Demystifying O-RAN Implementations under Resource-Constrained Hardware: An Experimental Evaluation

Juan Navarro[1], Alberto Castro[1], Carlos A. Astudillo[2], Claudina Rattaro[1*]

[1*]Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, Montevideo, 11300, Uruguay.
[2]Instituto de Computação, Universidade Estadual de Campinas, Av. Albert Einstein, 1251, Campinas, 13083-889, SP, Brazil.

*Corresponding author(s). E-mail(s): crattaro@fing.edu.uy;
Contributing authors: juan.navarro@fing.edu.uy; acastro@fing.edu.uy; castudillo@unicamp.br;

Abstract

Open Radio Access Network (O-RAN) has emerged as a disruptive paradigm for mobile networks by disaggregating traditional RAN functions into software-based components interconnected through open and interoperable interfaces. By enabling multi-vendor integration, cloud-native deployments, and programmability via rApps and xApps, O-RAN promises increased flexibility, innovation, and cost efficiency. However, due to the rapid evolution of specifications, software stacks, and experimental platforms, practical knowledge on how to experiment with O-RAN remains fragmented and difficult to access, particularly for laboratories with limited hardware resources. This paper presents a structured theoretical and practical study of the O-RAN ecosystem aimed at lowering the entry barrier to academic and small-scale research. First, we consolidate the architectural foundations of O-RAN, including its main components, interfaces, and control principles. Then, we comparatively evaluate five representative open O-RAN testbeds spanning simulated, hybrid, automated, and flexible emulated deployments, all analyzed from the perspective of reproducibility, deployment complexity, realism, and resource requirements. Rather than providing an exhaustive benchmark of O-RAN performance, the paper offers a reproducible, decision-oriented guide to help researchers select the most appropriate experimentation platform for their needs. Illustrative use cases are implemented on the most promising platforms to validate monitoring, control, and integration capabilities. All configurations and scripts are publicly released to support reproducibility and future experimentation on open and intelligent RAN solutions.

Keywords: Open RAN, O-RAN Alliance, O-RAN Testbeds, Resource-Constrained Experimentation, Open-Source Platforms, RAN Intelligent Controller

1 Introduction

Historically, mobile network operators have been highly dependent on a small number of equipment vendors that provide proprietary hardware and software solutions, often delivered as closed, black-box systems. This vendor lock-in has resulted in limited interoperability, high deployment costs, and slow innovation cycles. Although standardization bodies such as the 3rd Generation Partnership Project (3GPP) have played a key role in harmonizing radio technologies, most internal interfaces and implementation details remain proprietary. While network function virtualization significantly increased openness in the mobile core, the Radio Access Network (RAN) remained largely proprietary and vertically integrated.

Structural market asymmetries reinforce this situation. While hundreds of mobile network operators coexist worldwide, the global RAN equipment market is dominated by a small number of vendors, which account for most of the market share. These imbalances generate conflicting incentives: operators tend to favor open, interoperable, and disaggregated architectures, whereas vendors often promote end-to-end proprietary solutions. As a result, innovation has traditionally been constrained to what vendors decide to implement, leaving other actors—such as academia, small companies, and independent developers—with limited influence on real-world deployments and restricted access to experimental platforms.

The landscape began to change with the emergence of network virtualization and, later, with open-source software platforms such as OpenAirInterface (OAI), srsRAN, and Open5GS. These tools significantly lowered the barrier to entry for academic research and laboratory experimentation, enabling cost-effective deployments of LTE and 5G networks and fostering a substantial increase in scientific contributions. Nevertheless, despite these advances, commercial RAN deployments remained largely unchanged, preserving closed interfaces and vendor-specific implementations.

Building upon these developments, the concept of an open and disaggregated RAN gained momentum around 2014 through regional operator-led initiatives. These efforts merged in 2018 to form the Open Radio Access Network (O-RAN) Alliance, founded by major mobile operators. Its goal is to define open, interoperable, and intelligent RAN architectures. Today, the Alliance brings together operators, vendors, research institutions, and software communities. It is complemented by the O-RAN Software Community (OSC), hosted by the Linux Foundation, which develops open-source implementations aligned with O-RAN specifications.

O-RAN aims to virtualize and disaggregate RAN functions, interconnect them through standardized open interfaces, and introduce intelligence via the Non-Real-Time and Near-Real-Time RAN Intelligent Controllers (Non-RT RIC and Near-RT RIC). This architecture enables programmability through rApps and xApps, thereby allowing third parties to develop applications for optimization, automation, and AI-driven control. Despite rapid specification development and early deployments, O-RAN remains a relatively recent and evolving technology. Consequently, knowledge is scattered across technical documents, standards, experimental prototypes, and isolated laboratory experiences, making entry into the field challenging for new researchers.

In this context, the objective of this work is to facilitate experimental access to O-RAN by consolidating architectural knowledge and comparatively evaluating practical testbed implementations that can be deployed under constrained hardware conditions. Adopting a combined theoretical and experimental approach, this paper first presents a comprehensive overview of the O-RAN architecture, components, interfaces, and design principles. It then analyzes five representative O-RAN testbeds spanning simulated, hybrid, automated, and flexible emulated environments. Selected use cases are implemented on the most promising platforms to illustrate their practical applicability, flexibility, and constraints. Rather than ranking these solutions through an exhaustive performance benchmark, the goal is to provide a structured and experimentally grounded set of options that can guide researchers in selecting the most suitable platform for their specific needs.

The contributions of this work are threefold. First, we provide a structured and accessible consolidation of O-RAN architectural concepts, interfaces, and programmable control elements, aimed at reducing the entry barrier for researchers new to the field. Second, we comparatively evaluate five representative open O-RAN experimental platforms that can be deployed under constrained hardware conditions. Third, we provide a practical, decision-oriented analysis of the trade-offs among these platforms in terms of reproducibility, flexibility, realism, software maturity, and resource requirements.

Accordingly, the use cases presented in this paper are not intended as exhaustive performance benchmarks, but as demonstrative validation scenarios that reveal the practical strengths, limitations, and research affordances of the evaluated platforms.

This article extends our work presented at the 13th Wireless Days Conference (WD 2025) [1]. Compared with the conference version, the present manuscript expands the architectural and related-work discussion, incorporates an additional evaluated implementation, and broadens the experimental section with multiple demonstrative use cases and a more explicit comparative analysis of platform trade-offs.

