Int. J. Electron. Commun. (AEÜ) 187 (2024) 155515

Contents lists available at ScienceDirect

International Journal of Electronics and Communications

journal homepage: www.elsevier.com/locate/aeue

Enhancing 5G performance: A standalone system platform with customizable features

Emmanouil-Zafeirios G. Bozis [a], Nikos C. Sagias [a], Michael C. Batistatos [a], Michail-Alexandros Kourtis [b], George K. Xilouris [b,c], Anastasios Kourtis [b]

[a] Department of Informatics and Telecommunications, University of Peloponnese, Tripolis, Greece
[b] Institute of Informatics and Telecommunications, National Centre for Scientific Research “DEMOKRITOS” (NCSRD), Athens, Greece
[c] National and Kapodistrian University of Athens, Department of Port Management and Shipping, Dirfies Messapies,Greece

ARTICLE INFO

Keywords:
Software Defined Radio
5G
Open-source software
Open Air Interface
Linux

ABSTRACT

The fifth generation (5G) mobile networks have introduced 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 is increasing. This further drives the development of software for testing new ideas and different 5G and Beyond-5G use cases in academia and industry. In this paper we propose a flexible end-to-end (E2E) standalone 5G system platform based on the Open Air Interface software, which is fully reprogrammable and customizable. We created a set of Linux shell scripts, which improves the ease of use of the software, speeding up the process of implementing different connectivity scenarios. We present the main capabilities of the platform and the achievable throughput and latency for different bandwidth parts. Setting the DL/UL Transmission periodicity to 2 ms, we measured a latency of 6.78 ms for the mean RTT value, which is acceptable for many applications requiring low latency.

1. 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 research or private industrial mobile networks. The software for some of 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 the 5G protocol stack in COTS hardware is Amarisoft [3], but the software in this solution is proprietary. 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 academia. The design concepts and features of OAI code base are presented in [6] and [7]. OAI, in contrast to its alternatives, offers E2E 5G system implementation including the software for the user equipment (nrUE), CN and 5G base station (gNB). Furthermore, it has the maximum flexibility and customizability, but at the cost of increased difficulty in configuration and insufficient documentation for all use cases. The OAI RAN (gNB and UE) software components are executable programs that run on Linux OS and function as 5G software modems. The OAI 5G Core Network Functions (NF) act as interconnected services that run inside containers. Apart from the ability to transmit and receive 5G waveforms using SDR devices, the OAI has several software tools known as Phy Simulators [24], that can be exploited to develop a 5G system emulation platform. Thus, a 5G E2E OAI system emulation can be conducted using docker containers, deployed in a single host [8].

In this work we aim at developing a standalone 5G system platform based on open-source software and commodity hardware that facilitates testing 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 tools available and shell scripting, to implement our connectivity scenarios and automate the process of starting up processes and deploying docker containers.

E-mail addresses: mbozis@go.uop.gr (E.-Z.G. Bozis), nsagias@uop.gr (N.C. Sagias), mbatist@uop.gr (M.C. Batistatos), akis.kourtis@iit.demokritos.gr (M.-A. Kourtis), xilouris@iit.demokritos.gr (G.K. Xilouris), kourtis@iit.demokritos.gr (A. Kourtis).

https://doi.org/10.1016/j.aeue.2024.155515
Received 10 June 2024; Accepted 4 September 2024
Available online 5 September 2024

1.1. Related work

The establishment of 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, has been presented in [9]. Authors in [10] present a portable demonstrator 5G NSA system based on OAI software that integrates open-source software and commodity SDR devices in a modular way. A similar Network-in-a-Box solution for private deployments with evaluation results was presented in [11]. In [12] a demonstration and evaluation of a 5G system emulation platform using a single host in conjunction with other open-source tools were presented. A study on the use cases and the respective cost for Cellular Industrial Communication based on open-source software and different SDR devices, can be found in [13]. A standalone 5G testbed was proposed by authors of [14], that enables researchers to study the various threat-based features by capturing packets and RF signals from the physical layer and up. In [15], the open-source mobile network stacks were evaluated, including a 5G NSA testbed with srsRAN and OAI and a 5G SA testbed with srsRAN. The authors of [16] present a comparative analysis of TDD 5G Standalone testbeds, composed of different RAN components.

1.2. Motivation

Different 5G testbeds have already been proposed in the literature based on the available open-source solutions [9–15] and guidelines for the development have been discussed especially in [15–17]. However, implementing different connectivity scenarios still requires a lot of effort to configure and run [16,17]. The motivation for our work was to address this problem and improve the ease of use of the OAI software, by automating repetitive tasks and checks related to system configuration and use. Based on the available and already tested configurations in our work, the system can be expanded and upgraded, using extra hardware and/or software, to meet the needs of specific research.

1.3. Contribution

The contributions of this paper are as follows:

- To the best of our knowledge, this is the first study that reduces the difficulty of configuring and running the OAI 5G system implementation, by automating the process of starting up the OAI software components and conducting measurements. This is achieved with the shell scripts we developed, by incorporating open-source tools for the Linux OS. Therefore, the interaction with the OAI software is improved, automating the repetitive tasks and the necessary checks in the 5G protocol stack.
- With the proposed 5G testbed, the user can easily switch between RF simulation and wireless connection of the UE to the 5G network, using the same configuration file in gNB, thus facilitating quick experimentation and testing of new scenarios, prior to a wireless transmission. Exploiting this functionality, we expand the system throughput and latency measurements already presented in the literature [9–11,15–17], to bandwidth parts of 50 and 60 Mhz.

1.4. Paper structure

The paper is structured as follows: Section 2 discusses the design implementation of our proposed platform, its features and operation modes. Section 3 presents the tests performed, results, and analysis. Finally, the paper is concluded in Section 4 discussing the main findings and future work.

