

Performance Comparison between Monolithic gNB and DU-CU Split Deployments in 5G Networks

Michael Amistà

Apeiroon

Vicenza, Italy

michael.amista@apeiroon.com

Claudio E. Palazzi

Department of Mathematics

University of Padua

Padua, Italy

cpalazzi@math.unipd.it

Lorenzo Perinello

Department of Mathematics

University of Padua

Padua, Italy

lorenzo.perinello@phd.unipd.it

Gianluca Verin

Apeiroon

Vicenza, Italy

gianluca@apeiroon.com

Abstract—As mobile networks adopt increasingly heterogeneous and multi-vendor RAN solutions, the complexity of configuring gNBs grows due to diverse vendor-specific architectures and parameter sets. Among the most widely adopted platforms, srsRAN and OpenAirInterface (OAI) provide software-defined implementations of the 5G base stations (gNodeBs), supporting both monolithic and disaggregated (DU-CU) deployment models. Despite their widespread use, a systematic comparison of their performance in different architectural configurations remains limited in the literature. This work presents an experimental comparison of Next Generation Radio Access Network (NG-RAN) implementations. Specifically, the work presents an experimental evaluation of srsRAN and OAI in standalone (SA) 5G networks, considering both monolithic and DU-CU split deployments. Experiments are conducted on a reproducible Software Defined Radio (SDR) based testbed using identical radio and hardware configurations, measuring round-trip latency, TCP and UDP throughput, jitter, and packet loss. The results show that OAI achieves lower latency and higher throughput in monolithic deployments, while srsRAN provides more stable latency and significantly better uplink performance in DU-CU split configurations. These findings highlight that the optimal choice of open-source NG-RAN implementation depends on the target deployment architecture, offering practical guidance for researchers and practitioners designing 5G experimental platforms.

Index Terms—5G, NG-RAN, srsRAN, openAirInterface, test

I. INTRODUCTION

The continuous evolution of mobile networks toward 5G and beyond has driven a paradigm shift in Next Generation Radio Access Network (NG-RAN) design. The 5G New Radio (NR) specifications introduced by 3GPP enable unprecedented flexibility through virtualization, functional disaggregation, and software-defined implementations. These changes, while promising improved scalability and innovation, have also increased architectural complexity and configuration heterogeneity, especially in experimental and research environments. Open-source implementations of NG-RAN components have thus become essential for testing, education, and rapid prototyping. Among the available options, srsRAN and OpenAirInterface (OAI) stand out as two of the most mature and widely adopted open-source 5G SDR.

Despite their popularity, these implementations differ in architecture, configuration mechanisms, and functional split support. However, a systematic analytical and experimental

comparison between these two stacks, particularly considering both DU-CU split deployments, remains limited in the literature. Such an analysis is crucial for researchers and practitioners seeking to understand how open-source gNodeBs perform.

This paper presents an experimental comparison of the srsRAN and OAI 5G gNodeBs, considering monolithic and DU-CU split deployments. Our experimental campaign compares monolithic gNB and DU-CU split deployments on metrics such as round-trip latency, TCP and UDP average throughput, jitter, and packet-loss. The results clarify that OAI performs better in monolithic deployments and srsRAN in DU-CU split scenarios.

II. RELATED WORK

Research benchmarking open-source NG-RAN implementations has expanded rapidly with the maturation of srsRAN and OAI. However, because these stacks are relatively new, the literature remains dominated by deployment guides and single-stack characterizations, with few head-to-head evaluations. Alves *et al.* [1] present an experimental comparison of srsRAN and OAI under varying radio conditions, while Amini and Rosenberg [2] report systematic uplink/downlink variations across several TDD testbeds. Other studies examine single-stack limits or methodology. Håkegård *et al.* [3] quantify the performance boundaries of an srsRAN platform. Rouili *et al.* [4] compare SDR-based, emulated, and simulated OAI deployments, showing that non-hardware setups can misrepresent real over-the-air performance. Despite demonstrating the utility of these platforms, the literature lacks a systematic comparison evaluating srsRAN and OAI across monolithic and disaggregated (DU-CU split) topologies under identical conditions. Existing studies focus on monolithic setups or single-stack evaluations, leaving the performance interaction between architectural split and stack implementation unstudied. Furthermore, the literature notes that OAI is considerably more difficult to configure than srsRAN due to low-level physical settings [5].