The remainder of this article is organized as follows. Section 2 introduces the O-RAN architecture and its main components. Section 3 reviews related work on O-RAN testbeds and experimental platforms. Section 4 presents the evaluated testbeds and the selected use cases. Section 4 presents the evaluated testbeds and their main trade-offs. Section 5 introduces the selected demonstrative use cases, the measurement methodology, and the corresponding experimental observations. Finally, Section 6 summarizes the main findings and outlines directions for future research.

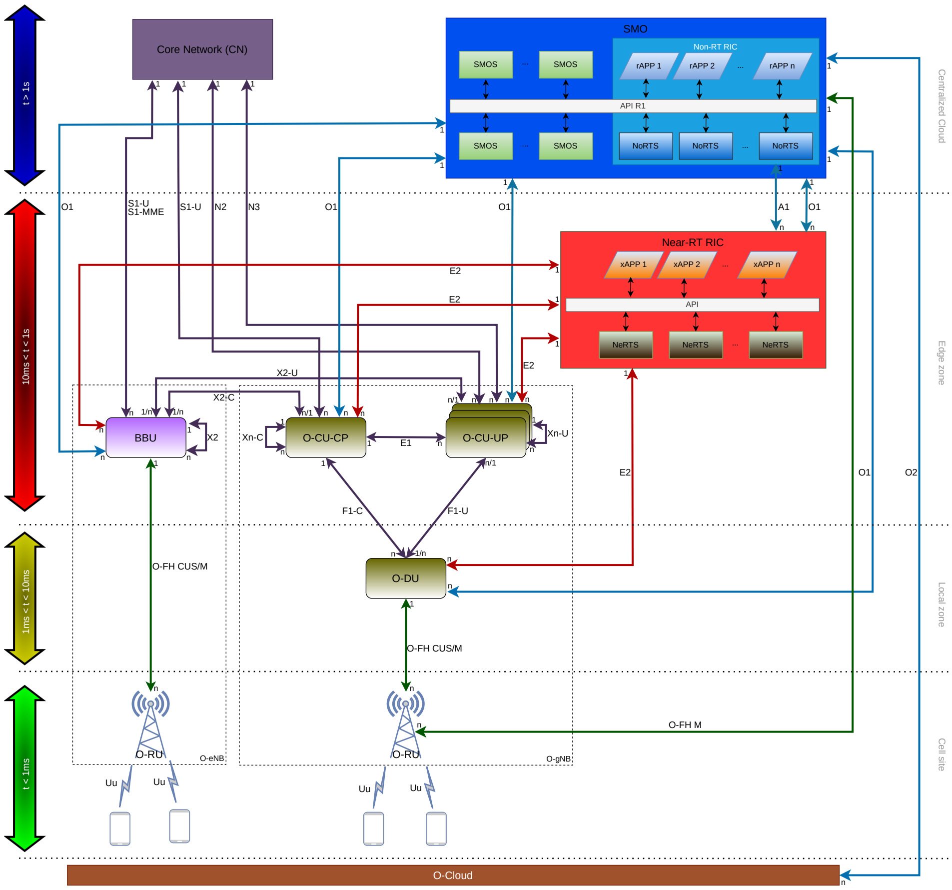


Fig. 1: O-RAN architecture showing the interaction among the Core, SMO, Non-RT RIC, Near-RT RIC, O-gNB/O-eNB, and O-Cloud through open interfaces (A1, O1, O2, E2, and O-FH). The figure highlights control-loop time scales (left) and deployment domains (right).

2 O-RAN Architecture

This section presents the O-RAN architecture, describing its main components, functional roles, and open interfaces. The architecture is illustrated in Fig. 1, which highlights the disaggregation of the RAN, the hierarchical control loops across different time scales (left), and the geographical deployment domains (right). The description is primarily based on O-RAN Alliance specifications [2], complemented by relevant literature [3–7].

2.1 Service Management and Orchestration (SMO) and Non-RT RIC

At the top of the O-RAN architecture, the Service Management and Orchestration (SMO) entity provides end-to-end management of the RAN, including FCAPS functions and lifecycle management of virtualized network functions hosted in the O-Cloud. The SMO typically operates at the regional or national level and constitutes the primary management and orchestration component of the architecture.

Within the SMO resides the Non-Real-Time RAN Intelligent Controller (Non-RT RIC), which operates with control-loop latencies exceeding 1 s. The Non-RT RIC hosts rApps and is responsible for policy-driven optimization, long-term analytics, and AI/ML model training. It interacts with the Near-RT RIC via the A1 interface and with managed network elements via the O1 interface. Internally, both the SMO and the Non-RT RIC follow a service-based design, exposing functionality through well-defined APIs to enable extensibility and vendor interoperability.

2.2 Near-Real-Time RIC

The Near-Real-Time RIC operates at shorter time scales, typically between 10 ms and 1 s, enabling closed-loop control of radio resources. It hosts xApps that monitor network conditions and apply near-real-time control actions, such as traffic steering, load balancing, and quality of service (QoS) enforcement. The Near-RT RIC interacts with O-RAN radio nodes through the E2 interface and receives policies and enrichment information from the Non-RT RIC via A1.

The Near-RT RIC is commonly deployed at the network edge and may control multiple E2 nodes. Its platform provides essential services, including data sharing, conflict mitigation among xApps, subscription management, and optional AI/ML inference support, ensuring the coordinated and safe execution of multiple control applications.

2.3 Open RAN Nodes: O-gNB and O-eNB

O-RAN generalizes 3GPP base stations into Open gNodeB (O-gNB) for 5G and Open eNodeB (O-eNB) for LTE. The O-gNB follows a disaggregated design composed of the Open Central Unit (O-CU), Open Distributed Unit (O-DU), and Open Radio Unit (O-RU). The O-CU is further split into control-plane (O-CU-CP) and user-plane (O-CU-UP) functions, interconnected via standard 3GPP interfaces (E1, F1, Xn).

The O-DU and O-RU are connected through the Open Fronthaul (O-FH) interface, with the 7.2x functional split being the preferred option in O-RAN. This split balances fronthaul bandwidth requirements, latency constraints, and processing capabilities by allocating higher-PHY functions to the O-DU and lower-PHY and RF processing to the O-RU. Both O-gNB and O-eNB expose E2 interfaces toward the Near-RT RIC and O1 interfaces toward the SMO, enabling unified control and management across LTE and 5G deployments.

