2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom) | 979-8-3503-3782-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/BLACKSEACOM58138.2023.10299755

Open Source 5G Core Network Implementations: A Qualitative and Quantitative Analysis

Rekha Reddy†, Michael Gundall†, Christoph Lipps† and Hans Dieter Schotten*†
†Intelligent Networks Research Group, German Research Center for Artificial Intelligence
D-67663 Kaiserslautern, Email: {firstname.lastname}@dfki.de
*Institute for Wireless Communication and Navigation, RPTU Kaiserslautern-Landau
D-67663 Kaiserslautern, mail: {lastname}@rptu.de

***Abstract*—Beyond 5G and 6G systems provide significant advances in network architectural design and deployment with the ever-increasing connected devices. In order to address the tremendous growth of cellular traffic, both hard- and software are facing increased scalability and flexibility requirements. Therefore, a comparative study on open-source 5G core network implementations is conducted that includes qualitative as well as quantitative requirements. While the qualitative comparison is based on licensing, programming language, deployment possibilities, and relevance, the qualitative metrics are resource utilization and the Round Trip Time experienced by end-user devices. In order to evaluate these metrics, each of the open-source projects is deployed as service based architecture in a virtual machine/container-based infrastructure. The results indicate implementation benefits and the trade-off between load consumption and latency evaluation.**

***Index Terms*—5G CN, Free5GC, OAI 5G CN, Open5GS, SBA**

I. Introduction

Automation of the evolved networks is driven towards optimizing service usage and accessing new use cases with widely varying requirements such as enhanced Mobile broad-Band (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and massive Machine Type Communication (MMTC). The introduction of Fifth Generation (5G) has influenced emerging applications such as Virtual Reality (VR), Autonomous Driving, and Industrial Internet of Things (IIoT). These technologies will significantly cause demand not only for communication resources but also for energy and computation power [1]. In addition, 5G presents a potential transition in the millimeter-wave (mmWave) spectrum, bandwidth allocation, and reallocation, as well as virtualization in the Core Network (CN) to accommodate the increasing growth for new applications and connected devices. By the use of Network Function Virtualization (NFV), Cloud-Native Network Functions (CNFs), and Software-Defined Networks (SDNs), 5G CNs are implemented as Virtual Network Functions (VNFs) on Commercial-off-the-Shelf (COTS) hardware.

The 5G system architecture is represented in terms of service-based interfaces applicable for control plane functions as well as point-to-point functions to connect the user plane. To accommodate the functional and service agility the Service Based Architecture (SBA) has attracted higher interest. Thus, providing greater flexibility to incorporate new functions and more dynamic agile architecture [2]. Moreover, 5G SBA addresses the separation of control and user plane functions, network slicing allowing multiple logical networks to support over one network, the distribution of cloud infrastructure closer to users, and the consolidation of user plane functions for mobile and fixed access networks [3]. Over the past years, 5G CNs are implemented as open-source platforms by various communities such as free5GC, Open5GS, and OpenAirInterface (OAI). Going forward, all these platforms are enhanced with Machine Learning (ML) techniques to study the system behavior, such as traffic forecast [4] or resource resilience and optimization [5]. While other investigations focus on enhancing the core design for improved authentication, session management, and security [1], [6], this work covers the quantitative and qualitative evaluation of the most relevant open-source software, namely free5GC, Open5GS, and OAI 5G CN, based on the following parameters.

- *Quantitative Analysis*: (a) Resource utilization analysis during the CN execution. (b) Performance and Round Trip Time (RTT) assessment of User Equipment (UE) in presence of multiple end-user instances.
- *Qualitative Analysis*: Platforms are reviewed based on licensing, infrastructure, programming language, and relevance.

The remainder of the paper is organized as follows: Section II presents the related work performed with a focus on enhancing the platforms. The standalone architecture of 5G is highlighted in Section III. A brief description of the open-source 5G projects is discussed in Section IV and the testbed design and specification are presented in Section V. Further, Section VI details both results and their interpretation achieved during the comparative study. Finally, Section VII concludes the paper and outlines future work.