2. Open source versatile 5G platform

2.1. Description of the platform

The proposed testbed encompasses 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 between 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 exploit the direct 10Gbps network connection between them.

The RF simulator is a server client add-on module, that bypasses the SDR devices and transports directly the baseband signal, as I/Q samples from one host to the other. 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 the two SFP+ports of the device. The two desktop PCs have the same 16 core Intel i9-12900 K CPU, but different RAM sizes of 16 Gb and 32 Gb each. Ubuntu 22 OS is installed on both PCs. We tested the system efficiency with two different Linux kernels: generic and lowlatency. The latter one was our final choice, mainly because with the low latency kernel we faced less stability issues in OAI running processes. The OAI software for the gNB and nrUE was downloaded from the Eurecom's repository [18] and compiled using the guide for 5G E2E standalone system implementation which is available in [19]. In our testbed we used the latest development branch of OAI RAN software, which at the time of installing the software components, was the integration_2024_w15.

The 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 used the frequency band 78 of 5G NR with BWPs of 51,106,133 and 162 Physical Resource Blocks (PRB) with a SubCarrier Spacing (SCS) of 30KHz. One PRB is made up of 12 consecutive subcarriers and lasts 1 slot. The values of PRBs correspond to a bandwidth part of 20,40,50 and 60 MHz, 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 User Plane Function (UPF). A functional description of the Network Functions (NF) can be found in section 6.2 of 3GPP TS 23.501 document [20].

All NFs run as services inside docker containers on the gNB host. The version of the OAI CN docker containers used in our testbed is 2.0.1. Apart from the NF containers, two more containers are deployed: OAI-EXT-DN, that serves as the external data network and mysql (database server), in which 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. The schematic diagram of the Core Network is depicted in Fig. 2.

For the 5G system to meet the real-time requirements, we made specific 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. For the 10Gbps network interface controllers, we have set the value of MTU to 9000 instead of 1500, that is the default. In CPU settings, CPU frequency scaling governor must be set to performance and lock processor frequency to maximum (only C0 state active). The software and hardware configuration are summarized in Table 1.

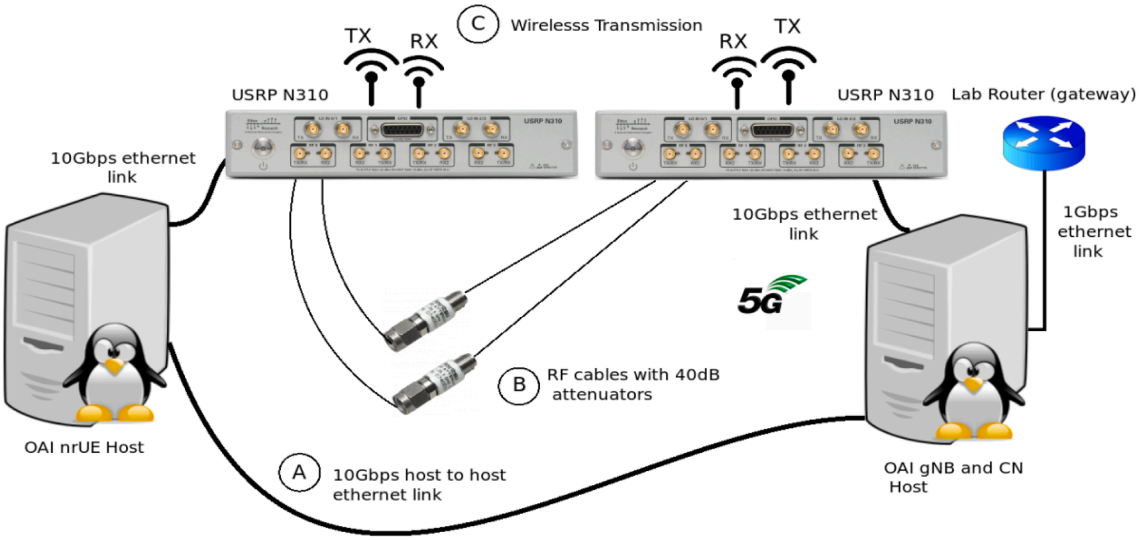

Fig. 1. Schematic Diagram of the 5G testbed.

