

Analysis of Real-Time Video Streaming and Throughput Performance Using the Open Air Interface Stack on Multiple UEs

Bharat Agarwal¹, Clemens Felber², Neel Pandeya², Florian Kaltenberger³, Marco Ruffini⁴, and Gabriel-Miro Muntean¹

¹School of Electronic Engineering, Dublin City University, Ireland, ²NI Corp., Austin, TX, USA

³Eurecom, Campus SofiaTech, France, ⁴Computer Science and Statistic Department, Trinity College Dublin, Ireland

¹bharat.agarwal2@mail.dcu.ie, gabriel.muntean@dcu.ie

Abstract—The Open Air Interface (OAI) is an open-source project that implements 3GPP standards using general-purpose x86 computer hardware and commercial off-the-shelf (COTS) software-defined radio devices, such as the Universal Software Radio Peripheral (USRP). This paper presents a reference implementation for connecting a COTS UE, such as the Quectel and the Google Pixel 6a, to a 5G Stand-Alone (SA) testbed, based on USRP radios, the USRP Hardware Driver (UHD) software, and the OAI 5G NR software stack. The testbed is used to test and analyse the performance of video streaming over a 5G NR network for different TDD formats. In order to perform this analysis, we register the 5G COTS UE with the OAI 5G Core Network (CN) via the OAI 5G gNB, stream real-time videos vary the channel conditions, monitor the modulation coding scheme (MCS) adaptation on the gNB side, and observe the resulting quality at the UE side. We detail the implementation of the testbed, and the obstacles encountered when establishing the RF connection over-the-cable (OTC) and over-the-air (OTA).

Index Terms—OAI, USRP, UHD, 5G, Video streaming testbed

I. INTRODUCTION

In 2020, the 5G mobile networks were formally introduced as possible replacement for the widely used 4G LTE networks. Release 15 of the 3rd Generation Partnership Project (3GPP) introduces the 5G new radio (NR) core network and interfaces like Next Generation Access Protocol (NGAP). These support increases in wireless data speeds, bandwidth, coverage, and connectivity in 5G wireless communications and decreases in round-trip latency and energy usage. Release 16 (Rel-16) [1] focuses on providing Ultra-Reliable low latency communications for mission-critical services across the various 5G versions. In addition, Rel-16 allows apps to be prepared for new vertical sectors and deployment situations [2]. New Release 17 (Rel-17) [3] features include a NR up to 71 GHz, an IoT solution with narrow band, and extended reality support to test and implement changes that will further increase 5G’s suitability for augmented reality, virtual reality, and mixed reality. The 3GPP standards organization classifies all releases into three phases [4]. Stage 1 is the “service needs” level. In stage 2, the focus is mainly on determining what kind of functionalities need support based on service requirements.

Stage 3 involves solutions to be deployed in the network. This paper presents a solution to setup the testbed using the latest 5G technologies for supporting stage 3 criteria.

In the context of the 5G network resource management [5], this paper presents an implementation of a 5G Standalone (SA) testbed, based on Universal Software Radio Peripheral (USRP) radios from National Instruments (NI), the USRP Hardware Driver (UHD), the OpenAirInterface (OAI) 5G NR software stack, and a 5G commercial off-the-shelf (COTS) wireless module User Equipment (UE).

This paper describes how to build the 5G SA testbed and connect Quectel and Google Pixel 6a COTS UE to the 5G SA network and run some use cases. The paper starts by looking at the related work in Section II. Section III describes each component involved in the testbed. Section IV details the implementation of the testbed, and presents the use cases that are examined. Finally, result analysis is provided in Section V, and then conclusions and future work directions are presented in Section VI.

II. RELATED WORK

To the best of the authors’ knowledge, no study has provided a 5G SA testbed based on OAI with novel use cases operating on top of the testbed in the literature. However, some studies have employed the OAI code base for hardware-accelerated Low-Density Parity Checking (LDPC) decoding, network slicing, and physical layer bench-marking. However, none of the work has explained how the testbed evolved, how to stream high-definition video, and how the MAC scheduler adapts to channel conditions along with throughput analysis for different TDD formats.

