Design and Prototype of A Virtualized 5G Infrastructure Supporting Network Slicing

Chin-Ya Huang *, Chung-Yin Ho *, Navid Nikaein † and Ray-Guang Cheng *

* Dept. of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taiwan
† Communication System Department, Eurecom, France
Email: chinya@gapps.ntust.edu.tw, nick133371@gmail.com, navid.nikaein@eurecom.fr, crg@mail.ntust.edu.tw

***Abstract*—In this paper, we investigate how OAI can be used on top of M-CORD, and present the deployment results demonstrating a virtualized 5G Radio Access Network (RAN)-Core infrastructure on top of M-CORD. Currently, both Mosaic5G and mobile central office re-architectured as a datacenter (M-CORD) utilizes OAI and the cloud RAN idea to virtualize some RAN functions and core network functions. Both Mosaic-5G and M-CORD enables virtualization and softwarization of 5G RAN-CORE infrastructure following SDN and NFV principles. In M-CORD, such a feature is extended to transport network to manage traffic in fronthaul and backhaul network segment. Specifically, both Software defined network (SDN) and (NFV) are applied not only to dynamically slice the network resources but also to manage traffic engineering in both RAN and core network aiming to support the requirements of network services. The deployment results demonstrate the procedure integrating virtualized OAI RAN and core network into M-CORD.**

***Index Terms*—M-CORD, OAI, Virtualization, SDN, NFV, 5G.**

I. Introduction

Network slicing is one of the key concept in 5G, which allows multiple logical (virtual) networks to be created on top of a common shared physical infrastructure. According to 5G Infrastructure Public Private Partnership (5G PPP) [1], network slicing is an end-to-end concept covering all network segments including Radio Access Network (RAN) and Core Network (CORE) among the others. A sliced 5G infrastructure should offer virtualized infrastructures, re-using existing mechanisms and tools, and seamless control and configuration of physical and virtual resources. The programmability, flexibility, and modularity of such infrastructure are enabled by software-defined networking (SDN) and network function virtualization (NFV).

Fig. 1 shows functional blocks and reference points in a NFV reference architecture framework defined by ETSI [2]. The functional blocks include Virtualised Network Function (VNF), Element Management System (EMS), NFV Infrastructure, Service, VNF and Infrastructure Description, and Operations and Business Support Systems (OSS/BSS), and NFV management and orchestration (MANO). A VNF is a virtualization of a network function in a legacy non-virtualized network such as 3GPP Evolved Packet Core (EPC) or RAN. The EMS performs the typical management functionality for one or several VNFs. Service, VNF and Infrastructure Description provides information regarding the VNF deployment template, VNF Forwarding Graph, service-related information, and NFV infrastructure information models.

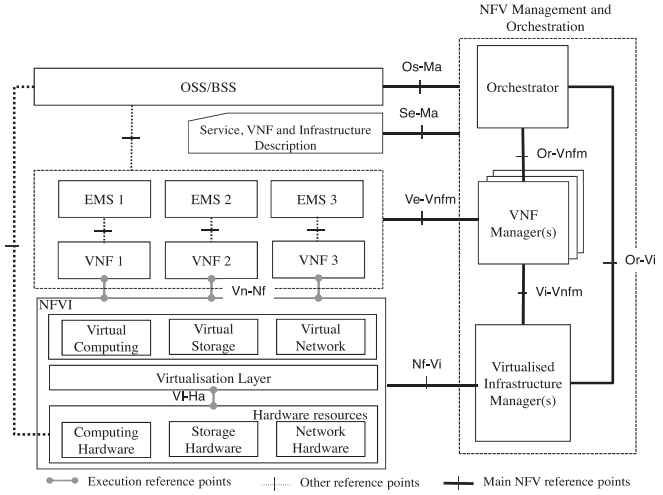


Fig. 1. NFV reference architecture framework [2].

