A Feasible 5G Cloud-RAN Architecture with Network Slicing Functionality

Chung-Nan Lee^{*}, Ming-Feng Lee[†], Jian-Min Wu and Wei-Chieh Chang National Sun Yat-sen University, Taiwan ^{*}E-mail: cnlee@mail.cse.nsysu.edu.tw Tel: + 886-7-5252000 ext. 4313 [†]E-mail:mflee@mail.nsysu.edu.tw Tel: + 886-7-5252000 ext. 4335

Abstract—The development of the fifth generation (5G) communication technology is in progress to satisfy the increasing demands of higher capacity, lower latency, and ubiquitous mobile access in the next generation mobile cellular networks. Furthermore, 5G networks shall be able to provide diverse and challenging requirements. As we know, Cloud-Radio Access Network (Cloud-RAN or C-RAN) architecture has been identified as a promising approach to 5G, and network slicing offers a serviceable way to meet diverse requirements for 5G. However, providing reliable, cost-effective, and quality of service guaranteed network slices under C-RAN architecture is one of the major challenges in 5G. In this paper, we present a feasible solution for managing network slices in the C-RAN architecture in order to facilitate the deployment of the 5G prototype. We make use of OpenStack, Docker and OpenAirInterface (OAI) to implement the C-RAN architecture with properties of flexibility, scalability and rapid deployment. We then validate the feasibility of the prototype by means of signal and data transmissions of Unmanned Aerial Vehicle (UAV).

I. INTRODUCTION

The fifth generation mobile communication system (5G) is the extension of 4G mobile communication system. The new generation of 5G communication networks adopts the high frequency and millimeter wave (5G-NR) to significantly increase the communication bandwidth. It is also a new generation of telecommunications network innovations from the previous generation of "closed systems" to "open source architecture". The requirements of 5G mentioned in the 5G white paper [3] of Next Generation Mobile Networks Alliance (NGMM Alliance) indicate that the resource and signal efficiency must be improved in 5G, and that IoT (Internet of Things) or MTC (Machine-type Communication) should be noted. The white paper also mentions that 5G must meet the needs of the following three major scenarios: higher bandwidth (Enhanced Mobile Broadband, eMBB) to achieve higher transmission speed; higher coverage (Massive Machine Type Communication, mMTC) to satisfy diverse services; and have lower latency (Ultra-Reliable and Low-Latency Communications, uRLLC) to improve service reliability.

The 5G mobile communication system is bound to connect a large number of emerging mobile devices and must face a heterogeneous network architecture with multiple access technologies. In order to cope with the rapid development of various technologies, a modular system needs to be established to deal with various situations. Since the traditional core network requires the use of complicated and expensive hardware devices, it is necessary to virtualize the network architecture to reduce the build-up cost of the density network.

Cloud Radio Access Network (Cloud-RAN or C-RAN) [2, 8] is a novel mobile communication network architecture. Its main concept is to divide the base station of mobile communication into two parts, including the Baseband Unit (BBU) responsible for the processing of the baseband signal, and Remote Radio Head (RRH), which is responsible for signal processing of radio frequency. With the maturity of Software Defined Radio (SDR) technology, BBU functions can be implemented by software. Therefore, the current technology trend is to combine cloud computing and virtualization technologies to dynamically allocate BBU computing resources.

In the future 5G networks, various different network access technologies and services providing different needs will be integrated for various user groups. For this purpose, network slicing technology [12] is a very useful tool. Network slicing is a function that can integrate all kinds of services into the underlying infrastructure of the same entity through virtualization technology, and has the ability to execute these services at the same time. Services are independent and do not interfere with each other. Users or service providers can form new services by combining several services. Among them, the network architecture formed by services is implemented through Network Function Virtualization (NFV). In order to allow a single network infrastructure to provide a variety of services, we can offer network slicing functionality through cloud computing, Software-Defined Networking (SDN), and NFV.

