Network Slicing Orchestration of IoT-BeC^3 applications and eMBB services in C-RAN

Salvatore Costanzo, Sylvain Cherrier and Rami Langar
University Paris-Est, LIGM-CNRS UMR 8049, UPEM, F-77420, Marne-la-Vallée, France
Emails: salvatore.costanzo@u-pem.fr, sylvain.cherrier@u-pem.fr, rami.langar@u-pem.fr

***Abstract*—In this demo, we present a Cloud Radio Access Network (C-RAN) prototype based on the Open Air Interface (OAI) platform, which enables efficient coexistence of Internet of Things (IoT) and Enhanced Mobile Broadband (eMBB) slices, sharing the same Radio Access Network (RAN).**

Our prototype aims at efficiently distributing the spectrum resources among multiple slices, while considering the inputs from a northbound Software Defined Network (SDN) application. By using real smartphones and IoT devices orchestrated by the BeC^3 (Behaviour Crowd Centric Composition) framework, we run experiments on stage to validate the feasibility of our prototype in configuring IoT and eMBB slices in real-time, while considering triggering events generated by the IoT network.

Keywords: *5G, C-RAN, Network Slicing, OAI, BeC^3.*

I. Introduction

The proliferation of Internet of Things (IoT) devices, requiring ubiquitous network access, brought new challenges for the mobile Radio Access Network (RAN). In fact, in several use cases, IoT traffic profiles are characterized by a multitude of short and bursty sessions, which may impact the performance of other mobile users, sharing the same RAN. To this end, the upcoming Fifth Generation (5G) mobile system has been designed with the aim to enable efficient isolation among heterogeneous services, by running them in independent logical networks, referred to as slices, on a common shared physical network infrastructure. According to 5G-PPP [1], the 5G network should be able to support up to three slice categories, denoted as: i) Enhanced Mobile Broadband (eMBB), ii) Massive Machine-Type communications (mMTC) and iii) Ultra-Reliable Low Latency communications (uRLLC). Accordingly, there is an increasing demand from the research community to have access to 5G testbed facilities, capable of enabling rapid proof-of-concept designs of Network Slicing solutions.

In this context, we present a Software Defined Network (SDN)-based prototype, which enables efficient coexistence of eMBB and IoT slices in the Cloud Radio Access Network (C-RAN) [2], that is considered as the reference architecture for 5G. In our demonstration, we will show the integration of our first C-RAN Network Slicing prototype, presented in [3], with an IoT framework, named Behaviour Crowd Centric Composition (BeC^3) [4]. We have designed a northbound SDN application, which enables network slicing orchestration of BeC^3 and eMBB services, while considering their Quality of Service (QoS) requirements. By using real smartphones and both real and virtual IoT devices, we carry out experiments on-stage to validate the feasibility of our prototype in supporting the creation and configuration of network slices on-demand, while considering their performance under different triggering events. The remainder of this paper proceeds as follows. In Section II, we describe our prototype architecture, while in Section III we provide an overview of our demonstration.

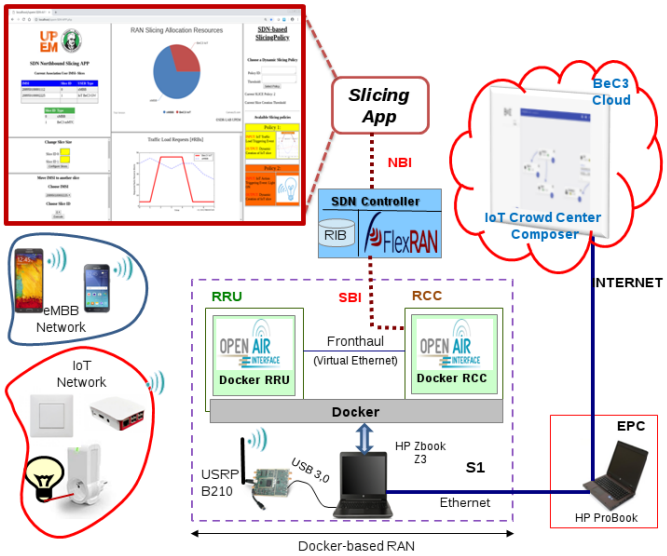

Fig. 1: Prototype Logic Architecture

II. Prototype Description

The logical building blocks of our prototype are depicted in Fig. 1. It consists of a C-RAN infrastructure connected to an IoT framework, named BeC^3. BeC^3 provides a crowd-centric architecture, which allows users to design IoT applications and compose interactions between IoT objects in a user-friendly manner. Note that BeC^3 makes use of D-LITe [4], a lightweight RESTful virtual machine that enables service creation, control and choreography among heterogeneous legacy IoT devices. The C-RAN infrastructure makes use of the Open Air Interface (OAI) software [5] for implementing the network nodes of the RAN and Evolved Packet Core (EPC).

Note that we have developed a virtualized version of the RAN components that relies on the Docker container technology [6]. Accordingly, the RAN consists of the Radio Remote Unit (RRU) node, which includes a remote radio transceiver (USRP B210), that is interconnected via a fronthaul interface

(FH) to the Radio Cloud Center (RCC) node, which carries out the baseband processing functions. The RAN Docker containers are running on an "Ubuntu 14.04" laptop, characterized by "Linux kernel" release 3.19.0-61-*lowlatency SMP PREEMPT*, $i7-4$-core CPU and Random Access Memory (RAM) of 16 GB. The EPC functions are running on a second "Ubuntu 16.04" laptop, characterized by "Linux kernel" release 4.7.0-1, $i5-vPro$, 4-core CPU and RAM of 4 GB. The EPC is interconnected to the RAN via a Gigabit Ethernet interface, refereed to as "S1", while it is connected to Internet through a second Ethernet interface. Note that the BeC^3 Cloud, i.e., the place where the crowd-central composition of IoT services is performed, is connected to the C-RAN architecture via the EPC, using legacy Internet connection.

