Demo – Closer to Cloud-RAN: RAN as a Service

Navid Nikaein[†], Raymond Knopp[†], Lionel Gauthier[†]
Eryk Schiller[*], Torsten Braun[*], Dominique Pichon[§],
Christian Bonnet[†], Florian Kaltenberger[†], and Dominique Nussbaum[†]
[†] Eurecom 06410 Biot Sophia-Antipolis, France firstname.name@eurecom.fr
[*] University of Bern lastname@iam.unibe.ch
[§] Orange Labs dominique.pichon@orange.com

ABSTRACT

Commoditization and virtualization of wireless networks are changing the economics of mobile networks to help network providers (e.g., MNO, MVNO) move from proprietary and bespoke hardware and software platforms toward an open, cost-effective, and flexible cellular ecosystem. In addition, rich and innovative local services can be efficiently created through cloudification by leveraging the existing infrastructure. In this work, we present RANaaS, which is a cloudified radio access network delivered as a service. RANaaS provides the service life-cycle of an on-demand, elastic, and pay as you go 3GPP RAN instantiated on top of the cloud infrastructure. We demonstrate an example of real-time cloudified LTE network deployment using the OpenAirInterface LTE implementation and OpenStack running on commodity hardware as well as the flexibility and performance of the platform developed.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Wireless Communication System.

Keywords

OpenAirInterface, LTE/LTE-A, Cloud Computing, Cloud-RAN, Virtualization, RANaaS.

1. INTRODUCTION

In the last few decades, radio access networks (RANs) have significantly evolved from analog to digital signal processing units, where hardware components are often replaced with flexible and reusable software-defined functions. In a pure software-defined radio (SDR) system, the entire radio function runs on a general-propose processor (GPP) and only requires analog-to-digital and digital-to-analog conversions, power amplifiers, and antennas, whereas in typical cases, the system is based upon a programmable dedicated hardware (e.g. ASIC, ASIP, or DSP) and associated control software. Thus, the flexibility offered by a pure SDR improves service life-cycle and cross-platform portability at the cost of lower power and computational efficiency (i.e. ASIC: 1x, DSP: 10x, GPP: 100x).

Virtual RAN extends this flexibility through abstraction (or virtualization) of the execution environment. Consequently, radio functions become a general-purpose application that operates on top of a virtualized environment and interacts with physical resources either directly or through a full or partial hardware emulation layer. The resulted virtualized software radio application can be delivered as a service and managed through a cloud controller [1]. This changes the economics of mobile networks towards a cheap and easy to manage software platforms. Furthermore, cloud environment enables the creation of new services, such as RAN as a service (RANaaS) [2, 3], and more generally, network as a service (NaaS), such as LTEaaS, associated with the cloud RAN (C-RAN) [4–6]. C-RAN systems replace traditional base stations with distributed (passive) radio elements connected to a centralized baseband processing pool. Decoupling of the radio elements from the processing serves two main purposes. Centralized processing has the benefit of cost reduction due to fewer number of sites, easy software upgrade, performance improvement with coordinated multi-cell signal processing. Also, the remote radio heads have a much smaller footprint than a base station with on site processing, allowing for simpler and cost-effective network densification. C-RAN can be realized in two main steps, namely:

- **Commoditization and Softwarization:** refers to a software implementation of network functions on top of GPPs with no or little dependency on a dedicated hardware (e.g. DSP, FPGA, or ASIC). In addition, the programmability in RF domain can be achieved with the Field Programmable Radio Frequency (FPRF) technology.
- **Virtualization and Cloudification:** refers to execution of network functions on top of virtualized (and shared) computing, storage, and networking resources controlled by a cloud OS. I/O resources can be shared among multiple physical servers, and in particular that of radio front-end through multi root I/O virtualization (MR-IOV).