This paper focuses on proposing a feasible C-RAN architecture with networking slicing functionality using open source software. We use OpenStack to provide computing and storage resources, so that we can freely expand hardware devices. At the same time, we use OpenAirInterface's protocol stack to build the EPC and set up the OAI BBU and RRH to achieve Cloud-RAN. In addition, through the use of the container technology provided by Docker, the programs needed to provide the service can be modularized and lightened, and the modules can be installed and distributed easily. This can reduce the complexity and time-consuming of deployment. Through this architecture, the system can provide RAN as a Service (RANaaS) [9, 20]. By arranging

and combining the above modules to provide new services, the C-RAN achieves the network slicing function. Our C-RAN system contains both edge cloud and core cloud. Enhanced Mobile Broadband (eMBB), a multimedia service that requires more bandwidth, is placed in the edge cloud, while Ultra-reliable and Low-latency Communications (uRLLC), which requires high reliability and ultra-low latency, is placed in the core cloud. This kind of placement makes resource scheduling more efficient. By building a Virtual Private Network (VPN) server, the hosts in different regional networks are connected to the Virtual Machines (VMs) to achieve system integration. Finally, we use several sets of application services to verify the feasibility and efficient of our C-RAN system.

II. RELATED WORK

Demestichas et al.[1] mentioned that in the 5G generation, the heterogeneous network would be a representative network. 5G will use the C-RAN architecture as its main development direction. C-RAN adopts two technologies: SDN and NFV. Both of these technologies can reduce the cost of deploy a base station and provide more flexibility in deployment. Nikaein et al. [20] described that as the Internet would grow explosively, the commercialization and virtualization of wireless networks have changed the economics of mobile networks, from specialized and customized hardware devices to the open software platform, hence setting up base stations are less costly and more flexible. In addition, innovative services can leverage existing infrastructure and create it efficiently through the cloud technique.

The main goal of the 5G architecture proposed by 5G white papers [3, 5] is to logically provide a variety of services in a single network infrastructure. The most intuitive method is to use C-RAN to provide large-scale device access and implement radio access network functions as required. Different from the traditional network architecture formed by cumbersome and numerous routers and switches, 5G simplifies the architecture of the core network, separates the control plane from the user plane, and implements an automatically configured network function. In addition, the 5G architecture achieves automatic slicing of various services through adaptive network operation and maintenance, and can maintain and terminate services, thereby reducing operating costs. When service providers provide services, they can reuse open function modules stored in the system and develop modules with their own uniqueness, and combine them into a complete set of services through agile networks.

The document in [8] mentions that Nokia have four solutions to build cloud telecommunications. The first one is to use IT virtualization technology to separate the hardware functions from the bottom through the NFV, and build the necessary functions in the software. The NFV is applicable to the processing of any data plane grouping and the control plane function in the mobile network infrastructure, and can form an Infrastructure as a Service (IaaS) cloud in a cloud platform. The second solution is the use of SDN technology to

virtualize the Local Area Network (LAN) of the data center, which can achieve the infrastructure management to provide the required application software, through the Service Level Agreements (SLAs) to manage the transmission of the virtual tenant network (VTN) and data center. In the third solution, the balance of the decentralized architecture [2] and centralized architecture must be considered when constructing the cloud system. It is recommended that operators can deploy centrally scheduled components to be responsible for the relatively high-level control functions. The fourth solution mentions that the development of data and applications in computing will be automated. Cross-network cloud management and services are very important, so they need two new features, one is the Global Network Orchestrator (GNO), to achieve the necessary automation. Another feature is Customer Experience Management (CEM), which is used to manage applications to optimize network endpoints to endpoints rather than to optimize individual network functions, while GNO and CEM do not interfere with each other's functions.

