Evaluating Open-Source 5G SA Testbeds: Unveiling Performance Disparities in RAN Scenarios

Mohamed Rouili*, Niloy Saha*, Morteza Golkarifard*, Mohammad Zangooei*,
Raouf Boutaba*,Ertan Onur†,Aladdin Saleh‡
{mrouili, n6saha, mgolkari, mzangooei, rboutaba}@uwaterloo.ca
eonur@metu.edu.tr, aladdin.saleh@rci.rogers.com
*University of Waterloo, Canada, †Middle East Technical University, Turkey, ‡Rogers Communications, Canada Inc.

Abstract—**Fifth generation (5G) standalone (SA) mobile networks are rapidly gaining prominence worldwide, and becoming increasingly prevalent as the telecommunication industry standard. Most published work concerning 5G applications relies on open-source 5G radio access network (RAN) simulation and emulation tools to evaluate various concepts, algorithms, and use cases. However, these tools are not always accurate in conveying a realistic representation of real-world RAN performance and expected quality of service (QoS). This paper discusses the deployment of a 5G SA testbed supporting three different RAN scenarios of real and simulated deployments using open-source software, commercial-off-the-shelf (COTS) hardware, and software defined radios (SDRs). We experimentally evaluate the performance of these scenarios for the RAN and quantify their differences in terms of computational resource utilization, throughput, latency, coverage, and power consumption. Specifically, we explore the emulation and simulation tools' ability to reflect realistic RAN performance and highlight the differences compared to the SDR-based deployment. Through this analysis, this paper provides insights into the performance of each approach and sheds light on the feasibility of using open-source software for 5G testing and experimentation.**

Index Terms—**5G, Testbed, Radio access networks, OpenAirInterface, UERANSIM**

I. INTRODUCTION

The development of fifth generation (5G) cellular networks aims to tackle the increasing demand for high-speed data transmission, minimal network latency, and seamless connectivity across a wide range of devices. These advancements are crucial to support emerging service categories standardized by the third-generation partnership project (3GPP), including ultra-reliable low latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC). Traditional QoS evaluation platforms for these applications such as proprietary and commercial testbeds are limited in scope and accessibility, most often only available to mobile network providers (MNOs) and partnered researchers. These limitations sparked a growing interest in open-source solutions within the research community as they provide an open, free, and cost-effective alternative for conducting 5G research and experimentation.

Available open-source RAN simulation tools such as UERANSIM [1] and RFsimulator [2] offer the ability to evaluate various 5G SA applications, algorithms, and use cases. However, despite their widespread use within the community, there is a lack of publicly available results and studies discussing their ability in providing a realistic representation of QoS performance compared to 5G testbeds including a complete implementation of 5G SA RAN protocol stack and radio units (RUs). To bridge this gap, we conducted a comprehensive study by implementing a practical 5G standalone (SA) deployment using OpenAirInterface (OAI) [3] and software defined radios (SDRs). This deployment was then compared against widely used emulation-based scenarios (OAI RFSimulator) and simulation-based scenarios (UERANSIM).

The choice of these scenarios comes from their different RAN protocol stack implementations. To the best of our knowledge, this work constitutes the first effort towards conducting an empirical study on available open-source solutions with a focus on three different RAN deployment approaches, and quantifying their differences in terms of different QoS requirements. The contributions of this study are as follows:

- We deploy a 5G SA testbed with three distinct RAN scenarios using open-source software and COTS hardware.
- We provide an empirical performance analysis of the three different scenarios in terms of different 5G SA QoS requirements namely, throughput, latency, computational resource usage, coverage, and power consumption.
- We provide key insights regarding the adoption of simulation tools to validate different 5G use cases.
- We offer a supporting Github repository [4] containing our experience installing, configuring, and debugging the deployed testbed, which can serve as a valuable resource for researchers seeking to deploy similar open-source solutions and allow further analysis and investigation.

II. BACKGROUND AND RELATED WORK

This section presents available open-source 5G solutions and applications in the literature.