Several tools can be applied to deploy the virtualized 5G infrastructure based on NFV MANO. The NFV MANO architecture comprises three major functional blocks: NFV Orchestrator (NFVO), Virtual Network and Function Manager (VNFM), and Virtualized Infrastructure Manager (VIM). The VIM is responsible for controlling and managing the NFV Infrastructure (NFVI) and abstract them into the physical hardware resources of compute, storage, and network. Existing solutions for VIM include OpenStack [3], VMware vSphere, CloudStack, Google Kubernetes VIM, etc. VNFM takes care of deploying, monitoring, scaling and removing Virtual Network Functions (VNFs) on a VIM. Juju [4] is one of the solutions for VNFM. The NFVO is in charge of network slice lifecycle management together with the VNF lifecycle (i.e., supported by the VNFM) and the NFVI resources (i.e., supported by the VIM). Juju-based orchestrator (JOX) [5], Open Baton [6], and Open Source MANO (OSM) [7] are possible solutions for NFVO.

Furthermore, network slicing for the LTE network is mainly composed of two egalitarian parts, which are a slice of the core network (CN) resources as well as services and a slice of the RAN, including eNB resources as well as services. Specifical-

ly, the former is a grouping of physical and virtual resources bundled together with the evolved packet core (EPC) services while the latter is for sharing of the wireless Resource Blocks (RBs) in frequency, time and space. One advantage of RAN slicing is to provide different levels of sharing and isolation according to the slice requirement, and then different service level agreement in 5G can be better satisfied [8]. SliceNet [9] is one of the approaches in realizing such a virtualized 5G infrastructure [10]. In [10], the authors realized the concept by leveraging OpenAirInterface (OAI) [11] and Mosaic-5G [12–14]. OAI is an open source project, developed to softwarize mobile network functions from the access network to the EPC of the mobile network. OAI is generally divided into two parts, the access-network software and the EPC software, and they are named as OAI-RAN and OAI-CN, respectively. The OAI-RAN can be either a monolithic evolved Node B (eNB) or a disaggregated eNB with Baseband Unit (BBU) and Remote Radio Unit (RRU). The OAI-CN implements Home Subscriber Server (HSS), the Mobility Management Entity (MME), the Serving Gateway (SGW or S-GW) and the Packet Data Network (PDN) Gate- way (PGW or P-GW). OAI is LTE release 10 compliant (with a subset of release 14) and can be deployed on standard Linux-based computing equipment. Mosaic-5G is another open source project that provides RAN and CN APIs on the top of OAI to enable monitoring, control, and programmability of RAN and CN module from a controller-domain through north-bound APIs. It provides FlexRAN [15] and LL-MEC [16] as two main controller for RAN, and edge/CN domain. In addition, it introduced a Juju-based orchestration for mobile network that support network slicing.

Mosaic-5G is leveraged in 5GPPP SliceNet project, where OpenStack, Juju, and JOX are adopted as VIM, VNFM, and NFVO, respectively. OpenStack is an open source software service framework, which provides service reservation and virtualization. OpenStack architecture is modular and pluggable, therefore using the most appropriate modules is allowed based on the need. Juju is in charge of installation, configuration and communication among services. However, it does not take the actual decisions for a particular service, but delegates service-specific decisions to a set of scripts called Charms instead. A Charm defines the ways to fetch, install and run service and the ways to fill up configuration files and to react to events. A collection of Charms linking services together is called a Bundle, which allows to deploy whole chunks of application infrastructure in one go. Further, Juju can be used to efficiently and quickly deploy, configure, scale, integrate and perform operational tasks in a wide selection of public clouds. JOX supports lifecycle management of network slices and the mobile network orchestration. Inside the JOX core, a set of services is used to operate and control each network slice, while support the necessary interplay between resource and service orchestration, VNFM and VIMs at the same time.

Mobile Central Office Re-architected as a Datacenter (M-CORD) [17] is another option to manage the RAN and CORE softwarization of the virtualized 5G infrastructure. This paper presents an approach to realize a slice-friendly virtualized 5G RAN-CORE infrastructure by leveraging existing platforms of OpenAirInterface and M-CORD. The remainder of this paper is organized as follows. Section II presents the M-CORD architecture to realize a virtualized RAN-CORE infrastructure. The design of OAI LTE over M-CORD and the prototype are described in Section III and IV, respectively. Section V concludes this paper.

