Design of a 5G Experimental Platform Based on OpenAirInterface

Quentin Douarre[1,2], El-Mehdi Djelloul[1], Pascal Berthou[3], Daniela Dragomirescu[2], and Philippe Owezarski[1(✉)]

[1] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
owe@laas.fr
[2] LAAS-CNRS, Université de Toulouse, INSA, Toulouse, France
qdouarre@insa-toulouse.fr, daniela@laas.fr
[3] LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France
berthou@laas.fr

Abstract. 5G has been designed for providing the appropriate services for a large range of applications requiring high throughput, low latency, a support for the IoT, or for Industry 4.0 business, etc. One of its strong statements is the sofwarization of most of its functions for providing more flexibility, and a support that can easily evolve for providing new services. In that context, OpenSource implementations of 5G functions arise. One of this implementation is the 5G OpenAirInterface (OAI). This paper then describes how a 5G experimental platform taking advantage of the 5G OAI software suite was designed and deployed at LAAS-CNRS. The aim of the platform is to be as generic as possible for being able to experiment and evaluate all 5G new mechanisms and protocols issued from researchers. This paper then specifically addresses how the compatibility issues between 5G OAI and the equipments (USRP, servers, operating systems, etc.) were fixed. The paper also proposes a performance evaluation of the 5G OAI platform and analyzes its limits.

Keywords: 5G · Experimental platform · network softwarization · OpenAirInterface

1 Introduction

Over the last few decades, several major evolutions impacted network design, as a huge number of new usages and applications requires wireless and mobile communications: smart-phones, IoT, vehicular networks, etc. This lead to the arising of multiple wireless technologies for local networks (Wifi, ...), long distance cellular networks (4G, 5G, ...), for the IoT (Sigfox, LoRA, ...), etc. Wireless networks present many limits nowadays because of the hertzien nature of the transmission support that provides only limited capacities compared with wired communication supports, together with a huge variability of available communication resources. Providing the multiple services requested by current applications in such a context is a tricky challenge.

S. Yu et al. (Eds.): TridentCom 2022, LNICST 489, pp. 87–103, 2023.
https://doi.org/10.1007/978-3-031-33458-0_7

Providing theses multiple Qualities of Services (QoS) on wireless and mobile networks is a hot topic that the 3GPP standardization organization is strongly considering nowadays, especially for 5G. The old monolithic architecture of "One-size-fits-all" kind passed away, and a service oriented one has to replace it. Network slicing is the main aspect for developing such architecture. A network slice is defined by 3GPP as a virtual network that provides the characteristics of a specific network. In other words, each Mobile Network Operator (MNO) splits its physical infrastructure in several slices dedicated to several service providers, as a slice for automotive, and another one for the health domains, etc. Thus, each slice could be devoted to a specific professional domain with its own QoS requirements. When a user equipment (UE) registers on the network, it then has to use a slice id to connect to the appropriate slice. Up to now, 3GPP standardized three kinds of slices/services: Enhanced Mobile Broadband (eMBB), ultra-reliable low latency (URLLC) and massive internet of things (mIoT). eMBB service provides high throughput, URLCC provides low latency communication for real-time applications/services, and mIoT is specifically dedicated for IoT services. Implementing slicing approach however requires high flexibility and programmability for 5G networks. It then takes advantage of the network softwarization approach, as NFV (Network Function Virtualization) and SDN (Software Defined network) paradigm. The first approach provides the flexibility of NF (Network function) thanks to virtualization, and the second one separates the control and data planes. Several prototypes based on NFV and SDN have already been proposed for the slicing in core networks (CN) [1] and the radio access network (RAN) [2].

5G also benefits from the analyses of previously deployed networks by operators and equipment manufacturers. The cost of updating or managing hardware devices when it is required to add new services for instance is dramatically high. In that context also, network softwarization provides strong advantages as updating or replacing some functions implemented in software (NFV) can be made remotely, and without any action on the hardware. Network management then becomes more flexible and cheaper. That's why the opensource market has decided to take an interest in the telecommunication sector under many aspects such as USRP or mobile networks. Several projects are born to propose 4G or 5G opensource implementations.

In this context, the research community takes advantage of this network virtualization aspect as it is then possible to assess their research proposals not only using simulator, but also by direct software implementations that can be rapidly integrated on the actual market networks. For this purpose, LAAS-CNRS lab in Toulouse, France decided to design and implement such a 5G experimental platform. This paper describes its design principles and first evaluation results. For that it takes advantage of the concepts of Software defined Radio techniques (SDR) and the open-source software OpenAirInterface (OAI) platform for the experimental assessment of 5G wireless networks.