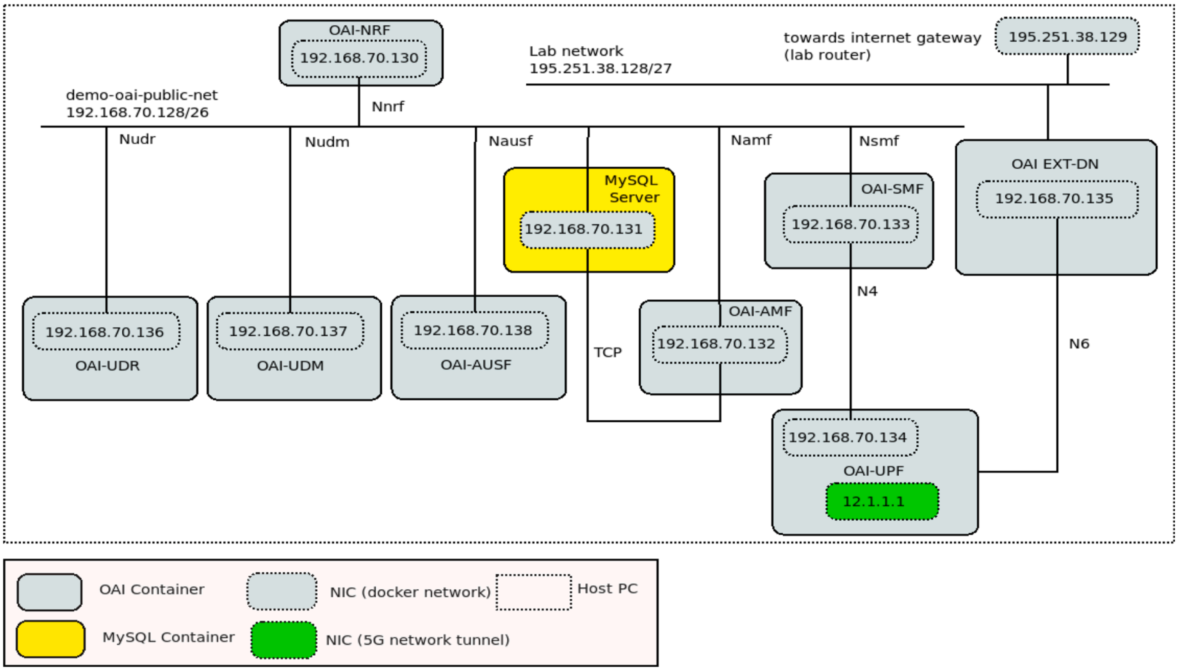

Fig. 2. Schematic Diagram of Core Network.

Table 1
Hardware and Software Configuration of Hosts.

n/a	*gNB and CN host*	*UE Host*
CPU	Intel(R) Core(TM) i9-12900 K	Intel(R) Core(TM) i9-12900 K
RAM (Gb)	32	16
OS	Ubuntu Linux 22.04 LTS	Ubuntu Linux 22.04 LTS
OS kernel	5.15.0–105-lowlatency	5.15.0–97-lowlatency
OAI RAN revision	integration_2024_w15	integration_2024_w15
OAI CN containers version	2.01	n/a
Docker version	25.0.3	n/a

2.2. *Platform operation modes*

The platform 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 the AMF container. These steps are made by running the bash shell script we developed, to automate the process. Then, the nr-softmodem program is launched. This program implements the 5G radio interface and the functions of gNB. If OAI software is interfaced to the SDR, the program initializes the device (USRP N310) which serves as a Radio Unit (RU). In the case of using the RF Simulator instead of the SDR devices, the nr-softmodem starts a udp server, listening for incoming connections from the UEs. Through this connection the I/Q data of 5G signal is transported to and from the gNB. Following the successful registration of a UE to the AMF, a PDU session is created. The nr-uesoftmodem program is the software implementation of the 5G modem that is executed on the

UE host. After the successful PDU session establishment, 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. To connect the host to the 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 shell scripts we developed to automate the process and are accessible from the software repository [21] we provide for this reason. In Fig. 3 the software component stack in each host is depicted. The configurable parameters of the OAI 5G protocol stack are stored in a set of configuration files in gNB and UE host. Each configuration file contains the parameters for a specific connectivity scenario. The user can adjust the parameters for one or all scenarios running the configuration script. We also created scripts to deploy the core network functions with predefined parameters, run the gNB and UE 5G modem programs and measure the bandwidth and latency, incorporating open-source tools like iperf, inetutils, speedometer and others. The communication of OAI software stack with the SDR device is done through the USRP Hardware Driver (UHD) Application Programming Interface (API). In section 3, we present the performed tests and results with the three available modes (A-C) of connecting UE to gNB, as depicted in Fig. 1.

3. Performed tests and results

3.1. Using RF simulator

We implemented 5 different scenarios with initial BWPs from 51 to 273 PRBs, exploiting the RF simulator mode of operation. The configuration of each scenario is shown in Table 2. The subcarrier spacing was set to 30KHz for all scenarios. 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, the OAI nrUE registered to the AMF and a PDU session was created. We checked the data connectivity creating traffic in Downlink and Uplink with the iperf tool. We exploited the RF Simulator tool for checking the E2E connection of the UE to the 5G network. Having checked the gNB configurations of all scenarios with the RF simulator, made the transition to wireless transmission easier. The values of maximum throughput or latency using the RF Simulator can be higher or lower than those measured in a real 5G transmission, being only related to the execution speed of the host computer. This happens because in the 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. As a matter of proof, we checked two different scenarios. When we increased the PRBs from 51 (scenario 1) to 106 (scenario 2) 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.

Table 2
GNB Configurations with RF Simulator.

Scenario	*Number of PRBs in BWP*	*Subcarrier spacing (KHz)*	*Band / No of Antennas*	*Downlink-Uplink Bandwidth (MHz)*
1	51	30	78 (SISO)	20
2	106	30	78 (SISO)	40
3	133	30	78 (SISO)	50
4	162	30	78 (SISO)	60
5	273	30	77 (2x2 MIMO)	100

3.2. Using the SDR devices with over the cable (OTC) transmission

For the first four scenarios (51,106,133 and 162 PRB in Table 2), we conducted measurements connecting the SDR devices to host PCs. As the number of PRB increases, the computational resources used are increased too. Thus the first scenario is less demanding and can be executed in less powerful host computers, while higher numbered scenarios require more computational resources. We excluded the fifth scenario with 273 PRB, because we didn't have enough computational resources to meet the real-time requirements of transmission. The RF link between the two SDRs was achieved using RF cables and 40 dB attenuators, to adjust the signal power to the acceptable level of the USRP inputs. To optimize the radio link, we also added another 15–18

