An Open, Programmable, Multi-vendor 5G O-RAN Testbed with NVIDIA ARC and OpenAirInterface

Davide Villa*§, Imran Khan*§, Florian Kaltenberger*‡, Nicholas Hedberg¶, Rúben Soares da Silva‖,
Anupa Kelkar¶, Chris Dick¶, Stefano Basagni*, Josep M. Jornet*, Tommaso Melodia*,
Michele Polese*, and Dimitrios Koutsonikolas*
*Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, U.S.A.
‡Eurecom, Sophia Antipolis, France, ¶NVIDIA, Inc, Santa Clara, CA, U.S.A., ‖Allbesmart, Castelo Branco, Portugal

***Abstract*—The transition of fifth generation (5G) cellular systems to softwarized, programmable, and intelligent networks depends on successfully enabling public and private 5G deployments that are (i) fully software-driven and (ii) with a performance at par with that of traditional monolithic systems. This requires hardware acceleration to scale the Physical (PHY) layer performance, end-to-end integration and testing, and careful planning of the Radio Frequency (RF) environment. In this paper, we describe how the X5G testbed at Northeastern University has addressed these challenges through the first 8-node network deployment of the NVIDIA Aerial RAN CoLab (ARC), with the Aerial Software Development Kit (SDK) for the PHY layer, accelerated on Graphics Processing Unit (GPU), and through its integration with higher layers from the OpenAirInterface (OAI) open-source project through the Small Cell Forum (SCF) Functional Application Platform Interface (FAPI). We discuss software integration, the network infrastructure, and a digital twin framework for RF planning. We then profile the performance with up to 4 Commercial Off-the-Shelf (COTS) smartphones for each base station with iPerf and video streaming applications, measuring a cell rate higher than 500 Mbps in downlink and 45 Mbps in uplink.**

***Index Terms*—Private 5G; GPU offloading; O-RAN**

I. Introduction

Fifth generation (5G) networks have introduced significant improvements in Radio Access Network (RAN) performance, with hundreds of Mbps average user data rates and extremely low latency [1]. 5G systems are also transitioning toward more open, intelligent, and programmable architectures [2]. The combination of the 5G capabilities as defined by the 3rd Generation Partnership Project (3GPP) and the openness, softwarization, and programmability brought along by the Open RAN paradigm and O-RAN ALLIANCE have the potential to transform how we deploy and manage wireless mobile networks. Such systems leverage network disaggregation, with the layer of the 5G RAN distributed across different network functions, i.e., the Central Unit (CU), the Distributed Unit (DU), and the Radio Unit (RU); softwarization, with the protocol stack functionalities implemented in software rather than dedicated circuits; and intelligent control, with closed-loop solutions for the RAN optimization and automation [3].

These solutions are associated with lower capital and operational expenditures, facilitated by a supply chain with increasing diversity and robustness [4], which includes open-source projects [5, 6]. Combined with higher spectrum availability, they have paved the way toward the private 5G systems, which complement public 5G networks for site-specific use cases (e.g., industrial control, events, warehouse automation, etc).

While there are significant upsides in the transition to software-driven, disaggregated, programmable systems, there are still several challenges that need to be addressed before they can deliver Key Performance Indicators (KPIs) aligned with traditional cellular systems. First, deploying end-to-end cellular systems is still a complex proposition, as automation and zero-touch provisioning and configuration are still far out in the radio domain. Second, while disaggregated solutions boast a diverse vendor ecosystem, they face challenges associated with end-to-end integration and interoperability across products [7–9]. Third, completely virtualized 5G deployments need to tackle the high computational complexity of the Digital Signal Processing (DSP) for the Physical (PHY) layer, which uses about 90% of the available compute when run on general-purpose CPUs. Finally, how to design Artificial Intelligence (AI) and Machine Learning (ML) solutions that generalize well across a multitude of cellular network scenarios remains an active area of research. Therefore, there is a need for a concerted effort that spans multiple communities (hardware, DSP, software, DevOps, AI/ML) toward the design and deployment of open, programmable, multi-vendor cellular networks and testbeds that can support private 5G use cases and requirements with production-level stability and performance.

In this paper, we introduce X5G, a private 5G network testbed deployed at Northeastern University in Boston, MA, and based on a combination of open-source and programmable components from the PHY layer up to the Core Network (CN). We discuss for the first time the integration of a PHY layer implementation based on Graphics Processing Unit (GPU) acceleration (i.e., NVIDIA Aerial) with higher layers from OpenAirInterface (OAI) [5]. Such integration leverages the Small Cell Forum (SCF) Functional Application Platform Interface (FAPI) for the interaction between the Medium Access Control (MAC) and PHY layers. It allows the inline acceleration of demanding PHY tasks on GPU, hardware that is well equipped with massive parallelization of DSP operations, enabling scalability and the embedding of AI/ML in the RAN. The platform, known as NVIDIA Aerial RAN CoLab (ARC), is deployed on

This work was partially supported by the U.S. National Science Foundation under grant CNS-2117814.