The technical details of the FPGA accelerator design and the method of integrating the accelerator with OAI, the foundation of our 5G NR platform, were contributed by the authors of [6]. The modulation and coding system, the transmit and receive gains of the RF front-end, resource usage in time and frequency, and these characteristics have been carefully selected by the authors of [7] to control the throughput of a 5G network. The research investigation of the network slice

function was presented in [8], along with creating the 5G testbed. The paper [9] provides an overview of an experimental mobile network testbed built on the Fraunhofer core and the OAI for the radio access network (RAN). To complete the integration and connection test for the 5G NR SA system, [10] uses the universal and USRP as the hardware base and the OAI platform as the software architecture. The analysis of the test findings examined the modulation mode, random access procedure, and UL and DL data of 5G NR.

Despite their excellent performances, most approaches mentioned above have various limitations. For instance, the approach in [6] does not explicitly explain how to integrate the OAI with the LDPC accelerator system, while [7] only performed the bench-marking of the OAI 5G Physical layer, there were no use-cases run on the top of the developed testbed. In addition, [8] only provides network slicing on the 4G CN, [9] uses the CN from Fraunhofer, and [10] also does not run any use-case on the developed testbed apart from connecting UE to the 5G internet. These shortcomings have motivated us to propose the SA testbed based on OAI 5G NR with novel use-cases running on top.

III. SYSTEM ARCHITECTURE AND PRINCIPLE

This section is divided into two parts: the USRP N310, and the network topology of the laboratory physical environment.

A. OAI with USRP N310

For the evaluation of this 5G implementation, the team collaborated with NI to configure the USRP N310 SDR devices with an open-source general-purpose computing platform running the OAI software stack. The USRP devices are an open platform and use the open-source UHD device driver. USRP SDRs are versatile radios that enable a wide range of experimentation. The OAI implementation discussed in detail in [11] enables evaluation of the capabilities and limitations of the E2E 5G system.

B. Laboratory Physical Environment

The method of merging the x86 based server and USRP is used in the lab. The x86 based server finishes complete 5G NR protocol stack processing from RRC to PHY, and up-down conversion finished by the USRP on the target band *i.e.*, n78. The network topology of our lab setup environment and the hardware configuration used in our paper are introduced here:

- 1) **USRP N310:** In our testbed for the software radio front end, we select USRP N310. It implements real-time and low-latency processing and connects to a high-end server through high-speed interfaces. In the testbed USRP N310 operate as 5G gNB. Other USRP devices such as the USRP B210, USRP N320, and USRP X410 can also be used as 5G gNB.
- 2) **x86 based Server:** The testbed contains 3 x86 servers: one 5G CN; one 5G gNB and the UE terminal. As the functions of each module are different, the requirements for hardware configurations are also different. The 5G CN has a lower computational load, and processes less

data, so the hardware configuration requirements are lower than for the 5G gNB. The Quectel COTS UE does not need a high-performance computer since the UE stack runs on the Qualcomm chipset.

- 3) **Quectel COTS UE:** A 5G module designed specifically for IoT/eMBB applications is the Quectel RM500Q-GL. It supports both 5G NSA and SA modes. The UE computer controls the module using the QCOM tool from Qualcomm by sending AT commands to operate the wireless module. The QFlash tool can be used to update the firmware of the Quectel module.
- 4) **Google Pixel 6a:** The Pixel 6a is an Android smartphone designed, developed, and marketed by Google as part of the Google Pixel product line. Compatible networks GSM/EDGE, UMTS/HSPA+, CDMA EVDO Rev A, WCDMA, LTE, LTE Advanced, **5G sub-6 / mmWave**.
- 5) **Miscellaneous Components:** The testbed developed and presented in this paper is over-the-cable (OTC) and over-the-air (OTA). For OTC operation, components such as fixed and variable attenuators, SMA cables, and 1:2 and 1:4 power splitters were used.
- 6) **System Network Topology:** The testbed topology is shown in Fig. 1. It consists of a 5G OAI CN system, a 5G OAI gNB system, and a 5G Quectel and Google Pixel 6a COTS UE. The USRP N310 connected to the gNB system via dual SFP+ 10 Gbps Ethernet links, and then to a switch, to connect with the 5G CN system. A laptop controls the Quectel COTS UE, with the Qualcomm device driver installed. The Quectel COTS UE is attached with gNB USRP using RF splitters and attenuators, as shown in Fig. 2.