The rest of this paper is as follows: Sect. 2 gives a short presentation of the elements of a software 5G platform, as OAI, required for implementing our 5G experimental platform. Section 3 details the equipments required for setting up

our 5G platform. It also mostly presents the software components that have been included, as well as the way they have been configured. This Sect. 2 also indicates how to fix the main encountered problems during the installation and configuration of such a platform, in order to provide an important document for engineers or researcher that would be interested in setting up such a platform for their own researches. Then, Sect. 4 shows some performance results gained on our 5G experimental platform. It especially illustrates the performance cost of using a 5G software platform. It can be seen as the "price" of flexibility, compared to the old fashion monolithic hardware only networks. Finally, Sect. 5 concludes this paper.

2 The 5G Opensource Platform

2.1 5G Standalone Architecture

There are several 5G network architectures. The 5G Non-Standalone architecture builds on existing LTE infrastructures such as ENB and EPC. It is therefore a kind of 4G++. The architecture we are interested in is the 5G-NR Standalone which characterizes the new generation of mobile networks. It is composed of several elements, each having a specific role. They are arranged as depicted on Fig. 1:

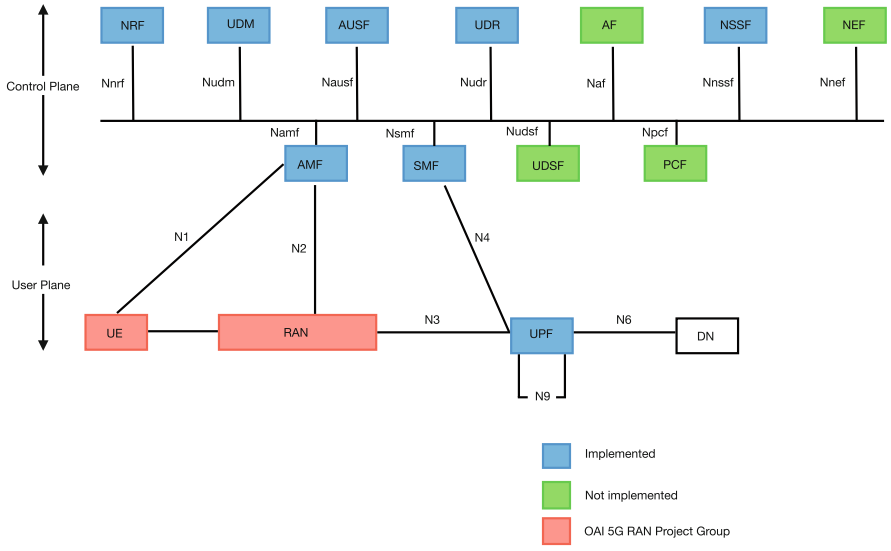

Fig. 1. 5G-NR Standalone Architecture

The UE represents the User Equipment. It is a connected object like a smartphone, a connected watch or a drone. At first, on our 5G experimental platform, it will be a Dell server. But it will be replaced later by a Google Pixel 6 cell-phone. Its radio interface communicates with the core network RAN. In 5G,

this operator equipment is called a GNB. It represents the entry point to the provider's network. Next comes the AMF. It is in charge of the good management of the registrations on the network, the handovers and the maintenance of the client databases. The SMF, the UPF and the UDM are responsible for the PDUsessions allowing to identify a communication and to fix the "rules" of transmission. The NRF controls the virtual functions of the network. It is essential, because in recent years the importance of virtualization as a research topic clearly appears. The AUSF takes care of the authentication of the users of the 5G network. It interacts closely with the AMF and UDM. The UDR centralizes detailed customer information such as their subscription. To finish with this little tour, there is the NSSF. This component of the core network supervises the management of slicing.

2.2 NGAP Protocol

During the design of this 5G experimental platform, the NGAP protocol proved to be the most important one. This protocol is in charge of the control plan between the GNB and the AMF. Thanks to it, we can see the inscriptions on the network, the reservations of resource in Uplink and Downlink, the request for PDUsession, etc.