Open source projects: These 5G network solutions for core and RAN components provide a cost-effective method to set up experimental testbeds for academic and industrial research. The open-source code can be adapted for various use cases and deployments, allowing the integration of new features. Researchers can replicate and enhance outcomes from earlier studies using these projects. Several open-source initiatives have played a significant role in advancing the core and RAN components. On one hand, notable contributions for

Core	SMF	AMF	PCF	UDM	NEF	UPF
Open5GS [5]	✓	✓	✓	✓	✗	✓
Free5GC [6]	✓	✓	✓	✓	✗	✓
OAI5G-CN [7]	✓	✓	✗	✓	✓	✓

RAN	PHY	MAC	RLC	PDCP	SDAP	RRC
UERANSIM	✗	✗	✗	✗	✗	✓
srsRAN [8]	✓	✓	✓	✓	✓	✓
OAI5G	✓	✓	✓	✓	✓	✓
RFSimulator	✓	✓	✓	✓	✓	✓

TABLE I: Comparison of open-source 5G implementations for the RAN and core

the core include projects such as OAI5G-CN, Open5GS, and FREE5GC. which provide full-stack implementations of the 5G core architecture with diverse functionalities and system compatibility. In our testbed, we leverage OAI's 5G-CN implementation as it offers seamless compatibility with the RAN implementations that we use for the experiments. On another hand, projects such as OAI 5G RAN, and SoftwareRadioSystemsRAN (srsRAN) provide 3GPP-compliant implementations of 5G SA and NSA RAN while UERANSIM offers a software simulation of the 5G SA gNB and UE functions.

Large-scale 5G testbeds: Large-scale measurement studies investigated 5G performance across large geographic areas [9] and specific terrains like industrial campuses [10]. These studies, crucial for understanding 5G's potential in various applications, have focused on network coverage, throughput, delay, energy consumption, and their effects on application QoS. However, their reliance on proprietary, commercial networks limits reproducibility for researchers without such access. Open-access testbeds like Colosseum [11] and POWDER [12], leveraging open-source OAI and srsRAN software and COTS SDRs, emerged as valuable options for validating 5G applications. Located in the US, these testbeds necessitate remote access for most users, presenting challenges. A significant limitation is the requirement for users to either ship their specific UEs or personally visit the sites for installation and testing, especially when these UEs differ from the standard SDR-based ones used in these testbeds.

Simulation vs real deployments: Table I outlines the network functions (NFs) of open-source core and RAN projects, with UERANSIM being widely used for evaluating 5G applications in areas like security, end-to-end slicing, and monitoring [13], [14], [15]. However, it lacks support for layer one and two protocols. In contrast, RFsimulator, OAI's gNB implementation with a simulated radio stack, includes layer two and three protocols and offers the potential for deploying and evaluating new radio functionalities in 5G networks. Unlike UERANSIM which does not allow fine-grain control over the radio stack due to its IP packet-based emulation, RFsimulator allows fine-grain control of the radio stack. For instance, He et al. [16] conducted fuzzing tests on the 5G NAS protocol leveraging RFsimulator. However, assessing various QoS requirements with both UERANSIM and RFsimulator may yield results that differ from commercial networks and open-source implementations like OAI with SDR devices and COTS UEs. To understand the limitations and capabilities of these tools, we study them with OAI's 5G SA gNB using SDR devices. We evaluate various QoS requirements and compare the results where applicable, providing essential insights for researchers and practitioners in selecting appropriate tools for their specific needs.

Performance profiling studies: To validate the accuracy of the RAN simulation tool SimuLTE, Wischhof et al. [17] deployed an experimental 4G testbed using OAI. They measured packet delay and inter-arrival time in various load scenarios, comparing results with SimuLTE simulations. The study showed SimuLTE's limitations in high-load scenarios and its inability to accurately model an LTE system-level network. While emphasizing the importance of experimental validation, the study is restricted to 4G networks and does not explore 5G NR deployments. Cuidi et al. [18] implemented a 5G SA testbed using OAI to analyze RAN component costs and the impact of different gNB configurations in SDR and emulated radio scenarios. Their study highlights protocol layer overheads and OAI deployment architectures but lacks details on the emulated radio software and its comparison to the B210 USRP in their SDR testbed. In another study, Sahbafard et al. [19] established a small-scale 5G SA testbed with B210 and N310 SDRs, assessing network performance within an office environment. They noted inadequate latency for URLLC applications and reduced uplink speeds, attributing these findings to OAI's limitations. However, they didn't explore CPU and memory profiling, which could elucidate differences between the two SDRs.