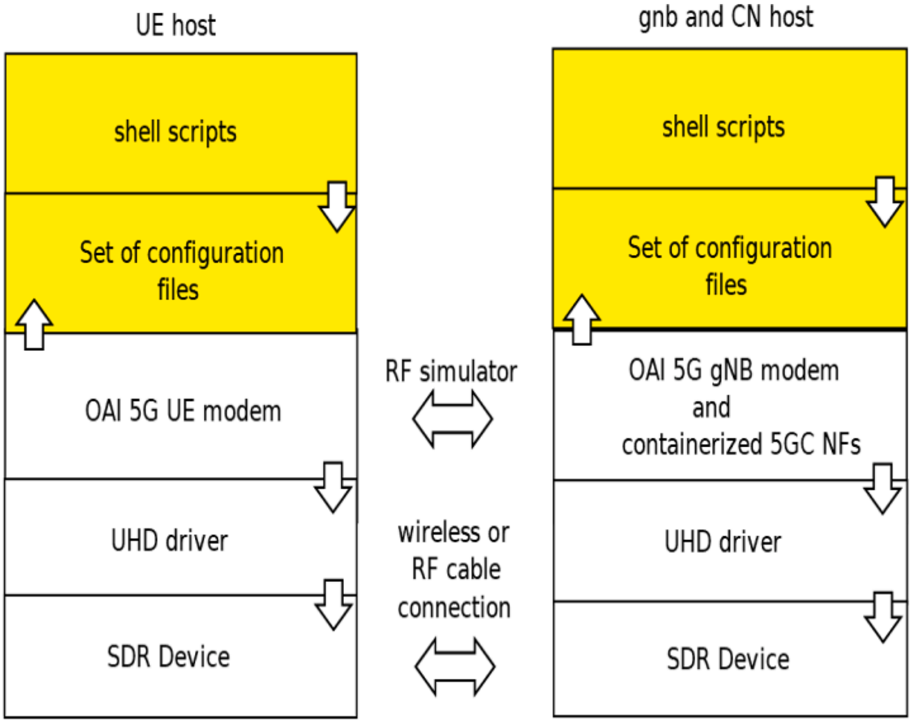

Fig. 3. Block diagram of testbed software components.

dB of attenuation in transmitter and receiver ports of the SDR adjusting the parameters att_tx and att_rx respectively in the gNB configuration file. In this setup the interference from other RF signals present in our lab is minimized. We measured 40 consecutive values of throughput in Uplink and Downlink by running the iperf client tool, generating TCP traffic with the default read/write buffer size of 128 Kb for 5 s on UE host and OAI-EXT-DN container, respectively. Since we have connected only one UE to the 5G network, it is possible that the MAC scheduler allocates all resource blocks for this UE. However the maximum achievable throughput both in uplink and downlink is affected by the selected TDD pattern and scheme of Physical Uplink Control Channel (PUCCH). PUCCH is designed to carry UCI (Uplink Control Information) which includes among other control elements, the acknowledgement of the received packets. The achievable data rate is further influenced by the congestion control in transferred data. which is applied by the selected in our tests TCP protocol. Thus the measured values are closer to the data rates that user experiences in the application layer, rather than those in the physical layer. Through the virtual NIC of the OAI-EXT-DN container, network traffic from the UE is routed to the physical NIC on the host PC and consequently to the router, providing internet connection. The mean values we measured for the achievable throughput for each scenario are presented in Table 3. The sampling rate of the baseband signal sent and received from the SDR device is 30.72 for scenario 1 and 61.44 Msamples/s for the other scenarios (2–4).

We used the ping command to access the Round Trip Time (RTT) in the 5G network. Ping works by sending an Internet Control Message Protocol (ICMP) Echo Request message from the core network (the network interface of oai-ext-dn container) and waiting for a reply from the UE in the uplink. Hence, the resulting time delay shows the RTT latency in the network. As we will show later, the value of RTT is affected by the selected TDD DL/UL slot configuration at the physical layer, depicted in Fig. 5.

The cumulative density functions of Downlink, Uplink throughput and roundtrip latency are depicted in Fig. 4. The achievable values of throughput increase with the increase of PRBs in the BWP, as more radio resources are available in MAC layer for scheduling in the Downlink and Uplink channels. As observed, the mean latency values are similar for all scenarios (Table 3). This was expected, as we kept the parameters of the numerology and TDD DL/UL slot configuration the same for all scenarios. Numerology is related to the duration of the OFDM symbols, which in our case of numerology $\mu = 1$, leads to a value of 20 slots/frame and a slot duration of 0.5 ms. The TDD DL/UL slot configuration is the assignment of slots and OFDM symbols to Uplink or Downlink transmission intervals. For a given slot configuration, the DL-UL Transmission Periodicity is defined as the periodicity of the DL-UL pattern and varies from 0.5 to 10 ms for all use cases. The DL-UL Transmission Periodicity was equal to 5 ms for all scenarios. The assignment of slots and OFDM symbols is depicted in Fig. 5. The parameters have a direct effect on the RTT latency value presented through the 5G protocol stack.

We managed to improve the mean RTT latency measured at the application layer with ping command by almost 3 ms, changing the physical layer parameter of DL-UL Transmission Periodicity from 5 to 2 ms. The CDF of the 40 consecutive values that we measured in scenario 2 is depicted in Fig. 6b, having a mean value of 6.78 ms compared to 9.88 ms (Table 3) in the initial setting of 5 ms for the DL-UL Transmission Periodicity. Our effort to adjust this parameter to the even lower value of 1 ms, led to an unexpected termination error of the gNB softmodem process, therefore this was the lowest RTT latency value achieved in our testbed. We also noticed a small increase of 6.3 % to the mean uplink throughput from 39,58 (Table 2) to 42.1 Mbps with no noticeable difference at the downlink throughput, because of the higher proportion of UL to DL slots in TDD slot configuration of Fig. 6a compared to that of Fig. 5.