II. Related Work

There has been recent research to improve the overall performance of the 5G architecture. $L^2$5GC architecture proposed by *Jain et al.* [7] is built based on free5GC to improve the control plane operations and reduce their impact on the data plane. $L^2$5GC eliminates message serialization, and overheads caused by HTPP requests to reduce the control plane inferences using scalable, flow-level packet classifiers while still being 3GPP-standard compliant. Each of the specified operations is compared to Free5GC, whereas the latency on the control plane was reduced by $\sim$2 times. A comprehensive tutorial about the Non-access stratum (NAS) and NG Application Protocol (NGAP) was conducted by *Dominato et al.* to understand the communication in 5G network segments and architecture components [6]. Functionality aspects such as authentication and identification procedures, data session establishment, and resource allocation along with the message flow between UE and Next Generation Node B (gNodeB) registration are reviewed. A my5GTester was developed for evaluating open-source 5G CN projects such as Free5GC, OAI 5G CN, and Open5GS using a black-box testing methodology. The Plat5G-Br project aimed to develop a 5G network-based disaggregation, virtualization, and interface standardization. A comparative study to select the most suitable platform to develop the 5G was carried out by *De Souza Neto et al.* between open-source, Magma, Open5GS, and Free5GC [1]. The platforms were analyzed based on infrastructure, documentation, community support, base projects, Control/User plane separation, and management. Based on the evaluation Magma was chosen for the further use case.

III. Standalone 5G Architecture

A typical cellular network architecture consists of Radio Access Networks (RAN) incorporated with the wireless channel and a backhaul network to connect mobile devices to the CN. The CN is responsible for providing Data Network and 'cellular services' to the UEs. A brief overview of the 5G architecture and the current state of the art of RAN are described in the following section.

A. 5G Core Architecture

Due to the significant benefits of SDN and NFV, the 5G CN functions are implemented as software rather than purpose-built hardware. Based on the concepts of microservices the CN is divided into control plane and user plane. The main network functionalities are Access and Mobility Management Function (AMF), Session Management Function (SMF), Authentication Server Function (AUSF), Policy Control Function (PCF), Unified Data Management (UDM), Application Function (AF), Data Network (DN), Network Slice Selection Function (NSSF), User Plane Function (UPF), RAN and UE. Compared to the point-to-point approach, the control plane utilizing CNF forms a service mesh providing inter-service communication using HTTP/REST API. In the user plane, the Packet Detection Rules (PDRs) are provisioned by UPF based on the decision made by SMF. Based on the priority, the UPF organizes the PDRs. The data is transferred between gNodeB and UPF over the so-called GPRS Tunnelling Protocol (GTP).

B. Radio Access Networks (RAN)

The major part of wireless communication is the RAN, which connects the UEs to the CN. It consists of Radio Unit (RU) and Processing Unit (PU) responsible for transmission, reception, radio management, and resource allocation. To fulfill the growing demand of users, the framework was adopted as RU and Distributed Unit (DU). Further, with the introduction of virtualization and an increase in the data application traffic, DU is cloudified containing a pool of network resources. This new architecture is referred to as Cloud Radio Access Network (CRAN) [8]. Additionally, the O-RAN Alliance addresses the radio access network domain and promised flexible, more open, and smarter architecture. This new framework leverages the functionalities of SDN and NFV technologies to integrate new interfaces [9].

IV. Open-source 5G platforms

The transition of 5G mobile networks is driven towards virtualization and SBA. A brief overview of existing open-source CN projects is presented within this section.

A. Free5GC

Free5GC is an open-source project built for the 5G CN. The initial implementation is based on NextEPC and further extends the concepts to the 5G CN defined in 3GPP Release 15 and beyond [10]. The standalone 5G using virtual machine based is released in Stage 3. Later, the approach is extended towards docker-based CN [11] and each of the components is individually deployed as a container.