Here is a non-exhaustive list of messages that NGAP carries with their role:

- **NGSetupRequest:** The GNB asks to associate with the AMF
- **NGSetupResponse:** The AMF accepts to connect with the GNB
- **InitialUEMessage:** The user has transmitted all the information necessary to the network for its use of the network. The GNB transfers to the AMF so that the latter updates these tables
- **UplinkNASTransport:** This message allows you to reserve resources on the uplink between the UE and the GNB
- **DownlinkNASTransport:** This message allows users to reserve resources on the downlink between the UE and the GNB
- **InitialContextSetupRequest:** The AMF ensures that it has the right information about the UE and the requested network access procedures.
- **InitialContextSetupResponse:** This is the answer brought by the UE to the verification launched by the AMF concerning its access to the network
- **UERadioCapabilityInfoIndication:** The UE announces its radio specificities to the network in order to have the most efficient and reliable communication possible
- **PDUSessionRessourceSetupRequest:** The UE requires the ability to transmit a data stream using the network infrastructure and specific quality of service parameters.
- **PDUSessionRessourceSetupResponse:** The SMF says that it has the capacity to make the reservation by respecting the parameters setting requested by the UE

The above messages indicate that the attachment of the device to the network is successful. With the help of Wireshark, it is possible to check the good

registration of the UE on the 5G network, the reservation of the resources and the good transfer of the data. This tool designed for sniffing the packets in transit on the network provides a great help to researchers and network managers.

2.3 The Main Communication Channels

The 5G standard defines several different types of communication channels. The following lists them in a non-exhaustive way, useful for understanding the rest of the report and possibly debugging errors.

- **PBCCH** (Packet Broadcast Control Channel): Control information and signaling related to packet services.
- **PCCCH** (Packet Common Control Channel): Includes logical channels for GPRS common control signaling. These sub-channels include: PRACH (Packet Random Access Channel), PPCH (Packet Paging Channel), PAGCH (Packet Access Grant Channel) et PNCH (Packet Notification Channel).
- **PDCCH** (Physical Downlink Control Channel): Consists of a set of resource elements that carry data from higher layers to the physical layer or from the physical layer to higher layers.
- **PRACH** (Physical Random Access Channel): An uplink channel used by the UE for connection request. The PRACH is used to transport data from the RACH transport channel.
- **PUSCH** (Physical Uplink Shared Channel): Used to transmit the uplink shared channel (UL-SCH) and L1 and L2 control information.
- **PDSCH** (Physical Downlink Shared Channel): A transport channel used for the transmission of user data, dedicated control and user-specific upper layer information and downlink system information.

2.4 The OAI 5G Project

The OpenAirInterface project is led by the OSA. Its objective is to implement future opensource cellular networks in order to demystify the inner workings of these networks and to make it easier for the global research community to contribute to their development.

OpenAirInterface 5G is divided into three main projects: 5G RAN, 5G Core Network and Mosaic5G. Mosaic5G adds a range of virtual functionalities to the network and proposes, in addition to the data and control planes, a management plane. However, Mosaic5G will be stopped in favor of a new project called Trirematics which will have the same objectives using more modern technologies to achieve them.

The first two projects are the ones we are particularly interested in. 5G RAN deals with the interactions between the UE and the GNB. It thus focuses on the lower layers of the OSI model.

Several architectures are supported, as depicted on Fig. 2. The one we are interested in is the standalone one. The project is still evolving. Soon, RAN slicing and Massive MIMO, two fundamental features of theorized 5G, will be

supported. 5G Core Network offers the management of the almost complete architecture of the 5G-NR Standalone, the highest level of functionality. In order to create all components virtually, Docker is used. As with the previous project, this one continues to be developed. Missing components like UDSF, AF will be coded (see Fig. 1 for more information about missing components), slicing management will be added in version 1.4, etc.

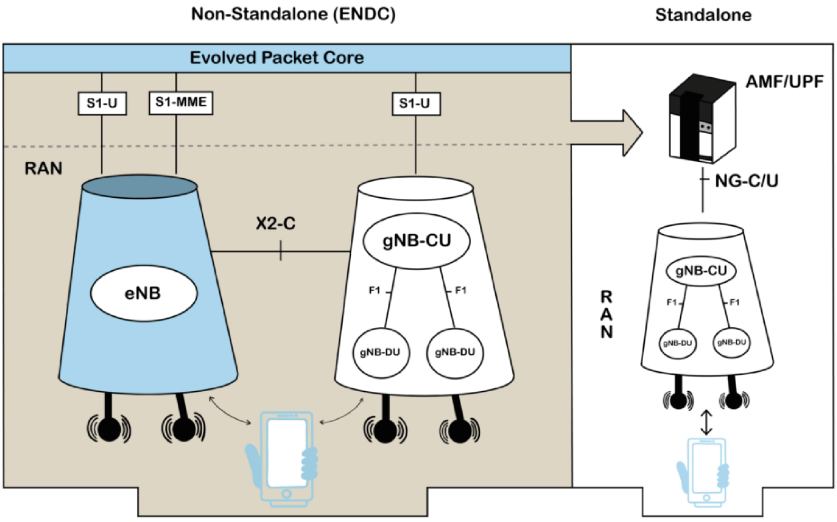

Fig. 2. Interactions, according to the chosen architecture, of the radio entities of the network