IV. SYSTEM SETUP PROCESS

The OAI-based 5G NR SA system includes three parts: CN, gNB, and COTS UE (Quectel and Google Pixel 6a), which are introduced in detail in this section.

A. Core Network Construction

The OAI 5G CN consists of four Docker containers: AMF, SMF, UPF, and NRF. To configure the OAI 5G CN system, first clone the OAI 5G CN from the OAI GitLab server, and then build the Docker containers, and then start them one-by-one. Run Wireshark to observe the packets exchanged between the various containers.

B. Construction of OAI gNB

The x86 server and the USRP N310 make up gNB, which is split into two components as part of the AN. The following subsection describes the x86 server's environment configuration, the installation of the UHD driver, and related configuration parameters.

- 1) **Environment Configuration:** There are various performance tuning optimizations that need to be performed on the Ubuntu operating system on the gNB system, such as increasing socket buffer and Ethernet ring buffer

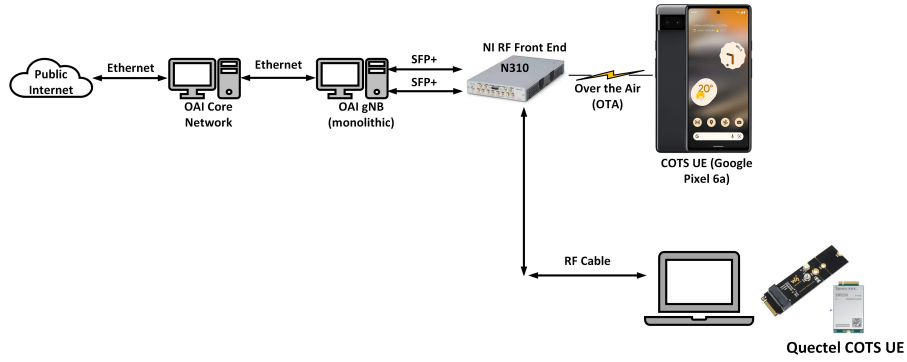


Figure 1: System architecture of the integrated 5G SA system using Quetel COTS UE, and Google Pixel 6a.

The variable attenuator at 30dB is the result of an experimental approach to get the attach procedure up and running.

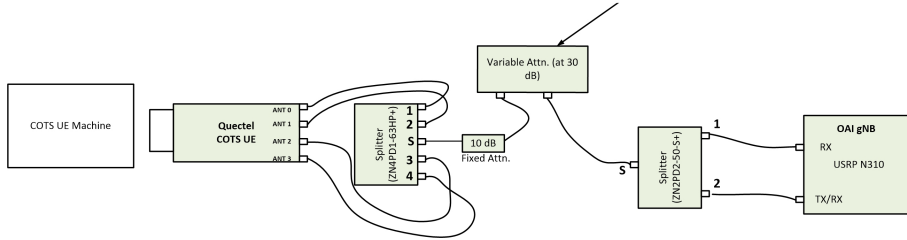


Figure 2: Connection between 5G gNB and Quetel COTS UE.

sizes, configuring CPU governors, and using a low-latency kernel, which can be installed from the package manager. In addition, it is necessary to disable specific CPU capabilities, such as the C-states, P-states, hyper-threading, and frequency scaling features, and these can be set in the BIOS configuration page at boot-time.

- 2) **UHD Driver Installation:** The UHD driver for the USRP radio should be build and installed from source code. After UHD is installed, connect the USRP to the x86 server, and run the `uhd_find_devices` command to check the connection, and then the `uhd_usrp_probe` command to verify the configuration of the radio.
- 3) **Parameter Configuration:** The `build_oai` script is used to build and install the gNB software from source code, and the `source oaienv` command is used to load environment variables for running the gNB stack. The Mobile Country Code (MCC), Mobile Network Code (MNC), and USRP IP address are set in the gNB configuration file, and the network interface addresses must match the CN configuration.

