Implementation of an SDN-Enabled 5G Experimental Platform for Core and Radio Access Network Support

Kostas Ramantas[1], Elli Kartsakli[1], Mikel Irazabal[1],
Angelos Antonopoulos[2(✉)], and Christos Verikoukis[2]

[1] Iquadrat Informatica S.L., Barcelona, Spain
{kramantas,ellik,mirazabal}@iquadrat.com
[2] Telecommunications Technological Centre of Catalonia, Barcelona, Spain
{aantonopoulos,cveri}@cttc.es

Abstract. Software Defined Networking (SDN) and Network Function Virtualization (NFV) are very promising technologies for future 5G mobile networks, enabling network operators to enhance the flexibility, efficiency and ease of management of their networks. Although SDN is typically associated with wired networks, it is lately making steady advances in cellular networks. In this paper, we present the architecture of an SDN-enabled 5G experimental platform that employs SDN to support both the core and the radio access network.

Keywords: SDN · 5G · Orchestration · Testbed

1 Introduction

Software Defined Networking (SDN) and Network Function Virtualization (NFV) have been envisioned as promising technologies to increase the efficiency of future networking and enhance the flexibility of network configuration and management [1]. In SDN-enabled networks, the control plane, which handles management operations, is logically centralized and physically decoupled from the data plane, thus enabling high network configurability and programmability. The network programmability and the capability of optimizing the resource allocation and utilization in a centralized way, made possible by the SDN paradigm, are expected to alleviate the burden of the data onslaught expected from data-intensive applications. Hence, SDN, along with network virtualization, are expected to play an important role in future 5G networks, paving the way for the implementation of novel concepts such as end-to-end slicing, thus enabling network operators to micro-segment traffic by application and increase network agility.

While SDN and NFV are pretty well understood in the context of wired networks and data centers, there are several advantages that can be obtained by the application of the SDN principles to the wireless domain as well. For instance, distributed protocols currently employed in cellular networks cannot optimally allocate resources, nor manage handoffs and load balancing between cells. In this context, SDN can enable the implementation of centralized handover and load balancing schemes [2], allowing mobile operators to optimally manage the limited spectrum resources.

M. E. Auer and T. Tsiatsos (Eds.): IMCL 2017, AISC 725, pp. 791–796, 2018.
https://doi.org/10.1007/978-3-319-75175-7_77

Several large-scale SDN testbeds have been designed enabling researchers to run their prototypes and experiments as well as to evaluate the network backhaul of future 5G architectures [3]. Small-scale platforms have also been proposed [4] using single-board computers as SDN switches to lower the deployment cost. Other works address the Radio Access Network (RAN) part of 5G networks, focusing on RAN Virtualization [5, 6], or flexible resource allocation [7]. However, these are largely conceptual and do not consider the associated implementation challenges of real-world RANs. A recent work [8] moves past the theoretical context and focuses on the prototyping challenges of applying SDN principles to solve the Radio Resource Management problem in 5G networks for a number of use cases (interference management, mobile edge computing, RAN sharing).

In this paper, we propose a novel architecture for the implementation of an SDN-enabled 5G experimental platform that integrates both the wired and wireless parts of a cellular network. Our testbed leverages state-of-the-art components from the OpenDaylight [9] and OpenAirInterface [10] open source projects, and employs the FlexRAN controller [8] for centralized SDN control of the RAN. The proposed setup aims to enable the experimentation and validation of novel algorithms and mechanisms in next generation 5G networks.

2 Architecture Description

The proposed testbed architecture, depicted in Fig. 1, considers the application of the SDN paradigm to both the core network and the RAN of a Long Term Evolution (LTE) system. More specifically, the core network and the RAN can be considered as the wired and the wireless part of the network, respectively. In continuation, we describe in detail the proposed testbed architecture and setup, and discuss open issues with respect to the integration of the wired and wireless components.