3 OAI 5G Installation Steps on Dell Servers and Google Pixel 6

This section describes the installation and configuration of the 5G experimental platform, based on OAI. This platform consists of a 5G Core Network (CN), a gNB and a UE. After describing the hardware required for the installation of the platform, the following parts will be dedicated to the installation steps of the different modules of the 5G network.

Interested readers can refer to a more detailed report for this 5G platform [3].

3.1 Material Required

The configuration chosen for the platform is as follows:

- The gNB and the CN are installed on the same machine
- The UE is installed on a second machine

Each machine is connected to a USRP to act as a radio module.
The couple (gNB+CN) is installed on the machine 1:

- A Dell Precision Rack 7920 18-core server
- An Ettus B210 (or X310) USRP connected via USB 3.0 (or 10G-Ethernet)
- Two broadband log-periodic antennas (850 MHz–6.5 GHz)

Regarding the UE on machine 2:

- A Dell Precision Rack 7920 18-core server
- An Ettus B210 (or X310) USRP connected via USB 3.0 (or 10 Gb/s Ethernet)
- Two broadband log-periodic antennas (850 MHz–6.5 GHz)

3.2 Software Configuration

Each Dell Precision Rack 7920 server must have the following features:

- Ubuntu 20.04 LTS
- Lowlatency kernel to download
- Disable C-States and P-States (except C0 and C1 which allow the cores to run at full speed but also to be able to stop in case of interruption)
- Disable CPU hyperthreading
- Configure each core as Governor Performance
- Specialize the cores manually according to the tasks

To download the low latency kernel for Ubuntu 20.04 LTS, enter the following command:

sudo apt-get install linux-image-lowlatency linux-headers-lowlatency

To be able to configure the CPUs operation correctly, the i7z package is needed. It allows the return of detailed information about the operation states.

sudo apt-get install i7z
sudo i7z

Now go to the BIOS to leave only the C0 state (operation at the maximum energy of the CPUs) and C1 for the stop of the CPUs (interrupts). When the servers are launched, press F2.

Then configure all the cores in performance mode: mode allowing to use the full computing power of the CPUs. First, install the cpufrequtils package which will allow changing the power consumption of the CPUs, and the operating frequencies.

Recommendation: We recommend using a USRP x310 for the gNB + CN part configured as 10 Gb ethernet. For this, at least a category 6 cable must be used.

Installation and Configuration of USRP. The installation and configuration of the USRPs are done by following the *Getting Started Guides*, available on the Ettus Research website at the following address:
https://files.ettus.com/manual/page_usrp_x3x0.html.

Installation of the 5G Network Core on Machine 1. On the machine, once the OS is installed, the chosen USRP must be configured. Ideally, for gNB+CN, a powerful USRP capable of handling a 10 Gb/s throughput is needed. For this, the Ettus X310 USRP is ideal, but the B210 could work as well.

Installation of the gNB on Machine 1. The main prerequisite for the installation of the gNB is the installation of UHD. This driver allows the management of the USRP.

Installation of the UE on Machine 2. For the UE, machine 2 is associated with a USRP (B210) that must be installed and configured on the machine.
It is advised to install the core on machine 2, allowing to symmetrize the roles of machines 1 and 2.

Then, on machine 2, it is enough to take again the installation of the gNB whose argument `--nrUE` installs the modules required for launching the UE.

NB: We have chosen to install the UE on the machine with the gray background and the GNB + heart network on the colored background.

Configuration of New Bands. To configure new bands, go to the folder: /openairinterface5g/targets/PROJECTS/GENERIC-NR-5GC/CONF

It contains some models made for several bands working with the USRPs b210 and x310 from Ettus. Between the two USRP models, there are differences in the way they are configured. For creating other configurations, these models can serve as a basis.