§Davide Villa and Imran Khan are co-primary authors.

a dedicated multi-vendor infrastructure with 8 servers for the CU and DU, 4 RUs that can be installed in a lab space, O-RAN 7.2 fronthaul and timing hardware, and a dedicated 5G CN. It delivers KPIs representative of 5G sub-6 GHz systems, with cell throughput north of 500 Mbps with 4 connected User Equipments (UEs) and a 100 MHz carrier. NVIDIA ARC and OAI are tools that can be readily used to develop 5G and beyond intelligent use cases, thanks to the combination of a performance that improves over most open-source, non-accelerated solutions while maintaining the openness and code accessibility typical of such systems. The rest of the paper is organized as follows. The FAPI-based integration is described in Section II. Section III concerns the network infrastructure. System performance is evaluated in Section IV via digital twinning and multiple Commercial Off-the-Shelf (COTS) UEs and applications. Section V draws conclusions and future work.

II. Full-stack Programmable gNBs with NVIDIA Aerial and OAI

Figure 1 shows the software architecture of the Aerial RAN CoLab (ARC) gNBs, following the basic O-RAN architecture split into CU, DU, and RU. The DU is further split into a DU-low, implementing Layer 1 (PHY, or L1), and into a DU-high, implementing Layer 2 (the MAC and the Radio Link Control (RLC)). They communicate over the 5G FAPI interface specified by the SCF [10]. The DU-low is implemented using the NVIDIA Aerial Software Development Kit (SDK) [11] on GPU accelerator cards, whereas DU-high and CU are implemented by OAI on general-purpose CPUs. In our setup, we combine the CU and the DU to a monolithic gNB deployed in a Docker container. Layer 1 is deployed in a separate container.

The FAPI defines two sets of procedures. Configuration procedures handle the management of the PHY layer and happen infrequently. On the contrary, slot procedures happen in every slot (i.e. every 500 μs for a 30 kHz subcarrier spacing) and determine the structure of each slot (DL and UL). In our case, Layer 1 acts as the master and sends a slot indication every slot. Layer 2 (i.e., the MAC, or L2) replies with a UL or DL request message that specifies what the PHY needs to do at every slot. The L1 may also send indication messages to the L2 indicating reception of data for the Random Access Channel (RACH), Uplink Control Indication (UCI), Sounding Reference Signal (SRS), checksums, or the user plane. To guarantee real-time performance, the communication between the L1 and L2 containers is implemented using the NVIDIA Inter-Process Communication (IPC) (NVIPC) library, a generic shared memory IPC solution capable of tracing the FAPI messages and exporting them to a pcap file analyzable with Wireshark.

On the southbound side, the NVIDIA Layer 1 uses the O-RAN 7.2 interface [12] to communicate directly with the O-RU, in our case from Foxconn. The O-RAN 7.2 transports frequency domain IQ samples with optional block floating point compression over a switched network allowing for flexible deployments. The protocol includes synchronization, control, and user planes. The S-plane is based on PTPv2. We use sync architecture option 3 [13], where the fronthaul switch provides synchronization to both DU and RU. The specification also includes a management plane, although it is currently not supported.

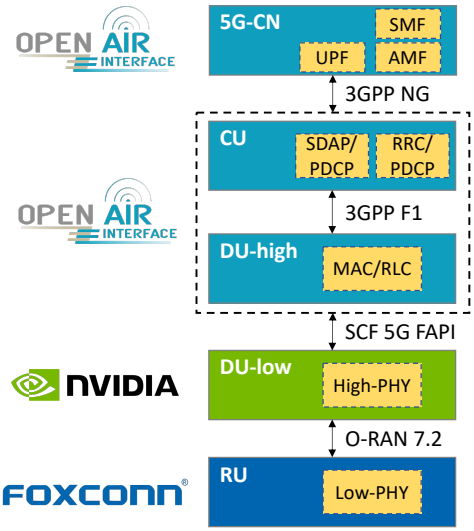


Fig. 1: Architecture of the gNB following O-RAN specifications and consisting of a Foxconn O-RU, an O-DU-low based on Nvidia Aerial SDK, an O-DU-high and an O-CU based on OAI with their corresponding interfaces.

