A Versatile 5G Standalone Testbed Based On Commodity Hardware

Emmanouil-Zafeirios G. Bozis
Department of Informatics and Telecommunications
University of Peloponnese
Tripolis, Greece
mbozis@go.uop.gr

Nikos C. Sagias
Department of Informatics and Telecommunications
University of Peloponnese
Tripolis, Greece
nsagias@uop.gr

Michael C. Batistatos
Department of Informatics and Telecommunications
University of Peloponnese
Tripolis, Greece
mbatist@uop.gr

Michail-Alexandros Kourtis
Institute of Informatics and Telecommunications
National Centre for Scientific Research “DEMOKRITOS” (NCSRD)
Athens, Greece
akis.kourtis@iit.demokritos.gr

George K. Xilouris
Institute of Informatics and Telecommunications
National Centre for Scientific Research “DEMOKRITOS” (NCSRD)
Athens, Greece
xilouris@iit.demokritos.gr
& National and Kapodistrian University of Athens,
Department of Port Management and Shipping,
Dirfies Messapies,Greece

Anastasios Kourtis
Institute of Informatics and Telecommunications
National Centre for Scientific Research “DEMOKRITOS” (NCSRD)
Athens, Greece
kourtis@iit.demokritos.gr

***Abstract*—The fifth generation (5G) mobile networks have adapted new features compared to the previous generation that resulted in increased overall throughput and decreased latency. At the same time the complexity of 5G-based standards and deployment scenarios has increased. The new features of the physical layer gather interest not only in the development of software, but also in the field of measurements and performance evaluation, among the research community. This paper proposes a flexible End-to-End (E2E) standalone 5G system platform that can be fully reprogrammable and customizable to serve as a basis for research that can be expanded to all aspects of a 5G system design. The main capabilities and use cases of the platform are presented and conclusions are made based on the conducted tests and measurements.**

Keywords—Software Defined Radio, 5G, open source software, Open Air Interface, Linux

I. INTRODUCTION

The implementation of 5G Radio Access (RAN) and Core Network (CN) based on commodity computer hardware and Software Defined Radio (SDR) devices facilitated the development of 5G systems for reasearch or private industrial mobile networks. The software for these implementations is available as open source making a significant contribution to the cost reduction combined with the use of commercial-off-the-shelf (COTS) hardware.

The open source software solutions for 5G RAN include srsRAN [1] and Open Air Interface (OAI) [2]. Another implementation of 5G system in COTS hardware is Amarisoft [3], but is closed source and offered as a combined software-hardware solution. For the 5G core network the main open source implementations are Open5GS [4] and OAI CN5G [5].

Among the open source solutions OAI is the one that is more active in development gathering interest in 3GGP 5G protocol stack implementation from the mobile network operators, the RF hardware vendors and the academia. The design concepts and features of OAI code base are presented in [6] and [7]. OAI in contrary to it’s competitors offers E2E 5G system implementation including the software for user equipment (nrUE), CN and 5G base station (gNB) having the maximum flexibility and customizability but at the cost of increased difficulty in configuration and not sufficient documentation for all use cases. All software components of OAI from UE and gNB softmodem programs to Core Network Functions (NF) can run on Linux OS or as programs inside containers. Apart form the ability to transmit and receive 5G waveforms using SDR devices, the OAI has several software tools known as Phy Simulators [8], that can be utilized to develop a 5G system emulation platfrom. Thus, a 5G E2E OAI system emulation can be conducted using docker containers deployed in a single host [9].

In this work we aim at devoping a standalone 5G system platform based on open source software and commodity hardware that facilitates the user to test different connectivity scenarios and use cases. For the implementation of the 5G protocol stack we selected to use the OAI for all 5G components including the gNB, CN and nrUE. We exploited the Linux ecosystem with the plethora of open source tools available and bash scripting, in order to implement our connectivity scenarios and automate the proccess of starting up processes and deploying docker containers.

A. Related Work

The establishment of the afforementioned open source software has led to the development of 5G prototype systems and testbeds. A series of experiments on a 5G non-standalone (NSA) platform using srsRAN and OAI have been conducted and the comparison between these two platforms in terms of experienced throughput and latency, have been presented in [10]. Authors in [11] present a portable demonstrator 5G NSA system based on OAI software that integrates in a modular way open source software and commodity SDR devices. A similar Network-in-a-Box solution for private deployments with evaluation results was presented in [12]. In [13] a demonstration and evaluation of a 5G system emulation platform using a single host in conjunction with other open source tools was presented. A study on the use cases and the respective cost

979-8-3503-1884-5/24/$31.00 ©2024 IEEE

for Cellular Industrial Communication based on open source software and different SDR devices can be found in [14]. A standalone 5G testbed was proposed by authors of [15] that enables researchers to study the various threat based features by capturing packets and RF signals from the physical layer and up. In [16] the open-source mobile network stacks were evaluated including a 5G NSA testbed with srsRAN and OAI and a 5G SA testbed with srsRAN.