During the configuration, several very important parameters need to be modified according to the band. For example:

- absoluteFrequencySSB
- dl_frequencyBand
- dl_offstToCarrier
- dl_carrierBandwidth
- initialDLBWPlocationAndBandwidth
- initialDLBWPcontrolResourceSetZero

The list goes on. The uplink band has to be configured the same way. At the bottom of the configuration files are the tx and rx attenuation, the gains which are essential parameters for the good configuration of the USRP and for the good management of the available radio resources.

To help configure some of the settings, refer to the two sites: https://5g-tools.com/5g-nr-gscn-calculator/ https://www.sqimway.com/nr_refA.php

NB: It is essential to start from a valid GSCN to find the corresponding ARFCN.

It is also possible to go to the file nr_mac_common.c to see the configurations entered in the code.

3.3 Debugging Phase

IMPORTANT: The use of Wireshark is highly recommended to follow the exchanges between the different entities and to detect errors or missing NGAP

exchanges. Go to the demo-oai interface. Set it up by filtering all exchanges containing NGAP.

Communication Problem Between gNB and CN

1. Make sure the tracking area code is set to 1
2. mcc = 208 et mnc = 99 (MCC identifies the country, MNC identifies the operator)

NB: The changes are mainly in the gNB configuration but also a little in the core configuration.

Problems of Identification of the UE

1. Ensure the consistency of the core record tables and the information passed to the UE.
2. Check in the configuration of the UE:
   - dnn = "oai";
   - nssai_sst = 1;
   - nssai_sd = 1;

NB: This setting is recommended on the gNB side because it is the one that is directly implemented in the core network. For selecting another one, change the initial CN configuration in the file **/oai-cn5g-fed/docker-compose/docker-compose-basic-nrf.yaml**

Ping Problems Between the UE and the GNB

1. Make sure that the identifier of the UE is well contained in the files of the databases oai-db, oai-db1, oai-db2
2. Make sure that on Wireshark, the NGAP packet PDUSessionRequest is sent so that the UE is assigned an address.

DNS Problems Core Network

1. Change DNS to:
   - DEFAULT_DNS_IPV4_ADDRESS = 127.0.0.53
   - DEFAULT_DNS_SEC_IPV4_ADDRESS = 8.8.8.8
2. Add these lines in the core network configuration in the file **/oai-cn5g-fed/docker-compose/docker-compose-basic-nrf.yaml** for the UDM and UDR:
   - REGISTER_NRF = yes
   - NFR_IPV4_ADDRESS = 192.168.70.130
   - NFR_PORT = 80
   - NRF_API_VERSION = v1
   - NRF_FQDN = oai-nrf

Problem Launching the gNB with the USRP x310 Configuration. Sometimes, the launching of the program crashes with an abort message. Just restart it to make it work again. This is a stability problem in the OpenAirInterface code that has not yet been fixed.

3.4 Google Pixel 6 Settings

First of all, SIM cards and their reader are required in order to be able to configure the phones correctly. All required information can be found at the following address: https://open-cells.com/index.php/sim-cards/.

Then change the firmwares of the phones so that our network is visible and it becomes possible to connect using the European 5G bands (n28a, n41, n78). The procedure for installing new firmware is described here:
https://developers.google.com/android/images
Install the firmware **12.1.0 (SQ3A.220705.001.B1, Jul 2022, EMEA/APAC carriers)**.

Then, go into the advanced settings of the phone. To do this, go to the phone application, enter *#*#4636#*#*, then choose NR-only.

At the same time, to program a SIM card, first download the Opencells uicc program.

IMPORTANT: At the imsi level, a MCC of 001 and a MNC of 01 must be entered. The key of the telephone and the operator can be found in the databases of the core network. We have already programmed for 6 UE. They use the following imsis: 001010000000001, 001010000000002, 001010000000003, 001010000000004, 001010000000005, 001010000000006.
NB: Remember to turn off the wifi so to easily connect to the 5G platform

On the side of GNB, it is necessary to change for each configuration the MCC to 001 and MNC to 01. This because the operators restrict the configurations with a MCC of 208 indicating France that would not make possible some tests out of the operator networks.

On the core network side, keep the SQL databases oai_db and oai_db2 up to date. Install sqlite and mysql on the servers to do this. The databases are already configured to host 6 different UE. The only thing to change is the SQN of the last 4 UE according to the value indicated during the configuration of the SIM card. For the 001010000000001 and 001010000000002 the configuration is already done. It is then necessary to modify the .yaml file called at the time of the launching of the heart to make appear the good values of the MCC and MNC in the various places.