Table I summarizes the main features and operational parameters of the ARC deployment in the X5G testbed. The protocol stack is aligned with 3GPP Release 15, and uses the 5G n78 Time Division Duplexing (TDD) band and numerology 1. The TDD pattern, which repeats every 2.5 ms, includes three downlink slots, one special slot (which is not used due to limitations in the Foxconn RUs), and an uplink slot.

ARC, comprising the Aerial PHY and the OAI higher layers, is open and can be extended with custom features and functionalities. The components implemented by OAI are published under the OAI public license v1.1 created by the OAI Software Alliance (OSA) in 2017 [14]. This license is a modified Apache v2.0 License, with an additional clause that allows contributors to make patent licenses available to third parties under Fair, Reasonable, And Non-Discriminatory (FRAND) terms, similar to 3GPP for commercial exploitation. This ensures that companies holding intellectual property in related areas can contribute. The usage of OAI code is free for non-commercial/academic research purposes. The Aerial SDK is available through an early adopter program [15].

TABLE I: X5G ARC deployment main features.

Feature	Description
3GPP Release	15
Frequency Band	n78 (FR1, TDD)
Carrier Frequency	3.75 GHz
Bandwidth	100 MHz
Subcarrier spacing	30 kHz
TDD config	DDDSU*
Number of antennas used	2 TX, 2 RX
MIMO config	2 layers DL, 1 layer UL
Max theoretical cell throughput**	525 Mbps DL, 94 Mbps UL

*Currently the special slot is unused due to limitations in Foxconn radios.
**The single user maximum theoretical DL throughput is 350 Mbps since we can schedule a maximum of 2 DL slots per user in one TDD period, as only 2 ACK/NACK feedback bits are available per user.

III. X5G Architecture

This section describes the X5G physical deployment that is currently located on the Northeastern University campus in Boston, MA[1]. The deployment spans a server room with a dedicated rack for the private 5G system and an indoor laboratory open-space area with benches and experimental equipment that provide a realistic Radio Frequency (RF) environment with rich scattering and obstacles. Figure 2 illustrates the hardware infrastructure that we deployed to support the X5G operations, including 8 ARC nodes, a dedicated CN, and a fronthaul infrastructure with timing and synchronization capabilities.

CU and DU. The 8 ARC nodes are deployed on Gigabyte E251-U70 servers, which come with (i) a half rack chassis, for deployment in RAN and edge scenarios; and (ii) a Broadcom PEX 8747 PCI switch, which enables direct connectivity between cards installed in two dedicated PCI slots, without the need for interactions with the CPU. Specifically, the two PCI slots are used for an NVIDIA A100 GPU, which serves as the compute resource for the NVIDIA Aerial PHY layer, and for a Mellanox ConnectX-6 Dx Network Interface Card (NIC), which is used for the 7.2 fronthaul interface and equipped with two QSFP ports. Through the PCI switch, the NIC can offload or receive packets directly from the GPU, enabling low-latency packet processing. The servers are also equipped with a 24-core Intel Xeon Gold 6240R CPU and 96 GB of RAM.

CN and Backhaul. For the CN, we deploy the micro-services-based 5G Core from OAI on a Dell R750 server with 56 cores, 256 GB RAM, and multiple network interfaces. Two additional CNs are available in the system, including Open5Gs and a commercial core from A5G, which we are integrating with ARC as part of our future work. A Dell S4112T-ON switch provides backhaul connectivity between the gNBs, the CN server, and the Internet, accessed through the Northeastern University switched network.

Fronthaul and Synchronization Infrastructure. The fronthaul infrastructure combines switching and synchronization capabilities. It features a Dell S5248F-ON switch providing QSFP ports for the connection to the Mellanox cards on the gNB, and 48 SFP+ ports for connectivity to the RUs. The switch acts as a boundary clock in the synchronization plane, and it receives Precision Timing Protocol (PTP) signals from a Qulsar QG-2 acting as grandmaster clock. The Qulsar unit is connected to a GPS antenna for precise class 6 timing, and generates both PTP and SyncE to provide frequency, phase, and time synchronization compliant with the ITU-T G.8265.1, G.8275.1, and G.8275.2 profiles. The switch is PTP-aware providing full on-path support, necessary to distribute phase synchronization.

RU. Currently, we deploy 2 Foxconn RPQN 4T4R RUs, operating in the $3.7 - 3.8$ GHz band, with 2 additional units being tested in the lab. The units have 4 externally mounted antennas, each antenna with a 5 dBi gain, and 24 dBm of transmit power. The Over-the-Air (OTA) transmissions are regulated as part of the Northeastern University Federal Communications Commission (FCC) Innovation Zone [16], with an additional transmit attenuation of 20 dB per port to comply with transmit power limits and guarantee coexistence of multiple in-band RUs in the same environment. As discussed in Section IV, we leverage two of these RUs for the study in this paper, deployed following an RF planning exercise. Plans are in place to procure and deploy Citizen Broadband Radio Service (CBRS) RUs in an outdoor location to complete the 8-node deployment.