2.4 O-Cloud

The O-Cloud is the distributed cloud infrastructure that hosts O-RAN network functions. It comprises geographically distributed compute nodes deployed at central data centers, edge locations, or cell sites. The SMO manages the O-Cloud through the O2 interface, which supports infrastructure management and deployment lifecycle operations. Time and frequency synchronization across O-Cloud sites is essential to ensure coordinated RAN operation.

2.5 O-RAN Interfaces

In addition to standard 3GPP interfaces, O-RAN defines a set of open logical interfaces that enable interoperability and programmability:

- **A1**: connects the Non-RT RIC and Near-RT RIC for policy management, enrichment information, and AI/ML model distribution.
- **E2**: links the Near-RT RIC with O-RAN radio nodes (E2 nodes), enabling monitoring and near-real-time control through E2AP and service models.
- **O1**: provides management and configuration connectivity between the SMO and managed elements.
- **O2**: enables management of the O-Cloud infrastructure and deployment of network functions.
- **O-FH**: defines the fronthaul interface between the O-DU and the O-RU, supporting the user, control, synchronization, and management planes.

2.6 Programmability via rApps and xApps

A key innovation of O-RAN is the introduction of programmability through rApps and xApps. rApps operate at non-real-time scales to define policies and optimization strategies based on global network knowledge. In contrast, xApps enforce these policies and perform fine-grained control in near-real time. This separation enables third-party development of control applications and fosters innovation beyond traditional vendor-driven solutions.

In summary, the O-RAN architecture introduces disaggregation, cloudification, and intelligence into the RAN through open interfaces and programmable control loops. These characteristics motivate the need for experimental platforms and testbeds capable of validating O-RAN concepts and supporting the development of rApps and xApps, which are reviewed in the following section.

3 Related Work

The rapid evolution of O-RAN has led to a growing body of literature addressing its architectural principles, experimental validation, and practical deployment challenges. Existing works can be broadly grouped into: (i) analyses of real-world deployments and lessons learned, (ii) experimental and enterprise-scale O-RAN testbeds, (iii) automation and orchestration frameworks, (iv) simulation-based O-RAN platforms, and (v) application-driven extensions beyond traditional mobile access. To better position this contribution, the revised discussion distinguishes more explicitly between enterprise-scale deployments, laboratory-oriented open platforms, automation-focused frameworks, and simulation-based O-RAN environments.

Early studies have highlighted the practical challenges of deploying O-RAN in real environments. In [8], the authors analyze multiple private 5G O-RAN deployments and report significant integration complexity across hardware and software components, limited vendor interoperability, and a lack of mature automation. Their findings emphasize that most deployments progressed incrementally, postponing complex features such as RIC integration to later phases, and underline the need for skilled personnel and CI/CD-based operational practices.

Several works focus on building experimental O-RAN testbeds with varying degrees of realism. High-performance and enterprise-scale platforms are presented in [9–12], which use advanced hardware accelerators, commercial radio units, and cloud-native orchestration to achieve stability and performance comparable to production systems. While these platforms provide valuable insights into large-scale O-RAN operations, they typically rely on costly infrastructure and often omit or only partially integrate RAN Intelligent Controllers (RICs), thereby restricting accessibility for academic and small-scale research.

In contrast, several testbeds emphasize openness, programmability, and accessibility. Platforms based on open-source stacks such as srsRAN, OpenAirInterface, and Open5GS are presented in [13–15], integrating Near-RT RICs and E2 interfaces to enable experimentation with xApps for KPI monitoring, slicing, and intelligent radio control. These works demonstrate the feasibility of O-RAN experimentation using COTS hardware and SDRs. However, they often provide limited documentation on deployment workflows or focus on a subset of architectural components. More recent contributions further reinforce the relevance of open and laboratory-oriented O-RAN experimentation platforms. For instance, recent work has presented open-source testbeds explicitly designed to support AI-enabled monitoring and control studies in realistic 5G/O-RAN environments [16].

Automation and orchestration have emerged as key enablers for reproducible O-RAN experimentation. The NIST blueprint [17] provides a comprehensive guide for deploying O-RAN testbeds using diverse software stacks and automated installation tools, significantly lowering the barrier to entry. Similarly, AutoRAN [18] introduces a zero-touch, cloud-native framework that abstracts heterogeneous hardware and software resources, enabling rapid deployment and reconfiguration of O-RAN systems. These efforts highlight the importance of Infrastructure-as-Code and declarative management for managing O-RAN complexity.

Beyond physical deployments, simulation-based approaches have been proposed to evaluate O-RAN concepts under controlled conditions. Extensions to the ns-3 simulator [19, 20] model O-RAN components and interfaces, enabling the study of AI-driven control and traffic steering without the cost and complexity of real hardware. While simulations offer flexibility and scalability, they abstract away many practical integration and performance issues encountered in real testbeds.

Finally, several works explore O-RAN in nontraditional application domains. Examples include UAV control [21], non-terrestrial networks [22], robotic teleoperation [23], and cell-free MIMO systems [15]. These studies demonstrate the versatility of O-RAN and its suitability for advanced use cases that require programmability and intelligent control. However, they often focus on specific subsystems rather than complete end-to-end platforms.

In summary, existing research demonstrates substantial progress in O-RAN experimentation, but also reveals fragmentation across platforms, uneven integration of RIC components, limited reproducibility, and a strong dependence on either costly infrastructure or highly abstracted simulations. This work builds on these efforts by providing a reproducible and practically oriented comparative study that jointly organizes architectural background, related experimental platforms, deployment requirements, reproducibility considerations, and platform-specific trade-offs under resource-constrained conditions. This broader perspective is also consistent with recent tutorial and survey efforts that organize O-RAN deployment solutions across simulated, emulated, and real testbeds, highlighting both the growing diversity of available platforms and the persistent challenges in interoperability, reproducibility, and deployment complexity [24].

4 Evaluation of O-RAN Implementations under Resource-Constrained Conditions

After introducing the O-RAN architecture and reviewing related experimental platforms, this section evaluates a set of representative O-RAN implementations from a practical perspective. Rather than targeting large-scale deployments or exhaustive performance benchmarking, the focus is placed on feasibility, reproducibility, and adaptability under limited hardware resources. This approach aligns with the main objective of this work: lowering the entry barrier for researchers and institutions seeking initial practical experience with O-RAN technologies.