III. TESTBED SETUP AND CONFIGURATION

We deployed a lab-scale end-to-end 5G SA testbed as shown in Figure 1. The testbed comprises three Intel i9 Dell PCs running Ubuntu 22.04 (low-latency kernel), which are used to host the OAI 5G core and three different RAN deployment scenarios, as shown in Figure 2. Table II provides a summary the testbed components. We consider three different RAN deployment scenarios as follows:

SDR-testbed: The first scenario is an end-to-end deployment involving an OAI 5G SA gNB (monolithic), a SDR (USRP X310), and a Google Pixel 7 Pro UE.

RFsimulator: In the second scenario, we use OAI's RFsimulator, which emulates the radio interface between the UE and the gNB. This deployment involves an OAI 5G SA gNB (monolithic) in RFsimulator mode and a simulated OAI UE.

UERANSIM: The third scenario uses the UERANSIM tool to simulate both the RAN and the UE, providing a completely virtualized 5G SA network environment.

IV. EXPERIMENTAL EVALUATION

For each of the three testbed deployments, we conduct the following key performance profiling experiments:

Throughput Analysis: We calculate the maximum theoretical throughput for the testbed based on 3GPP standards and compare it with effective downlink throughput using `Sockperf` [20].

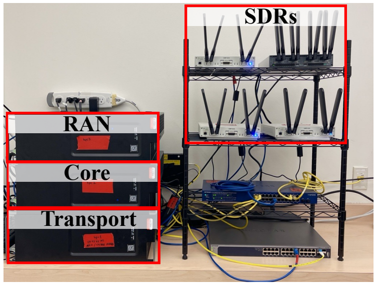


Fig. 1: Testbed overview

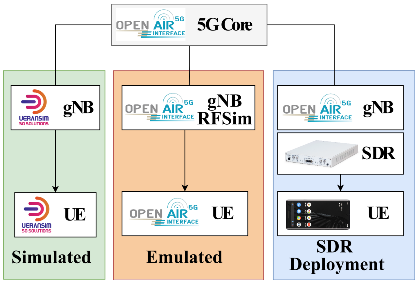


Fig. 2: RAN scenarios

Component	**Specification**
5GC	CPU i9-10980XE, RAM 32GB, OAI 5G-CN
gNB	CPU i9-10980XE, RAM 32GB, OAI 5G SA gNB (40 MHz, 106 PRBs, Band n78 TDD)
RF	Ettus USRP X310 UHD v4.3
UE	Google Pixel 7 Pro, Android 13
SIM	Sysmocomm sysmoISIM-SJA2

TABLE II: Testbed Setup

Computational Profiling: We isolate the gNB process and measure the CPU and memory utilization using the `ps` utility [21] under various bandwidth configurations.

Latency Analysis: We use `Sockperf` to measure the average round trip time (RTT) between the UE and the 5G core. We consider both idle and under-load traffic scenarios.

Power Consumption: We measure the CPU power consumption and the overhead induced by the different system states using `Powerstat` [22].

Network Coverage Analysis: We measure the reference signal received power (RSRP), reference signal received quality (RSRQ), and signal to interference plus noise ratio (SINR) at varying locations from the RU using the `Network cell info lite` tool [23] on our COTS UE.

To ensure statistical significance, each experiment is run 25 times per scenario, and average values are considered.

A. Throughput Analysis

We evaluate the three scenarios in terms of downlink (DL) throughput compared to the 5G NR standards set by 3GPP. First, we calculate the maximum theoretical data transfer rate (DR) for the testbed's RAN configuration based on the 3GPP TS 38.306 standard [24], the formula is given as follows:

$$\mathrm{DR} = v_{\mathrm{L}} Q_m f R_{\max} \frac{N_{\mathrm{PRB}}^{\mathrm{BW},\mu} 12}{T_s^{\mu}} (1 - \mathrm{OH}) \tag{1}$$

where $R_{\max} = 948/1024$ is the max low-density parity check (LDPC) code rate, v_{L} is the maximum number of supported MIMO transmission layers, Q_m is the maximum supported modulation order, f is the scaling factor, $N_{\mathrm{PRB}}^{\mathrm{BW},\mu}$ is the maximum possible resource block (RB) allocation in bandwidth BW, μ is the numerology, T_s^{μ} is the orthogonal frequency division multiplexing (OFDM) symbol duration, and OH is the overhead for frequency range 1 (FR1).