Our prototype makes use of the SDN FlexRAN controller [7], that enables remote control of the OAI MAC Layer, through a specific southbound interface (SBI), based on Google Protobuf [7]. In this demo, we employ the V2.0 version of FlexRAN, which provides a set of Northbound RESTful APIs, enabling the split of the spectrum resources in different radio slices. On the top of the FlexRAN controller, we have implemented a northbound "Slicing APP" application, which enables the users of our prototype to run network slicing experiments in a user-friendly and abstracted way. Finally, our prototype is completed by an Android smartphone, which streams YouTube videos, and a set of EnOcean IoT devices [8].

III. Demo Overview

The demonstration consists of the following steps. Firstly, we set-up a Mobile Virtual Network Operator (MVNO), which irradiates a 4G carrier, using the C-RAN radio infrastructure. Such a MVNO will provide 4G connectivity to one 4G Android smartphone and one IoT Gateway (GW), equipped with a 4G module. Note that the IoT GW is in turn connected to two Raspberry single-board computers (Pi 3 Model B) through a WiFi link. The first Raspberry computer, referred to as "Watt Emulator", generates IoT watt measurements periodically, while sending them to a remote "BeC^3 Virtual Display" following the procedure described in Fig. 2 (steps 1a-4a). The second Raspberry computer, referred to as "EnOcean GW" handles one EnOcean "Wall Switch", which in turn controls two EnOcean actuators, i.e., two "Smart Plugs" equipped with a Led light. By pushing the "Wall Switch", a radio command is generated and sent to the "BeC^3 Cloud", going firstly through the "EnOcean GW" using the EnOcean Radio Protocol (step 1b), and then through the IoT GW and C-RAN network architecture respectively (steps 2b-4b). Finally, the BeC^3 Cloud decodes the received radio command and sends back an instruction to the actuators (steps 5b-8b), e.g., "Turn ON/OFF" the Led light.

By means of our SDN APP (whose GUI is depicted in Fig. 1), we can (re-)allocate radio resources to each slice employing either a static or a dynamic approach. Specifically, we firstly consider a baseline (BL) scenario wherein both eMBB and IoT devices are served by the same slice and then a second dynamic scenario (DS), wherein upon a triggering

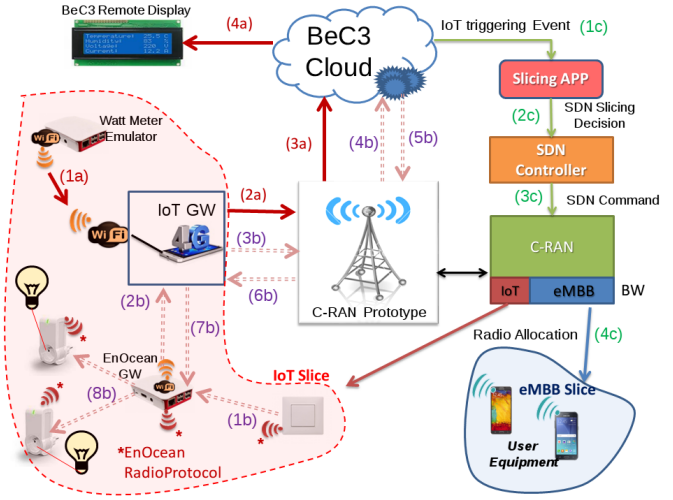

Fig. 2: Resource Slicing Allocation Process

event (step 1c), e.g., an IoT peak session, the SDN Controller automatically instantiates a new slice (IoT slice), while moving the IoT users to that slice (steps 2c-3c). Finally, a number of radio resource blocks (RBs) is automatically allocated to the new IoT slice (steps 4c). By varying the traffic load generated by the IoT devices, we will evaluate the performances of both slices under the "BL" and "DS" scenario, respectively. Note that an IoT peak session can be generated from continual pressing of the "Wall Switch" or by employing an appropriate bursty traffic profile at the "Watt Meter" emulator.

Moreover, the "SDN Slicing APP" can be triggered by a specific state of the IoT devices. For instance, we will show how the state "ON" of a virtual "Alarm" object can trigger the creation of an IoT slice in real-time, while triggering the destruction of such a slice when its state is set to "OFF". Note that the "SDN Slicing APP"can easily handle a different scenario, wherein higher priority is given to the eMBB traffic over delay-tolerant IoT traffic.

Note that a 90 second video of our demonstration has been made available at [9], while a more detailed version can be found at [10].

Acknowledgment

This work was supported by the FUI "SCORPION" project (Grant no. 17/00464) and the CNRS "PRESS" project (Grant no. 239953).

References

[1] "*5G PPP 5G Architecture*", White Paper, version 2.0, Dec 2017.
[2] China Mobile Research Institute, "*C-RAN The Road Towards Green RAN*", White Paper, Oct. 2011.
[3] S. Costanzo, et al., "*A Network Slicing Prototype for a Flexible Cloud Radio Access Network*", 15th IEEE Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, Jan. 2018.
[4] "*BeC^3: Behaviour Crowd Centric Composition for IoT applications*", available online at http://bec3.com/en/
[5] http://www.openairinterface.org/
[6] http://www.docker.com
[7] "*FlexRAN Platform*", available online at: http://mosaic-5g.io/flexran/
[8] https://www.enocean-alliance.org/
[9] https://youtu.be/WikBAh_qJZ8
[10] "*DEMO: C-RAN Network Slicing Prototype for IoT and eMBB services*", available online at: https://sdr-lab.u-pem.fr/slicing-eMBB-IoT.html