B. OAI 5G CN

OAI 5G CN is an open-source implementation of the 3GPP specifications of 5G CN. It consist of network elements such as AMF, AUSF, Network Repository Function (NRF), SMF, UDM, Unified Data Repository (UDR), UPF, NSSF, Network Exposure Function (NEF), and PCF. The configuration is based on containers and provides flexibility to test the deployment using different RAN and UE simulators. The minimalistic deployment, which is used in this work, consists of CNF with AMF, SMF, NRF, vpp-UPF, UDR, UDM and AUSF [12].

C. Open5GS

The NR/LTE network is configured using the Open5GS project. The CN built is equipped with gNB/eNB and USIM to provide a private 5G network. Nodejs and React are used for the WebUI for adding subscriber information and performing the simulation [13].

D. RAN and UE simulators

To evaluate the deployment and functionality of the different 5G CNs, a RAN including UEs is a necessity. Here, there are multiple solutions available such as OAI RF simulator, gnbsim, UERANSIM [14], My5g-RANTester, omec-gnbsim, and commercial UE. For further use-case examination, UERANSIM is used, as it provides the flexibility to operate with all 5G platforms applicable in this current study.

UERANSIM is the world's first open-source 5G standalone UE and gNodeB simulator developed for testing and studying 5G systems. UE functionality with 3GPP access and RAN with the central unit are implemented via UDP protocol. There are various features supported in NAS and NGAP layers through this open-source project. Among others, these are primary authentication and key agreement, security mode control, identification, generic UE configuration update, initial and periodic registration, paging, PDU session resource setup and release, or initial UE messages.

V. Infrastructure Setup and Testbed design

The structure of the 5G SBA provides the feasibility to leverage virtualization benefits. The 5G CN is deployed with different strategies based on the open-source 5G platforms. The system requirements, software dependencies, and methodology for building the 5G CN, RAN, and UE are discussed in the following section.

A. Infrastructure

The platform's infrastructure depends on the elements required to be deployed into the network. They are implemented as CNF in a container-based system, using Docker. The virtual machines are created for each 5G CN with the interface using Oracle VirtualBox and each consisting of 8GB RAM, 4 virtual CPUs, with a disk of 40TB.

B. Core Network Deployment and UERANSIM

Initially, Docker and Docker Swarm software dependencies are installed. Each of the project repositories is locally cloned to build the setup. Additional requirements, such as GTP that is required by free5GC, are installed manually. Each virtual machine is evaluated individually with UERANSIM.

C. Overall Testbed Setup

Figure 1 indicates the overall picture of the testbed. The setup consists of a physical host with Ubuntu 22.04 enabling the hypervisor layer, whereas virtual machines are created using Oracle Virtual Box. The system contains three virtual machines called VM1, VM2, and VM3, each of those linked to a specific open-source implementation. Furthermore, each of the 5G CN functions runs as a separate container. During the evaluation process, the data is extracted with the application of Prometheus [15], cAdvisor [16], and Grafana [17] which are well-known monitoring and visualization tools.

VI. Evaluation and Results

The performance analysis is carried out on each of the open-source projects implemented on the testbed. The overall system equipped with the 5G network, RAN, UE, data extractors, and visualization is subjected to regress evaluation. Table I presents the summarized results and the comparative analysis results obtained in this work. The comparison is investigated as follows:

A. Qualitative Analysis

In order to select the most suitable open-source project for specific cases, several qualitative requirements have to be taken into account. First of all, licensing is a relevant topic, since it restricts the use and distribution of open-source software. Here, each of the 5G CN implementations is based on a different license, making it valid to integrate the requirement in the study. While the free5GC project is based on Apache 2.0 license that does not belong to the so-called copyleft licenses, Open5GS comes with the AGPL-3.0 license. Copyleft fuels open-source research, because each software that is based on a project including a copyleft license has to be made publicly available. On the other hand, if modified software should be distributed commercially, this is a constraint. Thus, OAI 5G CN is based on a heterogeneous