B. *Contribution*

With the proposed testbed the user can conduct both 5G SA system emulation and wireless communication to test different scenarios and use cases. Based on the initial concept the system can be expanded and upgraded using extra hardware and software tools to meet the needs of a specific research. The key benefits of our testbed are:

- gNB configuration (Frequency Band, Bandwidth Parts (BWPs), Time Division Duplex (TDD) slot configuration, numbers of antennas, etc) can be initially tested using the RF Simulator prior to a wireless transmission with SDR devices, thus facilitating quick experimentation and as well as debugging possible errors.

- the process of implementing a specific scenario, starting up the software components, generating network traffic and conducting measurements is automated with the bash scripts we developed by incorporating open source tools for the Linux OS.

- the cost level of the platform is the lowest possible depending only on the cost of COTS hardware, as all software components are available for free.

C. *Paper Structure*

The paper is structured as follows: Section II discusses the design implementation of the our proposed platform, it’s features and operation modes. Section III presents the performed tests, results and analysis. Finally the paper is concluded in Section IV discussing the main findings and future work.

II. OPEN SOURCE VERSATILE 5G PLATFORM

A. *Description of the Platform*

The proposed testbed encompass two powerful desktop personal computers (PC), two USRP N310 SDR devices, RF signal attenuators and all the necessary RF and network cables for the interconnection of the devices, as depicted in Fig. 1. The OAI gNB and CN are installed on the first PC host, while the second one runs the nrUE. In the case of running OAI software with RF Simulator, the two hosts are connected with a 10Gbps network connection.

The RF simulator is a server client add on module that bypasses the SDR devices and transports the baseband signal as I/Q samples directly from host to host. The USRP devices are connected to the hosts with a separate 10Gbps network connection through cat6 ethernet cables. The HG version of the pre-built FPGA image is installed in both SDR devices and supports a maximum network speed of 10Gbps in port 1 of SFP+ ports. The two desktop PCs have the same 16 core Intel i9-12900K CPU but different RAM size of 16Gb and 32Gb respectively. Ubuntu 22 OS is installed in both PCs. We tested the system efficiency with two different Linux kernels: 6.2.0-35-generic and 5.15.0-87-lowlatency. The later one was our final choice mainly due to the fact that with the generic kernel we faced stability issues and errors in running processes. The OAI software for the gNB and nrUE was downloaded from the Eurecom’s repository [17] and compiled using the guide for 5G E2E standalone system implementation which is available in [18]. In our testbed we used the latest develop branch of OAI RAN software, which at the time of installing the software components, was the integration_2023_w40.

The aforementioned hardware and software setup allows for two distinct operations of the OAI 5G SA system. The first option is to transmit and receive the RF signal with the SDR devices and the other one is to use the RF simulator which transports the I/Q samples over the direct ethernet connection between the two hosts. The benchmark tests we conducted with USRP driver (UHD) run successfully up to 62.50 Msamples/sec between the host and the SDR device. We configured the gNB to use frequency band 78 of 5G NR with 51 and 106 Physical Resource Blocks (PRB) with a sub carrier spacing (scs) of 30KHz which corresponds to a bandwidth of 20 and 40MHz respectively. Currently OAI supports scs of up to 30KHz in FR1 5G NR bands and we used this setting for all our tests.

The OAI core network implementation compounds the Access and Mobility Management Function (AMF), the Session Management Function (SMF), the Unified Data Management (UDM), the Unified Data Repository (UDR), the Authentication Server Function (AUSF), and the Serving and Public Data Network Gateway (SPGWU). The latest was firstly intended as a 4G Core Network User Plane Function but now serves also as a 5G User Plane Function (UPF). A functional description of the Network Functions (NF) can be found in section 6.2 of [19].