C. Building the Terminal UE

1) **Quetel COTS UE:** For the terminal UE, we used the Quetel COTS UE. The COTS UE has a PCIe connector that sits on the top of the M.2 adapter, and is connected via the host computer's USB port. We used a test SIM card from Open Cells, and used the USIM programmer software to configure the SIM card. The essential parameters for the SIM are Key (Secret Key for USIM) and OPC (Secret Key for the operator). It is critical that these values match with the database available in the CN. We use the QCOM tool to send AT commands to

the COTS UE to begin operation and attach to the 5G gNB. The relevant AT commands are:

- 1) `AT+QMBNCFG="select","ROW_Commercial"`—→this is needed to unlock the Quetel (when done, unplug/replug to reboot the module)
- 2) `AT+QNWPREFCFG="nr5g_band"`—→to see what 5G bands are configured
- 3) `AT+QNWPREFCFG="mode_pref"`—→to see the mode
- 4) `AT+QNWPREFCFG="mode_pref", nr5g`—→to set the mode to SA
- 5) `AT+QNWPREFCFG="nr5g_disable_mode",0`—→ to enable 5G operation
- 6) `AT+CGDCONT=1,"IP","oai","0.0.0.0",0,0`—→to specify the PDP context parameters for a specific context cid (be sure that there is not another APN available by default in the USIM, such as IMS, and if it exists, then remove it with the command `AT+CGDCONT=cid`)

2) **Google Pixel 6a:** Google Pixel 6a COTS UE has Android 13 with open cells sim card. To connect the Pixel 6a with OAI stack some of the modifications need to be done:

- 1) Open cell sim card must be programmed with IMSI 001010000000001, where MCC 001 and MNC is 01. Note the COTS UE should automatically read MCC and MNC as shown in Fig.3.
- 2) In Settings under Edit Access Point option change APN from default to OAI, and select NR in Bearer option as shown in Fig.3.

D. Test Platform Integration Process

After each part is prepared, the whole system can be integrated. We have divided the whole integration process in

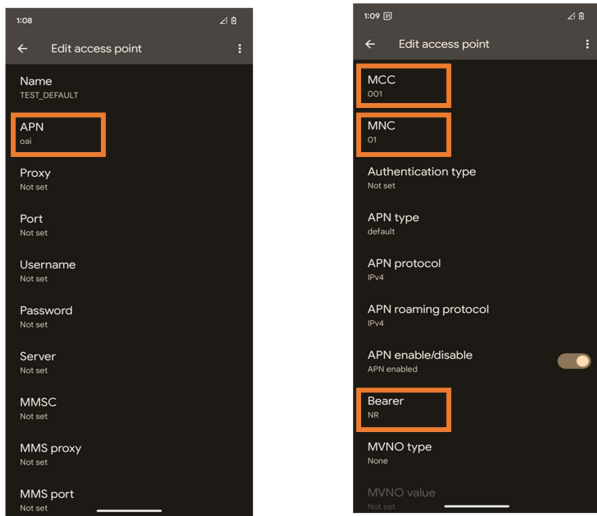


Figure 3: Setting of the Google Pixel 6a device.

three sessions as shown in Fig. 4.

- 1) **Start the 5G CN** using the command “`python3 ./core-network.py -type start-mini -fqdn no -scenario 1`”. In this command `start-mini` corresponds to CN with only SMF, AMF, UPF and NRF containers. Once the CN is on, the demo-oai network interface will be available in Wireshark. Open the demo-oai interface to see the packets being exchanged between the different containers.
- 2) **Start the gNB**: Connect the USRP to the gNB server and check whether the connection is successful with the “`uhd_find_devices`” command. Load environment variables in the OAI root directory, and invoke the gNB software. Once the gNB is up and running, the NGAP setup request and response message will be exchanged between the gNB and AMF container.
- 3) **Start the terminal UE**
 - 1) **Quectel COTS UE**: Turn on the Quectel COTS UE by sending the AT command “`AT+CFUN=1`”. This command enables full functionality in the COTS UE.
 - 2) **Google Pixel 6a**: There is no such AT command is required to turn on the Pixel 6a.