[1]X5G website: https://x5g.org.

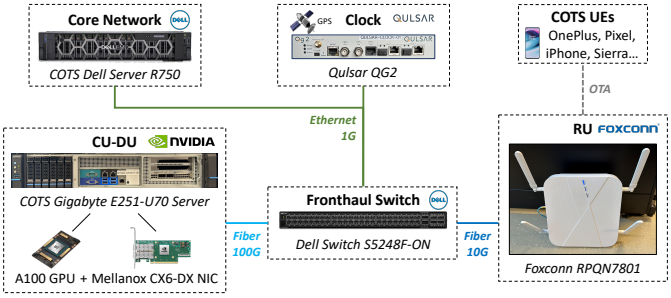


Fig. 2: Hardware architecture of the X5G deployment.

UE. Finally, we use COTS 5G UEs from OnePlus (AC Nord 2003) and Sierra Wireless (EM9191) to connect OTA to X5G.

IV. Experimental Analysis

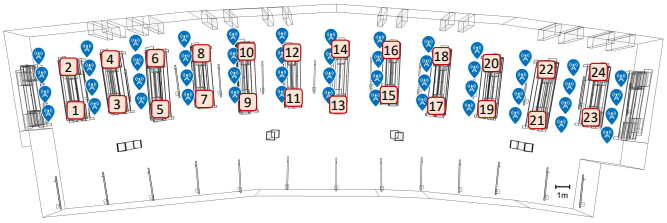
We present (i) an RF planning exercise to identify suitable locations for the RUs deployment, through a ray-tracing-based digital twin framework of our indoor laboratory space; and (ii) a set of experiments that profile the performance of the system with varying channel conditions and different number of UEs.

A. RF Planning with Ray-tracing

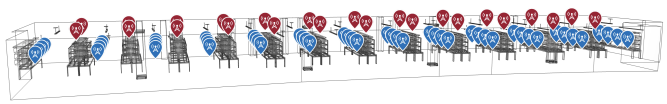
The purpose of the RF planning study is to determine an optimal location for the RUs within our indoor laboratory space in the Northeastern University ISEC building in Boston, MA. We leverage ray-tracing in a detailed digital twin representation of the laboratory space, to achieve high fidelity between the real-world environment and the simulated one. We analyze how to deploy 2 RUs by considering the Signal to Interference plus Noise Ratio (SINR) between the RUs and the UEs as the objective function in the optimization problem. We restrict the optimization space by using a grid of 24 possible RU locations and 52 UE test points and optimize through exhaustive search.

First, we leverage our digital twin framework, developed in [17], to create a 3D representation of our laboratory environment through the Sketchup modeling software. We then import the model in the MATLAB ray-tracer and define the locations of RUs and UEs as shown in Figure 3a (from a top perspective) and in Figure 3b (from a side view). The 24 possible RUs locations (2 for each bench) are shown in red, and the 52 test points for the UEs are in blue. They are arranged in a 4×13 grid positioned in a semi-arc of a circle. Tables I and II summarize the parameters for our ray-tracing model. In this context, we consider the RUs as transmitter nodes (TX) transmitting to all the UEs, which are considered only as receiver nodes (RX), i.e., we focus on optimizing the downlink rather than the uplink.

The ray-tracer generates a 24×52 matrix $\mathbf{C}$ where each entry $c_{i,j}$ corresponds to the channel information between RU_i and UE_j. From these results, we derive relevant parameters such



(a) Site viewer top view.



(b) Site viewer side view.

Fig. 3: Site viewer with RU (red icons) and UE (blue icons) locations.

TABLE II: Parameters of the Matlab ray-tracing study to locate RU locations.

Parameter	Value
RU antenna spacing	0.25 m
RU antenna TX power (P_{RU})	24 dBm
RU antenna gain (G_{RU})	5 dBi
RU antenna pattern	Isotropic
RU TX attenuation (A_{RU})	$[0-50]$ dB
Number of RU locations	24 in a 2×12 grid
RU height	2.2 m
UE number of antennas	2
UE antenna spacing	0.07 m
UE antenna gain (G_{UE})	1.1 dBi
UE noise figure (F_{UE})	5 dB
Number of UEs	52 in a 4×13 grid
UE height	0.8 m
Environment material	Wood
Max number of reflections	3
Max diffraction order	1
Ray-tracing method	Shooting and bouncing rays