Given the TDD configuration for the SDR-testbed and RFsimulator scenarios, we determine the maximum theoretical throughput to be 122 Mbps. We normalize our measurement results with respect to this calculated theoretical throughput value in the subsequent bar chart analysis. Next, we measure the throughput in the three scenarios. Figure 3 illustrates the experimental to the theoretical throughput ratio of three deployment scenarios. Using an SDR device and COTS UE, we achieve 95% of the maximum theoretical throughput. During data transmission, a guard period overhead is added to facilitate the transition between Tx and Rx. Additionally, there are symbols dedicated to uplink control channels, such as the Sounding Reference Signals (SRSs). This causes the throughput to be a little less than the estimated theoretical value. In comparison, the current RFsimulator implementation achieves only around 60% of the maximum theoretical throughput which is significantly lower than the SDR-Testbed due to the radio emulation using a considerable amount of bandwidth resources (*cf.*, Section IV-B).

UERANSIM's implementation does not include a PHY layer. Instead, it employs the radio link simulation (RLS) protocol to emulate the radio interface, substituting the conventional radio transmissions within the RAN with a UDP-based link connecting the gNB and UE. Consequently, RB allocation is not possible leading UERANSIM to saturate the entirety of the available 1 Gbps link, achieving an unrealistically high throughput that does not account for the limitations imposed by the PHY layer observed in the other scenarios.

B. Computational Profiling

We analyze the computational resource consumption of the RAN by varying the downlink traffic rate from 20 to 140 Mbps for a single UE and plotting the CPU and memory usage of the gNB process.

Memory: Figure 4 demonstrates low and stable memory usage across all three scenarios. The SDR-testbed exhibits the highest average memory consumption of approximately 2 GB followed by RFsimulator at 1.6 GB, while UERANSIM's memory usage is negligible. Notably, memory usage remains consistently low regardless of the traffic rate.

CPU: Figure 5 shows that UERANSIM peaks at about 30% consumption of a CPU core which is significantly less computationally expensive when compared to the SDR-testbed and RFsimulator. The observed discrepancy can be attributed to the substantial CPU demand of the PHY layer compared to the upper layers of the RAN, as confirmed by a relevant study [18]. This disparity arises from the PHY layer's essential CPU-intensive functions, including processing incoming baseband signals, data preparation for upper layers, and UE transmission. Moreover, RFsimulator exhibits the highest CPU consumption, consistently maintaining a usage of 140% irrespective of variations in the downlink traffic rate, whereas