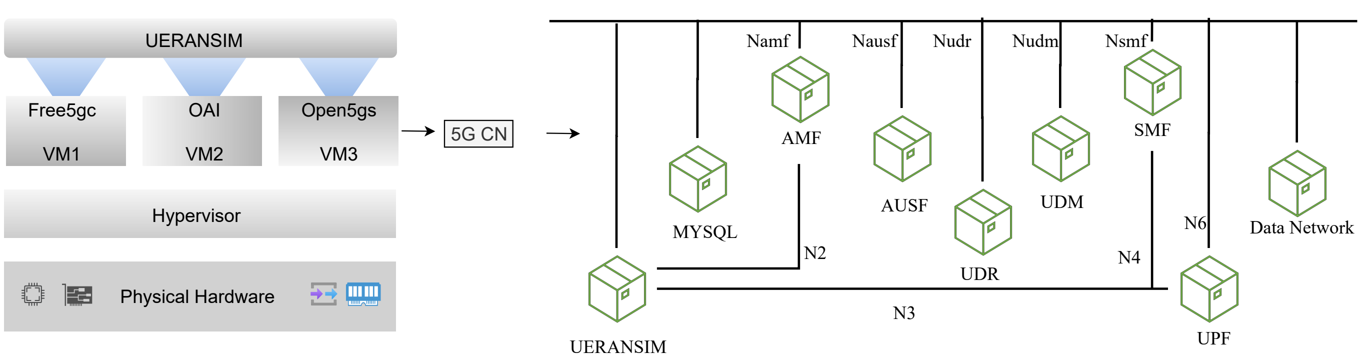


Figure 1: 5G testbed architecture comprising of the physical layer (physical hardware and hypervisor) and Virtual layer with open-source 5G CN implementation deployed as container

TABLE I: Qualitative and Quantitative Comparison of open-source Implementations

Specifications	free5GC	OAI 5G CN	Open5GS
License	Apache 2.0	OAI Public License V1.1	AGPL-3.0
Programming Language	Golang	C++	C
Infrastructure	Virtual Machine, Container	Container	Virtual Machine, Container
Relevance Ranking	3	1	2
No of UE instances	Average CPU Load Utilization in [%]		
1 UE	2.05	3.94	2.04
2 UEs	2.33	4.02	1.41
4 UEs	2.42	4.15	1.44
6 UEs	3.82	4.39	1.67
8 UEs	2.00	4.87	1.72
No of UE instances	Average Round Trip Times in [ms]		
1 UE	24.928	20.4577	23.911
2 UEs	25.618	16.04	23.2332
4 UEs	24.9146	15.643	24.0244
6 UEs	23.7929	20.171	24.63
8 UEs	24.382	35.165	24.95

license, called OAI Public License V1.1 which is based on the Apache 2.0 license for research and non-profit usage and extended for commercial use. Thus, it is also feasible for big companies to contribute to these kinds of projects, without having conflicts with their own patents.

Furthermore, each 5G CN implementation is realized in a different programming language. While OAI 5G CN and Open5GS are programmed hardware-close (C/C++), free5GC is based on Golang which benefits from the use of web services. The infrastructure of the 5G CN platforms is deployed either as a virtual machine or container-based service. Free5GC and Open5GS provide adaptability for virtual machines and container deployment. While OAI 5G CN supports only the container environment. Both the implementation procedures have positive and negative benefits. Wherein, the container provides better isolation and performance while the virtual machine will have additional overhead. In the case of free5GC virtual machine implementation, on the occurrence of a failure, the entire stack of the CN should be restarted, while in the case of a container environment, individual network functions can be restarted, while maintaining other services operational.