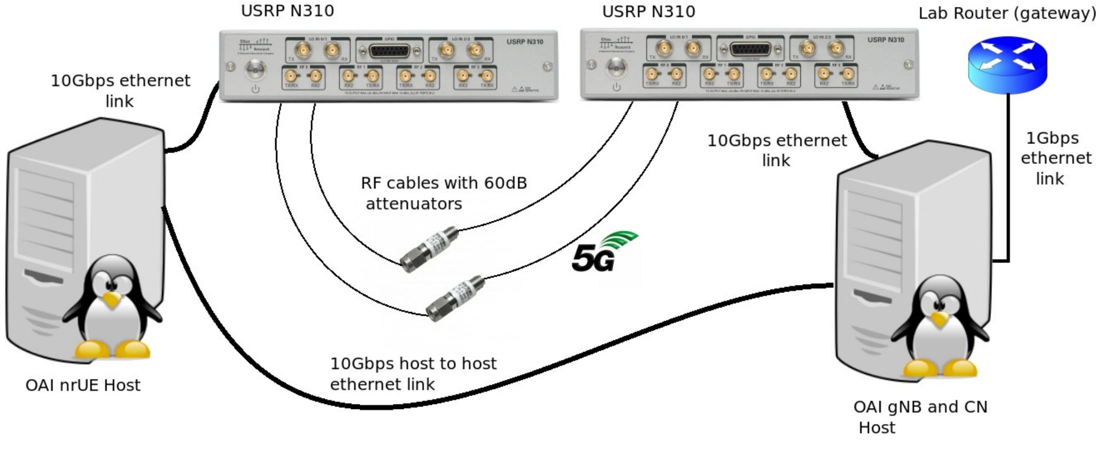

Fig. 1. Schematic Diagram of the 5G testbed

All NFs run as microservices inside docker containers on the gNB host. The version of the OAI CN docker containers used in our testbed is 1.5.1. Apart from the aforementioned containers, two more containers are deployed: EXT-DN that serves as the external data network and myqsl, which is the database server where the data for authentication subscription, access and mobility and AMF access registration are stored. The configuration of all CN containers is stored in a YAML formatted file and the deployment is done with the docker-compose tool.

In order for the 5G system to meet the real-time requirements, we made several configurations to the BIOS and OS of host PCs. We disabled hyper-threading in BIOS and we increased the Linux OS kernel socket and ethernet ring buffers. Also for the 10Gbps network interface controllers, the value of for MTU should be set to 9000 instead of 1500 that is the default. In regards to CPU settings, CPU frequency scaling governor must be set to performance and lock processor frequency to maximum (only C0 state active).

B. Platform Operation Modes

The user starts the operation by deploying the 5G Core NF containers on the gNB host. Then it configures the Linux OS parameters, the CPU settings for maximum performance and checks connectivity to AMF container. The afforementioned steps are done by running the bash shell script we developed to automate this process. Then, user runs the nr-softmodem program. This program implements the 5G radio interface and all the other functions of gNB. It starts by making all necessary configurations to the NG Interface and initializing the SDR device (USRP N310). In the case of using the RF Simulator, the nr-softmodem starts a udp server listening for incoming connections from the UEs. Through this connection the I/Q data is transported to and from the gNB. Following a successful registration of a UE to the AMF a PDU session is created. The nr-uesoftmodem program is the 5G software modem executed on the UE host. After the successful PDU session esatblishment, nr-uesoftmodem creates a tunnel-type network interface with the assigned IP address from the CN (IP range 12.1.1.2-254). Through this interface the network packets traverse the 5G protocol stack. In order to connect the host to internet through the OAI 5G network, the default gateway is set to 12.1.1.1. All tests and measurements can be conducted executing the bash scripts we developed to automate the process [20].

III. PERFORMED TESTS AND ANALYSIS

A. Using RF Simulator

We checked 3 different initial BWPs using the RF simulator transporting the I/Q samples directly form gNB host to UE host as shown in Table I. We configured in gNB the subcarrier spacing to 30KHz. In each case only 1 initial BWP was configured, which remained the same for the whole connection time of UE to the 5G network. In all tested configurations OAI nrUE registered to the AMF and a PDU session was created. We checked the data session creating traffic in downlink and uplink with the iperf tool. In the case of 100MHz bandwidth we encountered process errors when we tried to use very high throughput consuming all of our available computational resources. We don't present any measurements of maximum throughput or latency for this case.

TABLE I. GNB CONFIGURATIONS WITH RF SIMULATOR

Number of PRBs in BWP	*Subcarrier spacing (KHz)*	*Band / No of Antennas*	*Downlink-Uplink Bandwidth (MHz)*
51	30	78 (SISO)	20
106	30	78 (SISO)	40
273	30	77 (2x2 MIMO)	100

The reason is that in RF simulator mode the timing of frames is not exact, as it is the case when the OAI software is interfaced to an SDR device. In fact when we increased the PRBs form 51 to 106 keeping all other parameters the same, we experienced a higher latency denoting that the software was running slower due to increased computational load in the case of the 106 PRB BWP.