as the thermal noise (N) and the path loss (PL) to compute the Received Signal Strength Indicator (RSSI) $R_{i,j}$ for UE_j connected to RU_i, as follows:

$$R_{i,j} = P_{RU,i} + G_{RU,i} - A_{RU,i} - PL_{i,j} + G_{UE,i}. \quad (1)$$

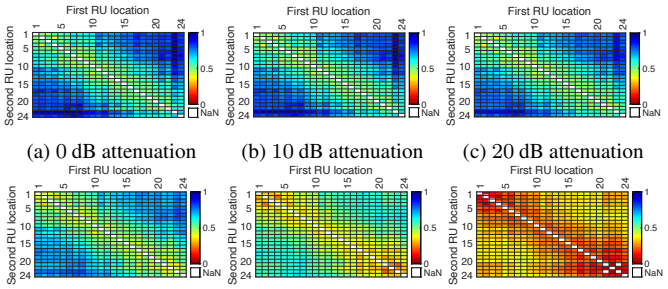
Then, considering the linear representation of $\hat{R}_{i,j}$, the SINR $\Gamma_{i,j}$ can be represented as

$$\Gamma_{i,j} = \frac{\hat{R}_{i,j}}{NF_{UE,i} + \sum_{u=1,u\neq i}^{M} \hat{R}_{u,j}}. \quad (2)$$

Here, M is the number of RUs being deployed (e.g., $M = 2$ in the rest of the paper). The SINR $\Gamma_{i,j}$ considers the interference to the signal from RU_i to UE_j due to downlink transmissions of all other $M-1$ RUs being deployed.

In this initial study, our objective is to place 2 RUs, resulting in $M = 2$, i.e., one transmits and the other creates interference. We test all possible pairs of RUs locations, for a total of 276 combinations. For each combination, we assign each UE to the RU with the best SINR value, generating a matrix $\mathbf{\Gamma}_{\max}$ of size 276×52. Finally, we calculate the average SINR $\mathbb{E}(\Gamma)$ value for all UEs in each possible deployment, a vector with 276 average SINR values, which is used as a *score* to evaluate suitable pair of locations for the RUs.

We test this algorithm with different values of the attenuation A_{RU}, from 0 to 50 dB in 10 dB increments. Figure 4 visualizes the normalized values for $\mathbb{E}(\Gamma)$ for all the pairs of possible RU locations, for the different attenuation values. Additionally, Table III provides the best RU locations including the minimum and maximum values of the corresponding combinations. As expected, RUs that are more distant exhibit high average SINR values, as they are less affected by interference. However, it is important to note that the score also considers coverage. Consequently, the optimal combination of locations identifies RUs which are further apart but not necessarily the most distant pair. Considering these results, for the experiments in Section IV-B, we selected a TX attenuation of 20 dB which exhibits a good trade-off between coverage and the range of the average SINR values. Moreover, during real-world testing, we observed that a 20 dB attenuation leads to increased system stability and reduced degradation compared to lower attenuation values, resulting in improved overall performance, as it reduces the likelihood of saturation at the UE antennas. Therefore, the current locations of the RUs in our laboratory space are [6,23].



(a) 0 dB attenuation (b) 10 dB attenuation (c) 20 dB attenuation

(d) 30 dB attenuation (e) 40 dB attenuation (f) 50 dB attenuation

Fig. 4: Heatmap results of the normalized average SINR $\mathbb{E}(\Gamma)$ with 2 RUs.

TABLE III: Best RUs and $\mathbb{E}(\Gamma)$ range values.

A_{RU} [dB]	RUs with best SINR	[Min, Max] $\mathbb{E}(\Gamma)$ [dB]
0	[8, 23]	[6.08, 23.33]
10	[6, 23]	[5.71, 22.66]
20	[6, 23]	[5.00, 21.03]
30	[8, 23]	[3.58, 17.82]
40	[7, 24]	[0.19, 12.94]
50	[8, 20]	[-6.23, 6.63]

B. Experiment Results

To evaluate X5G, we initially assess its stability through stress tests. We find that each gNB with no UE attached has an uptime of multiple days and that continuous downlink and uplink data exchange can be sustained for at least 8 hours with one or two UEs. Finally, we successfully connect and exchange traffic with 8 UEs simultaneously to a single gNB, using a mix of different UEs, including smartphones and 5G modem boards.