II. M-CORD Architecture

A. Central Office Re-architected as a Datacenter (CORD)

CORD combines NFV, SDN, and the elasticity of commodity clouds to bring datacenter economics and cloud agility to the Telecom Central Office [18]. CORD lets the operator manage their Central Offices using declarative modeling languages for agile, real-time configuration of new customer services. Major service providers like AT&T, SK Telecom, Verizon, China Unicom and NTT Communications [19] are already supporting CORD.

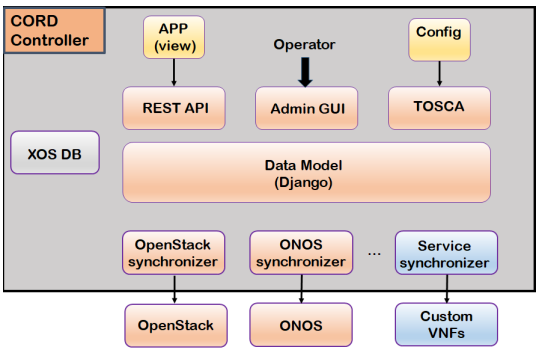


Fig. 2. Architecture of CORD platform [18].

Fig. 2 shows the general CORD platform. The CORD platform provides interface to provision and scale services, instantiate and control service instances, and monitor service performance as well as behavior. The most important function of CORD is mediating all inter-service dependency. REST API is the entry point of North-bound application, and it can also perform the configuration. Admin graphic user interface (GUI) is used for operator to visually manage VNF service. Further, TOSCA is responsible for specific configuration for each VNF that operator provides. Data model is used to define a service model which manages tenant-specific (per-service-instance) state. It contains clean, abstract and declarative representation of the system. Moreover, a synchronizer acts as the link between the data model and the functional half of the system. Synchronizers are processes continuously checking the changes of the Tenant model and applying the tenant models to the running instances. Although various containers are available to create virtual machine instances, in this paper, TenantWithContainer is used for implementation based on the characteristic of the synchronizers. Additionally, synchronizers have three parts, the synchronizer python program, model policies which enact changes on the data model, and a play-book (typically Ansible) that configures the underlying system.

Furthermore, OpenCORD has released the new version 6.0 which supports using kubernetes to hypervise the VNFs, and all the VNFs on CORD platform will be containers. However, for kubernetes, our current work was based on the stable version CORD 4.1 and 5.0. In these versions, the hypervisor of VNFs is OpenStack, and the VNF is a KVM-based virtual machine on the orchestration, and will keep on the track since the version 6.0 is not stable enough to work with.

B. Mobile CORD (M-CORD)

Furthermore, M-CORD (Mobile CORD) enables 5G on CORD to fulfill the requirement of mobile network. Based on CORD, both access and core network in 5G are virtualized and disaggregated aiming to optimize the mobile network. Hence, M-CORD is a good platform to explore 5G capabilities, including hardware, software and resource utilization enhancement, customized services and increased QoE support, and agile and cost-efficient deployment. Fig 3 shows the main concept of M-CORD. With respect to NFV architecture shown in Fig. 1, OpenStack, CORD controller, network functions and the controller in M-CORD map to VIM, VNFM, a VNF and the EMS in NFV architecture, respectively.

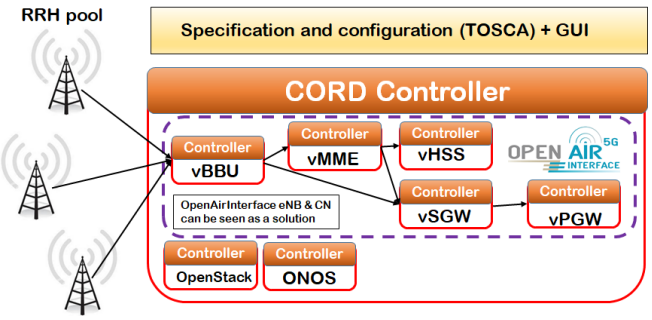


Fig. 3. M-CORD architecture [17].