III. EXPERIMENTAL ASSESSMENT

The objective of the experiments is to compare the srsRAN and OAI gNB and DU-CU split implementations under iden-

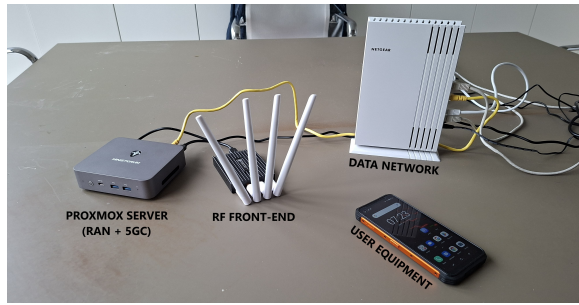


Fig. 1. 5G SA private network experiments testbed

tical test conditions. For each RAN implementation, ten repetitions of TCP uplink, TCP downlink, UDP uplink, and UDP downlink throughput tests are performed. Periodic ICMP requests are issued to measure round-trip latency, while all measurements follow a consistent methodology that ensures that any performance differences can be attributed solely to the underlying RAN architecture. The code used to automate the configuration of the experiments is publicly available in the repository [6].

A. Experiments Testbed

Fig. 1 shows a compact 5G standalone (SA) setup where experiments were carried out. The setup is designed to reproduce the essential components of a 5G system, emulating the behavior of real 5G networks. The testbed consists of an UE device acting as the end-user terminal, responsible for generating traffic and initiating procedures such as registration, the establishment of the PDU session, and data transfer. Its interaction with the rest of the system enables observation of latency, connectivity, and protocol behavior under controlled conditions.

The NG-RAN stack under evaluation is deployed on the Proxmox server and communicates with the UE through a USRP B210 that acts solely as an RF front-end. While the USRP handles the analog transmission and reception of NR signals, all PHY, MAC, RLC, PDCP, RRC, and higher-layer procedures are executed on the server itself. Although not equivalent to a commercial gNB, this architecture accurately reproduces the operations required for high-fidelity laboratory experiments. The Proxmox host runs both the gNB and the 5G Core (5GC). Open5GS, the chosen implementation of 5GC, is deployed inside a virtual machine (VM1), where AMF, SMF, UPF, NRF, and PCF operate as independent SBA services communicating via standard HTTP/2 interfaces. A commercial router provides the external data network and serves as the end point for user-plane flows exiting the UPF, enabling realistic end-to-end data-path evaluation.

To ensure predictable real-time performance during high-throughput tests, the Proxmox server, equipped with an Intel Core i3-N305 (8 physical cores at 3.80 GHz) and 16 GB RAM, uses CPU isolation. Four of the eight cores are reserved exclusively for the gNB stack, shielding the RAN processes from OS scheduling and background tasks. Furthermore, the

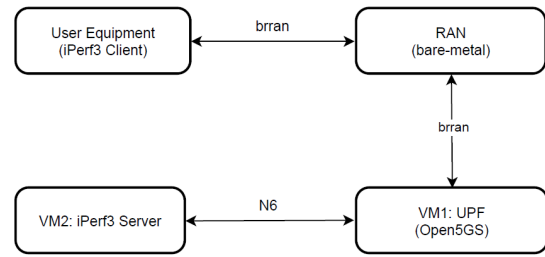


Fig. 2. End-to-end data flow in the 5G SA experiments testbed

gNB stack (srsRAN or OAI) runs at bare-metal, avoiding virtualization overhead that could interfere with the performance evaluation of the RAN components under test. In DU-CU split experiments, one isolated core is dedicated to CU processing, while three cores execute DU operations, reflecting the heavier computational requirements of real-time baseband execution. The remaining four cores support virtual machines and auxiliary services. All experiments use identical physical and protocol-layer parameters: FR1 band n78 at 3.5 GHz, 20 MHz channel bandwidth, 30 kHz subcarrier spacing, and a single-UE configuration. A standard 3GPP TDD frame pattern is used to balance the uplink and downlink capacity. The constant maintenance of frequency, bandwidth, numerology, and UE load ensures that observable performance differences arise from the RAN implementation itself rather than from variations in radio configuration.

B. Measurements

Performance evaluation relies on an end-to-end 5G SA topology in which the gNB runs directly on the Proxmox bare-metal host, the 5GC executes in VM1, and a dedicated traffic generator runs in VM2. As shown in Fig. 2, all components are interconnected using Proxmox virtual bridges that emulate N1, N2, N3, and N6 interfaces. The RAN node connects to the brran bridge, which carries both NGAP and GTP-U traffic to the core; after UPF processing, user-plane packets exit through the N6 bridge toward VM2 and the external router.