The establishment of C-RAN can integrate access networks of large and small areas, but it is difficult to allocate resources. This is an NP-hard problem, so finding the best method is a big challenge. The use of the Gale-Shapley (GS) algorithm in [4] to match D2D pairs in cellular User Equipments (UEs) have been proven to be a stable method. Spectrum sharing [10, 21] or resource sharing [13] can also be used in addition to static resource allocation. Zakrzewska and Iversen [13] investigated the situation of resource blocking rates. The survey results show that the cloud network architecture can flexibly allocate shared resources to reduce resource blocking rates. Pompili et al. [11] proposed a method of resource reconfiguration, which divides each base station into groups and dynamically and efficiently adjusts the demand according to the usage of each user. This method can make resource allocation more efficient. In [17], the authors proposed a Dynamic Resource Allocation (DRA) method under the Heterogeneous Cloud Radio Access Network (H-CRAN) model. The authors designed a clustering algorithm to group RRHs. They used multi-point techniques to eliminate the interference in each set, and finally developed a joint frame structure. The DRA can significantly reduce interference effects in a simulated environment. Hu and Wang [7] proposed user-centric local mobile cloud-assisted (UC-LMC) resource allocation method to assist H-CRAN. This method is mainly for resources-poor UEs to use other UE's spare resources through the offloading procedure. This method can improve the user experience and improve the utilization of resources. Yang et al. [16] mentioned that Information-centric network (ICN) can be used in the C-RAN. In this method, the information stored in the BBU pool for a certain period of time is allowed to be read by the user quickly, so as to reduce the transmission blocking rate. Elias et al. [23] formulated the optimal resource calendaring problem in C-RAN using Integer Linear Programming (ILP) and then devised effective heuristics producing close-to-optimum solutions. Simulations show that the proposed approach can effectively improve the performance of C-RAN scheduling. Zuo et al. [24] first considered the energy efficiency-based user association problem in the C-RAN environment with massive MIMO empowered, and then proposed three user association algorithms. Their simulation results show that the proposed methods achieve better energy efficiency.

Under the heterogeneous C-RAN, interference management will be a very important issue. In [17], the inter-layer interference coordination technology and Cooperative Radio Resource Allocation (CPRA) are considered, especially Interference Collaboration (IC) and Beamforming (BF) to suppress inter-layer interference. Based on IC and BF, a CPRA optimized model was proposed. The results show whether better performance depends on the configuration of Heterogeneous Cloud Radio Access Network (H-CRAN), including the number of RRHs. In [14], Cloud Empowered Cognitive Inter-cell Interference Coordination (C2-ICIC) is proposed. This method is to use C-RAN architecture and inter-departmental coordination to adopt Phantom Cell architecture [15], mainly to limit the inter-area transmission interference of signal from each channel. This method is proved that in the ultra-density environment, it can increase 100% gain. In [22], the authors proposed a user weighted probability-based algorithm for the heterogeneous C-RAN architecture to achieve effective spectral resources and interference management.

In terms of network slicing, Son and Yoo [12] proposed a complete End-to-End (E2E) network slicing architecture. Three kind of service slices are listed: Ultra High Definition (UHD) slice, phone slice, and massive IoT slice. These services have their own service module support. For example, in the UHD slice, the UE performs network connection and packet transmission through the three service modules, Digital Unit (DU), Core, and cache provided by NFV. And then through the VPN tunnel of the UHD slice, UE communicates with the back-end Content Delivery Network (CDN) module, which can provide users with a complete set of multimedia audio and video services. The same applies to the phone slice. The UE transmits the data to the system through the DU. The voice packet is provided in the entire system through the VPN Server to provide the routing function, and finally sent to the core cloud for processing. The massive IoT slice has the characteristics of a large number of devices. Its DU's ability to process multiple files is necessarily different from other services. Therefore, it has a dedicated and specific processing program. In summary, the network slicing is based on the service module as its constituent unit, and the service is provided by the combination of these modules. These service modules have strong replaceability and portability. In the 5G architecture, network slicing has the ability to quickly build and quickly deploy.





Figure 2. Network slicing of the proposed C-RAN Architecture

III. SYSTEM OVEWVIEW

Figure 1 shows the overall system architecture. We use OpenStack's built-in SDN Controller to configure network resources such as IP addresses, virtual routes and network bandwidth. The part of C-RAN is to modularize and combine the services required by the core network through SDN. It becomes a service oriented core, which can also be regarded as a software-defined node. By virtualizing the EPC, the eNB can be plug and play (PnP).