Network functions like virtual Baseband Unit (vBBU), vEPC, and several mobile edge services are running on the virtualized platform from OpenStack. M-CORD uses the NFV platform to reach the service orchestration from XOS [20], and SDN service from ONOS [21]. To fit the bandwidth requirement and service feature of network slicing, each protocol module of RAN runs on one or multiple virtual machines on top of OpenStack. Moreover, Ansible is adopted in M-CORD for service automation. The playbook in Ansible describes the configuration management and the fundamental of services deployment on large-scale environment.

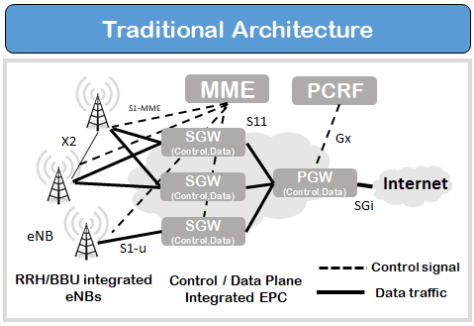
The design of M-CORD architecture splitting RAN functions into Remote Radio Heats (RRHs) and vBBU is similar to the architecture of cloud RAN (C-RAN) which is proposed to utilize the cloud resources softwarizing the mobile network functions [22]. From the viewpoint of RAN in C-RAN, the BBU and RRHs are centralized processed, and next generation fronthaul interface (NGFI) is defined for the BBU and RRH interface for packet transmission [22]. In NGFI architecture, some BBU functions are splitted and integrated into RRH to enhance system performance in latency [15, 23]. More specifically, IF4p5 and IF5 split-point are two approaches splitting the baseband functions [15, 24], and currently OAI support both of them. Specifically, IF4p5 is known as RU-RAU split, and OFDM symbol generator is the split-point which indicates in IF4p5, the received or transmitted resource elements in the usable band are simply compressed [24]. On the other hand, IF5 split-point is divided into baseband and RF, and then only RF functions are implemented in RRH. In addition to RAN, the C-RAN also consists of the core network including home subscriber server (HSS), mobility management entity (MME), serving gateway (SGW) and packet gateway (PGW).

The Juju store uses OAI C-RAN as an example to realize the RAN virtualization [10], and M-CORD also follows the concept of C-RAN to integrate some OAI RAN functions into CORD in addition to run core network functions in CORD. However, according to the concept of C-RAN, one challenge in applying M-CORD to 5G is the latency caused from message exchanging among virtual machines and the resource allocation optimization. One scenario of 5G is to support applications with ultra low latency known as uRLLC services, and thus M-CORD might encounter the timing violation in some baseband protocol procedures.

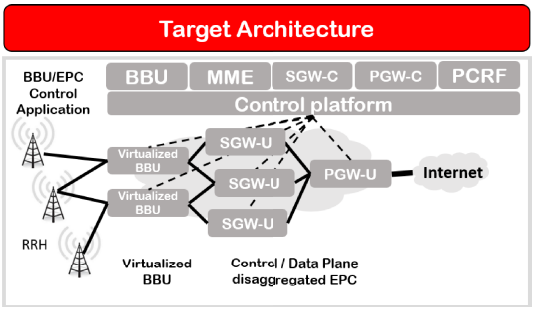
III. OAI LTE OVER M-CORD

Following the concept of C-RAN, both Mosaic and M-CORD utilize OAI to virtualize and implement 5G network function to flexibly service different applications. Fig. 4 illustrates the network infrastructures between traditional network and M-CORD. In traditional network, both eNBs and a EPC supports the functions of RAN and core network, individually. The RRHs and BBUs are integrated into eNBs while HHS, MME, SGW and PGW are integrated into EPC for LTE network. On the other hand, mobile network functionalities are disaggregated via virtualization before running on top of commodity hardware. Therefore, functions of core network and RAN can be disaggregated. Further, based on IF4p5 or IF5 split-point, some functions of RAN can be integrated into CORD together with the functions of core network. Under this condition, virtualize Radio Access Network (vRAN) can cooperate with virtualized core network functions to support end-to-end network slicing aiming to adaptively manage network resources and provide more efficient 5G services. Fig. 4 shows the scenario of vRAN, in which the traditional eNB was split into virtualized BBU (vBBU) and RRH.