B. Using the SDR Devices

For the first two configurations of gNB (51 and 106 PRB in Table II) we conducted measurements connecting the SDR devices to hosts. The RF link between the two SDRs was achieved using RF cables and 60 dB attenuators to adjust the signal power to the acceptable level of USRP inputs and to minimize the interference from other signals. In the case of 273 PRBs our computer hardware could not meet the real-time requirements and no measurements were possible to be made. The maximum throughput in uplink and downlink was optained running the iperf server and client applications on UE host and oai-ext-dn container. The later plays the role of the external data network. The values we measured for the achievable throughput are presented in Table II.

TABLE II. MAXIMUM THROUGHPUT VALUES

No. of PRBs in BWP	*Sampling Rate (Msamples /sec)*	*RSRP (dBm)*	*Band / No of Antennas*	*Downlink-Throughput (Mbps)*	*Uplink-Throughput (Mbps)*
51	30,72	-107	78 (SISO)	30.1	16.0
106	61.44	-112	78 (SISO)	66.7	30.9
273	n/a	n/a	77 (2x2 MIMO)	n/a	n/a

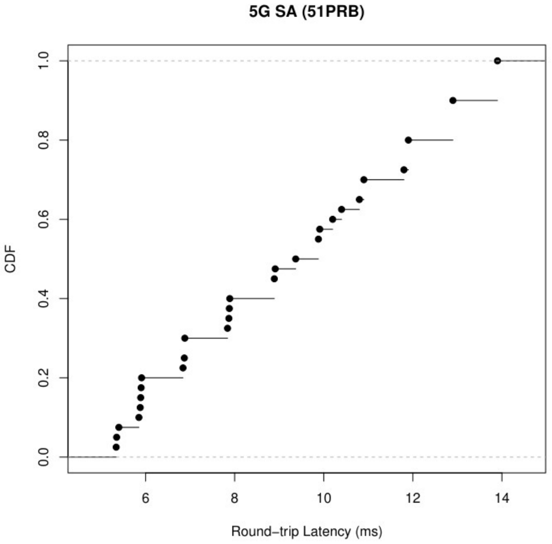

Fig. 2. CDF of round-trip latency for 51 PRB Bandwidth

We also measured 40 values of the round-trip latency using the ping command from the UE host to the lab router (gateway). The cumulative distribution functions (CDF) of these values are depicted in Figs 2 and 3 for the case of 51 and 106 PRB bandwidth part respectively.

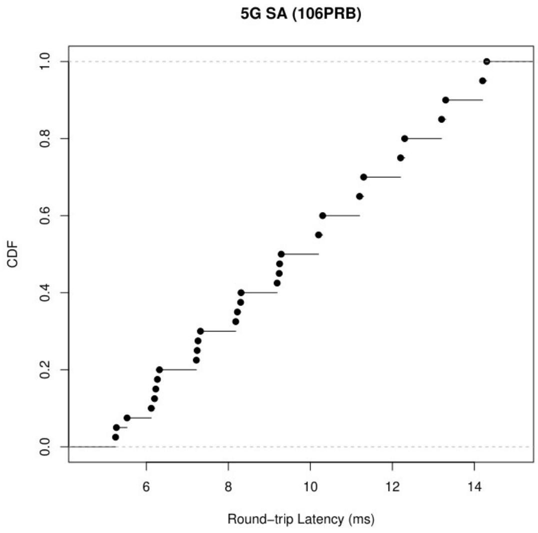

Fig. 3. CDF of round-trip latency for 106 PRB Bandwidth

The mean values of latency were very close in both cases (9.45 and 9.77 ms), since the TDD DL/UL slot configuration and numerology that affects the latency remained the same.

IV. Conclusions And Future Work

We showcased a versatile 5G SA system capable of conducting both emulations and wireless connection tests. By incorporating open source tools into bash scripts, we managed to startup all 5G componets, make all necessary network configurations and finally check connectivity throught the 5G protocol stack and perform measurements. We provide a repository [20] containing all the configuration files and bash scripts we developed, facilitating reproducibility of our testbed. We will keep this repository updated as we will expand the testbed with new connection scenarios in the future.

However there are still some limitations to our approach. The OAI software does not support all possible modes of operation and has some known bugs. For example the list of supported COTS UEs is limited and newer models with Android 13 OS are not supported for the time being. Furthermore, the configuration of system parameters is challenging and the documentation for some use cases is limited.