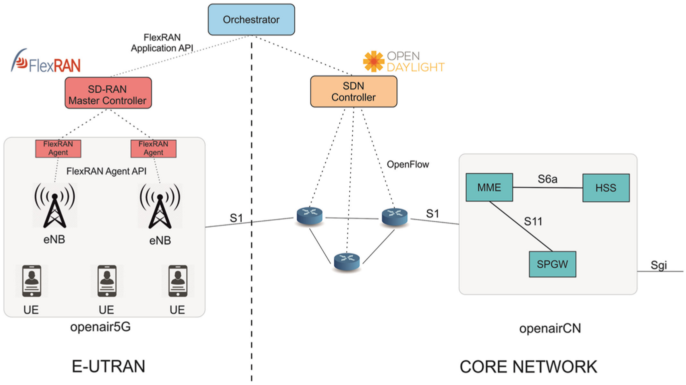


Fig. 1. Testbed architecture

A. *Wired Domain*

A key feature of the SDN technology is the ability to significantly cut on network costs by replacing expensive specialized hardware devices by generic programmable components. The proposed SDN-enabled experimental platform exploits this feature by employing low cost SDN switches to build a transport network, controlled by OpenDaylight. Specifically, the SDN switch has been implemented on an Odroid C2 board that includes a fast 64-bit 1.5 Ghz quad core ARM CPU, 2 GB DDR3 SDRAM, and a Gigabit Ethernet interface (Fig. 2). The packet switching is handled by Open vSwitch, which implements the switching functionality via a kernel module. This module had to be specifically built for the Odroid C2, as it was not included in the Linux distribution provided by the manufacturer. Finally, for the VLAN switch we employed the TL-SG105E 5-Port Gigabit Switch from TP-Link.

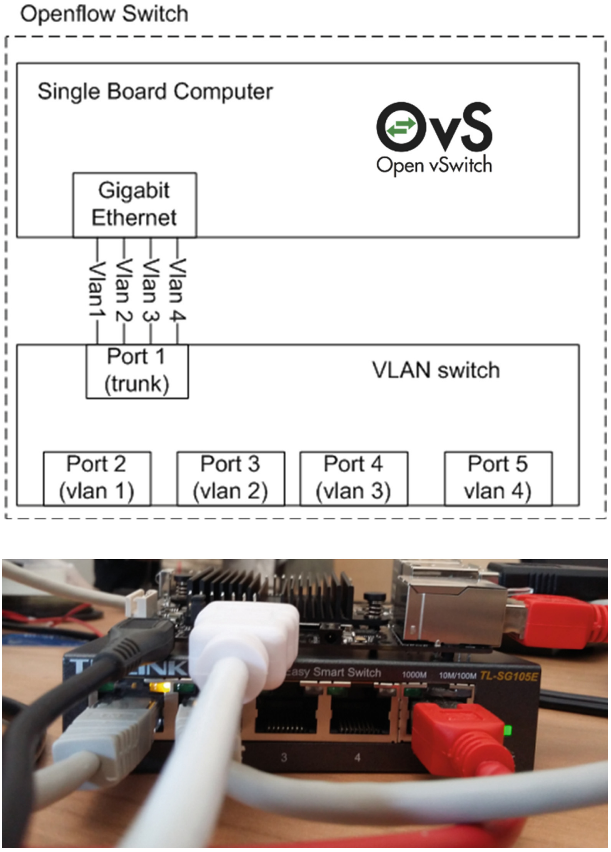
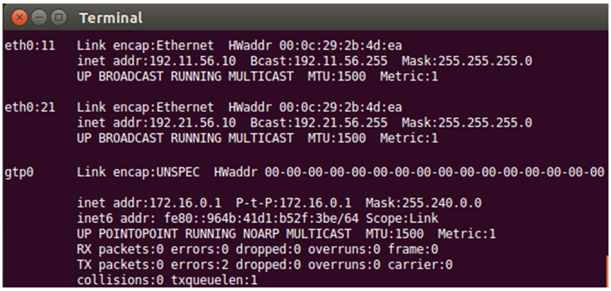


Fig. 2. SDN switch. Architecture (up) and implementation (down)