The iPerf3 server runs on VM2 and is attached exclusively to the N6 bridge. This isolates user-plane throughput generation from the VM of the core network and ensures that any processing bottleneck observed during tests reflects the behavior of the gNB stack rather than limitations in the core or virtualization environment. Once the UE completes the registration and establishment of the PDU session, it initiates TCP or UDP iPerf3 sessions to VM2, generating controlled downlink and uplink traffic flows.

iPerf3 provides application-layer throughput, retransmission counts, jitter measurements, and packet-loss statistics. Its ability to operate in both TCP and UDP modes and to use parallel streams makes it suitable for stressing the gNB under progressively increasing load. Bidirectional throughput is evaluated by running separate downlink and uplink sessions, while parallel pings from the UE to VM2 provide round-trip time (RTT) samples during each test. Traffic follows the entire

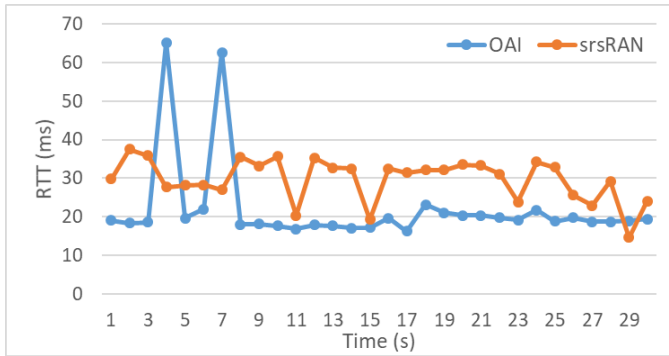


Fig. 3. Monolithic gNB - srsRAN vs OAI RTT evolution over time

TABLE I
GNodeB PERFORMANCE: MONOLITHIC VS DU-CU SPLIT

Measurement	Monolithic		DU-CU split	
	srsRAN	OAI	srsRAN	OAI
Round-Trip-Time (ms)	30	24	31	31
TCP DL Throughput (Mbps)	38.8	46.1	36.2	36.7
TCP UL Throughput (Mbps)	7.7	13.4	19.8	9.7
UDP DL Throughput (Mbps)	41.7	46.3	43.1	41.5
UDP UL Throughput (Mbps)	8.4	12.5	20.9	9.3

path of UE → gNB → 5GC → iPerf3, ensuring that all RLC, PDCP, GTP-U, and virtualization layers are exercised.

The experiments focus on five key metrics that together characterize both the real-time utilization of RAN resources and its end-to-end service quality. First, throughput is measured bi-directionally via iPerf3 streams between each UE and the Proxmox server. Second, round-trip latency is measured by issuing periodic ICMP (Internet Control Message Protocol) echo requests (ping) from the UE to the iPerf3 server, providing information on transmission delays introduced by RLC buffering, HARQ retransmissions, and GTP-U encapsulation. In addition, jitter is evaluated in UDP mode by observing variation in inter-arrival time of packets at the server side, highlighting instability caused by scheduling delays or congestion in the user plane. Finally, packet-loss statistics are collected from UDP test runs, quantifying the resilience of each RAN implementation to transient PHY errors, queue overflows, or missed retransmissions. Together, these metrics provide a comprehensive view of how each software stack performs under tightly controlled, reproducible conditions.

IV. RESULTS

A. Monolithic gNodeBs

Table I summarizes the end-to-end metrics measured for the monolithic gNB experiments. The two implementations show clearly different behavior across latency and throughput. The measured RTTs average 30 ms for srsRAN and 24 ms for OAI. As shown in Fig. 3, the RTTs of srsRAN exhibit a wide distribution (observed span 14-37 ms), indicating higher jitter and less consistent timing, while OAI shows a lower

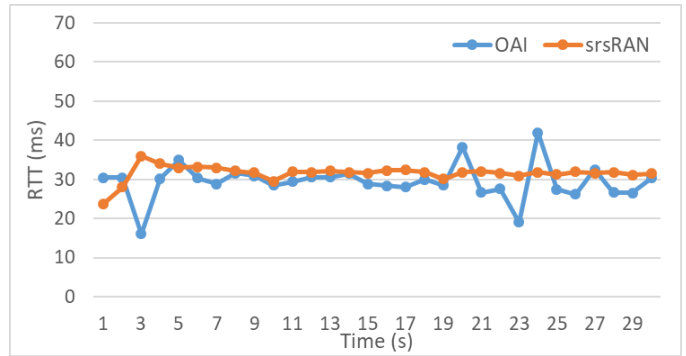


Fig. 4. DU-CU split - srsRAN vs OAI RTT evolution over time

mean, and, after a few large start spikes (observed up to 65 ms), the RTTs cluster tightly around 17-22 ms. Thus, although srsRAN occasionally attains lower minimum delays, OAI provides more stable and generally lower latency once the system stabilizes.