For future work we are planning to expand our testbed capabilities performing different connectivity scenarios with MIMO transmission and greater BWPs. Furthermore we plan to connect a COTS UE device and adjust the system parameters to improve the KPIs.

V. Acknowledgement

This research initiative is supported by the European Union’s Horizon Europe Framework Program for Research and Innovation, under the OASEES project (Grant Agreement No. 101092702). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union, which cannot be held responsible for them.

References

[1] srsRAN Project, https://www.srslte.com/ , Accessed Nov 1, 2023.

[2] Open Air Interface, https://openairinterface.org/, Accessed Nov 1, 2023.

[3] Amarisoft, https://www.amarisoft.com/, Accessed Nov 1, 2023.

[4] Open5GS, https://open5gs.org/, Accessed Nov 1, 2023.

[5] Repository for the Open Air Interface Core Network development, https://gitlab.eurecom.fr/oai/cn5g, Accessed Nov 1, 2023.

[6] Kaltenberger, Florian & Silva, Aloizio & Gosain, Abhimanyu & Wang, Luhan & Nguyen, Tien-Thinh. (2020). OpenAirInterface: Democratizing Innovation in the 5G Era. Computer Networks. 176. 107284. 10.1016/j.comnet.2020.107284.

[7] W. T. Han and R. Knopp, "OpenAirInterface: A Pipeline Structure for 5G," *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, Shanghai, China, 2018, pp. 1-4, doi: 10.1109/ICDSP.2018.8631835.

[8] Introduction to 5G Phy Simulators in Open Air Interface, https://openairinterface.org/introduction-to-5g-ran-phy-simulators-in-openairinterface/ Accessed 1 Nov 2023

[9] OAI full stack 5G NR RF simulation with containers, https://gitlab.eurecom.fr/oai/openairinterface5g/-/tree/develop/ ci-scripts/yaml_files/5g_rfsimulator.

[10] R. Mihai, R. Craciunescu, A. Martian, F. Y. Li, C. Patachia and M. -C. Vochin, "Open-Source Enabled Beyond 5G Private Mobile Networks: From Concept to Prototype," *2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Herning, Denmark, 2022, pp. 181-186, doi: 10.1109/WPMC55625.2022.10014829.

[11] P. Matzakos, H. Koumaras, D. Tsolkas, M. Christopoulou, G. Xilouris and F. Kaltenberger, "An open source 5G experimentation testbed," *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, Athens, Greece, 2021, pp. 1-2, doi: 10.1109/MeditCom49071.2021.9647647.

[12] A. Aijaz, B. Holden and F. Meng, "Open and Programmable 5G Network-in-a-Box: Technology Demonstration and Evaluation Results," *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, Tokyo, Japan, 2021, pp. 369-371, doi: 10.1109/NetSoft51509.2021.9492719.

[13] E. -Z. G. Bozis, M. C. Batistatos and N. C. Sagias, "Overview Of SDR Platforms Based On Open Source Software: A 5G System Emulation With Open Air Interface," *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*, Tripolis, Greece, 2022, pp. 1-5, doi: 10.1109/PACET56979.2022.9976373.

[14] A. I. Grohmann, M. Seidel, C. Lehmann, T. Höschele, M. Reisslein and F. H. P. Fitzek, "5G on the Cheap: Configurable Low-Cost Cellular Industrial Communication," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, 2022, pp. 1-6, doi: 10.1109/ICECCME55909.2022.9988703.

[15] A. Gabrielson, K. Bauer, D. Kelly, A. Kearns and W. M. Smith, "CUE: A Standalone Testbed for 5G Experimentation," *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, 2021, pp. 745-750, doi: 10.1109/MILCOM52596.2021.9653117.

[16] M. Chepkoech, N. Mombeshora, B. Malila and J. Mwangama, "Evaluation of Open-Source Mobile Network Software Stacks: A Guide to Low-cost Deployment of 5G Testbeds," *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*, Madonna di Campiglio, Italy, 2023, pp. 56-63, doi: 10.23919/WONS57325.2023.10061896.

[17] 4G LTE and 5G Radio Access Network code repository Eurecom, https://gitlab.eurecom.fr/oai/openairinterface5g/ Accesed Nov 1 2023.

[18] Open Air Interface 5G NR SA tutorial with OAI nrUE , https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/NR_SA_Tutorial_OAI_nrUE.md?ref_type=heads

[19] 3GPP TS 23.501 Version 17.10.3.0 Release 17, availabe at: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/17.10.00_60/ts_123501v171000p.pdf

[20] Repository for scripts and configuration files of OAI 5G testbed, https://github.com/mbozis/OAI_5G_scripts