Many architectures have been proposed to support RANaaS ranging from a partially accelerated to a fully software-defined RAN. Recent efforts have shown the feasibility and efficiency of a full software implementation of the LTE RAN functions over GPPs. Two software implementations of the fully functional LTE/LTE-A already exist, namely Amarisoft[1] and OpenAirInterface[2] delivered as open-source. A full GPP approach to RAN brings the cloud and virtualization even closer to the wireless world allowing us to build a cloud-native RANaaS along with the following principles [1, 7]:

- **Mircoservice Architecture:** breaks down the network into a set of horizontal functions that can be combined together, assigned with target performance parameters, mapped onto the infrastructure resources (physical or virtual), and finally delivered as a service.[3]

[1] www.amarisoft.com
[2] www.openairinterface.org
[3] Microservice architecture is in opposition to the so-called "monolithic" architecture, where all functionality is offered by a single logical executable, see http://martinfowler.com/articles/microservices.html. It has to be noted that the microservice architecture supports the ETSI NFV architecture [8], in which each VNF can be seen as a service.

- **Scalability:** monitors the RAN events (e.g. workload variations, optimization, relocation, or upgrade) and automatically adds/removes resources without any degradations in the required network performance (scale out/in).
- **Reliability:** shares the RAN contexts across multiple RAN services while keeping the required redundancy (making RAN stateless), and distributes the loads among them.
- **Placement:** optimizes the cost and/or performance by locating the RAN services at the specific geographic area subjected to performance, cost, and availability of the RF front-end and cloud resources.
- **Real-time Service:** offers a direct access to real-time radio information (e.g. radio status, statistics) for low-latency and high-bandwidth service deployed at the network edge [9].

In this work, we present RANaaS describing the service life-cycle of an on-demand, elastic, and pay as you go 3GPP RAN instantiated on top of the cloud infrastructure. The RANaaS service life-cycle management is a process of network design, deployment, resource provisioning, runtime management, and disposal that allows to rapidly architect, instantiate, and reconfigure the network components and their associated services. The proof-of-concept demonstrator is built upon the OpenAirInterface LTE software implementation, low latency Linux kernel, Linux containers, OpenStack, Heat orchestrater, and Open vSwitch as well as commodity PCs and National Instrument/Ettus USRP B210 RF front-end. It proves the feasibility of a real-time cloudified LTE network deployment using commodity hardware by providing a multimedia service to a commercial LTE-compatible user equipment.

2. FEASIBILITY STUDY OF CLOUD-RAN

To demonstrate the feasibility of C-RAN, we rely on the OpenAirInterface (OAI) LTE implementation upon which different C-RAN testbeds can be built [10]. In this section, we present the profiling results of OAI base band unit under various physical resource block (PRB) allocations, modulation and coding schemes (MCS), and for different virtualization technologies [11]. The experiments are performed using a single user with no mobility, in FDD SISO mode with AWGN channel, and full buffer traffic.

Fig. 1 compares the baseband processing time for a GPP platform with different virtualized environments, namely Linux Containers (LXC), Docker, and KVM, on the SandyBridge architecture (3.2 GHz). It can be observed that processing load is mainly dominated by the uplink and increases with growing PRBs and MCSs. Furthermore, the ratio and variation of downlink processing load to that of uplink also grows with the increase of PRB and MCS. While on average processing times are very close for all the considered virtualization environments, both GPP and LXC have slightly lower variations than that of DOCKER and KVM, especially when PRB and MCS increase. The results also reveals how the computing resources have to be provisioned to meet the real-time requirements.

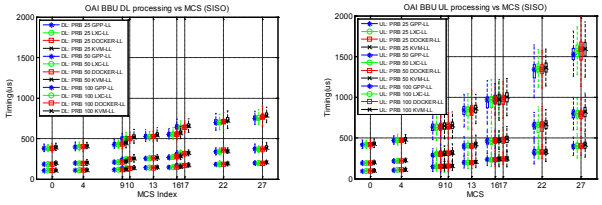

Figure 1: BBU processing time in downlink (left) and uplink(right) for different virtualized environments.

By analyzing processing time for a 1 ms LTE sub-frame, the main conclusion for the considered reference setup (FDD, 20 MHz, SISO, AWGN) is that 2 cores at 3 GHz are needed to handle the total processing of an eNB. One processor core for the receiver processing assuming 16-QAM on the uplink and approximately 1 core for the transmitter processing assuming 64-QAM on the downlink are required to meet the HARQ deadlines for a fully loaded system. With the AVX2 optimizations, the computational efficiency is expected to double and thus a full software solution would fit with an average of 1 x86 core per eNB.