Following the configuration shown in Fig. 4, we first install M-CORD platform using **Cord-in-a-Box** scenario, and then integrate the OAI EPC and BBU with the M-CORD platform to realize the virtualized RAN-CORE infrastructure. Furthermore, we also create a script to support the deployment automation according to the implementation procedure described in Fig. 5. M-CORD basically implements mobile network functions on top of CORD platform and services and network functions are running as instances which are managed by OpenStack. Specifically, Django data model will be firstly used to update the service model to meet different



(a) Traditional network.



(b) M-CORD.

Fig. 4. Scenario of traditional network and the M-CORD [17].

service requirement. To make sure the operation in the whole system, it needs to be synchronized first. With the developed synchronizer, the required network functions can be created via the modification of Ansible. Additionally, the backend resources can be managed after the services are up to run in the CORD.

LTE over M-CORD Implementation Procedure

1. Change configuration by Django data model (REST API).
2. Synchronize request by the synchronizer framework (activates data model).
3. Generate recipe by synchronizer plugin and Ansible Template.
4. Operate on resources by modifying Ansible.
5. Manage backend resources.

Fig. 5. Procedure of implementing BBU on M-CORD.

IV. Deploy of OAI LTE functions with M-CORD

This sections describes the results of implementing OAI LTE in M-CORD. Specifically, the RAN is implemented via OAI and the virtualized functions of RAN are integrated with M-CORD in which contains the running functions of core network. In our implementation, we adapt the IF4p5 split point as the interface to split the RU and RAU parts. Moreover, we design a script to assist the configuration of OAI LTE over M-CORD which can support the automation in deploying M-CORD.

Fig. 6 presents the scenario porting OAI in M-CORD to realize the 5G virtualized RAN-CORE infrastructure. Based on the CORD architecture, the LTE virtualized network functions, BBU, MME, HSS and SPGW are implemented in the compute node of OpenStack. Open Virtual Switch (OVS) manages the

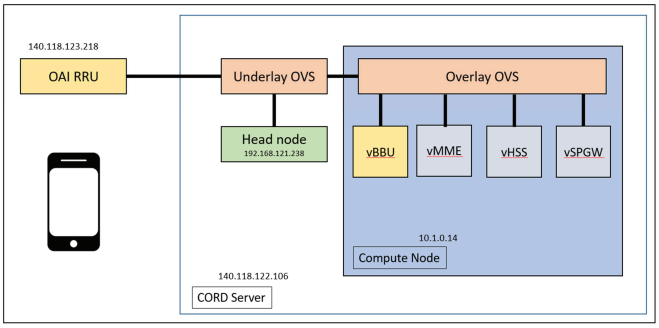


Fig. 6. OAI LTE in M-CORD scenario.

information exchange in the compute node and the CORD in the CORD. OVS has been applied to support traffic flow monitoring and management in SDN. Through OVS, the OAI RRU can communicate with the instances in the CORD server aiming to support LTE services. To realize the virtualized 5G M-CORD architecture, we first install M-CORD platform, and then integrate the OAI EPC and BBU. Furthermore, we also create a script to support the deployment automation according to the implementation procedure described in Fig. 5.