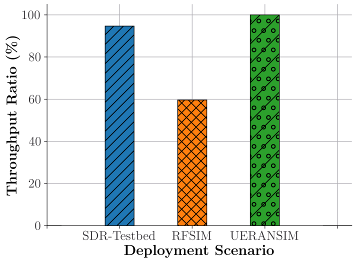


Fig. 3: Effective throughput under different deployments

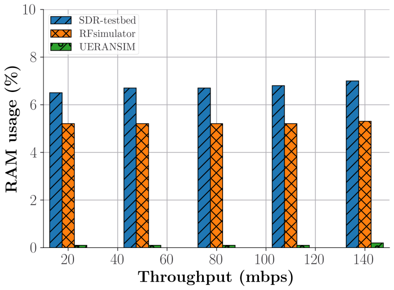


Fig. 4: RAM utilization under different deployments

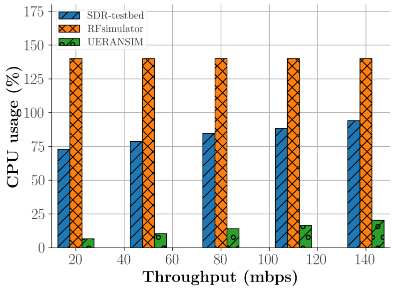


Fig. 5: CPU utilization under different deployments

the SDR-testbed reaches a peak usage of approximately 94%. The `ps` utility reports the percentage of CPU usage per core. The results observed in RFsimulator signify that the process is multi-threaded, i.e., the load is occupying more than one full CPU core and 40% of another. RFsimulator substitutes the conventional low-level driver responsible for transmitting I/Q samples to an SDR device, which modulates the baseband signal based on the provided I/Q data to the desired frequency as in the SDR-testbed. In the emulation, the signal exchange occurs directly between the gNB and UE, thereby resulting in a higher CPU utilization.

C. Round trip time analysis

In this experiment, we quantify the average round trip time (RTT) between the UE and the core for all three deployments. We consider two network scenarios: idle and under-load provided by the `Sockperf` tool which gradually congests the network with heavy traffic. This evaluation enables us to examine the influence of bandwidth utilization on latency.

Idle network scenario: In Figure 6 we plot the average RTT while adjusting the downlink traffic rate. The SDR-testbed achieves the highest RTT at approximately 15 ms, considerably worse compared to the 7 ms delay observed in RFsimulator and the 0.9 ms delay in UERANSIM. The observed variations in RTT stem from multiple factors, with the primary one being delays associated with the radio channel. These delays include frame alignment, TX preparation time, payload transmissions, and HARQ retransmissions, which depend on channel conditions, available resources, and transmission errors. Backhaul and core delays also contribute to the measured RTTs; however, as the same 5G core is utilized in all three deployments, these delays are comparable and have a minimal impact on the measured latencies.

These findings highlight the latency overheads introduced by the radio channel in the SDR-testbed and radio channel emulation in RFsimulator. To enhance channel conditions, we carefully positioned the user equipment (UE) in close proximity to the RAN, used offset tunning, and fine-tuned relevant radio parameters. Despite our diligent efforts, sub-optimal channel conditions persist due to inherent limitations in the OAI implementation and reported noise issues, particularly in the uplink (UL) of the X310 SDR device. The presence of interference from neighboring wireless devices and surrounding networks further exacerbates signal degradation.

Under-load scenario: The under-load scenario using `Sockperf` generates heavy traffic and sends it through the network for a fixed time using a fixed frequency of messages per second (MPS). Figure 7 shows the obtained results, the average RTT for the SDR-testbed is around 220 ms while RFsimulator achieves 250 ms. Similar to the throughput measurements, UERANSIM depicts no considerable change from the Idle scenario given the lack of PHY layer processing which leads to maintaining very low latency.

The high latency observed in the SDR-testbed and RFsimulator primarily stems from network congestion as a result of traffic flow exceeding the available bandwidth. As previously mentioned, RFsimulator uses a considerable portion of the available bandwidth to transfer the I/Q data, leaving limited bandwidth for effective data transmission.

D. Power consumption analysis

We analyze the power consumption of the CPU to understand the overheads introduced by the deployment of different RAN components, the measurements were taken over four distinct system states: (1) idle, (2) gNB deployed and running, (3) gNB and UE deployed and connected, and (4) gNB and UE connected with generated traffic flow. The results obtained for each scenario in terms of CPU power consumption are plotted in Figure 8. These trends provide valuable insights into the power draw of the different RAN scenarios.

Figure 8 illustrates the obtained results. In the idle state, where no components are deployed, the power usage remains consistent across all three scenarios, indicating negligible system-specific overhead. However, upon deploying the gNB, we notice an increase for all scenarios, with UERANSIM exhibiting a minimal change while the SDR-testbed and RFsimulator experience a noticeable rise. Upon deploying the UE and establishing a connection, RFsimulator exhibites a large increase. This highlights the overhead associated with simulating the RF environment and PHY layer processing. Next, we generate traffic between the UE and gNB which results in the largest increase for all scenarios.