Next, to evaluate X5G KPIs, we perform two different types of experiments: (i) throughput test with iPerf; and (ii) MPEG-DASH video streaming. We configure an edge server within the campus network with minimal latency ($1-2$ ms). For the throughput test, we transmit Transmission Control Protocol (TCP) traffic both in downlink (DL) and uplink (UL) directions

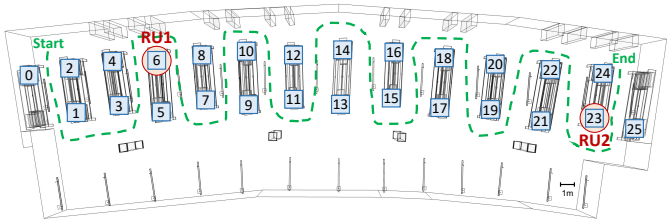
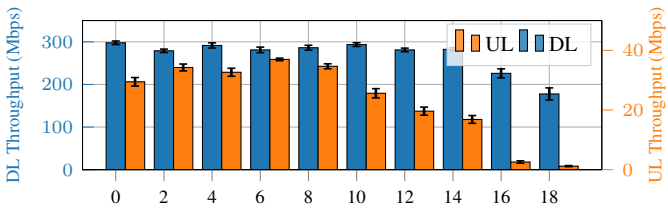
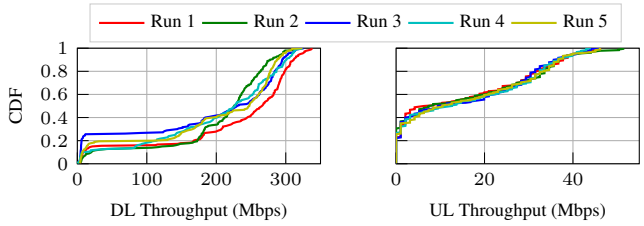


Fig. 5: Node locations considered in our experiments: RUs (red circles in 6 and 23); possible static UEs (blue squares); and mobile UEs (green dashed line).



(a) Static



(b) Mobility

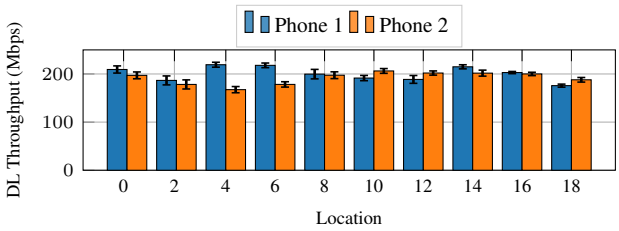
Fig. 6: Performance profiling with one UE and RU, for static and mobile cases.

for 40 seconds with varying UE numbers. For video streaming, we configure the server with ffmpeg [18] to advertise 5 profiles together, with the resolutions of 1080P (250 Mbps, 100 Mbps), 720P (50 Mbps, 25 Mbps), and 540P (10 Mbps) to the UE. We use OnePlus Nord Phone [19] as UE, iperf3 for generating backlogged traffic, and Google's ExoPlayer [20] as the video client application. For all 1 RU tests, we use the RU at location 6 depicted in Figure 5. For each set of experiments, we do 5 runs and plot the mean and 95% confidence interval of the metrics.

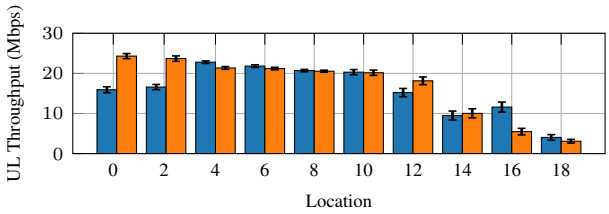
1 UE, static, iPerf. In the first set of experiments, we profile the performance of 1 UE per 1 RU. We select 10 static locations (0,2,4,...,18) around the lab, starting from near the RU to far away, as depicted in Figure 5. As shown in Figure 6a, the UE is able to achieve an average throughput of 300 Mbps in DL and 38 Mbps in UL in the best possible locations. As expected, the average throughput decreases as we move further away from the RU. At location 18, which is the furthest from the RU, we observe an average throughput of approximately 177 Mbps in the downlink direction and 1.5 Mbps in the uplink direction.

1 UE, mobile, iPerf. Next, we measure throughput for 3 minutes while walking in a zig-zag pattern around the lab, as indicated by the dashed green line in Figure 5. The results are plotted in Figure 6b. It is noticeable that even during mobility, the UE is able to achieve 300 Mbps in DL and 40 Mbps in UL. Additionally, we observe that when the UE gradually moves outside the range of the RU towards the end of the trace, the connection drops, causing 20% of throughput results to be 0.