Last but not least, relevance is an important criterion, since it describes the success rate of each project. Following, the number of scientific papers referring to the corresponding project and number of contributors, OAI 5G CN is the most relevant, followed by Open5GS and free5GC. But in recent years, Open5GS and free5GC have gained attention, since it provides the feasibility to deploy the CN both as a virtual machine and container service. Nevertheless, based on the analysis, OAI 5G CN followed by Open5GS, and then Free5GC will be the relevant order to acknowledge the growing research works.

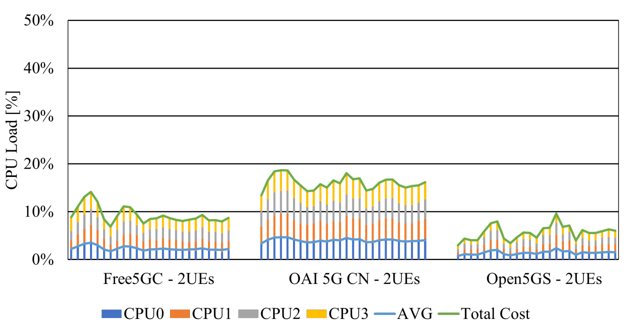
B. *Quantitative Analysis*

The resource consumption is investigated in the proposed testbed. To avoid additional inference caused by multiple virtual machines, individual projects are evaluated at a given point in time. Algorithm 1 describes the load evaluation procedure. Firstly, the dependencies and monitoring tools are initialized. Docker-compose YAML file is deployed and 5G CN is created. The end user performs activities while accessing the external network after registration. In parallel, a RTT test is carried out. The overall time taken for the UE to reach the data network is calculated. During this activity, the CPU load utilization is captured and audited for a comparative study. The evaluation procedure is repeated for multiple UE instances such as 1, 2, 4, 6, and 8.

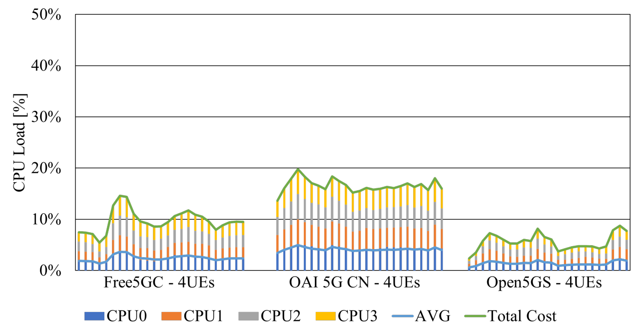
Figure 2 presents the statistics of the load consumption on each CPU, average load, and total cost occurred for the aforementioned number of UEs. The average CPU utilization ranges approximately between 2-4%, 3.5-5%, and 1.4-2.5% for free5GC, OAI 5G CN, and Open5gs respectively. Figure 2(a) shows the load during runtime of 2 UEs, Open5GS consumes fewer resources compared to OAI 5G CN and free5GC. Similarly, Figure 2(b) and (d), indicate that resource usage grows higher in the case of OAI 5G CN implementation with the growth of the UE count, while Open5GS remains stable without higher fluctuations. But the behavior of free5GC varies during the presence of 6 UEs, initially, the high load is caused due to the regress registration of the end-user, which gradually

Algorithm 1 Load Evaluation Procedure

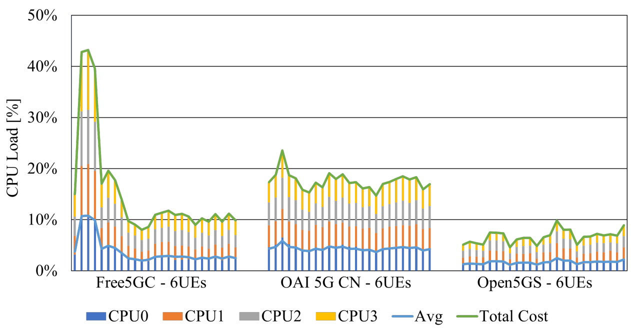
Procedure for Load Evaluation :
Main:
    Start the monitoring tools
    Start the CN
    Initialize UE
    Perform end-user activities
    Run RTT test
    Extract the Load Utilization
close;