In the proposed C-RAN, OpenStack that has powerful scalability provides computing resources and storage resources. When necessary, more computers or hosts can be added at any time to become new computing nodes and provide more computing and storage capabilities. In addition, OpenStack can also provide load balancing so that it can be more efficient and more flexible when using resources. In this system, virtual routers are built using the SDN controller provided by OpenStack itself, and a virtual area network is configured. Whenever there is a new instance or Docker machine generated, it can easily join the network. Next, we use the Docker machine to request OpenStack's resources to create an instance of the Docker suite. This approach allows different instances, and even Docker of different hosts, to use unified management when using resources. When the Docker machine is installed, we can use the Docker command to create a container, upload or download a container that has already been built, and set the startup parameters for the container (for example, network interfaces, libraries, etc.). Finally, for cross-segment requirements, we connect the container to the container through a VPN server that is set up on a physical machine or on a virtual machine.

Figure 2 is the network slice architecture proposed for our C-RAN. Its main components are base stations, including two sets of OAI eNBs and a commercial small cell, two sets of OpenStack cloud systems, and the establishment of virtual routers through the built-in SDN tools. Then, we need to create a program for the Docker container to run the service through the Docker machine, and finally connect the modules and the server outside the system through the router and the VPN server.

IV. C-RAN DEPLOYMENT USING OPEN SOURCE SOFTWARES

The system consists of two sets of cloud systems: core cloud and edge cloud. The major difference between these two types of cloud is that the hardware performance is different. Because of the differences in the hardware, and then affect the service items configured in the two cloud systems. According to the three major types of services in the 5G white paper, our cloud system provides different modules and resources for different types of services. In this system, the core cloud is classified as a service module that is in favor of uRLLC, and the edge cloud is the service module that is mainly responsible for eMBB.

A. Core Cloud

Compared with edge cloud, the core cloud of the back-end has a large storage space, and usually the user has less access to the core cloud. On the hardware side, the core cloud can allow equipment with poor performance of computation, memory or temporary storage, so it is suitable for setting up service modules such as database, data backup system, postimaging, or TCP server. The data transmission of the core cloud is not required to meet immediateness. The core cloud architecture is shown in Figure 3. OpenStack is used as a function to provide computing and storage resources, and by building virtual routers to provide SND's network selfconfiguration. Therefore, the underlying modules are automatically assigned with IP address, bandwidth and network resources, which can be a basis for data transmission. In order to achieve the fast configuration of the network slice in the 5G architecture, the module needs to be enabled within a short time and it is easy for users or service providers to add, delete, modify, and do other actions. The proposed system uses docker machine for module building, because it is a lightweight virtual machine technology. This not only provides the advantages of quick installation, the ability to upload containers to Docker, but also brings portability features to these modules.



B. Edge Cloud

Compared with the core cloud, the edge cloud features a good access rate and network bandwidth, and is also better

than the core cloud in terms of computing performance. When users enjoy services, they will frequently access services within the edge cloud. Like the core cloud, edge cloud also provides resources through OpenStack. The difference lies in the hardware devices. The edge cloud has enough memory and computing power. The UDP server or image stitching and other service modules that need real-time computing will be built on the edge cloud, which can reduce the user interface and data transmission route, and then improve the immediacy of service. The architecture of the edge cloud is shown in Figure 4. OpenStack is used as a role to provide resources, and virtual routers are deployed through the SDN controller and allocated network resources. Then we build docker machine to maintain the service module. In order to guarantee the immediacy of data transmission, UDP servers that can process a large number of packets, and OAI EPC and OAI eNodeB that are responsible for providing UE network resources are all built in the edge cloud. As a result, when the UE transmits data, the number of routes that the packet needs to pass through will be reduced, and the real-time nature of transmitting packets will be further achieved.