The OAI software periodically reports the HARQ retransmissions. HARQ stands for Hybrid ARQ (Automatic Repeat reQuest), which implies the combination of error correction and ARQ. Thus in the case of retransmission the UE MAC entity combines the data from the previous transmission which are stored in a buffer, with the current data it just received, and then decodes the combined data, as described in the 3GPP TS 38.321 document [22]. In Table 4 we present the percentage of HARQ retransmissions during one test of maximum throughput with the iperf tool for 5 s. As observed, there is only a percentage of first HARQ retransmissions, especially during the measurement of throughput in Downlink.

3.3. *Connecting SDR devices wirelessly*

We replaced the RF cables connecting the two SDR devices with 5G multi-band omni directional antennas in a line-of-sight distance of approximately 1 m between them. The use of the host PC connected to the SDR device as the UE, has the drawback of difficulty in positioning the UE in different spots, because of the weight and size of the equipment. This was not an issue for this study, since it is not focused on the testbed provided coverage. We performed the same tests as with the RF cables and the results are presented in Table 5.

The cumulative density functions of Downlink, Uplink throughput and roundtrip latency are depicted in Fig. 7. In Table 6 we present the percentage of HARQ retransmissions during one throughput test with the iperf tool for 5 s.

There is a higher rate of HARQ retransmissions in the case of wireless transmission, especially during the throughput measurements in Uplink, compared to the OTC transmission. There is also a small percentage of second HARQ retransmissions occurring, affecting the achievable throughput. The increased number of retransmissions in wireless operation comes from the multipath interference within the indoor environment, due to the reflection of RF signals to walls, people and other objects present in the lab room. We did not include the RSRQ or SNR/SINR values measured in wireless channel, because the version of OAI software used in our tests (Table 1), reports only the BER and the number of HARQ retransmissions as indicator to represent the signal quality.

As an overall comparison of the achievable KPIs in the two transmission cases (OTC and wireless), the throughput experienced at the application level was higher for all tested scenarios in the OTC case, while the latency had no notable difference in either OTC or wireless transmission. Finally, there were no third HARQ retransmissions in both cases. Consequently, there were not any errors in DLSCH and ULSCH transport channels, during all tests. In addition, we tested the 5G connection for 1 h, during which period it remained stable for all tested scenarios, and we did not experience any UE disconnection or

Table 3
Mean Throughput and RTT Latency Values with OTC connection.

Scenario	*No. of PRBs in BWP*	*Sampling Rate (Msamples /s)*	*Band / No of antennas*	*RSRP (dBm)*	*DL Through-put (Mbps)*	*UL Through-put (Mbps)*	*RTT Latency (s)*
1	51	30.72	78 (SISO)	−107	38.92	21.10	9.36
2	106	61.44	78 (SISO)	−112	68.30	39.58	9.88
3	133	61.44	78 (SISO)	−97	87.93	53.36	9.54
4	162	61.44	78 (SISO)	−97	97.88	67.01	9.89