Comparing results for different virtualization environments, we can conclude that containers (LXC and Docker) offer near bare metal runtime performance, while preserving the benefits of virtual machines in terms of flexibility, fast deployment, and migration. Due to the fact that containers are built upon modern kernel features such as `cgroups`, `namespace`, `chroot`, they share the host kernel and can benefit from the host scheduler, which is a key to meet real-time deadlines. This makes containers a cost-effective solution without compromising the performance.

3. C-RAN PROTOTYPE

Fig. 2 presents a proof-of-concept prototype of the RANaaS including the evolved packet core (EPC) and home subscriber server (HSS) services. The prototype has three top-level components, namely a web service, OpenStack, and a Heat stack. The web service features a user interface (UI) for network providers (e.g. MNO, MVNO) to manage their services. A service manager (SO) provides supporting services for the UI and requests the creation of the service from the service orchestrater (SO). SO is in charge of end-to-end life-cycle management of the RANaaS instance through an interaction with Heat. Typically, the OpenStack cloud manages large pools of computing, storage, and networking resources throughout a local *nano data-center*. Our OpenStack worker is deployed on top of Ubuntu 14.04 with the low latency kernel (3.17). To meet the strict timing requirements of RAN, the newly introduced SCHED_DEADLINE scheduler is used, which preempts the kernel to allocate the requested runtime (i.e. CPU time) at each period to meet requested deadlines. The OpenStack installation includes Heat orchestrater whose mission is to create a human- and machine-accessible service for managing the entire life-cycle of the virtual infrastructure and applications within OpenStack. Heat implements an orchestration engine to manage multiple composite cloud applications described in a form of text-based templates, called Heat Orchestration Templates (HoTs) and organized as the Heat stack, which is a stack of virtualized entities (e.g. network, LXCs). The Heat Template specifies LTE network elements with required networking wrapped up for a particular (business) domain. Thus, HoT manages the service instantiation of each LTE network function provided by OAI with desired granularity. The LTE network functions such as eNB, EPC, HSS are automatically programmed through the image flavor and meta-data provided to the LXC-based VMs.

4. DEMO DESCRIPTION

The considered demonstration scenario is depicted in Fig. 3, and consists of 1 commercial LTE-enabled smart-phone (Samsung Galaxy S5), 1 Intel-based mini-PC (Intel i7 at 3.2 GHz, 8 GB RAM, 250 GB SSD) running 1 OAI soft eNB, 1 OAI soft EPC, and 1 OAI HSS as a Heat stack under the control of OpenStack, National Instrument/Ettus USRP B210 RF front-end, and 1 Faraday cage. Various setups are possible ranging from an all-in-one to all in a physically separated entities, which are deployment-specific. For the demo, we plan to demonstrate an all-in-one setup, in which the

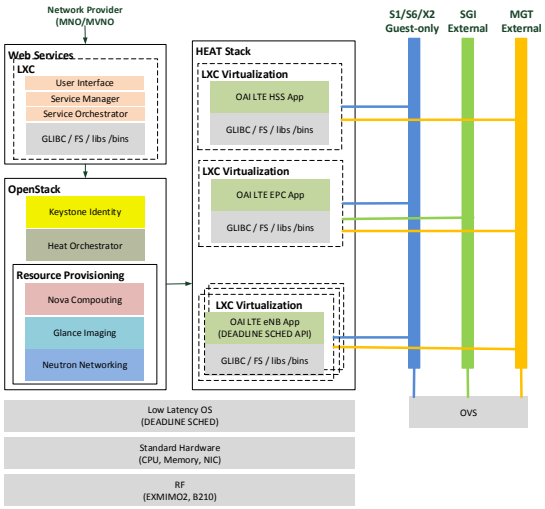

Figure 2: RANaaS prototype.