C. VPN

VPN runs through all module blocks in the entire architecture. We install the Ubuntu 16.04 virtual machine on Hyper-V, set a static IP address (available for DNS queries) and install the PPTP-Linux suite to set up a VPN server. Whether it is a Host, instance, VM, or Docker machine, as long as the PPTP client parameters are set, it can be connected to the VPN server. After that, regardless of whether the module is located in any or different regional networks, data transmission can be performed according to the IP address assigned by the VPN server. By combining the VPN server with the core cloud and the edge cloud, it can be demonstrated that this system architecture is capable of integrating crossplatform cloud systems. In addition, the VPN server can solve the disadvantage that the commercial access network cannot be arbitrarily adjusted or modified. As long as the EPC of the commercial access network is added to the VPN area network, or the packet is transmitted to the VPN area through an extra machine, it can combine commercial and non-commercial network. As shown in Figure 5, the instance or docker machine in the core cloud and edge cloud can be connected to the VPN server and a new IP address of the network can be configured. The commercial access network can also pass through or add a new packet forwarding server. This is a new routing mechanism to send packets to the VPN area network. In this way, regardless of whether the UE is in an edge cloud or a commercial access network, the UE can enjoy the resources of the edge cloud and the core cloud, and even cross-segment data transmission.



V. SYSTEM VERIFICATION AND EXPERIMENTAL RESULTS

After the completion of the above three system blocks, in order to prove the feasibility of the proposed system architecture, we create a set of video streaming broadcast services. First, we build a Tomcat server in the core cloud and used WebSocket technology [18] to coordinate the control signals of the terminal equipment. Apache Tomcat is a Java Servlet container developed by the Jakarta project under the Apache Software Foundation. WebSocket is a TCP fullduplex communication protocol [18]. In WebScoket's API, as long as the two parties complete the handshake, a continuous connection between the two can be established directly, and two-way data transmission is performed. Reference [19] mentions that WebSocket technology is a very good choice for services that require continuous transmission. The environment of the system is described in Table 1. For the parameter setting of the OAI eNB, we use the default 125 dbi of the tx_gain, 90 dbi of the rx_gain and TM 1 (single antenna port, port 0) of the ue_TransmissionMode.

Table 1. System development environment.				
Development Environment				
•	Operating System: Ubuntu 16.04			
٠	CPU: Intel® Core™ i7-4790 CPU @ 3.60GHz			
•	Main Memory: 20.0 GB			
•	USRP: Ettus USRP B210			
•	Development Software: Android, OpenStack, JAVA, Docker, MySQL, WebSocket, Tomcat			

A. Experiments

The purpose of the three kinds of experiments is to verify the feasibility of the proposed architecture. The experiments are to use the Tomcat server of the core cloud to exchange control signals with the application program on the mobile phone, and then use the UDP server of the edge cloud to serve the video stream captured by the mobile phone to other mobile phones. The network slicing function implemented includes four services: online instant messaging, video streaming service, face detection, and UAV control signal processing. Since the transmission protocol used for online instant messaging is the TCP protocol, mainly focuses on reliability, and therefore no performance test on the transmission. Due to the well-known software Iperf cannot measure the downlink of multiple UEs at the same time, we use the NC Express and in-house Android App to test downlink, uplink, and packet loss of the video streaming performance. The reason for adopting NC Express is that the software is a scalable measuring application that is ready to be found on Google Play and can simultaneously measure the downlink of multiple UEs. However, NC Express does not provide the function of measuring uplink, so we also use the in-house Android app to test the uplink and downlink.

The effectiveness of face detection is mainly the observation of the operation time. The part of the UAV control signal processing focuses on measuring the delay time of packets, which is one of the important indicators to measure reliability and low latency services.

B. Results

B.1. Video Streaming Service (Uplink/Downlink/Packet Loss)

We set up a server on the VM and use an internal network to reduce unnecessary interference. All tested UEs are connected to the video server that co-exists with EPC, and the path of the packet will only pass through one router. The UEs used in the experiment are Sony Xperia z3+. When the NC Express test is performed, the software downloads 10 times in two minutes, and each download takes one or two seconds extra time to finish. If another mobiles start downloading before the mobile phone is finishing its download, there is an opportunity that they will occupy all the bandwidth and cause the mobile phone to fail to make another request. Moreover, when NC Express tests multiple UEs, if some UEs occupy all the bandwidths, other UEs will not be able to receive the packets until these UEs end. As a result, different UEs use the maximum bandwidth at different points in time, which may result in higher total downlink when using NC Express testing than using in-house app testing. The results are listed in Tables 2 and 3. They shows that more UEs consume more bandwidth and also have more packet losses.