The selected implementations were chosen based on three criteria: (i) availability of open-source code and public documentation, (ii) the possibility of deployment using modest computational resources, and (iii) relevance within the current O-RAN research ecosystem. In all cases, the original implementations reported in the literature were first reproduced and subsequently adapted as needed to address missing components, resolve compatibility issues, or explore alternative deployment options.

For clarity and ease of reference, each evaluated case is labeled as *Implementation 1*, *Implementation 2*, and so forth. The analysis progressively moves from fully simulated environments to hybrid simulation–real integration, then to automated single-node deployments, and finally to a flexible emulated testbed that integrates real and simulated components. All implementations were executed on a workstation featuring an Intel i7-12700 processor and 16 GB of RAM. To complement the practical comparison of the evaluated platforms, we also observed the host-level resource footprint during representative runs. For the srsRAN-based implementations, RAM usage remained approximately constant at about 3.3 GB in the evaluated setup. CPU usage was more variable and intensive, ranging from an average load of about 60% of one CPU core

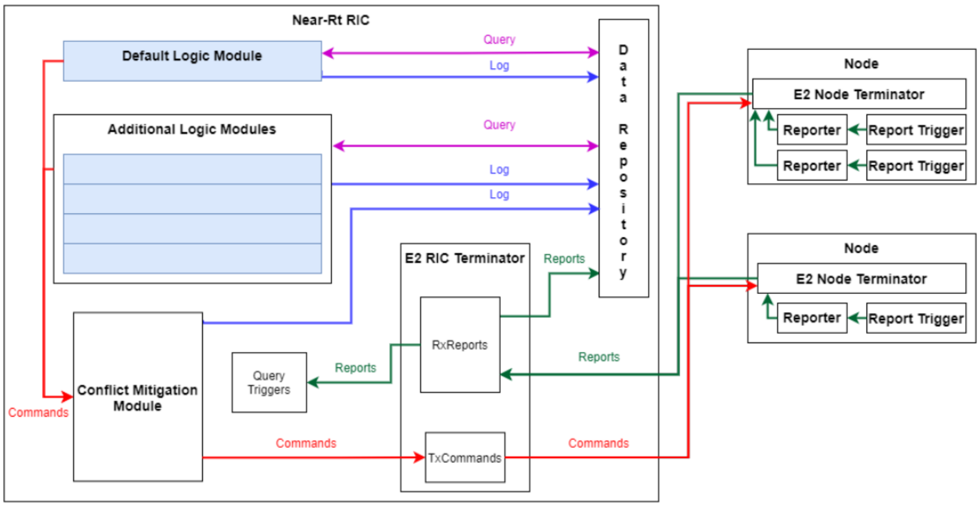


Fig. 2: Implementation 1. Near-RT RIC with its modules and their internal communication, as well as the E2 nodes and their connection to the Near-RT RIC.

when the system was running without an active UE, up to approximately 120% in the host-level CPU metric during throughput experiments. In the observed runs, the OAI-based implementations exhibited lower CPU usage than their srsRAN counterparts. This tendency is consistent with recent experimental results reported in the literature for O-RAN testbeds [25]. For the simulation-based implementations, host-level CPU and RAM usage were not systematically monitored, since these metrics depend strongly on the specific scenario under study, including the number of simulated nodes, the traffic configuration, the mobility model, and the enabled protocol-stack features. As a result, these metrics are highly scenario-dependent and less representative of the platform itself. In addition, scaling purely simulation-based studies is often more straightforward through access to shared computing infrastructure than building and operating fully controlled experimental environments involving SDR hardware, real interfaces, and end-to-end integration across multiple components.

4.1 Simulated and Hybrid O-RAN based on NS-3

4.1.1 O-RAN Implementation 1: Fully Simulated in NS-3

The first implementation relies on the NIST ns-3 O-RAN module [19], which enables fully simulated O-RAN experiments. It models a logical Near-RT RIC composed of logic modules (xApps), a data repository, conflict mitigation, query triggers, and E2 termination. Simulated nodes include reporters and triggers that generate metrics for the RIC. The framework also supports machine learning inference through integration with ONNX or PyTorch. The architecture is shown in Fig. 2.

Key Insights: This solution offers a highly flexible, fully simulated environment well-suited for algorithm development and xApp validation. Its main drawbacks are the

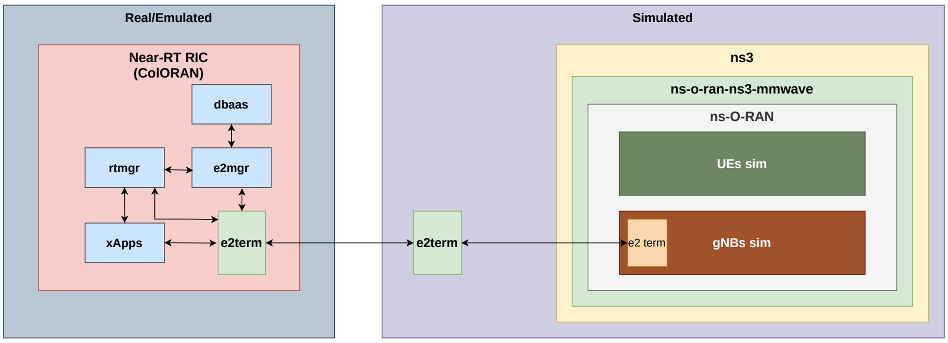


Fig. 3: Implementation 2. In the real or emulated world, the Near-RT RIC is deployed, while the rest of the RAN is simulated within ns-3.

steep learning curve associated with ns-3 and C++, and the fact that it currently supports only LTE-based E2 nodes. Extending this framework toward 5G simulators, such as 5G-LENA, remains an open research direction.