In comparison to the idle system state, our findings suggest that UERANSIM and the SDR-testbed only exhibit significant

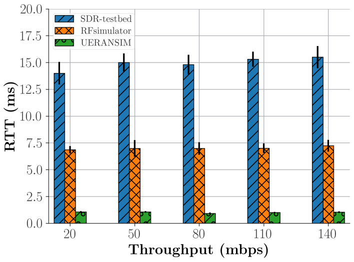


Fig. 6: RTT measurements in an idle network scenario

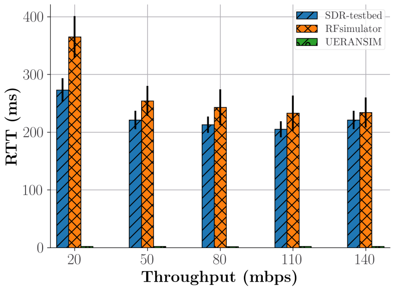


Fig. 7: RTT measurements in a congested network scenario

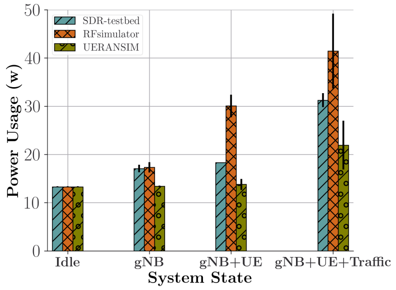


Fig. 8: CPU Power consumption under different system states

power usage during traffic processing. However, Rfsimulator experiences a notable increase upon establishing the gNB/UE connection, emphasizing the impact of emulating the RF environment and I/Q data transfer. Additionally, the SDR-testbed demonstrates more efficient power usage compared to RFsimulator suggesting that emulating RAN functions, whether singular or multiple, does not necessarily result in lower power consumption in contrast to a real implementation.

E. Signal strength evaluation

To discern the effects of radio channel conditions on network performance, specifically contrasting SDR-testbed and RFsimulator, we analyze the radio coverage and its impact on UE throughput. We consider multiple locations with different distances between the gNB and UE, under conditions of both clear Line-of-Sight (LoS) for optimal radio propagation and obstructed LoS, resulting in significant path loss attenuation.

We measure the RSRP, RSRQ, SINR, and the corresponding UE throughput at six locations within the lab area. Locations L0, L1, L2, and L3 have a clear Line of Sight (LoS). In contrast, L4 and L5 have a broken LoS. Figure 9a shows a significant variance of the median RSRP values, with the strongest signal at L0 (-65 dBm) and the weakest at L5 (-120 dBm). Further measurements were not possible due to -120 dBm representing the cell's edge where the connection is lost. RFsimulator's channel model emulates a constant RSRP of -45 dBm, which exceeds the RSRP value measured at L0 when the UE was placed directly in front of the SDR. This highlights the difference with the SDR-testbed. The presence of obstacles, such as a wall at L4 that breaks the LoS, impacts signal strength even at short distances.

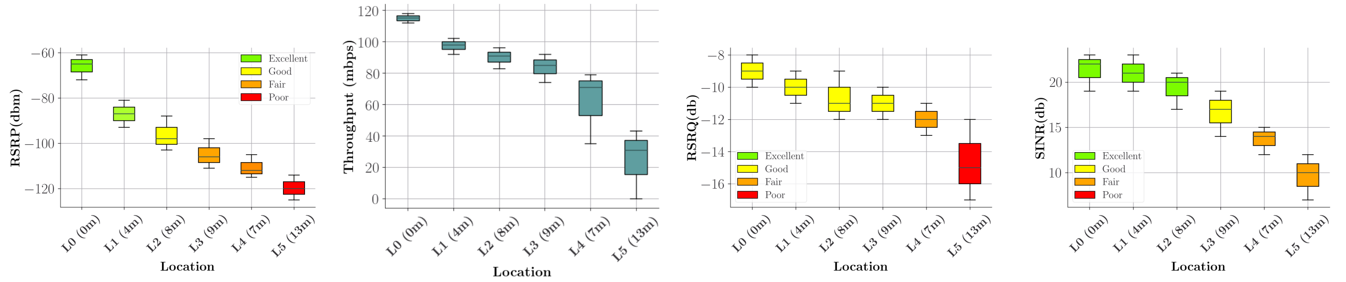
Figure 9c shows the median RSRQ values at different locations. The results indicate a presence of interference on the radio link, which contributes to sub-optimal channel conditions. Similarly, the median SINR values, depicted in Figure 9d, provide insight into the signal quality, considering both the desired signal strength and unwanted interference plus noise. The results indicate a relatively stable signal quality across the different locations, particularly at shorter distances. RFsimulator's implementation does not include native RSRQ and SINR monitoring. The measured RSRP, RSRQ, and SINR values explain the variance in the UE throughput as shown in Figure 9b. Along with the decline in the achieved throughput, we notice a large deviation from the median values in L4 and L5 when the LoS is broken compared to previous locations.