3.5 5G Experimental Platform Galery

This section aims at illustrating with a pictures gallery how the 5G experimental platform looks like, with additional comments on the pictures to explain what equipment implements what component (Figs. 3, 4, 5 and 6).

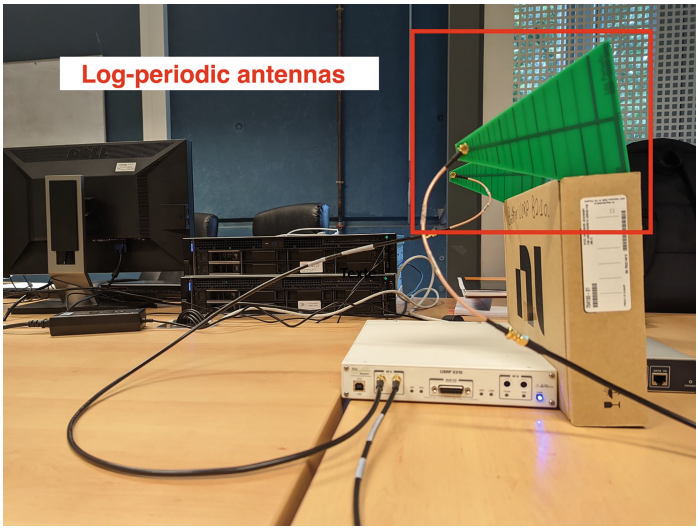

Fig. 3. Log-périodic antennas operating on the 850 MHz–6.5 GHz band

Fig. 4. Dell Precision Rack 7920 serveurs

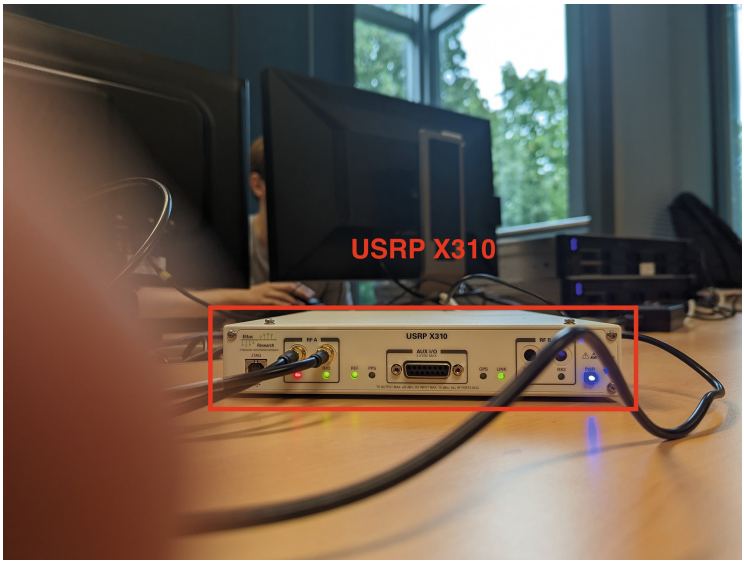

Fig. 5. Operating USRP X310

4 Experimental Evaluation

4.1 Simple IPerf Tests

In order to evaluate the performances of our 5G experimental platform, the IPerf tool was selected for generating and transferring packets. IPerf is a software tool for transferring IP packets from a source to a destination machines. In our case, we selected to transfer packets downlink from gNB to the UE.

Figures 7 and 8 show the evolution of the transfer throughput according to time. Two frequencies bands have been considered. The experiment makes the size of data to transmit and the bandwidth vary. The size of the data to transmit is noted N, the bandwidth is noted B. Frequencies bands considered are the n41 band whose central frequency is 2.5 GHz, and n78 band whose central frequency is 3.5 GHZ. These two bands have been selected as they are the most used ones, currently.

iPerf Results on the n41 Band - 2.5 GHz
Figure 7 shows that the transfer of small size packets, on a limited bandwidth, perfectly works. The theoretical throughput of 0.1 Mbits/s is reached.