2 UEs, static, iPerf. We test the performance of 2 UEs for



(a) Downlink



(b) Uplink

Fig. 7: Performance profiling for a single RU and two UEs.

a single RU. We position the UEs at the same static locations as in the previous 1 UE static case. The results are plotted in Figure 7. We observe that, in most cases, the UEs are able to share bandwidth fairly. The best achievable mean aggregate throughput from both UEs is around 400 Mbps in DL and 44 Mbps in UL. This shows that the total cell throughput can be higher than the single UE throughput. As discussed in Table I, this is due to a limitation in the number of transport blocks that can be acked in a single slot for a *single* UE, a condition that will be relaxed in our future work. Therefore, scheduling multiple UEs improves the resource utilization of the system.

1 UE, static-mobile, video streaming. We place the UE at three static locations at different distances from the RU: location 8 (close); 12 (mid); and 16 (far). We run each video session for 3 minutes and plot the mean bitrate over 5 runs, as well as the rebuffer ratio. As expected, the average bitrate decreases and the rebuffer ratio increases as the UE moves farther from the RU. We also observe that the UE can achieve a 1080p resolution and a steady mean bitrate of around 180 Mbps in all the static cases. Note that, unlike test results achieved through backlogged traffic, the mean bitrate for video streaming is found to be lower. The video client intermittently fetches segments (causing flows to be short), which depends on parameters, e.g., video buffer and segment size. Thus, the throughput sometimes does not ramp up to the fullest during that short period and the client ABR algorithm downgrades the bitrate based on the estimate it gets. This is due to a slow Modulation and Coding Scheme (MCS) selection loop in the OAI L2, which will be improved as part of our future work. Nonetheless, this shows that our setup can support up to 8K HDR videos that require $150 - 300$ Mbps bitrates according to YouTube guidelines [21]. During mobility, the average bitrate is found to be 120 Mbps, and the rebuffer ratio increases to 15%. This is once again because the UE moves away from the RU, gradually entering low-coverage regions and eventually disconnecting.

1-4 UEs, static, iPerf. We also extend our evaluation to multiple UE scenarios. At a fixed location (4), we compare the performance with varying numbers of UEs (1 to 4) connected to our network. The mean throughput and 95% confidence

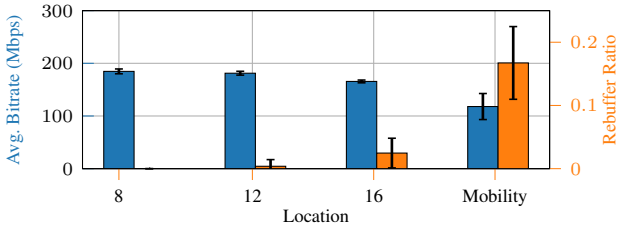


Fig. 8: Video Streaming Performance

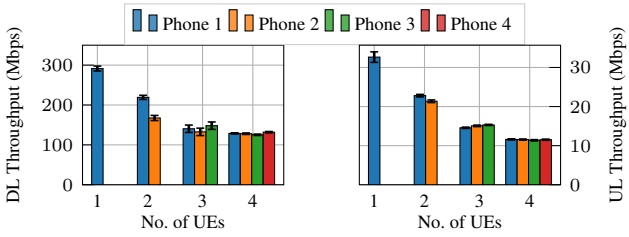


Fig. 9: Performance profiling with multiple UEs.

intervals are plotted in Figure 9. We observe that the UEs are able to achieve steady throughput in all the cases. The combined throughput also increases with the increasing number of UEs connected. With four UEs the aggregate throughput is found to be 512 Mbps in DL and 46 Mbps in UL.

2 RUs, 1 UE per RU, static, iPerf. Finally, we evaluate performance with 2 RUs by connecting phone 1 to $RU1$ and phone 2 to $RU2$. We select three pairs of locations (0,25), (2,23), (4,21) for the UEs so that they are far from each other. The overall average results are listed in Table IV. We observe that the impact of interference is dominant in the DL direction with each UE experiencing up to 40% reduction in throughput whereas UL performance remains stable, despite a high SINR experienced by the UEs (average of 22.62 dB, reported by OAI), also aligned with the value predicted by the ray-tracer (24.12 dB). We plan to analyze the impact of interference on the protocol stack in detail as part of our future work.

TABLE IV: Performance profiling with 2 RUs and 1 UE per RU.

UE	DL Throughput [Mbps]	UL Throughput [Mbps]
Phone 1	179.32 ± 27.29	35.34 ± 5.26
Phone 2	174.09 ± 27.42	33.97 ± 2.23

V. Conclusions and Future Works

We introduced X5G, an open, programmable, and multi-vendor private 5G O-RAN testbed deployed at Northeastern University in Boston, MA. We demonstrated the first integration of NVIDIA Aerial, a PHY layer implementation on GPUs, with higher layers based on OAI, resulting in the creation of the NVIDIA ARC platform. We provided an overview of the ARC software and hardware implementations, designed for an 8-node deployment. Additionally, we conducted a ray-tracing study using our digital twin framework to determine the optimal placement of X5G RUs, and we discussed platform performance with varying numbers of COTS UEs and various application tests, such as iPerf and video streaming.