TCP results further highlight implementation differences. In downlink, OAI achieves 46.1 Mbps on average compared to srsRAN's 38.8 Mbps (an increase of 19%), with OAI also showing a smaller run-to-run spread; srsRAN's DL TCP measurements vary more (one low at 31.5 Mbps contributed to the larger variance). In uplink TCP, OAI again outperforms srsRAN, achieving 13.4 Mbps versus 7.7 Mbps (74% higher); OAI's uplink rates are both larger and more stable (coefficient of variation 7.8% vs 15% for srsRAN), indicating a more efficient uplink data-path and scheduler in this testbed.

UDP measurements show the same pattern in both directions, but add timing and loss detail. The average UDP throughput of the downlink is 46.3 Mbps for OAI and 41.7 Mbps for srsRAN (11% higher for OAI); OAI's DL UDP runs are tightly clustered, while srsRAN shows a wider spread driven by an outlier. The packet-delay variation (jitter) in DL favors OAI: mean jitter 0.20 ms (range 0.143-0.264 ms) versus srsRAN's 0.39 ms (range 0.306-0.447 ms), i.e., roughly 50% reduction in average jitter under OAI. Downlink loss statistics also favor OAI: average loss 0.22% (peaking at 0.54%) compared to srsRAN's average 0.96% (driven by a single 3.3% spike). In the uplink UDP tests, OAI achieves 12.5 Mbps compared to srsRAN's 8.4 Mbps (49% higher), and again delivers lower and more consistent jitter (mean 3.6 ms, range 1.2–5.4 ms) versus srsRAN (mean 7 ms, range 2.9–8.7 ms). In particular, uplink UDP packet loss was 0% for both implementations across the ten runs, so the main differentiators in UL are throughput and timing stability, rather than reliability.

B. Disaggregated gNodeBs: DU-CU Split

Table I also reports the aggregate results for the DU-CU split experiments. Both implementations exhibit very similar median RTTs (31 ms), but the distributions and timing stability differ markedly. Fig. 4 shows that srsRAN's RTTs are tightly clustered (aside from an early low of around 23.8 ms, nearly

every sample falls in a narrow 30-36 ms band), producing low jitter and highly predictable latency in the split. The OAI split instantiation matches the same median, but shows pronounced outliers (observed down to 16.1 ms and up to 42 ms), indicating intermittent buffering or scheduling interactions across the fronthaul that produce occasional high-latency excursions.

The downlink TCP and UDP rates are essentially comparable between the stacks in the split configuration. Downlink TCP averages 36.2 Mbps for srsRAN and 36.7 Mbps for OAI, i.e., virtually identical mean throughput; the per-run stability differs somewhat (srsRAN's mean driven by two high runs and one low outlier). Downlink UDP favors srsRAN slightly (43.1 vs 41.5 Mbps), with srsRAN showing a compact range (42.7-43.9 Mbps) while OAI's DL UDP is a bit more spread (39.9-42.9 Mbps). Packet-loss behavior in DL is mixed: srsRAN's mean loss sits around 0.89% (range 0.31-1.50%), while OAI's mean is higher (1.344%) but driven by a single large outlier (4.5%); excluding that outlier would reduce OAI's mean below srsRAN's. The most striking differences appear in uplink performance. For TCP uplink srsRAN achieves 19.8 Mbps with very low variation, whereas OAI averages 9.7 Mbps (less than half of srsRAN's rate). Uplink UDP shows the same pattern: srsRAN 20.9 Mbps (tight 19.5-21.4 Mbps) versus OAI 9.3 Mbps (6.3-10.7 Mbps). Jitter in the uplink is also substantially better under srsRAN: mean jitter 2.16 ms (tight 1.96-2.44 ms) compared to OAI's mean 9.25 ms (wide 5.79-20.42 ms). Both platforms record 0% packet loss across all uplink UDP runs, so throughput and timing stability, rather than reliability, distinguish the implementations in the uplink.