Fig. 6. LAAS' 5G OAI experimental platform

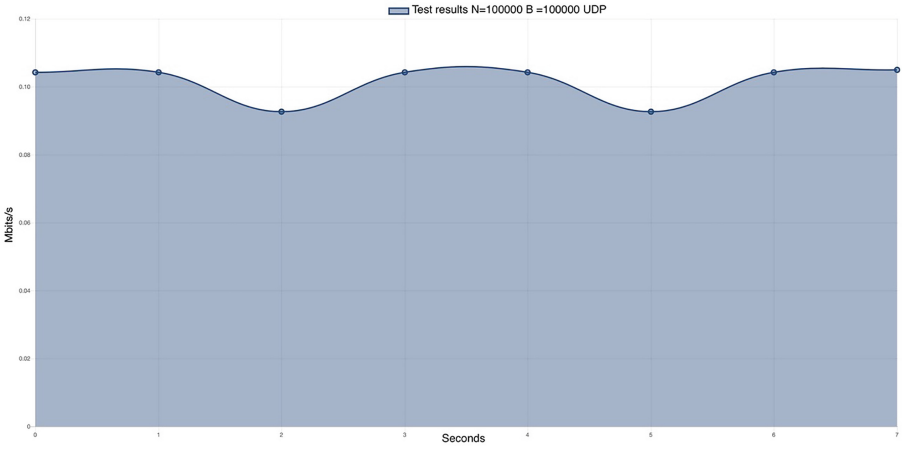

Fig. 7. Throughput on the n41 band with $N = 0.1$ Mbits and $B = 0.1$ MHz

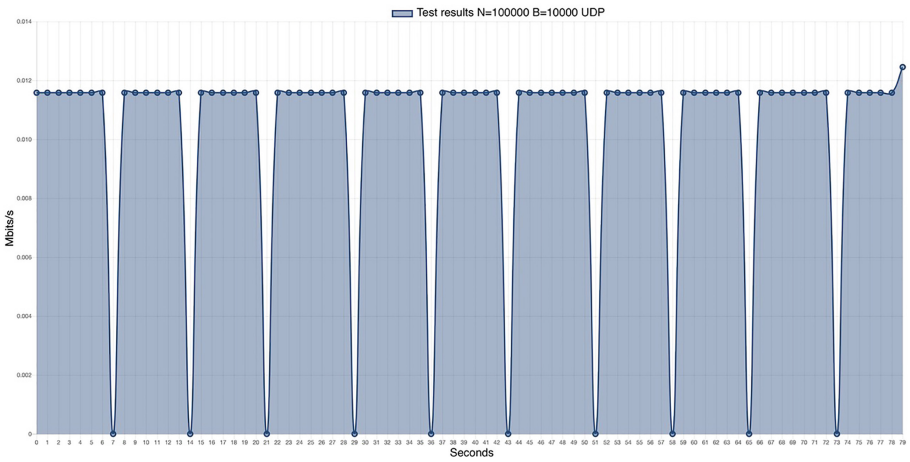

Fig. 8. Throughput on the n41 band with $N = 0.1$ Mbits and $B = 10$ kHz

Figure 8 shows that the transfer of small size packets, on a limited bandwidth, perfectly works. The theoretical throughput of 0.01 Mbits/s is reached.

4.2 File Transfers Evaluation

Current implementation of the 5G OAI platform cannot support cases where the uplink and downlink frequencies bands are different. This is the case for the n28 band whose central frequency on the uplink is 722 MHz, and the central frequency downlink is 780 MHz. It was then impossible to test this frequency band.

For n41 and n78 bands, the following experiments have been performed:

- n41 band: Transfer of a 100 Mbytes file with throughput measurement
- n78 band: Transfer of a 100 Mbytes file with latency measurement

The throughput was measured using Wireshark. This tool does not provide directly the latency measurement, but a simplifying hypothesis that was used consists in considering the latency as the half of the Round Trip Time (RTT). RTT is the time for a packet to go to the destination and going back to its source.

The throughput and latency measurements obtained for the transfer of the 100 MBytes file are given in Table 1.

Table 1. Performance indicators for the transfer of a 100 MBytes file on bands n41 and n78

	Débit moyen	Latence moyenne
n41 band	7.3 Mbits/s	21.2 ms
n78 band	9.3 Mbits/s	15.5 ms

The results were expected: when the frequency increases, the throughput increases as well, and the latency is reduced. The 3GPP norms indicate optimal throughputs of 10 Gbits/s for the eMBB slice, and an optimal latency of 1 ms. It was impossible on our platform to reach such values by a high degree of magnitude.

This can be explained by several factors:

- The limited emission power of our antennas (as we are in the public domain, we are not allowed to generate 5G signals that could interfere with the commercial 5G services).
- In the public domain in which the experiments took place, the environment is uncontrolled: there is a significant noise and frequent interferences from the commercial cellular networks.
- Our servers appear to have too limited computing power.