Next, we plan to complete the deployment of the 8 X5G gNBs comprising a mix of indoor and outdoor locations for more realistic experiments and a comprehensive development of UE handover procedures. We will also target the integration of RUs supporting bands for 5G New Radio (NR) Frequency Range 2 (FR2). We will develop pipelines for automatic deployment and management of workloads via container platforms like Kubernetes and Red Hat OpenShift. Furthermore, we will perform the integration with the O-RAN Software Community (OSC) RAN Intelligent Controller (RIC). Our aim is to enable full control of the RAN and to ease dynamic changes of network behavior by enhancing the capabilities of X5G for the research community of next-generation wireless networks.

References

[1] A. Narayanan, M. I. Rochman, A. Hassan, B. S. Firmansyah, V. Sathya, M. Ghosh, F. Qian, and Z.-L. Zhang, "A Comparative Measurement Study of Commercial 5G mmWave Deployments," in *IEEE Conference on Computer Communications*, May 2022.

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, Second quarter 2023.

[3] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "NeutRAN: An Open RAN Neutral Host Architecture for Zero-Touch RAN and Spectrum Sharing," *IEEE Transactions on Mobile Computing*, 2023.

[4] S. Pongratz, "Open RAN Market Opportunity and Risks," Dell'Oro Group report, November 2021.

[5] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "OpenAirInterface: Democratizing innovation in the 5G Era," *Computer Networks*, vol. 176, p. 107284, 2020.

[6] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "SrsLTE: An Open-Source Platform for LTE Evolution and Experimentation," in *Proceedings of ACM WiNTECH*, NYC, New York, October 2016.

[7] Meticolous Research, "5G Testing Market." [Online]. Available: https://www.meticulousresearch.com/product/5g-testing-market-5482

[8] P. Bahl, M. Balkwill, X. Foukas, A. Kalia, D. Kim, M. Kotaru, Z. Lai, S. Mehrotra, B. Radunovic, S. Saroiu, C. Settle, A. Verma, A. Wolman, F. Y. Yan, and Y. Zhang, "Accelerating Open RAN Research Through an Enterprise-Scale 5G Testbed," in *Proceedings of ACM MobiCom '23*, Madrid, Spain, October 2023.

[9] B. Tang, V. K. Shah, V. Marojevic, and J. H. Reed, "AI Testing Framework for Next-G O-RAN Networks: Requirements, Design, and Research Opportunities," *IEEE Wireless Communications*, vol. 30, no. 1, pp. 70–77, 2023.

[10] Small Cell Forum, "5G FAPI: PHY API Specification," techreport 222.10.04, November 2021.

[11] A. Kelkar and C. Dick, "NVIDIA Aerial GPU Hosted AI-on-5G," in *IEEE 4th 5G World Forum (5GWF)*, October 2021, pp. 64–69.

[12] O-RAN Alliance, "O-RAN working group 4 (open fronthaul interfaces wg) control, user and synchronization plane specification," techreport O-RAN.WG4.CUS.0-R003-v12.00, June 2023.

[13] O-RAN Working Group 4, "O-RAN Fronthaul Control, User and Synchronization Plane Specification 7.0," ORAN-WG4.CUS.0-v07.00 Technical Specification, July 2021.

[14] OpenAirInterface Software Alliance. [Online]. Available: https://openairinterface.org

[15] NVIDIA Corporation, "NVIDIA Aerial SDK." [Online]. Available: https://developer.nvidia.com/aerial-sdk

[16] FCC: Engineering & Technology Bureau, "FCC designates new innovation zones for advanced wireless technology research and innovation," Public Notice: FCC-21-92, August 2021.

[17] D. Villa, M. Tehrani-Moayyed, C. P. Robinson, L. Bonati, P. Johari, M. Polese, and T. Melodia, "Colosseum as a Digital Twin: Bridging Real-World Experimentation and Wireless Network Emulation," *IEEE Transactions on Mobile Computing*, pp. 1–17, In press 2024.

[18] *FFMPEG*. [Online]. Available: https://ffmpeg.org/ffmpeg.html

[19] *OnePlus*. [Online]. Available: https://www.oneplus.com/us/nord-specs

[20] *ExoPlayer Dash Formats*. [Online]. Available: https://exoplayer.dev/dash.html

[21] *Guidelines for Video Bitrates*. [Online]. Available: https://support.google.com/youtube/answer/1722171?hl=en#zippy=%2Cbitrate