V. EXPERIMENTAL RESULTS AND ANALYSIS

We discuss the findings and their analysis for four use cases. First, we discuss how to use the Quectel COTS UE to access the internet and browse the web. Second, we cover how to stream high-definition video from the server (CN) to the client (Quectel COTS UE). Third we cover was how the video quality fluctuates depending on the channel conditions between the Quectel UE and the gNB. Lastly, we perform throughput analysis for two different TDD formats using Google Pixel 6a COTS UE.

A. System Configuration

A Time Duplex Divison (TDD) system with 7 DL, 2 UL and 1 SL is integrated with the channel bandwidth of 40 MHz with

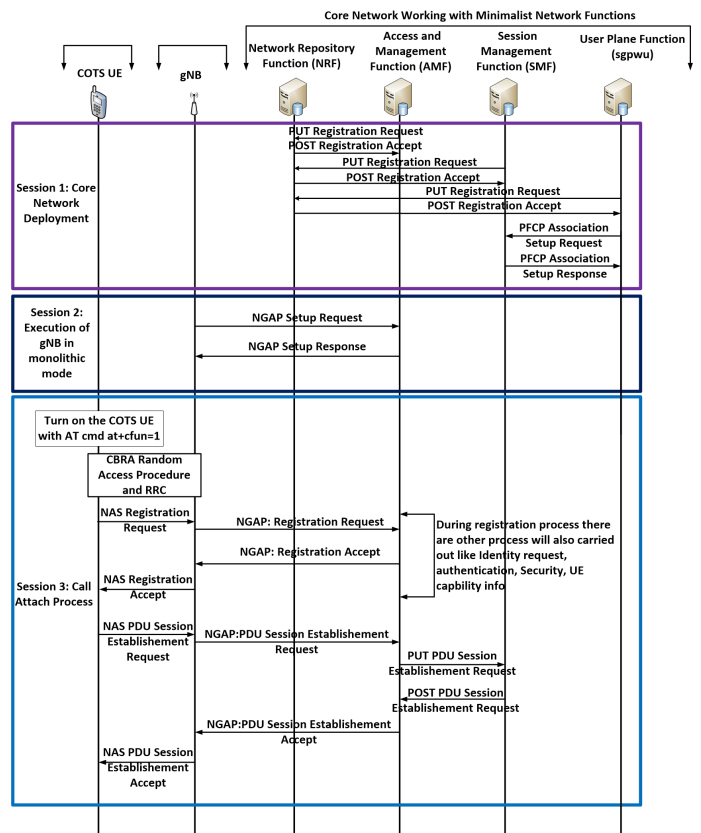


Figure 4: Integration process explained via different sessions

a sub-carrier spacing of 30 KHz. The Antenna Configuration is 2x2 TX and RX. The sample rate in USRP processing is 122.88 Msps.

B. Iperf and Ping Test

After a successful connection we run the DL and UL iperf and Ping test. We have achieved DL iperf throughput of 200 Mbps and 7 Mbps for UL. For ping test, we achieved average round trip time latency of 12.392 msec.

The commands to run DL and UL iperf test are:

- 1) DL iperf test: Server on COTS UE = “`iperf -s -i 1 -u -B 12.1.1.x`”, Client on CN = “`docker exec -it oai-ext-dn iperf -c 12.1.1.x -u -b yM -bind 192.168.70.134`”
- 2) UL iperf test: Server on CN = “`docker exec -it oai-ext-dn iperf -s -i 1 -u -B 192.168.70.134`”, Client on COTS UE = “`iperf -c 192.168.70.134 -u -b yM -bind 12.1.1.x`”
- 3) Ping Test: Run this commands on CN “`docker exec -it oai-ext-dn ping 12.1.1.x`”

C. Internet Connection to Quectel COTS UE

For connecting the Quectel COTS UE with the internet we need to modify the SMF and SPGWU docker container on the 5G CN computer. We need to configure SMF with the correct Domain Name System (DNS) and enable Network Address Translation (NAT) in SPGWU (by default) in order to enable the UE to connect to the Internet. Enter the correct DNS in the SMF configuration file in the follow-

ing fields “*DEFAULT_DNS_IPV4_ADDRESS*=‘8.8.8.8’” and “*DEFAULT_DNS_SEC_IPV4_ADDRESS*=‘8.8.8.8’”.