For the LTE core network, we employed the OpenAirInterface EPC (openairCN) software [11]. OpenairCN is an open source implementation of the 3GPP specifications of EPC [12], i.e., it includes the implementation of the Mobility Management Entity (MME), the Home Subscriber Server (HSS), the Service Gateway (S-GW) and the Packet Gateway (P-GW). The openairCN has been deployed on Ubuntu 16.04 LTS with kernel version 4.7 on top of a hypervisor machine (VMware software). The different EPC components have been inter-connected via virtual interfaces, as depicted in Fig. 3. The first two interfaces (i.e., eth0:11, eth0:21) facilitate the S11 communication between the MME and the S-GW, while the third one (i.e., gtp0) supports GPRS Tunneling. Finally, the S1 communication between EPC and E-UTRAN is realized by the real network interface of the Host machine.



```
eth0:11   Link encap:Ethernet  HWaddr 00:0c:29:2b:4d:ea
          inet addr:192.11.56.10  Bcast:192.11.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth0:21   Link encap:Ethernet  HWaddr 00:0c:29:2b:4d:ea
          inet addr:192.21.56.10  Bcast:192.21.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

gtp0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00

          inet addr:172.16.0.1  P-t-P:172.16.0.1  Mask:255.240.0.0
          inet6 addr: fe80::964b:41d1:b52f:3be/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:2 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
```

Fig. 3. Virtual interfaces for EPC components for S11 and GTP communication.

B. *Wireless Domain*

For the wireless domain of our testbed, we leverage components from the OpenAirInterface (OAI) platform [10, 11], which, to the best of our knowledge, is the most complete open-source LTE software implementation. The RAN is emulated with the oaisim emulator from the OAI project. Oaisim implements the entire protocol stack of all relevant LTE and LTE-Advanced protocols and supports the emulation of up to 16 UEs and 4 eNBs in a single computer. Our testbed employs a desktop computer equipped with a 6th generation Intel core i5 CPU and 16 GB RAM memory for the RAN emulation.

As it can be observed in Fig. 1, a FlexRAN Master Controller [8] is employed for the SDN-enabled centralized control of the RAN. In our setup, the FlexRAN Master Controller is deployed on Ubuntu 16.04 LTS distribution. FlexRAN is a flexible and programmable platform which separates the RAN control and data planes and supports the design of real-time RAN control applications. In FlexRAN, the RAN plane separation is made by the FlexRAN Agent API, which acts as the southbound API, having the eNB data plane on one side and the FlexRAN control plane on the other side. The control plane follows a hierarchical design with the FlexRAN Master Controller connected to a number of FlexRAN Agents, one for each eNB. The Master Controller controls the underlying infrastructure (eNBs) and orchestrates their operation, facilitating multiple advanced use cases, such as centralized load balancing and handover control.

C. *Client Traffic emulation*

As already mentioned, the proposed testbed implementation emulates the RAN by employing oaisim, which has two modes of operation, namely, full PHY emulation and PHY abstraction [13]. PHY abstraction mode allows us to emulate a large-scale system with multiple UEs and eNBs with low computational complexity and, therefore, low emulation time. In order to run more realistic emulation scenarios, oaisim has the possibility to read scenario descriptions from well-structured (OSD) files which include environment/system, topology, application and emulation configurations. The client traffic is emulated with the OpenAirInterface Traffic Generator (OTG) [14]. OTG is a realistic packet-level traffic generation tool for emerging application scenarios, which is capable of accurately emulating the traffic of different application scenarios, such as Voice over IP (VoIP), online gaming and machine type communication (MTC) among others.

Our 5G platform also supports wired network clients, which are emulated with Mininet [15]. Mininet is a network emulator that runs a collection of end-hosts, switches, routers, and links on a single Linux system. It uses lightweight virtualization (i.e., Linux containers) to emulate a complete network in a single machine. Apparently, Mininet-based networks cannot exceed the available resources (e.g., CPU or bandwidth) on a single server. Mininet hosts can run arbitrary Linux programs and traffic generators. Another powerful feature of Mininet is its ability to interact with real networks, by bridging virtual host interfaces with physical interfaces. Then, virtual hosts can send/receive packets to/from the physical SDN transport switches.