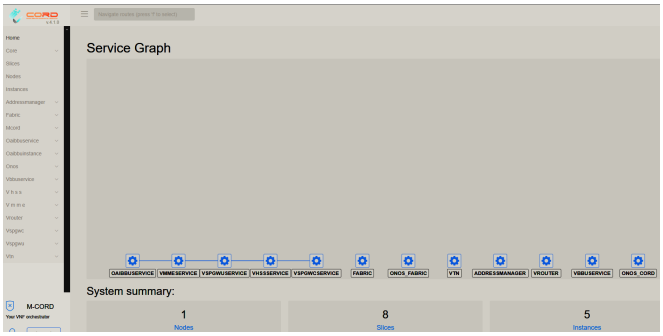


Fig. 7. Service graph of M-CORD with OAI services.

Fig. 7 presents the successful integration of OAI EPC and BBU service on the top of M-CORD platform. Specifically, the synchronizer needs to be firstly developed because it defines the service dependency, related policy, and also network topology of the VNF service. Further, to bringup the network, we modify the Ansible playbook (yaml), insert the custom image, define network type (TOSCA). By using GUI to get a VNF service, the deployment playbook will be triggered, and the service model can be found from the synchronizer in further. Note that the playbook of contribution is under open source [25]. Finally, the VNF from specific TOSCA configuration can be generated.

OpenStack plays an important role on M-CORD platform to realize virtualization, and then OpenStack also works on creating web-page GUI offered by the Horizon. When the network is created, by clicking on the web-page as shown in Fig. 8, more detail of each VNF can be checked for further instance status monitoring. As seen in Fig. 9, the connection between the vEPC and vBBU successfully is created by

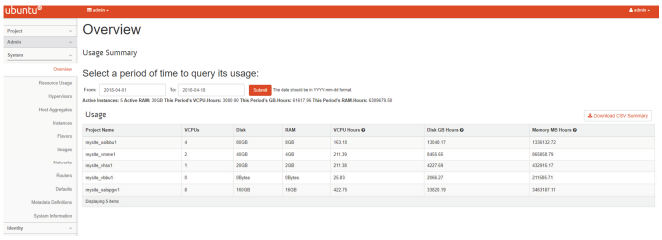


Fig. 8. OpenStack resource observation.

the well defined service dependency. Currently, we test the scenario with a M-CORD EPC with an eNB in different host in which the M-CORD server contains HSS, MME, SPGW-C and SPGW-U, and the computer performs as an eNB. The latency for the end to end latency for M-CORD deployment adds around 10 ms comparing to the Legacy OAI deployment. This additional latency issue due to virtualization of an EPC.

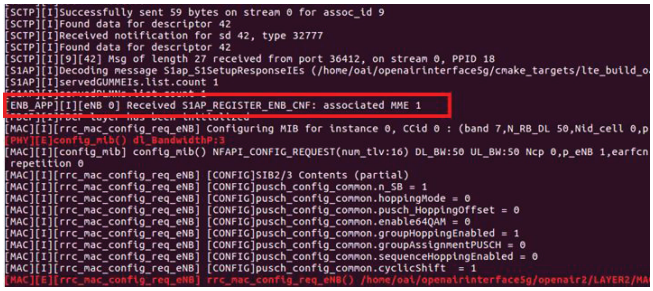


Fig. 9. Illustration of vBBU successfully connected vEPC.

V. Conclusion

M-CORD is a feasible platform virtualizing RAN and core network with the assistance of SDN and NFV. Similar to Mosaic-5G, M-CORD could support various mobile services by dynamically slice network resources for them. This paper addresses the concept of M-CORD and the demonstrates the integration of LTE and M-CORD. Specifically, OAI is applied to support the virtualization and implementation of LTE over CORD. On top of M-CORD, more network services can be integrated into M-CORD to provide various 5G services in near future.