4.1.2 O-RAN Implementation 2: (Hybrid) NS-3 with Real Near-RT RIC Integration

The second implementation is based on the ns-O-RAN framework [20], which enables interaction between ns-3 simulations and a real Near-RT RIC. A fork of the *mmWave* module (*ns-o-ran-ns3-mmwave*) extends ns-3 to support E2 interfaces in simulated gNBs and UEs. The *ns-O-RAN* module further enables the termination of multiple E2 interfaces within the simulation. An additional component, *o-ran-e2sim*, provides E2 termination and connects the simulation to the Near-RT RIC. The Colosseum Near-RT RIC (ColORAN) [26], a reduced version of the OSC RIC, is used to control the simulated RAN. The overall architecture is depicted in Fig. 3.
Key Insights: This hybrid approach significantly expands experimentation possibilities by combining a real RIC with a simulated RAN. However, it relies on outdated versions of both ns-3 and the Near-RT RIC, limiting adaptability and making the platform relatively rigid. The ColORAN Near-RT RIC implements only a minimal subset of Near-RT RIC functions, limited to E2 management, E2 termination, message routing, and basic data storage, and therefore does not represent a full OSC Near-RT RIC deployment.

4.2 NIST Automation Framework

4.2.1 O-RAN Implementation 3.1: srsRAN + OSC Near-RT RIC

The third implementation is based on the NIST *O-RAN-Testbed-Automation* framework [17], which automates the deployment of a nearly complete O-RAN testbed on a single node. This setup integrates Open5GS as the 5G Core, srsRAN for the gNB and UE, and the OSC Near-RT RIC deployed as a Kubernetes-based microservice infrastructure. The RIC includes key modules such as `e2mgr`, responsible for managing

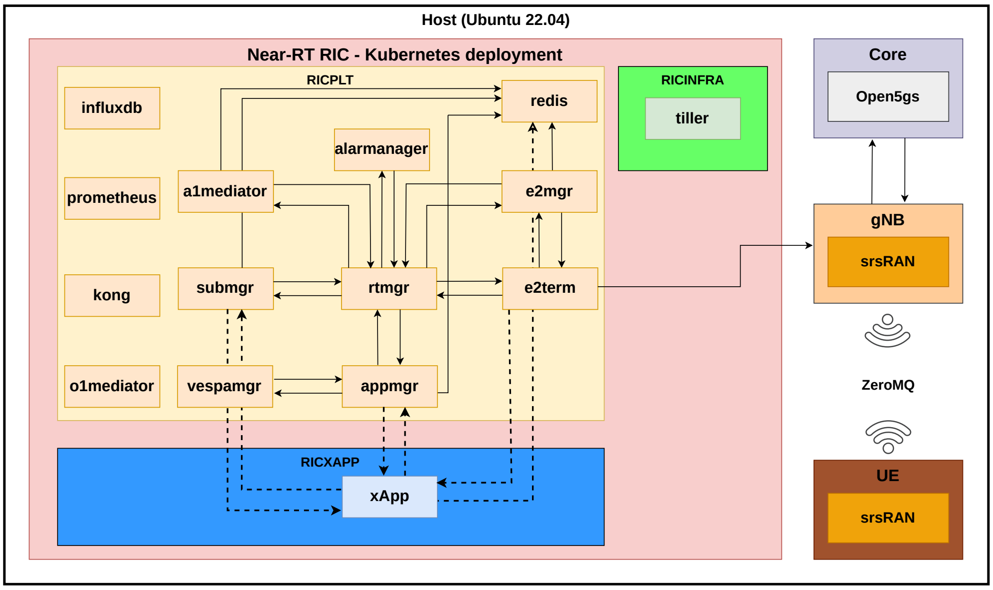


Fig. 4: Implementation 3.1. On the host, the OSC Near-RT RIC is deployed as a Kubernetes infrastructure comprising several pods across three namespaces. In addition, the srsRAN gNB and UE are installed directly on the host, together with the Open5GS core.

E2 connections with RAN nodes; `e2term`, which terminates the E2 interface; `appmgr`, which manages the lifecycle of xApps; Redis for configuration and state storage; and Prometheus for internal monitoring. Communication between the gNB and the RIC takes place over the E2 interface, while the radio channel is emulated using ZeroMQ. The overall architecture is illustrated in Fig. 4. The evaluated Near-RT RIC deployment includes a set of reference xApps provided by the O-RAN Software Community, covering monitoring, control, and experimentation functionalities. These xApps were actively exercised to validate E2-based data collection, inter-xApp interaction, and near-real-time control capabilities.

Key Insights: This implementation proved straightforward to deploy and reproduce, leveraging mature, widely adopted open-source components. However, interoperability issues were encountered during the deployment of certain xApps (notably KPIMON), primarily due to version mismatches between srsRAN and the OSC Near-RT RIC. As part of our experimental campaign, we identified and publicly reported interoperability issues affecting the deployment of certain xApps within the NIST automation framework[1]. These issues underscore the challenges posed by independently evolving software components and the importance of strict version alignment in reproducible O-RAN experimentation.

[1] https://github.com/usnistgov/O-RAN-Testbed-Automation/issues/2

4.2.2 O-RAN Implementation 3.2: srsRAN, OSC Near-RT RIC, and OSC Non-RT RIC Extension

As an extension of Implementation 3.1, the optional deployment of the OSC Non-RT RIC (Release L) was evaluated. While this framework introduces support for A1 policy management, rApps, and information coordination services, several pods failed to operate correctly in practice. Moreover, conflicts with the Near-RT RIC deployment were observed. In the evaluated setup, several pods did not operate reliably and conflicts with the Near-RT RIC deployment were observed. Therefore, rather than drawing a broad conclusion about the ecosystem as a whole, we limit our observation to stating that this particular release and configuration did not yet provide a sufficiently stable basis for reproducible experimentation in our environment. However, this should not be interpreted as a general statement about the entire ecosystem, since recent studies report ongoing progress toward more complete SMO implementations, particularly around O1-based closed-loop control and AI/ML-assisted operation [27].

4.2.3 O-RAN Implementation 4: OAI and FlexRIC

This configuration provided by the NIST automation framework integrates Open5GS, OAI gNB, and UE with E2 support, and the FlexRIC controller [28]. In contrast to the OSC Near-RT RIC, FlexRIC is not a full RIC implementation but a lightweight software development kit for building specialized SD-RAN controllers that can act as Near-RT RICs. FlexRIC follows a modular architecture composed of a server–agent model and supports custom service models and internal applications. Regarding service-model support, both OSC and FlexRIC support E2SM-KPM in the evaluated setups, while their support for E2SM-RC differs in scope and maturity. FlexRIC also provides additional custom service models, including MAC-, RLC/PDCP-, and slice-oriented instrumentation, which increase flexibility but also move the platform away from a strictly OSC-aligned reference path. Conversely, service models such as E2SM-CCC and E2SM-QoE were not available in the evaluated versions.