D. Streaming of High Resolution Video from Server to Client

For streaming the high resolution video from server to client, we use the VLC media player. We setup VLC as a server on the 5G CN system, and as a client on Quectel COTS UE system. The video used has a resolution of 1080x720p with a frame rate of 30 fps. The steps for configuring VLC as a server are listed below.

- 1) Under *Media* option open the *Stream* option
- 2) Under *Stream* option, specify the video to be streamed, and click on the *Stream* and then *Next* buttons.
- 3) Under *New destination*, add UDP (legacy), and click the *Next* button.
- 4) In the address bar, add the IP address of the Quectel COTS UE (to know what IP address allocated, see the SMF logs on the CN machine using the command “*docker logs oai-smf*”), and then click *Next*.
- 5) Uncheck the *Activate Trans-coding* option, and select the *Video-H.264+MP3(MP4)* option under the *Profile* field, and click the *Next* button.
- 6) Finally, click on the *Stream* button, and the video should start streaming to the particular IP address and port number specified.

Once the VLC server is ready, run the following command in the 5G CN system “*sudo ip route add 12.1.1.x via 192.168.70.134 dev demo-oai*” (12.1.1.x is the IP address allocated to Quectel COTS UE, 192.168.70.134 is the IP address of the UPF docker container, and demo-oai is the bridge to which all dockers in CN are connected). Open VLC on the client side and follow the steps listed below.

- 1) Under *Media*, select the Open Network Stream option.
- 2) Under *Please enter a network URL field*, enter the address “*udp://@:1234*”.

After completing these steps, the video should start streaming from the server to client.

E. Varying channel conditions between gNB and COTS UE

For varying the channel conditions between the gNB and the Quectel COTS UE, we used a variable attenuator, as shown in Fig. 2. We vary the attenuation from 40 dB to 80 dB, and we note the corresponding change of the Modulation Coding Scheme (MCS) used by the gNB, and we observe the video frames being lost. Based on the channel quality indicator (CQI) reported by UE, the MAC scheduler adjusts the MCS as shown in Fig. 6a. As the amount of attenuation between the gNB and Quectel COTS UE increases, the MCS is adjusted to lower values, and the number of lost video frames increases. This occurs because as the MCS value is lowered, the number of bits per symbol decreases, and as the channel has less dynamic range due to the increased attenuation, the channel throughput decreases. At 80 dB attenuation, the connection was lost between the gNB and Quectel COTS UE.

F. Throughput Analysis

For the throughput analysis, we use two different TDD formats with Pixel 6a COTS UE. The connection procedure of the Pixel 6a is exactly similar to Quectel UE, but the only difference is that this test is performed Over the Air (OTA), no AT commands are used to turn on/off the Pixel 6a. The system parameters used in this analysis is similar to describe in Section V.A except Channel Bandwidth, as we run for five different bandwidths i.e., 40, 60, 80 and 100 MHz, Antenna Configuration, we perform analysis for both 1x1 and 2x2 MIMO, and TDD format, we use to two different TDD formats i.e., 7 DL, 2UL and 1 Special slot as shown in Fig.5 (a) and 3 DL, 1 UL and 1 Special slot as shown in Fig.5 (b).

G. How to Change TDD pattern in OAI?

The TDD patten can be changed in OAI by changing this specific values of the gNB configuration value. The variables are that need to be changed:

- 1) *dl – UL – TransmissionPeriodicity*: Periodicity of the DL-UL pattern. If *dl – UL – Tx Periodicity -v1530* is configured, *dl – UL – Tx Periodicity* is ignored.
- 2) *Number of DL Slots*: Number of consecutive full DL slots at the beginning of each DL-UL pattern.
- 3) *Number of DL symbols*: Number of consecutive DL symbols in the beginning of the slot following the last full DL slot.
- 4) *Number of UL slots*: Number of consecutive full UL slots at the end of each DL-UL pattern.
- 5) *Number of UL symbols*: Number of consecutive UL symbols at the end of the slot before the first UL slot.