References

[1] 5GPPP Architecture Working Group, "5G Architecture White Paper: View on 5G Architecture", Dec., 2017, [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf

[2] ETSI GS NFV 002 V1.1.1, "Network Functions Virtualisation (NFV); Architectural Framework ", Oct. 2013, [Online]. Available online at http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

[3] OpenStack Foundation Report, "Accelerating NFV Delivery with OpenStack Global Telecoms Align Around Open Source Networking Future", 2016, [Online]. Available: https://www.openstack.org/assets/telecoms-and-nfv/OpenStack-Foundation-NFV-Report.pdf

[4] Juju, Apr., 2018, [Online]. Available: https://jujucharms.com/

[5] K. Katsalis and others, "JOX: An Event-ariven Orchestrator for 5G Network Slicing", in *Proc. IEEE/IFIP Network Operations and Management Symposium*, pp. 1-9, Apr. 2018.

[6] "Open Baton", Apr. 2018, [Online]. Available: https://openbaton.github.io/

[7] "Open Source MANO", Apr., 2018, [Online]. Available: https://osm.etsi.org/

[8] C.-Y. Chang and N. Nikaein, "RAN Runtime Slicing System for Flexible and Dynamic Service Execution Environment", *IEEE Access*, vol. 6, pp. 34018-34042, Jul. 2018.

[9] "SliceNet, Deliverable 3.1 - Design and Prototyping of SliceNet Virtualised Mobile Edge Computing Infrastructure", Apr., 2018, [Online]. Available: https://slicenet.eu/deliverables/

[10] N. Nikaein and others, "SliceNet, Deliverable 3.1 - Design and Prototyping of SliceNet Virtualised 5G RAN-Core Infrastructure", May, 2018, [Online]. Available: https://slicenet.eu/deliverables/

[11] N. Nikaein and others, "OpenAirInterface: A Flexible Platform for 5G Research", in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33-38, Oct. 2014.

[12] "Mosaic5G.io", Apr., 2018, [Online]. Available: http://mosaic-5g.io/

[13] N. Navid, "Mosaic5G - Agile network service delivery platforms", in *4th OpenAirInterface Workshop*, Nov. 2017, [Online]. Available: http://www.eurecom.fr/en/publication/5368/detail/mosaic5g-agile-network-service-delivery-platforms

[14] N. Navid and others, "Mosaic5G - Agile network service delivery platforms for 5G Research", in *ACM SIGCOMM Computer Communication Review*, vol. 48, pp. 1-6, Jul. 2018, [Online]. Available: http://www.eurecom.fr/en/publication/5609/download/comsys-publi-5609.pdf

[15] C.-Y. Chang and others, "FlexCRAN: A Flexible Functional Split Framework over Ethernet Fronthaul in Cloud-RAN", in *Proceedings of 2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1-7.

[16] A. Huang and others, "Low Latency MEC Framework for SDN-based LTE/LTE-A Networks", in *Proceedings of 2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1-6.

[17] B. Snow, "Mobile CORD (M-CORD) Open Reference Solution for 5G", 2017, [Online]. Available: http://sites.ieee.org/netsoft-2017/files/2017/07/Netsoft2017_Keynote_Snow.pdf

[18] L. Peterson, "CORD: Central Office Re-architected as a Datacenter", Oct. 2016, [Online]. Available: http://opencord.org/wp-content/uploads/2016/10/BBWF-CORD.pdf

[19] R. Harwahyu and others, "Optimization of random access channel in NB-IoT", *IEEE IoT Journal*, vol. 5, no. 1, pp. 391-402, Feb. 2018.

[20] ON.Lab/ONF, "XOS", [Online]. Available: http://guide.xosproject.org/

[21] ON.Lab/ONF, "ONOS (Open Network Operating System)", [Online]. Available: https://onosproject.org/

[22] China Mobile Research Institute, "White Paper of Next Generation Fronthaul Interface", Jun. 2015.

[23] C.-Y. Chang and others, "Impact of Packetization and Functional Split on C-RAN Fronthaul Performance", in *Proceedings of 2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1-7.

[24] OpenAirInterface (OAI), "NGFI Whitepaper", Oct. 2015, [Online]. Available: http://www.openairinterface.org/?page_id=1695

[25] N. Ho, "NTUST OAI CORD", [Online]. Available: https://gitlab.com/NickHoB/NTUST_OAI_CORD