There are a substant and an and an and an area and an area and and a					
Number of UEs	Downlink (Mbps)	Packet Loss Rate (%)			
1	5.57	0.00			
2	4.58	0.29			
4	3.03	1.70			

Table 2. Using NC Express testing on the internal network.

Table 3 Vising in	-house Android App testin	g on the internal network.

Number of UEs	Downlink (Mbps)	Uplink(Mbps)
1	4.28	4.22
2	2.70	2.55
4	1.50	0.81

B.2. Face Detection (Operation Efficiency)

We install the YOLO [6] face detection module in docker container on OpenStack and docker container on PC respectively, to test the computing performance. The two test scenarios use exactly the same hardware device. The difference between the two scenarios is that the docker machine uses OpenStack resources to build VMs or uses physical machine. This experiment uses in house APP installed in mobile phone to take pictures and then send them to the face detection module (YOLO library) in the edge cloud. After the detection is completed, the face position of the original image will be marked and then transmitted back to the mobile phone for display. In addition, the identified images and data will be saved to the image database in the core cloud that will be further analyzed later. The time required for the recording module to process one frame at a time is shown in Table 4.

Table 4. Processing time of the face detection module.					
Environment	Docker on	Docker on PC			
	OpenStack				
Processing Time (ms)	913.68	910.95			

B.3. Signal of UAV (Packet Delay)

We insert the OAI dongle and SIM card into the Raspberry Pi on the UAV to connect to our C-RAN system, then we perform two experiments. The first one is to test how much time delay will be needed when the mobile phone sends a control signal to the UAV. The second is the time delay of packet transmission when the UE on the UAV returns images to provide other users for viewing. A total of 50 measurements are taken in this experiment and averaged after recording. The result of the measurement is 19.71 milliseconds. The experiment of the video packet delay is to carry out 100 packet transmission, and the final average delay time is 41.81 milliseconds. Although the delay time of the UAV transmission control signal is not easily perceived by the human, the tactile IOT is expected to achieve a time delay of less than 1 millisecond. Due to the limited capacity of current hardware devices and underlying communication system, we have not yet been able to meet this requirement.

VI. CONCLUSIONS

In this paper, we proposed a C-RAN based on the 5G architecture, which is implemented using open source software. The system can not only be quickly and costeffectively built, but also can provide network slicing functionality. Using VPN as a bridge to solve equipment or virtual machines on different network segments. It can even integrate various types of network access methods such as Wi-Fi, Soft-Defined Radio (OAI), and commercial small cell to provide a cross-platform solution. The network access of the plug-and-play C-RAN architecture is further achieved. It also achieves higher network coverage and higher overall traffic. Based on the concept of network slicing, we implement four different application services, including social messaging software for online instant messaging, video streaming services, YOLO face detection modules, and the handling of UAV control signals. Through these practical and performance tests, it can be verified that the proposed framework is feasible and has a certain quality of service. The current network slicing function in our C-RAN system is mainly based on VM/instance as the basic unit. In the future, we shall allow the system to adaptively configure modules and hardware resources so that the network function can be sliced elastically.

ACKNOWLEDGEMENT

This research was supported in part by the Ministry of Science and Technology of Taiwan under contracts No. MOST-106-2221-E-110-018 and MOST 107-2221-E-036-MY3.

REFERENCES

- P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong and J. Yao, "5G on the Horizon: Key Challenges for the Radio-Access Network." *IEEE Vehicular Technology Magazine*, 8(3), pp. 47-53, 2013.
- [2] Ericsson, "CLOUD RAN." Ericsson White paper, 2015.
- [3] R. E. Hattachi and J. Erfanian, "5G White Paper." NGMN White Paper, 2015.
- [4] T. D. Hoang, L. B. Le and T. Le-Ngoc, "Distributed Energyefficient Resource Allocation for Device-to-Device Communications Underlying Cellular Networks." *IEEE Transactions on Vehicular Technology*, pp. 6972-6986, 2016.
- [5] Huawei, "5G Network Architecture A High-Level Perspective." *Huawei White Paper*, 2016.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unified, Real-time Object Detection." arXiv preprint arXiv:1506.02640, 2015.