In this setup, the radio channel is simulated using OAI's RFSimulator. The corresponding architecture is shown in Fig. 5. The evaluated FlexRIC deployment includes a set of reference xApps for monitoring and control, ranging from KPI collection based on the E2SM-KPM service model to deeper MAC, RLC, PDCP, and GTP-layer instrumentation using custom FlexRIC service models, as well as control and control-monitoring xApps enabling near-real-time interaction with E2 nodes.

Key Insights: This solution is computationally less demanding and more customizable than the OSC Near-RT RIC in the evaluated setup, which makes it attractive for low-resource experimentation and low-latency control studies. This observation is consistent with the literature describing FlexRIC as a streamlined software framework for building specialized SD-RAN controllers [28], and should be interpreted within the scope of the evaluated setup rather than as a universal ranking across all deployments.

During the evaluation, a minor issue affecting the deployment/integration workflow was identified and reported by the authors[2]. The experiments reported in this

[2] https://github.com/usnistgov/O-RAN-Testbed-Automation/issues/3

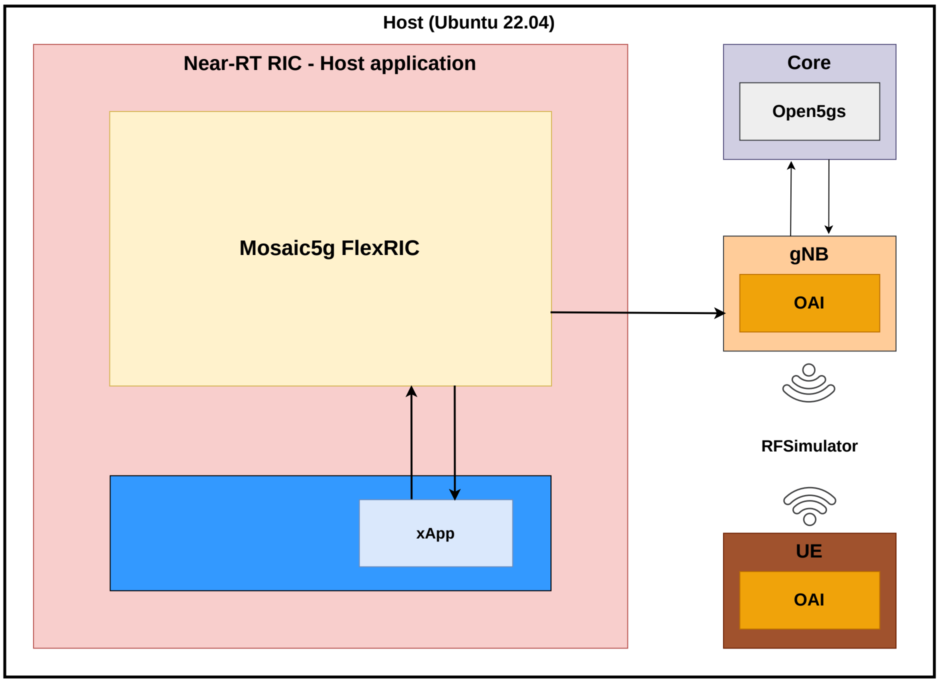


Fig. 5: Implementation 4. On the host, the OAI FlexRIC Near-RT RIC is installed. Additionally, the OAI gNB and UE are installed directly on the host, along with the Open5GS core.

manuscript were conducted using the corrected version; therefore, the presented results were not affected by the issue.

4.3 O-RAN Implementation 5: Flexible Emulated Testbed

Its goal is to provide a flexible and realistic O-RAN testbed that can be adapted to varying hardware availability, ranging from a single commodity workstation to distributed deployments with Software-Defined Radios (SDRs) and commercial-off-the-shelf (COTS) user equipment.

This testbed builds upon experience gained with the NIST automation framework, complemented by updated documentation from the srsRAN project and prior experimental work [29]. Unlike fully automated solutions, this implementation prioritizes architectural flexibility and software version compatibility over deployment simplicity.

The core network is based on Open5GS, while the RAN relies on the srsRAN gNB with E2 support. User equipment can be emulated using srsRAN UE instances or implemented using real COTS devices connected via SDR hardware. For the Near-RT RIC, both the OSC implementation and a FlexRIC-based controller are supported. Depending on the experiment, all components can be deployed on a single host or distributed across multiple machines. The radio channel can be emulated using ZeroMQ,

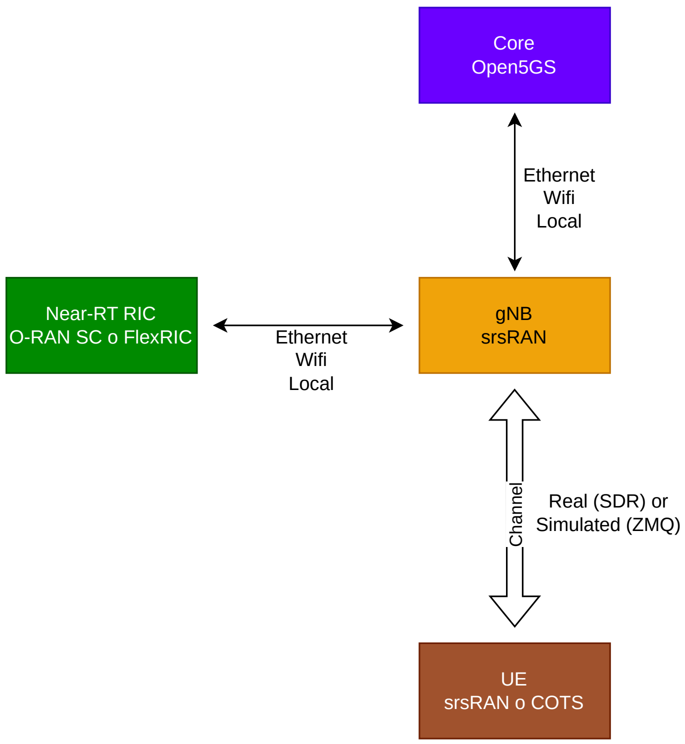


Fig. 6: Architecture of the solution proposed in Implementation 5. This solution not only supports multiple options for the Near-RT RIC and the UE, but also enables each component to be deployed independently, allowing all elements to run on a single host, on separate hosts, or in any combination in between.

implemented via SDRs, or realized over-the-air when COTS devices are used. Fig. 6 summarizes the supported deployment configurations.