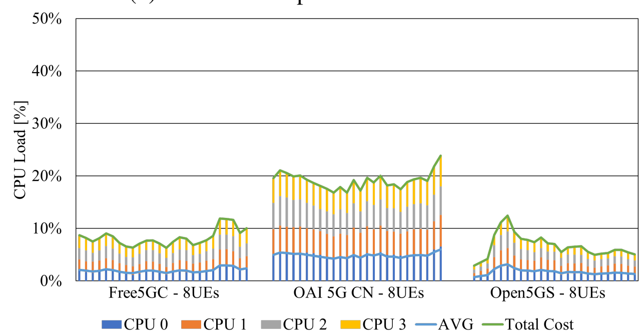
(a) Load consumption with 2 UE instances



(b) Load consumption with 4 UE instances



(c) Load consumption with 6 UE instances



(d) Load consumption with 8 UE instances

Figure 2: Statistics of the load Utilization on each CPU core operating with free5GC, OAI 5G CN, Open5GS with UE instances of 2, 4, 6, and 8.

reduces and remains stable through the latency analysis. The overall metric details for average CPU load utilization for each of the instances are presented in Table I.

The steps followed for the RTT evaluation are described in Algorithm 2. Initially, UE is registered into the CN. During the analysis, the overall time taken for the UE to reach the external network and receive the answer is recorded consecutively. The process is repeated for *n* running in the system.

Algorithm 2 RTT Evaluation Procedure

Procedure for RTT Evaluation :
n ← *number of UE*
i ← *1*
Loop:
    Access the external network
    Receive Response
    Record samples for every 1 second
    if *(i == n)* **then**
        i ← *i+1*
        goto *Loop*
    end if
close;

Figure 3 indicates the box plot of the RTT metrics generated during the presence of a single end-user. OAI 5G CN provides an efficient network performance compared to free5GC and Open5GS when viewed on a large scale of data samples recorded. But in contradiction, response time in multiple

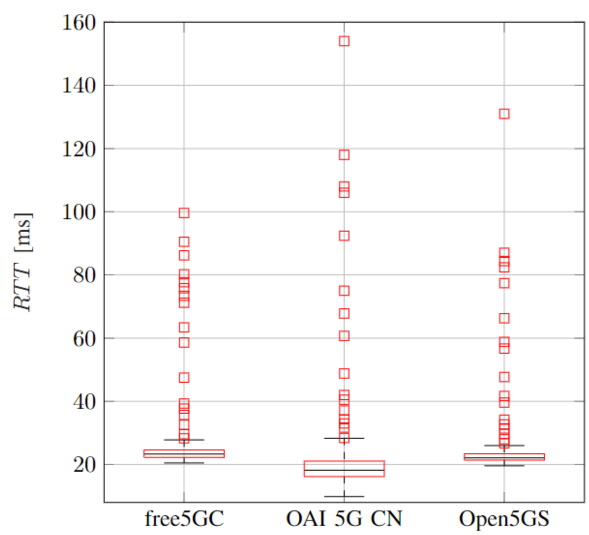


Figure 3: Round Trip time of the open-source projects indicating the statistics generated during the presence of a single end-user.

instances grows higher in OAI 5G CN compared to other implementations. As stated in Table I, the average time taken by free5GC, OAI 5G CN, and Open5GS are 23.7-25.6ms, 15.6-35ms, and 23.23-24.9ms respectively.

To briefly summarize the overall performance analysis, there is always a trade-off between CPU load utilization and achieved latency. In the case of OAI 5G CN, the load consumption grows higher, while providing better performance. Open5GS maintains its consistency throughout the evaluation irrespective of the UE scaling. Finally, free5GC performs similarly to Open5GS with moderate load fluctuations and sustainable response time for all end-users.