V. DISCUSSION

Findings and insights: RFsimulator's high CPU utilization and throughput limitations are due to handling a large volume of low-latency I/Q samples, utilizing a significant portion of the available bandwidth. On the other hand, UERANSIM's simulation achieves user plane data latency of less than 1 ms, suitable for testing URLLC applications. However, all three scenarios can support multiple eMBB applications with less stringent latency requirements. The SDR-based testbed needs further optimization in terms of latency to support 5G use cases such as URLLC. We found that memory is not a bottleneck in any of the examined scenarios, as modern hardware systems have sufficient capacity to support all deployment options. Additionally, there is a positive correlation between CPU usage and power consumption of the RAN functions. Therefore, optimizing CPU usage would lead to a decrease in RAN operation costs. Through the signal strength measurements, we found that the UE maintains consistent connectivity within the lab area as long as the LoS is maintained. However, the signal strength degradation experienced when increasing the distance from the SDR inevitably leads to considerable drops in the achieved throughput.

Applications and use cases: RFsimulator and UERANSIM offer cost-effective, easy-to-deploy, and stable alternatives to SDR-testbed deployments. However, their channel models, despite offering various options for wireless propagation simulation, may not accurately reflect the complexity of real-world radio environments. Simplifications and assumptions inherent in these models can lead to unrealistic outcomes.

For instance, [13] demonstrated the impact of DDoS attacks on network slices, recording a peak latency of 0.4ms ($\sim 0.8\text{ms}$ RTT), which is unrealistically low for a congested network, as our findings in Section IV-C suggest. Furthermore, the authors of [14] assessed E2E 5G slicing performance with UERANSIM, observing 0.1 ms idle and 48 ms under load latencies, and 57 mbps throughput in a scenario involving 11 UEs across 69 slices. Replicating these results in an SDR testbed, with a complete gNB protocol stack would be



(a) Distribution of RSRP over different locations

(b) UE Throughput variance over different locations

(c) Distribution of RSRQ over different locations

(d) Distribution of SINR over different locations

Fig. 9: Signal strength evaluation at 6 different locations

challenging due to the limited number of resource blocks and overheads introduced by layer 1 and layer 2 protocols. In a different approach, He et al. [16] chose RFsimulator over an n310 SDR to avoid stability issues attributed to the radio channel, which we encountered on the UL with the main OAI SDR-based release. This implies that while a more realistic performance is achieved through the utilization of a real testbed deployment, a trade-off prioritizing stability in certain 5G tests and use cases could be necessary.

Testbed limitations: Our SDR-based end-to-end testbed allows us to evaluate different QoS requirements and offers a more reliable and realistic implementation and depiction of gNB behavior when compared to RFsimulator and UERANSIM. However, it is important to note that the testbed is built on the OAI 5G SA project, which is in active development. During the testing, we encountered the following notable limitations — a) gNB instability and crashes under heavy uplink traffic, which was partially ameliorated by tuning the modulation and coding schemes, and b) RRC connection inconsistencies, requiring UE restart or SIM card re-insertion.

VI. CONCLUSION

This paper presents an empirical study of three open-source 5G SA network testbeds with varied RAN protocol stack architectures. We conducted performance profiling experiments, evaluating throughput, latency, CPU and memory usage, energy consumption, and radio coverage. The results demonstrate significant variations in these testbeds' abilities to replicate realistic network conditions and QoS.

While RFsimulator and UERANSIM are useful for 5G network simulation, they have limitations in radio channel and physical layer modeling, as well as in simulating UE and gNB behavior. These issues necessitate further development to enhance their accuracy and fidelity. Researchers are advised to be cautious in interpreting results from these simulations due to possible deviations from real-world scenarios.