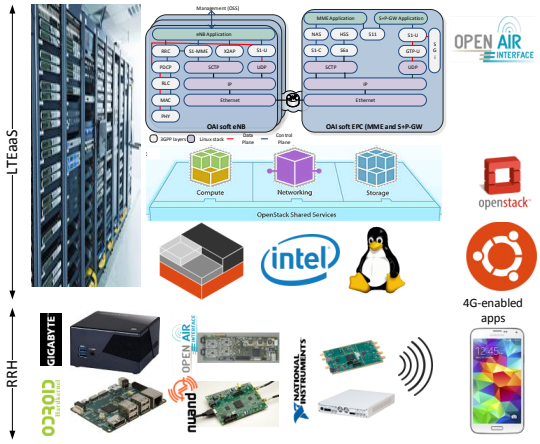

Figure 3: Hardware and software components

LTE eNB, EPC, and HSS functions are performed inside the same cloud infrastructure.

The demonstration will be deployed in FDD 2x2 MIMO mode with 5 MHz channel bandwidth. The target frequencies will be band 7 (Europe) in a controlled indoor radio environment. In the proposed demonstration, we will assess the following objectives

- feasibility of real-time C-RAN under different conditions,
- ease of deployment and service life cycle management,
- rapid service provisioning at the network edge.

The demonstration will be presented live and obtained results will be gathered on-the-fly. We will discuss the critical issues of C-RAN, show the life-cycle management procedure, and assess the performance of the network.

5. LESSON LEARNT

The experiments allowed us to closely observe and evaluate the behavior of a cloudified LTE network under different conditions. Therefore, based on the analysis of the results, a set of high level conclusions can be drawn. FDD LTE HARQ requires a round trip time (RTT) of 8 ms that imposes an upper-bound of less than 3 ms for the eNB TX/RX processing. Failing to meet such a deadline has a serious impact on the user performance. A virtualized execution environment of the BBU pool must provide the required runtime within the requested deadline. Containers proved to be more adequate for GPP RAN as they offer near-bare metal performance and provide direct access to the RF hardware. Virtual machines, in particular KVM, also provide very good performance, but require low latency mechanisms to access (virtualized) I/O resources (passthrough). In the case of containers, real-time or low latency kernels are required on the host. In the case of full virtualization (e.g., KVM), both the hypervisor and the guest OS must support real time/low latency tasks (different techniques are required for hypervisors type 1 and 2).

In addition, the cloudified network has to natively support scale in/out for resource provisioning and sharing across multiple instances as well as load balancing to deal with cell load variations. For this purpose, the network, when instructed by the service orchestrator, has to adjust the number of attached terminals (e.g. trigger handover or detach users) and/or to limit the maximum throughput (e.g., MCS, transmission mode) to align with the available resources (processing, storage, and networking). In terms of reliability, the terminal and network contexts have to be shared among different replicated network instances.

Acknowledgement

The research and development leading to these results has received funding from the European Research Council under the European Community Seventh Framework Programme (FP7/2014- 2017) grant agreement 318109 MCN and 612050 FLEX project.

6. REFERENCES

[1] Bill Wilder. *Cloud Architecture Patterns*. O'Reilly, 2012.

[2] MCN. Mobile cloud networking project, `http://www.mobile-cloud-networking.eu`.

[3] iJoin. Interworking and joint design of an open access and backhaul network architecture for small cells based on cloud networks, `http://www.ict-ijoin.eu/`.

[4] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi. Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development*, 2010.

[5] China Mobile Research Institute. C-ran the road towards green ran, 2013. White paper, v3.0.

[6] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud ran for mobile networks - a technology overview. *Communications Surveys Tutorials, IEEE*, 2014.

[7] MCN D2.5. Final overall architecture definition. Technical report, 2015.

[8] ETSI. Network functions virtualisation (nfv), white paper, 2014.

[9] Milan Patel, Jerome Joubert, Julian Roldan Ramos, Nurit Sprecher, Sadayuki Abeta, and Adrian Neal. Mobile-edge computing. *ETSI*, 2014.

[10] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Gaddab. Demo: Openairinterface: an open lte network in a pc. In *Mobicom*, 2014.

[11] I. Alyafawi, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, and N. Nikaein. Critical issues of centralized and cloudified lte fdd radio access networks. In *ICC*, 2015.