VII. Conclusion and Outlook

In order to provide greater flexibility to integrate new functions and dynamic agile architecture, mobile networks are undergoing a transition from point-to-point communication to SBA. This adaptation leads to the creation of multiple communities to develop open-source 5G CNs addressing the main profiles of 5G, i.e. enhanced Mobile broad-Band (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and MMTC. To understand the behavior of open-source projects, a comparative study is conducted on free5GC, OAI 5G CN, and Open5GS. A testbed is designed, and the corresponding projects are deployed in separate virtual machines. Each virtual machine consists of 5G system running network functions in a container environment. Platforms are individually investigated based on resource utilization and performance in the presence of multiple end-user devices. The evaluation is performed in a broad aspect of quantitative and qualitative performance. While the qualitative comparison is based on licensing, programming language, deployment possibilities, and relevance, the qualitative metrics are resource utilization and the RTT experienced by end-user devices.

Acknowledgment

This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 16KIS1283, AI-NET PROTECT and Förderkennzeichen 16KISK003K, Open6GHub). The authors alone are responsible for the content of the paper.

References

[1] F. J. De Souza Neto, E. Amatucci, N. A. Nassif, and P. A. Marques Farias, "Analysis for Comparison of Framework for 5G Core Implementation," in *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, 2021, pp. 1–5. DOI: 10.1109/ICISCT52966.2021.9670414.

[2] G. Brown, "Service-based architecture for 5g core networks," *Huawei White Paper*, vol. 1, 2017.

[3] Ericsson, *Evolving from EPC to 5GC: A guide for smooth evolution into the cloud-native 5G Core*, Stockholm, Sweden, 2022.

[4] S. Baradie, R. Reddy, C. Lipps, and H. D. Schotten, "Managing the Fifth Generation (5G) Wireless Mobile Communication: A Machine Learning Approach for Network Traffic Prediction," in *Mobile Communication - Technologies and Applications; 26th ITG-Symposium*, 2022, pp. 1–6.

[5] R. Reddy, S. Baradie, M. Gundall, C. Lipps, and H. D. Schotten, "CPU Resource Resilience in Wireless Mobile Communications: Design and Evaluation on COTS Virtual Distributed Platform," in *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2022, pp. 43–48. DOI: 10.1109/BlackSeaCom54372.2022.9858313.

[6] L. B. Dominato, H. C. de Resende, C. B. Both, J. M. Marquez-Barja, B. O. Silvestre, and K. V. Cardoso, "Tutorial on communication between access networks and the 5G core," arXiv, 2021. DOI: 10.48550/ARXIV.2112.04257.

[7] V. Jain, H.-T. Chu, S. Qi, C.-A. Lee, H.-C. Chang, C.-Y. Hsieh, K. K. Ramakrishnan, and J.-C. Chen, "L25GC: A Low Latency 5G Core Network Based on High-Performance NFV Platforms," in *Proceedings of the ACM SIGCOMM 2022 Conference*, ser. SIGCOMM '22, Amsterdam, Netherlands: Association for Computing Machinery, 2022, pp. 143–157. DOI: 10.1145/3544216.3544267.

[8] S. K. Singh, R. Singh, and B. Kumbhani, "The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6. DOI: 10.1109/WCNCW48565.2020.9124820.

[9] B. Brik, K. Boutiba, and A. Ksentini, "Deep Learning for B5G Open Radio Access Network: Evolution, Survey, Case Studies, and Challenges," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 228–250, 2022. DOI: 10.1109/OJCOMS.2022.3146618.

[10] Free5GC, https://www.free5gc.org/, Jan. 2023.

[11] Free5GC git, https://github.com/free5gc/free5gc-compose, Jan. 2023.

[12] OAI 5G CN, https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed, Jan. 2023.

[13] Open5GS, https://open5gs.org/, Jan. 2023.

[14] UERANSIM, https://github.com/aligungr/UERANSIM, Jan. 2023.

[15] Prometheus, https://prometheus.io/.

[16] cadvisor, https://github.com/google/cadvisor, Oct. 2022.

[17] Grafana, https://grafana.com/.