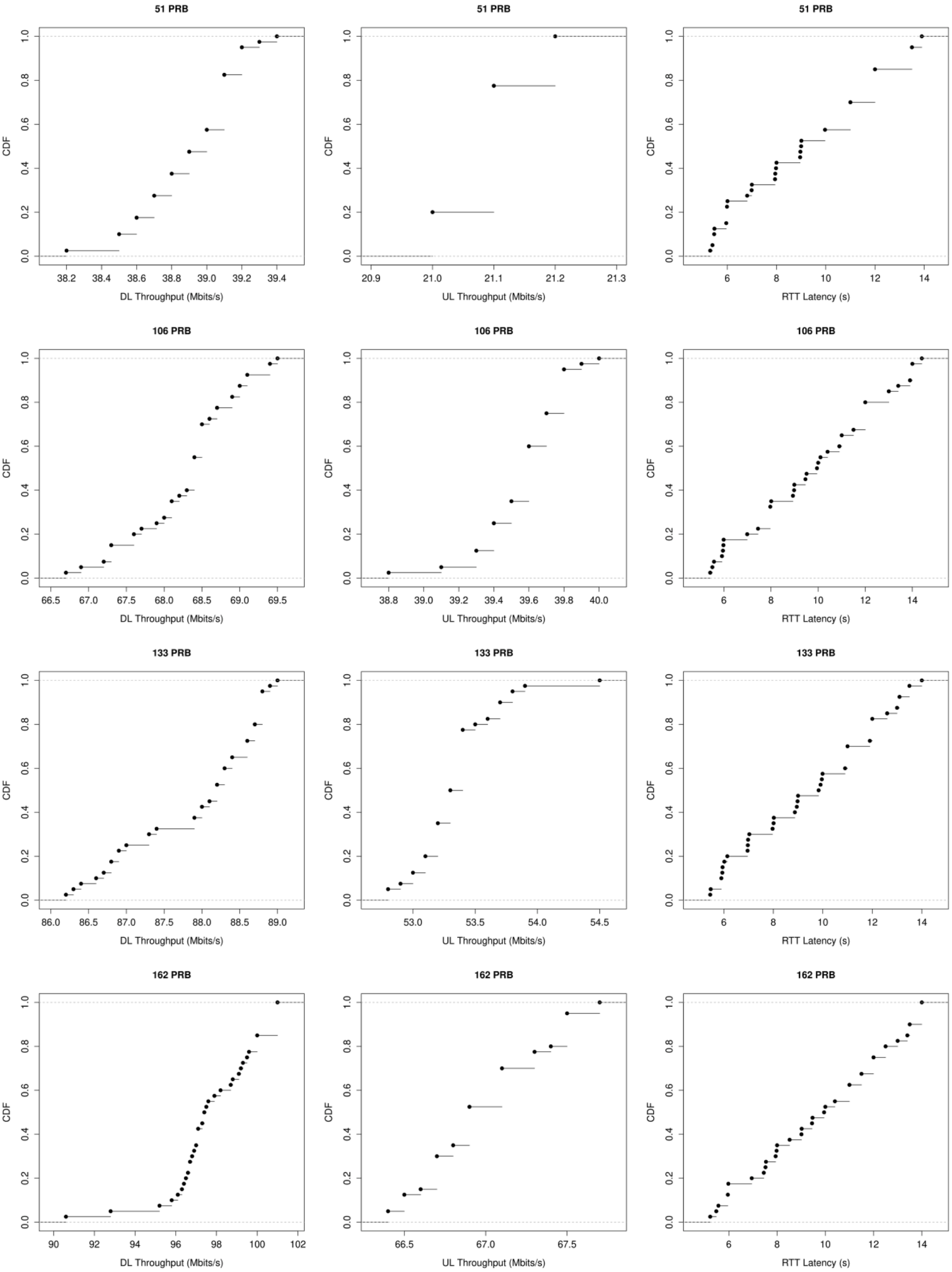

Fig. 4. CDF of Throughput and RTT latency for scenarios 1 to 4 with OTC transmission.

performance degradation.

4. 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 shell scripts, we managed to startup all 5G components, make all necessary network configurations and finally check connectivity through the 5G protocol stack and perform measurements. We provide a repository [21] containing all the shell scripts we developed, and the

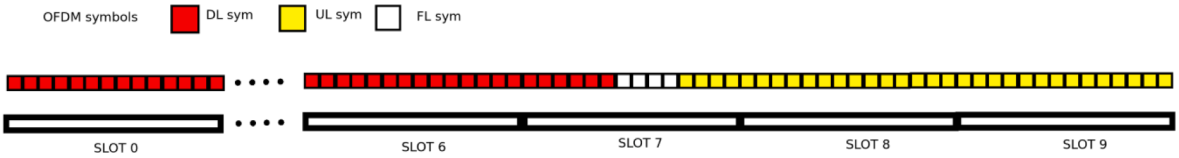

Fig. 5. TDD DL/UL slot configuration.

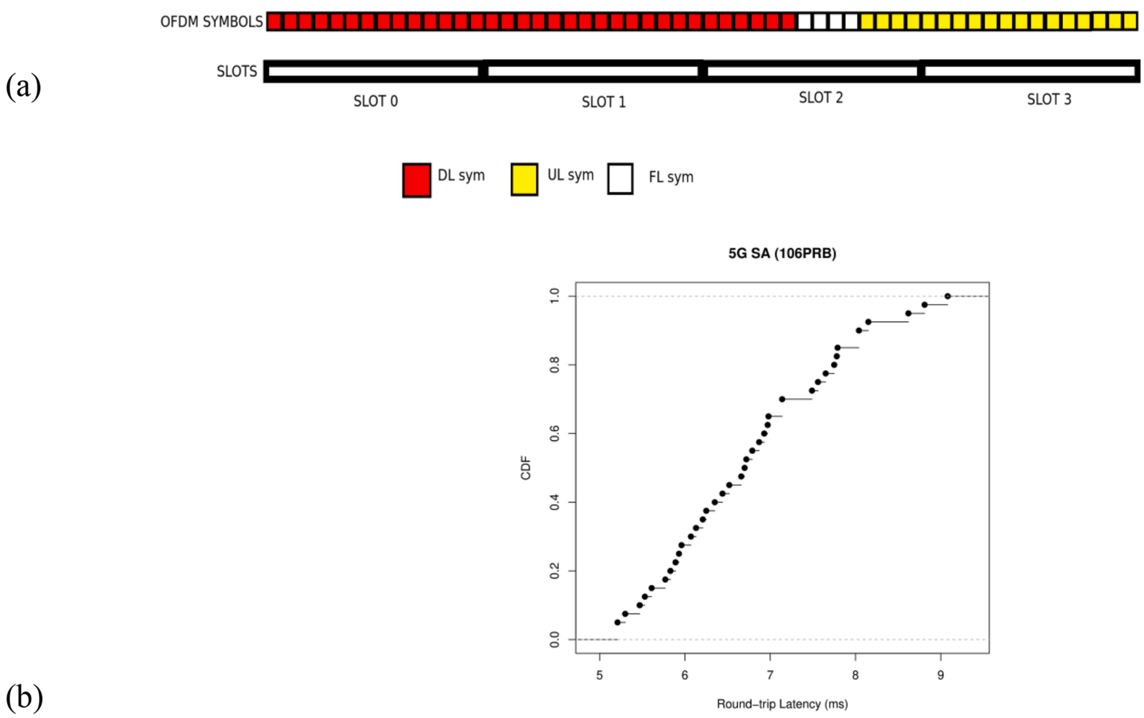

Fig. 6. Effect of lower DL-UL Transmission periodicity value on RTT latency (a) Slot configuration for 2 ms DL-UL Transmission periodicity (b) CDF of RTT Latency.

Table 4
Retransmissions with OTC Transmission.

Scenario	*Total Transmissions*	*First HARQ retx (%)*	*Second HARQ retx (%)*	*Third HARQ retx (%)*
	During Downlink* max *throughput test			
1	8202	6.44	0	0
2	6844	8.50	0	0
3	6709	7.14	0	0
4	5347	8.70	0	0
	During Uplink* max *throughput test			
1	6058	0.59	0	0
2	5711	1.02	0	0
3	5347	1.87	0	0
4	6374	3.55	0	0

configuration files used, facilitating reproducibility of our testbed. The shell scripts improve the ease of use of OAI software, reducing the number of repetitive tasks that need to be done to perform a specific connectivity scenario.