Future work will focus on conceptualizing and validating various 5G use cases and applications across different testbeds, and comparing these open-source deployments with large-scale commercial 5G networks. This will allow for a more thorough understanding of how deployment environments affect application performance.

ACKNOWLEDGEMENTS

This work was supported in part by Rogers Communications Canada Inc. and in part by a Mitacs Accelerate Grant.

REFERENCES

[1] UERANSIM, https://github.com/aligungr/UERANSIM, 2023.
[2] RFSIMULATOR, https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/radio/rfsimulator/README.md, 2023.
[3] OAI5G, https://gitlab.eurecom.fr/oai/openairinterface5g, 2023.
[4] perfEval-5G, https://github.com/mrouili/perfEval-5G, 2023.
[5] Open5GS, https://github.com/open5gs/open5gs, 2023.
[6] Free5GC, https://github.com/free5gc/free5gc, 2023.
[7] OAI-CN5G, https://gitlab.eurecom.fr/oai/cn5g, 2023.
[8] SRSRAN, https://github.com/srsran/srsRAN_Project, 2023.
[9] A. Narayanan, X. Zhang, R. Zhu *et al.*, "A variegated look at 5g in the wild: Performance, power, and qoe implications," in *ACM SIGCOMM*, 2021, p. 610–625.
[10] J. Rischke, P. Sossalla, S. Itting, F. H. P. Fitzek, and M. Reisslein, "5g campus networks: A first measurement study," *IEEE Access*, vol. 9, pp. 121 786–121 803, 2021.
[11] L. Bonati, P. Johari, M. Polese *et al.*, "Colosseum: Large-Scale Wireless Experimentation Through Hardware-in-the-Loop Network Emulation," in *DySPAN*, December 2021.
[12] J. Breen, A. Buffmire, J. Duerig *et al.*, "POWDER: Platform for open wireless data-driven experimental research," in *WiNTECH*, Sep. 2020.
[13] M. S. Khan, B. Farzaneh, N. Shahriar, N. Saha, and R. Boutaba, "Slicesecure: Impact and detection of dos/ddos attacks on 5g network slices," in *IEEE FNWF*, 2022, pp. 639–642.
[14] N. A. Mohammedali, T. Kanakis *et al.*, "Performance evaluation for end-to-end slice management in 5g/b5g cellular networks," in *IEEE SoftCOM*, 2022, pp. 1–6.
[15] N. Saha, A. James, N. Shahriar, R. Boutaba, and A. Saleh, "Demonstrating network slice kpi monitoring in a 5g testbed," in *IEEE/IFIP NOMS*, 2022, pp. 1–3.
[16] F. He, W. Yang, B. Cui, and J. Cui, "Intelligent fuzzing algorithm for 5g nas protocol based on predefined rules," in *ICCCN*, 2022, pp. 1–7.
[17] L. Wischhof, F. Schaipp, S. Schuhbäck, and G. Köster, "Simulation vs. testbed: Small scale experimental validation of an open-source lte-a model," in *IEEE PIMRC*, 2020, pp. 1–7.
[18] C. Wei, A. Kak, N. Choi, and T. Wood, "5gperf: Profiling open source 5g ran components under different architectural deployments," in *ACM SIGCOMM Workshop on 5G-MeMU*, 2022, p. 43–49.
[19] A. Sahbafard, R. Schmidt, F. Kaltenberger, A. Springer, and H.-P. Bernhard, "On the performance of an indoor open-source 5g standalone deployment," in *IEEE WCNC*, 2023.
[20] Sockperf, https://github.com/Mellanox/sockperf, 2023.
[21] ps, https://manpages.ubuntu.com/manpages/trusty/en/man1/ps.1posix.html, 2023.
[22] Powerstat, https://launchpad.net/ubuntu/+source/powerstat, 2023.
[23] Network Info Cell lite, https://www.m2catalyst.com/network-cell-info-lite, 2023.
[24] 3GPP, "User equipment (ue) radio access capabilities," 3GPP, Technical Specification (TS) 28.552, 01 2023, version 17.3.0.