Key Insights: Among the evaluated platforms, this implementation offers the highest degree of practical realism, since it supports combinations of emulated and real components, including SDR-based radio access and COTS user equipment. This characterization is relative to the other platforms considered in this study and should not be interpreted as full equivalence to production mobile-network deployments. Its main strength lies in bridging purely emulated and real deployments within a single experimental framework. However, this flexibility increases configuration complexity and sensitivity to software version changes, requiring careful integration and validation. Despite these challenges, the platform provides a practical pathway for transitioning from exploratory O-RAN experimentation toward more realistic laboratory deployments, while still remaining distinct from carrier-grade production environments.

4.4 Summary

The evaluated implementations cover a broad spectrum of O-RAN experimentation approaches, from automated single-node deployments to hybrid and fully simulated environments, and finally to flexible emulated testbeds. Automation frameworks ease reproducibility but often suffer from interoperability and version-alignment issues. Lightweight controllers such as FlexRIC offer flexibility and lower deployment overhead, albeit at the cost of diverging from official OSC implementations in some respects. Simulation-based solutions enable highly controlled experimentation, while flexible emulated platforms provide the closest approximation to real laboratory deployments under limited resources.

Together, these observations show that meaningful O-RAN experimentation is feasible across a wide range of hardware capabilities. Rather than constituting an exhaustive benchmark, this comparative analysis is intended to support platform selection and to facilitate reproducible experimentation. All configurations and scripts used in this work are made publicly available through an institutional repository[3] to support reproducibility and further research.

To facilitate comparison across the evaluated platforms, Table 1 summarizes their main characteristics, highlighting trade-offs between realism, flexibility, and hardware requirements.

5 Use Cases and Experimental Results

This section presents a set of representative use cases designed to experimentally validate the most promising O-RAN testbed implementations identified in the previous section. From the five evaluated platforms, three implementations were selected based on their maturity, flexibility, and suitability for realistic experimentation under resource-constrained conditions: Implementation 4 (emulated O-RAN with FlexRIC), Implementation 1 (fully simulated O-RAN in ns-3), and Implementation 5 (hybrid O-RAN deployment with emulated and COTS user equipment). For each implementation, a dedicated use case was defined to highlight its distinctive capabilities and to assess its functionality, performance, and programmability. The experiments are referred to as Use Case 1, Use Case 2, and Use Case 3, respectively.

5.1 Experimental Scope and Measurement Methodology

The goal of the experimental section is not to provide an exhaustive benchmark of O-RAN system performance, but to validate the practical suitability of representative platforms for experimentation under constrained hardware conditions. Throughput values were obtained from application-layer traffic generators and contrasted with xApp-reported KPIs and RAN logs when applicable. Radio-layer indicators such as

[3] https://gitlab.fing.edu.uy/mobile-optical-networks/testbed/mobile-network

Table 1: Comparison of the evaluated O-RAN testbed implementations.

Impl.	Stack (Core / RAN)	RIC	Radio / Channel	Advantages	Limitations
1 (ns3-ORAN)	Fully simulated ns-3	Logical RIC (xApps, data repository, conflict mitigation)	Fully simulated in ns-3	Highly flexible simulation environment; suitable for algorithm and xApp development; supports ML integration via ONNX/PyTorch	Requires strong ns-3 and C++ expertise; steep learning curve; simulation-only; LTE-only support
2 (ns-O-RAN)	ns-3 simulation / modified mmWave module	ColORAN (reduced OSC RIC)	Simulated in ns-3	Combines a real Near-RT RIC with a simulated RAN; enables flexible and hybrid experimentation	Relies on outdated ns-3 and RIC versions; rigid architecture; limited adaptability
3.1 (NIST)	Open5GS / srsRAN	OSC Near-RT RIC (Kubernetes-based)	ZeroMQ (simulated channel)	Simple and automated deployment; relies on widely adopted open-source tools; official OSC Near-RT RIC	Sensitive to software version changes; interoperability issues across independently evolving components
3.2 (NIST ext.)	Same as Impl. 1	OSC Non-RT RIC (Release L)	ZeroMQ	Introduces rApp support and A1 policy management	Several pods fail to operate correctly; conflicts with the Near-RT RIC; not mature for stable experimentation
4 (NIST–OAI)	Open5GS / OAI gNB & UE	FlexRIC (lightweight SDK)	RFSimulator (OAI)	Lightweight and customizable; supports low-latency xApps; strong integration with OAI	Not an official OSC implementation; divergence from O-RAN reference architecture; learning curve for FlexRIC
5 (srsRAN-based)	Open5GS / srsRAN	OSC Near-RT RIC or FlexRIC	ZeroMQ or real over-the-air channel	Highly configurable; scalable from single-host setups to distributed deployments; supports emulated and real UEs	Increased configuration complexity; higher hardware requirements for large-scale or multi-UE scenarios

SNR, BER, MCS, and scheduler-related parameters were collected through E2-based monitoring and validated against gNB-side observations where possible. Host resource usage was measured at the operating-system level during execution. Accordingly, the reported use cases are intended as demonstrative validation scenarios that expose monitoring, control, integration, and deployment trade-offs. When repeated runs were performed, this is explicitly stated in the corresponding use case description. In all cases, the reported observations are interpreted within the scope of functional and comparative validation, rather than as carrier-grade performance bounds.