However, with the currently available hardware in our testbed, we could not test BWPs higher than 60 MHz and the MIMO operation. Furthermore, the use of OAI UE implementations puts a limitation on wireless operation, because of the size and weight of the equipment Thus testing the coverage and measuring the KPIs in different SINR conditions will be easier using a COTS UE device.

For future work we are planning to expand our testbed capabilities, upgrading the computer and network hardware, to test MIMO transmission and wider BWPs. Furthermore, we plan to connect a COTS UE device and adjust the system parameters to improve the KPIs.

CRediT authorship contribution statement

Emmanouil-Zafeirios G. Bozis: Writing – original draft, Software,

Table 5
Mean Throughput and RTT Latency Values In Wireless Transmission.

Scenario	*No. of PRBs in BWP*	*Sampling Rate (Msamples /s)*	*RSRP (dBm)*	*Band / No of Antennas*	*DL Throughput (Mbps)*	*UL Throughput (Mbps)*	*RTT Latency (s)*
1	51	30.72	−93	78 (SISO)	24.08	16.87	9.33
2	106	61.44	−97	78 (SISO)	62.98	34.49	9.87
3	133	61.44	−106	78 (SISO)	81.61	51.16	9.19
4	162	61.44	−106	78 (SISO)	89.50	59.17	9.49

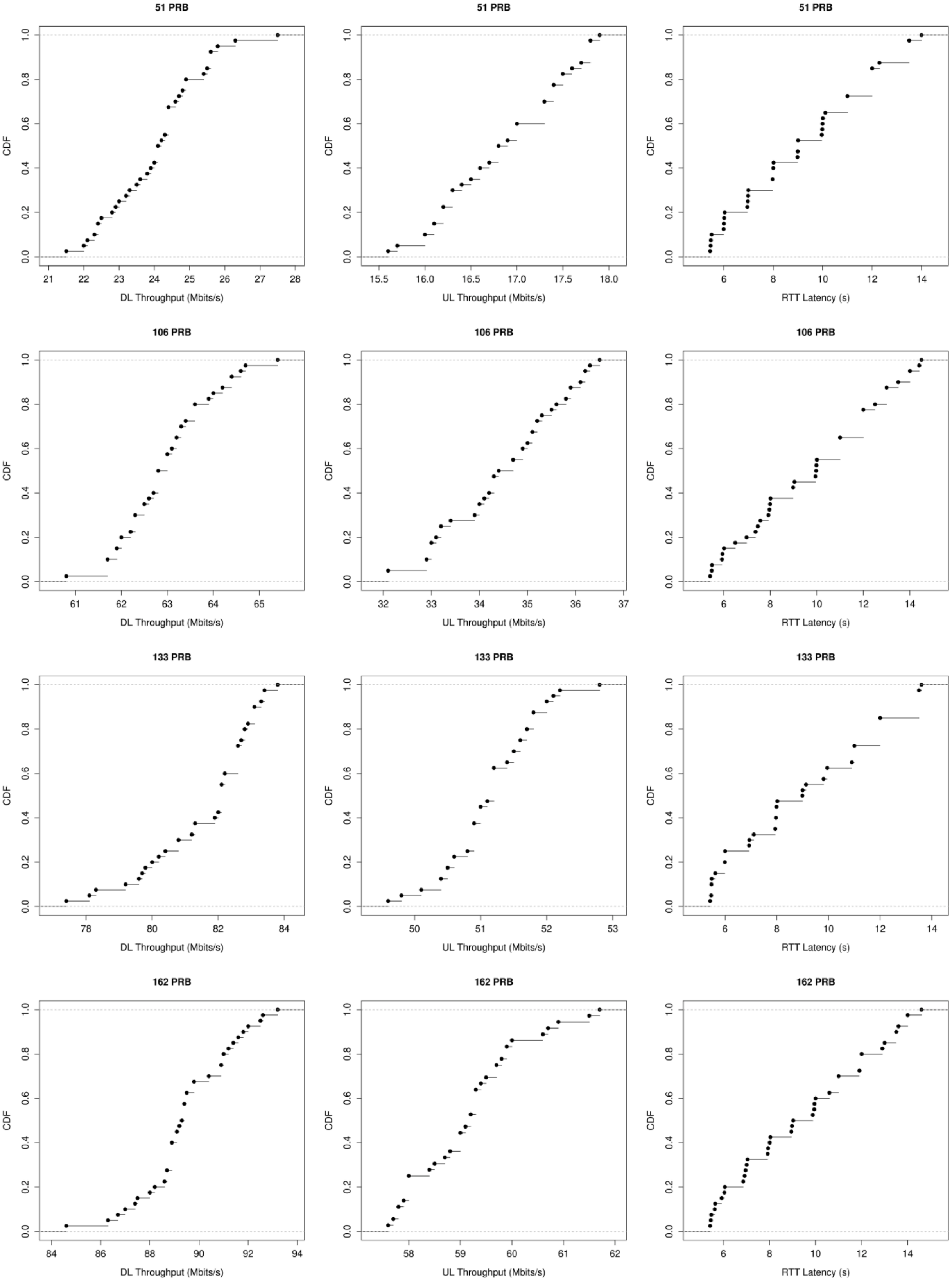

Fig. 7. CDF of DL, UL Throughput and RTT latency for scenarios 1 to 4 with wireless transmission.

Table 6
Retransmissions In Wireless Transmission.