Fig. 5 (a) shows the TDD format in which the value of variable *dl – UL – TransmissionPeriodicity* is 5msec, Number of DL Slots 7, Number of DL symbols 6, Number of UL slots 2, Number of UL symbols 4. Similarly the second different TDD format is shown in Fig. 5 (b).

H. Throughput for TDD pattern 1:

Fig. 6b shows the throughput achieved for the TDD pattern 1 as shown in Fig. 5 (a) for the different bandwidths. The throughput achieved for bandwidth 40 MHz (106 PRBs) is around 72 Mbps, for 60 MHz (162 PRBs) 120 Mbps, for 80 MHz (208 PRBs) 250 Mbps and for 100 MHz (272 PRBs) around 245 Mbps, respectively. Achieving less throughput at 100 MHz bandwidth, still a point for further analysis.

I. Throughput for TDD pattern 2:

Fig. 6c shows the throughput achieved for the TDD pattern 2 as shown in Fig. 5 (b) for the different bandwidths. The throughput achieved for bandwidth 40 MHz (106 PRBs) is around 120 Mbps, for 60 MHz (162 PRBs) 140 Mbps, for 80 MHz (208 PRBs) 250 Mbps and for 100 MHz (272 PRBs) around 245 Mbps, respectively. For BW 80 and 100 MHz we achieved same throughput for both TDD patterns. Achieving same throughput at both different TDD patterns, still a point for further analysis.

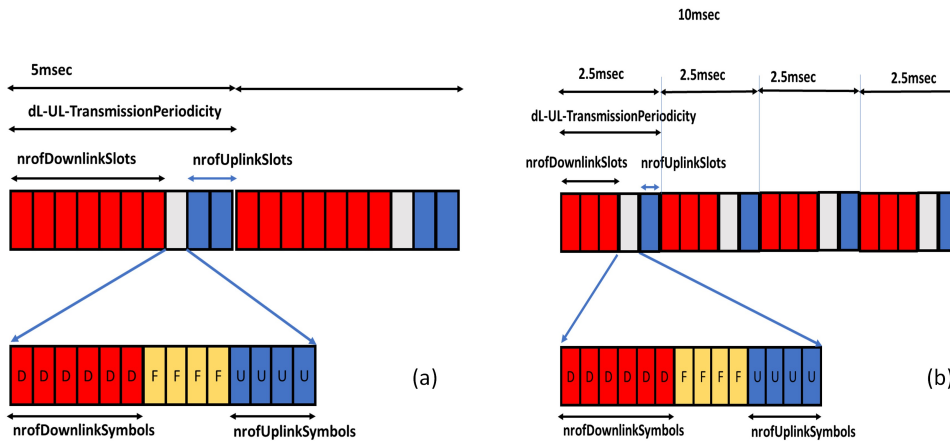
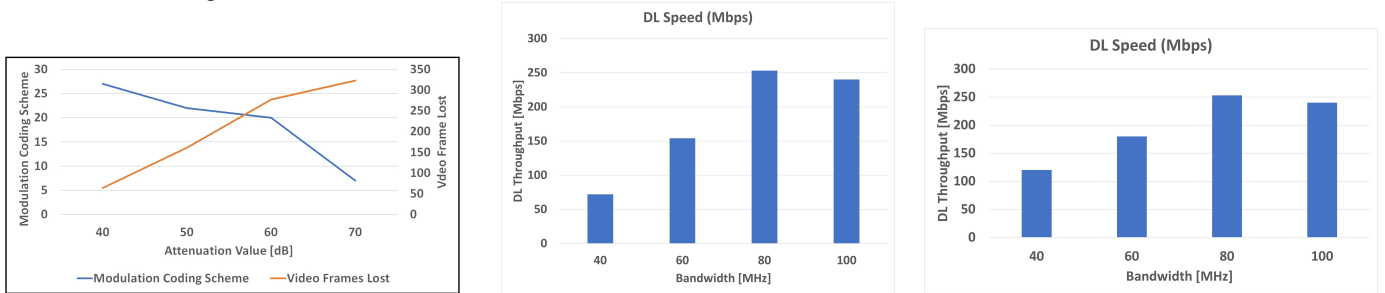


Figure 5: Different TDD formats (a) 7 DL, 2 UL and 1 SL, and (b) 3 DL, 1 UL and 1 SL.