D. *Wired/Wireless orchestration*

The consolidation of the two SDN controllers for the wired and wireless domains would give the potential of centralized core/RAN orchestration, taking advantage of the global network view from both SDN controllers. Our current efforts are focused on the implementation of a centralized orchestrator, which works by consolidating APIs form both OpenDaylight and FlexRAN controllers. This will allow the design of applications which can manage, monitor and control the heterogeneous infrastructure. Such applications can be used to coordinate radio access functions for interference management, mobile edge computing, as well as RAN sharing and virtualization in heterogeneous networks. For example, FlexRAN implements centralized resource scheduling and allocation as well as centralized interference cancelation, which potentially increases the performance in high load and high interference scenarios.

3 Conclusion and Future Work

In this paper, we have presented in detail the architecture and design decisions of a 5G experimental platform that supports SDN control in the wired and wireless domains. The proposed testbed leverages components from major open source projects, such as OpenAirInterface and OpenDaylight. In the future, we plan to deploy a wired/wireless orchestrator and incorporate hardware UEs and eNBs, in a hybrid setup. Taking advantage of centralized network knowledge and analytics, provided by the proposed platform, will allow us to implement traffic engineering across the heterogeneous infrastructure. Moreover, it will enable the implementation and validation of end-to-end network slicing and micro-segmenting traffic of specific applications (network virtualization).

Acknowledgement. This work has been supported by the research projects CellFive (TEC2014-60130-P), 5G-AURA (675806), AGAUR (DI059-2016), 5GSTEPFWD (722429), IoSense (692480) and 5G-PHOS (761989).

References

1. Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Commun. Mag. **51**(2), 114–119 (2013)
2. Li, L., Mao, Z., Rexford, J.: Toward software-defined cellular networks. In: Proceedings of European Workshop on Software Defined Networking, pp. 7–12 (2012)
3. Huang, T., Yu, F.R., Zhang, C., Liu, J., Zhang, J., Liu, J.: A survey on large-scale software defined networking (SDN) testbeds: approaches and challenges. IEEE Commun. Surv. Tutor. **19**(2), 891–917 (2017)
4. Kim, H., Kim, J., Ko, Y.B.: Developing a cost-effective OpenFlow testbed for small-scale software defined networking. In: 16th International Conference on Advanced Communication Technology, Pyeongchang, pp. 758–761 (2014)
5. Tseliou, G., Adelantado, F., Verikoukis, C.: Scalable RAN virtualization in multi-tenant LTE-A heterogeneous networks. IEEE Trans. Veh. Technol. **65**(8), 6651 (2016)
6. Costanzo, S., et al.: OpeNB: a framework for virtualizing base stations in LTE networks. In: IEEE International Conference on Communications (ICC), pp. 3148–3153 (2014)
7. Tseliou, G., Samdanis, K., Adelantado, F., Pérez, X.C., Verikoukis, C.: A capacity broker architecture and framework for multi-tenant support in LTE-A networks. In: IEEE International Conference on Communication (ICC), pp. 1–6 (2016)
8. Foukas, X., Nikaein, N., Kassem, M., Marina, M., Kontovasilis, K.: FlexRAN: a flexible and programmable platform for software-defined radio access networks. In: International Conference on emerging Networking EXperiments and Technologies (CoNEXT) (2017)
9. OpenDaylight: Open Source SDN Platform. https://www.opendaylight.org/
10. Navid, N., Marina, M.K., Manickam, S., Dawson, A., Knopp, R., Bonnet, C.: OpenAirInterface: a flexible platform for 5G research. SIGCOMM Comput. Commun. Rev. **44**(5), 33–38 (2014)
11. OpenAirInterface Platform. http://www.openairinterface.org/, Repository: https://gitlab.eurecom.fr/oai/
12. 3GPP, Network Architecture. Technical Specification 23.002, 3GPP, December 2011. http://www.3gpp.org/DynaReport/23002.htm
13. Latif, I., Kaltenberger, F., Nikaein, N., Knopp, R.: Large scale system evaluations using PHY abstraction for LTE with OpenAirInterface. In: Workshop on Emulation Tools Methodology and Techniques, Cannes, France, March 2013. http://www.eurecom.fr/en/publication/3925/download/cm-publi-3925
14. Hafsaoui, A., Nikaein, N., Wang, L.: OpenAirInterface traffic generator (OTG): a realistic traffic generation tool for emerging application scenarios. In: 2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Washington, D.C., pp. 492–494 (2012)
15. Mininet. http://mininet.org/