Scenario	*Total Transmissions*	*First HARQ retx (%)*	*Second HARQ retx (%)*	*Third HARQ retx (%)*
	During Downlink **max** ***throughput test***			
1	8378	8.53	0.13	0.00
2	7061	9.84	0.08	0.00
3	6751	9.11	0.03	0.00
4	7149	4.70	0.03	0.00
	During Uplink **max** ***throughput test***			
1	4975	5.95	0.00	0.00
2	5266	8.28	0.23	0.00
3	6152	5.43	0.00	0.00
4	5410	8.19	0.00	0.00

Conceptualization. **Nikos C. Sagias:** Writing – review & editing, Supervision, Conceptualization. **Michael C. Batistatos:** Writing – review & editing. **Michail-Alexandros Kourtis:** Writing – review & editing. **George K. Xilouris:** Writing – review & editing. **Anastasios Kourtis:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code and the configuration files of the 5G testbed are available at the repository https://github.com/mbozis/OAI_5G_scripts

Acknowledgment

The research work presented in this article has been supported by the European Commission under the Horizon Europe (2021-2027) Framework Program for Research and Innovation, through funding of 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 Commission, which cannot be held responsible for them. A preliminary version of this work has been published in the Panhellenic Conference on Electronics & Telecommunications (PACET) 2024 [23].

References

[1] srsRAN Project, https://www.srslte.com/, Accessed 4 Aug 2024.
[2] Open Air Interface, https://openairinterface.org/, Accessed 4 Aug 2024.
[3] Amarisoft, https://www.amarisoft.com/, Accessed 4 Aug 2024.
[4] Open5GS, https://open5gs.org/, Accessed 4 Aug 2024.
[5] Repository for the Open Air Interface Core Network development, https://gitlab.eurecom.fr/oai/cn5g, Accessed 4 Aug 2024.
[6] Kaltenberger F, Silva A, Gosain A, Wang L, Nguyen T-T. Openairinterface: democratizing innovation in the 5G era. Comput Netw 2020;176:107284. https://doi.org/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, https://doi.org/10.1109/ICDSP.2018.8631835.
[8] OAI full stack 5G NR RF simulation with containers, https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/ci-scripts/yaml_files/5g_rfsimulator/README.md, Accessed 4 Aug 2024.
[9] Mihai R, Craciunescu R, Martian A, Li FY, Patachia C, Vochin M-C, et al. "Open-source enabled beyond 5G private mobile networks: from concept to prototype". In: 2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC), Herning, Denmark; 2022. p. 181–6. 10.1109/WPMC55625.2022.10014829.
[10] Matzakos P, Koumaras H, Tsolkas D, Christopoulou M, Xilouris G, Kaltenberger F. "An open source 5G experimentation testbed". In: 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece; 2021. p. 1–2. 10.1109/MeditCom49071.2021.9647647.
[11] Aijaz A, Holden B, Meng F. "Open and programmable 5G network-in-a-box: technology demonstration and evaluation results". In: 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan; 2021. p. 369–71. 10.1109/NetSoft51509.2021.9492719.
[12] Bozis E-ZG, Batistatos MC, Sagias NC. "Overview Of SDR platforms based on open source software: a 5G system emulation with open air interface,". In: 2022 Panhellenic Conference on Electronics & Telecommunications (PACET), Tripolis, Greece; 2022. p. 1–5. 10.1109/PACET56979.2022.9976373.
[13] Grohmann AI, Seidel M, Lehmann C, Höschele T, Reisslein M, Fitzek FHP. "5G on the cheap: configurable low-cost cellular industrial communication". In: 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives; 2022. p. 1–6. 10.1109/ICECCME55909.2022.9988703.
[14] Gabrielson A, Bauer K, Kelly D, Kearns A, Smith WM. "CUE: A standalone testbed for 5G experimentation". In: MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA; 2021. p. 745–50. 10.1109/MILCOM52596.2021.9653117.
[15] Chepkoech M, Mombeshora N, Malila B, Mwangama J. "Evaluation of open-source mobile network software stacks: A guide to low-cost deployment of 5G testbeds". In: 18th Wireless on-Demand Network Systems and Services Conference (WONS) Madonna Di Campiglio, Italy; 2023. p. 56–63. 10.23919/WONS57325.2023.10061896.
[16] Amini, M., & Rosenberg, C. (2024). Performance Analysis and Comparison of Full-Fledged 5G Standalone Experimental TDD Testbeds in Single & Multi-UE Scenarios. arXiv preprint arXiv:2407.02341.
[17] Seidel M, Grohmann AI, Sossalla P, Kaltenberger F, Fitzek FHP. "How to get away with openairinterface: a practical guide to 5G RAN configuration". In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain; 2023. p. 1–6. 10.1109/ICECCME57830.2023.10252787.
[18] 4G LTE and 5G Radio Access Network code repository Eurecom, https://gitlab.eurecom.fr/oai/openairinterface5g/, Accessed Aug 202.
[19] 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, Accessed 4 Aug 2024.
[20] 3GPP TS 23.501 Version 17.10.3.0 Release 17, available at: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/17.10.00_60/ts_123501v171000p.pdf, Accessed 4 Aug 2024.
[21] Repository for scripts and configuration files of OAI 5G testbed, https://github.com/mbozis/OAI_5G_scripts, Accessed 4 Aug 2024.
[22] 3GPP TS 38.321 Version 16.5.0 Release 16, available at: https://www.etsi.org/deliver/etsi_ts/138300_138399/138321/16.05.00_60/ts_138321v160500p.pdf, Accessed 4 Aug 2024.
[23] Bozis E-ZG, Sagias NC, Batistatos MC, Kourtis M-A, Xilouris GK, Kourtis A. "A versatile 5G standalone testbed based on commodity hardware". In: 2024 Panhellenic Conference on Electronics & Telecommunications (PACET), Thessaloniki, Greece; 2024. p. 1–4. 10.1109/PACET60398.2024.10497086.
[24] Introduction to 5G Phy Simulators in Open Air Interface, https://openairinterface.org/introduction-to-5g-ran-phy-simulators-in-openairinterface/, Accessed 4 Aug 2024.