(a) MCS and Video Frames variation with attenuation. (b) Throughput for different bandwidths for TDD pattern 1. (c) Throughput for different bandwidths for TDD pattern 2.

Figure 6: Testing results

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a system reference design to evaluate the performance of video streaming on a standards-compliant 5G NR SA testbed, based on open-source technologies. It shows how the COTS UE like Quectel and Google Pixel operate with a gNB and CN to stream high-resolution videos over the 5G SA network in real-time and at high quality levels. The testbed uses a x86 server and USRP SDR radio platform for the hardware functions, and the OAI software for the 5G stack. Because the testbed is based on open standards, COTS equipment, and open-source software, it provides a highly flexible testing environment with granular low-level visibility into all network components, and is cost-effective. The testbed also offers an ideal starting point for additional studies involving aspects such as network slicing, FR2 operation, resource optimization, and other applications.

VII. ACKNOWLEDGEMENTS

The Science Foundation Ireland's support via grants 18/CRT/6224 (D-REAL) and 12/RC/2289_P2 (INSIGHT) is gratefully acknowledged.

REFERENCES

- [1] K. Flynn, "A Global Partnership," Release 16. [Online]. Available: <https://www.3gpp.org/release-16>.
- [2] M. Ivezic <https://5g.security/author/marin/>, "Introduction to 3GPP and 3GPP 5G Releases 15, 16 and 17," 04-Oct-2020. [Online]. Available: <https://5g.security/5g-technology/5g-3gpp-releases-15-16-17>.
- [3] K. Flynn, "A Global Partnership," Release 17. [Online]. Available: <https://www.3gpp.org/release-17>.
- [4] K. Flynn, "A Global Partnership," CT progress on Rel-16 Stage 3. [Online]. Available: <https://www.3gpp.org/news-events/2100-ct-progress-on-rel-16-stage-3>.
- [5] B. Agarwal, M. A. Togou, M. Marco and G. -M. Muntean, "A Comprehensive Survey on Radio Resource Management in 5G HetNets: Current Solutions, Future Trends and Open Issues," in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2495-2534.
- [6] E. A. Papatheofanous, D. Reisis, and K. Nikitopoulos, "LDPC hardware acceleration in 5G Open Radio Access Network platforms," *IEEE Access*, vol. 9, pp. 152960-152971, 2021.
- [7] S. S. Nakkina, S. Balijepalli and C. R. Murthy, "Performance Benchmarking of the 5G NR PHY on the OAI Codebase and USRP Hardware," *Int. ITG Workshop on Smart Antennas (WSA)*, 2021, pp. 1-6.
- [8] A. Shorov, "5G testbed development for network slicing evaluation," *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, 2019.
- [9] M. Vilakazi, C. R. Burger, L. Mboweni, L. Mamushiane, and A. A. Lysko, "Evaluating an evolving OAI testbed : Overview of options, building tips, and current performance," *Int. Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021.
- [10] Y. Gao, X. Zhang and H. Yuan, "Integration and Connection Test for OpenAirInterface 5G Standalone System," *IEEE Int. Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2021.
- [11] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "OpenAirInterface: Democratizing innovation in the 5G era," *Computer Networks*, vol. 176, pp. 107284, July 2020.
- [12] B. Agarwal, M. A. Togou, M. Ruffini and G. -M. Muntean, "QoE-Driven Optimization in 5G O-RAN Enabled HetNets for Enhanced Video Service Quality," in *IEEE Communications Magazine*, vol. 61, no. 1, pp. 56-62, January 2023, doi: 10.1109/MCOM.003.2200229.