4.3 Experiments with the Google Pixel 6 Smartphone

With the Google Pixel 6 connected to our 5G OAI network, it was possible to tun the following tests and applications:

- Sending and receiving of e-mails
- Have an access to the Internet
- Perform audio and video streaming

In an empirical way, Fig. 9 illustrates that the use of the Google Pixel 6 smartphone with the 5G OAI network is very smooth, with the possibility of viewing several Internet pages simultaneously. Concerning the streaming, it was possible to view a 4K video.

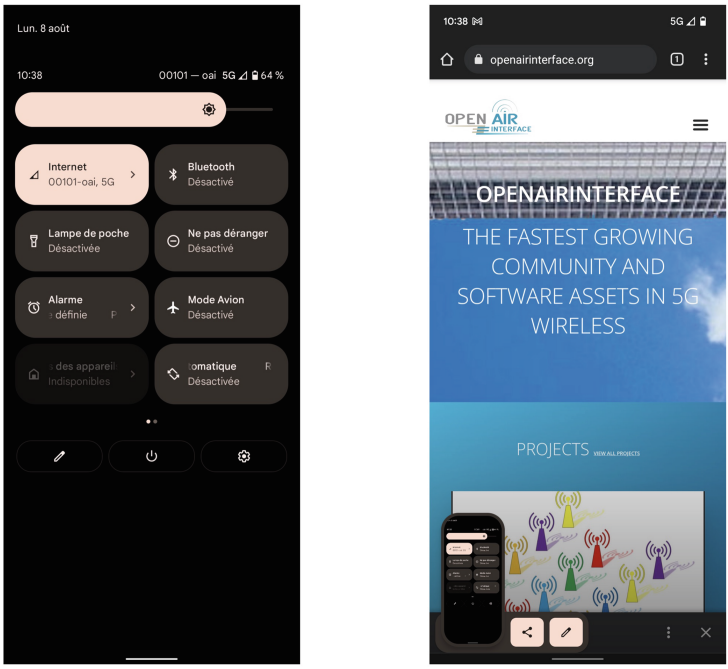

Fig. 9. The Google Pixel 6 gets Internet access through the 5G OAI network

5 Conclusion

5G has been designed for providing the appropriate services for a large range of applications requiring high throughput, low latency, a support for the IoT, or for Industry 4.0 business, etc. One of its strong statement is the sofwarization of most of its functions for providing more flexibility, and a support that can easily evolve for providing new services. At that time, the commercial 5G services are far from providing all the 5G promises, 5G operators just providing with their 5G networks more throughput for the users than a 4G network. The softwarization process is at its early age, and OAI is certainly the most advanced 5G software platform nowadays. As explained in this paper, 5G OAI is far from being completely developed and debugged. It was however possible to deploy a 5G experimental platform in our lab. This was a 6 months long study of the OAI documentation, OAI debugging, and OAI testing. But, as a result, our platform works, at least for simple services. For instance, the modules for 5G slicing are not available yet. The use of the smartphone is satisfactory with current applications, and with a single user. That is the limit of the current 5G OAI. The performances with all the softwarized components are limited compared to 5G objectives. That's what our evaluation experiments exhibited. The current limit is mostly due to the computing power of our servers. Let us recall that our servers are very powerful compared to computer standards: they have 18 cores,

large RAM, large communication buses, etc. Computing power would be today the cost of network flexibility.

For this reason, a significant effort has to be provided for optimizing the codes, but also by optimizing some of the normalized 5G mechanisms and protocols. This is the case for instance for 5G resources allocation [4]. Optimizing 5G mechanisms and protocol is part of our future works. We will also increase the size of the platform in order to be able to create several cells with several UEs in each of them. Last, the platform will be moved in an anechoic room to allow us to generate more powerful signals, and for avoiding being impacted by external noise and interferences.

References

1. Qazi, Z.A., et al.: A high performance packet core for next generation cellular networks. In: ACM SIGCOMM (2017)
2. Foukas, X., et al.: Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture. In: ACM MobiCom (2017)
3. Douarre, Q., Djelloul, M.: The 5G in Opensource: Installation of an experimental Platform using OpenAirInterface 5G. LAAS report, September 2022
4. Oussakel, I., Owezarski, P., Berthou, P., Houssin, L.: Toward radio access network slicing enforcement in multi-cell 5G system. J. Netw. Syst. Manage. (JNSM) **31**, 8 (2022)