are possible with OAI but need to be tested with oaisim-rru):

- **First scenario:**
  - 2 UEs connecting to each other. (to analyze the behavior of each one of them)
  - 2 UEs accessing the internet at the same time
- **Second scenario**
  - 1 UE and two RRHs (to see to which RRH the UE is going to be connected)
    We can have also a pool of two BBUs in this scenario, so we can see which BBU is going to be activated when the UE tries to connect.

The two scenarios are illustrated by the figure 10.

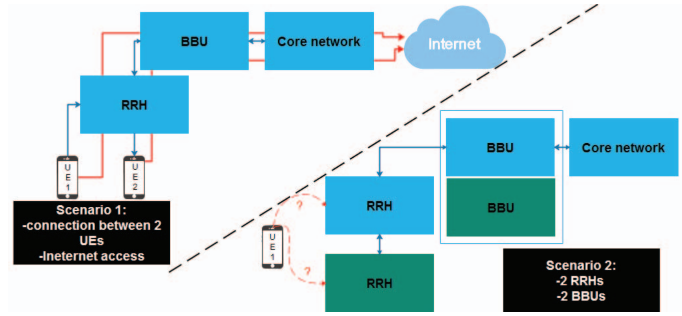

Figure 10. 2 scenarios for further work

5. Conclusion and perspectives

In this paper we have deployed a virtualized experimental setup of a C-RAN using the OAI platform. We have first introduced the context of our work which is the C-RAN and we presented an overview of the OAI emulation platform, its structure, components, etc. Then we have explained the deployment and the configuration of our C-RAN architecture using OAI. Finally, some scenarios have been presented to be tested in the future with the deployed system. In future work, we plan to stabilize the execution of the system and run different scenarios related to C-RAN, UE offloading and Machine to Machine (M2M) communications that are essential in future 5G networks.

Acknowledgments

We would like to thank the EURECOM Team, for their help throughout the deployment of our system.

References

[1] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Ghaddab, ”OpenAirInterface 4G: an open LTE network in a PC”, Eurecom

[2] R. Wang, Y. Peng, H. Qu, W. Li, H. Zhao and B. Wu, ”OpenAirInterface-An Effective Emulation Platform for LTE and LTE-Advanced”, Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on, August 2014

[3] The OpenAirInterface Platform,[Online], Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/home

[4] The OpenAirInterface Platform,”How to Connect OAI eNB (USRP B210) with COTS UE”, [Online], Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectCOTSUEwithOAIeNBNew

[5] C. Yeoh, M. Mokhtar, A. Rahman and A. Samingan, ”Performance Study of LTE Experimental Testbed using OpenAirInterface”, Advanced Communication Technology (ICACT), 2016 18th International Conference on, March 2016

[6] N. Nikaein, M. Marinay, S. Manickamy, A. Dawsony, R. Knopp and C. Bonnet, ”OpenAirInterface: A Flexible Platform for 5G Research”, Eurecom

[7] The OpenAirInterface Platform,[Online], Available:http://openairinterface.eurecom.fr/expressmimo2

[8] The ettus research platform, [Online], Available: https://www.ettus.com/product/details/UB210-KIT