C. Implications

The experiments conducted across monolithic and DU-CU split architectures demonstrate that performance advantages are highly dependent on the chosen deployment model. In the monolithic gNB architecture, OAI achieves a lower average RTT (24 ms) with minimal jitter once stabilized, whereas srsRAN exhibits a higher average RTT (30 ms) and greater variation. OAI also consistently outperforms srsRAN in both downlink and uplink TCP throughput, delivering higher mean data rates with smaller deviations. Similarly, in UDP tests, OAI provides superior throughput, lower jitter, and comparable or slightly lower packet-loss percentages. Since all protocol layers are co-located in the monolithic architecture, OAI's more mature PHY/MAC integration translates directly into improved latency and throughput, making it the preferred choice for single-node deployments focused on maximizing user-plane performance. In contrast, the DU-CU split architecture favors srsRAN. Although both implementations achieve similar average RTTs, srsRAN delivers highly consistent latency (approximately 31 ms) with very low jitter, while OAI exhibits occasional high-latency spikes. Under split deployment, srsRAN also matches or exceeds OAI in both downlink and uplink TCP throughput, providing more predictable and sustained performance. UDP measurements further highlight srsRAN's advantages, with higher throughput, sub-millisecond jitter, and minimal packet loss, whereas OAI

experiences reduced throughput and elevated jitter, particularly in uplink transmissions due to fronthaul processing. These results indicate that srsRAN's scheduler optimization, buffer management, and fronthaul implementation are better suited for distributed RAN deployments.

V. CONCLUSION AND FUTURE WORK

This work compared srsRAN and OpenAirInterface through controlled experiments. Both implementations were found to support the main procedures required for a functional gNodeB, with differences that largely reflect project-specific design choices. The experimental evaluation on the FR1 band testbed showed clear behavioral distinctions: in monolithic deployments, OAI achieved lower RTTs and higher, more stable TCP/UDP throughput; under the DU-CU split, downlink performance between stacks became similar, while srsRAN consistently outperformed OAI in uplink throughput, jitter, and latency stability. Future extensions include evaluating multi-UE and multi-cell scenarios, testing on more powerful hardware, and exploring additional SDR platforms and FR2 bands. Experimenting with other O-RAN splits, such as incorporating a separate radio unit, would broaden the understanding of disaggregated deployments. The configuration generator can be expanded to support FDD and FR2 numerologies, additional devices, and more deployment templates, increasing its usefulness across a wider range of research setups.

ACKNOWLEDGMENT

Partial financial support for this work is provided by the Department of Mathematics of the University of Padua, through its SID funding program. Furthermore, this research also falls within the framework of the project "Sustainable Mobility Center - Centro Nazionale per la Mobilità Sostenibile – CNMS" Spoke 8 "MaaS & Innovative Services" funded by the European Union under Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S su alcune Key Enabling Technologies", NextGenerationEU, project code CN00000023, thematic area: Sustainable Mobility (Mobilità Sostenibile).

REFERENCES

- [1] R. P. Alves, J. G. A. da S. Alves, M. R. Camelo, W. O. de Feitosa, V. F. Monteiro, and F. R. P. Cavalcanti, "Experimental comparison of 5g sdr platforms: srsran x openairinterface," 2024. [Online]. Available: <https://arxiv.org/abs/2406.01485>
- [2] M. Amini and C. Rosenberg, "Performance analysis and comparison of full-fledged 5g standalone experimental tdd testbeds in single & multi-ue scenarios," 2024. [Online]. Available: <https://arxiv.org/abs/2407.02341>
- [3] J. E. Håkegård, H. Lundkvist, A. Rauniyar, and P. Morris, "Performance evaluation of an open source implementation of a 5g standalone platform," *IEEE Access*, vol. 12, pp. 25 809–25 819, 2024.
- [4] M. Rouili, N. Saha, M. Golkarifard, M. Zangooei, R. Boutaba, E. Onur, and A. Saleh, "Evaluating open-source 5g sa testbeds: Unveiling performance disparities in ran scenarios," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, 2024, pp. 1–6.
- [5] M. Seidel, A. I. Grohmann, P. Sossalla, F. Kaltenberger, and F. H. Fitzek, "How to get away with openairinterface: A practical guide to 5g ran configuration," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2023, pp. 1–6.
- [6] M. Amista, "gnb config generator," <https://github.com/michaelamista/gnb-config-generator>, 2024, source code repository, accessed July 21, 2026.