- [7] H. Hu and R. Wang, "User-centric Local Mobile Cloud-Assisted D2D Communications in Heterogeneous Cloud-RANs." *IEEE Wireless Communications*, 22(3), pp. 59-65, 2015.
- [8] Nokia, "Reinventing Telcos for the Cloud," *Nokia White paper*, 2014.
- [9] A. Outtagarts, L. Roullet, B. Mongzon-Cazavet and G. Aravintha, "When IT Meets Telco: RAN as a Service." 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), pp. 422-423, 2015.
- [10] M. Peng, H. Xiang, Y. Cheng, S. Yan and H. V. Poor, "Inter-Tier Interference Suppression in Heterogeneous Cloud Radio Access Networks." *IEEE Access*, 3, pp. 2441-2455, 2015.
- [11] D. Pompili, A. Hajisami and T. X. Tran, "Elastic Resource Utilization Framework for High Capacity and Energy Efficiency in Cloud RAN." *IEEE Communications Magazine*, 54(1), pp. 26-32, 2016.
- [12] H. J. Son and C. Yoo, "E2E Network Slicing Key 5G Technology: What Is It? Why Do We Need It? How Do We Implement It?" 2015. Available at <u>http://www.netmanias.com/en/?m=view&id=blog&no=8325</u>
- [13] A. Zakrzewska and V. B. Iversen, "Resource Sharing in Heterogeneous and Cloud Radio Access Networks." 2012 IV International Congress on Ultra Modern Telecommunications and Control Systems, pp. 34-39, 2012.
- [14] S. A. R. Zaidi, D.C. McLernon, M. Ghogho and M. A. Imran, "Cloud Empowered Cognitive Inter-cell Interference Coordination for Small Cellular Networks." *IEEE International Conference on Communication Workshop* (ICCW), pp. 2218-2224, 2015.
- [15] S. Mukherjee and H. Ishii, "Energy Efficiency in the Phantom Cell enhanced Local Area architecture." Wireless Communications and Networking Conference (WCNC), 2013.
- [16] C. Yang, Z. Chen, B. Xia and J. Wang, "When ICN Meets C-RAN for HetNets: an SDN Approach," *IEEE Communications*

Magazine, 53(11), pp. 118-125, 2015.

- [17] Z. Yu, K. Wang, H. Ji, X. Li and H. Zhang, "Dynamic Resource Allocation in TDD-Based Heterogeneous Cloud Radio Access Networks," *China Communications*, 13(6), pp. 1-11, 2016.
- [18] I. Grigorik, "WebSocket Browser and Protocols, Chapter 17." 2013.

Available at https://hpbn.co/websocket

- [19] L. Yang and G. Yang, "Real-Time Wireless Signal Testing and Analyzing System Based on Websocket." *Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, 2016.
- [20] N. Nikaein, R. Knopp, L. Gauthier, E. Schiller, T. Braun, D. Pichon, C. Bonnet, F. Kaltenberger and D. Nussbaum, " Demo – Closer to Cloud-RAN: RAN as a Service." Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom), 193-195, 2015.
- [21] K. Kim, Y. Jin, D. Keum, S. Choi, M. Mueck and V. Ivanov, "Operation of Reconfigurable Mobile Device for Spectrum Sharing." Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 639-642, 2015.
- [22] I. Al-Samman, R. Almesaeed, A. Doufexi, M. Beach and A. Nix, "User Weighted Probability Algorithm for Heterogeneous C-RAN Interference Mitigation." *IEEE International Conference on Communications (ICC)*, 2017.
- [23] J. Elias, F. Martignon, M. Morcos, L. Chen and T. Chahed, "Radio Resource Calendaring in Cloud-based Radio Access Networks." *Wireless Days (WD)*, 2018.
- [24] J. Zuo, J. Zhang, C. Yuen, W. Jiang and W. Luo, "Energy Efficient User Association for Cloud Radio Access Networks", *IEEE Access*, pp. 2429 – 2438, May 2016.