5.2 Use Case 1: Emulated O-RAN with Near-RT RIC Monitoring

Use Case 1 is based on Implementation 4 and evaluates an emulated end-to-end O-RAN deployment integrating an Open5GS core, an OpenAirInterface (OAI) gNB, and multiple OAI user equipments (UEs). The wireless channel between the gNB and the UEs is emulated using the OAI RFSimulator, while Near-Real-Time RAN control is enabled through FlexRIC. A monitoring xApp is deployed to collect radio and traffic-related Key Performance Measurements (KPMs) via the E2 interface. Fig. 7

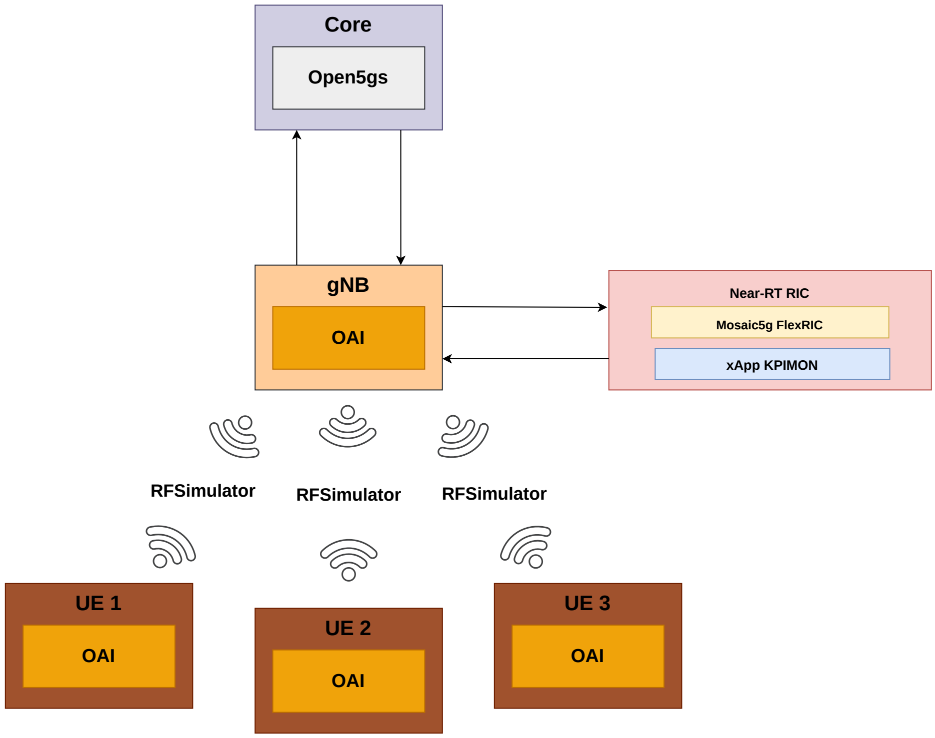
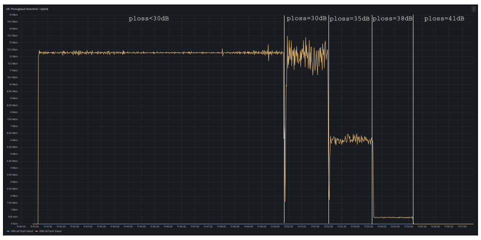


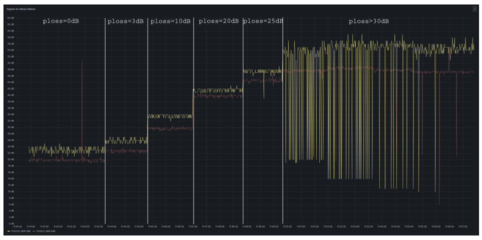
Fig. 7: Architecture of Use Case 1, comprising the mobile network core, one gNB, three UEs, and the Near-RT RIC.

illustrates the architecture of the experimental setup. The xApp subscribes to multiple KPMs, including uplink and downlink throughput, signal quality metrics (RSRP, RSSI, SNR), and scheduler-related parameters. These metrics are aggregated and visualized through a monitoring dashboard.

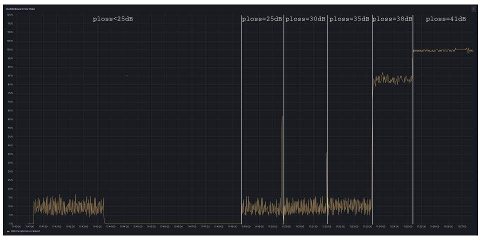
The experimental validation was organized into three stages. First, the system was observed under idle conditions to verify UE detection and baseline KPI reporting. In this stage, the xApp correctly detected connected UEs and reported near-zero traffic values, while radio metrics remained stable and consistent with gNB logs. Second, controlled uplink and downlink traffic was generated between selected UEs and the core network to validate the consistency of throughput and scheduler-related measurements. In these experiments, the reported throughput values closely matched the traffic generated at the application layer, while variations in modulation and coding schemes, PRB allocation, and HARQ statistics evolved coherently with the traffic load. Third, the emulated radio channel was progressively degraded (using the OAI RFSimulator) to assess whether the monitored KPIs coherently reflected link impairments. Starting from the default channel configuration, the path loss was increased according to the following sequence: 0 dB, 3 dB, 10 dB, 20 dB, 25 dB, 30 dB, 35 dB,



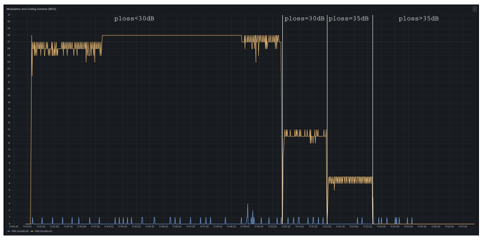
(a) Uplink throughput



(b) Signal-to-noise ratio



(c) Bit error rate



(d) Modulation and coding scheme

Fig. 8: Impact of increasing path loss on radio performance metrics in Use Case 1: (a) uplink throughput, (b) signal-to-noise ratio, (c) bit error rate, and (d) selected modulation and coding scheme.

38 dB, and 41 dB. During the experiment, a constant uplink traffic of 12 Mbps was generated from UE1 toward the core network. Each path-loss level was maintained for sufficient time to allow the monitored KPIs to stabilize before representative values were recorded, and the full experiment was repeated across multiple runs to verify consistency. Since the purpose of this use case is demonstrative, the reported results are intended to show coherent and repeatable functional trends rather than derive a statistical channel model.

To avoid excessive detail, only a subset of the metrics collected by the xApp was analyzed, namely uplink throughput, signal-to-noise ratio, bit error rate, and the selected modulation and coding scheme, as summarized in Fig. 8. The reported results show coherent and repeatable trends between the configured path loss and the monitored KPIs: increasing path loss leads to lower SNR, lower-order modulation and coding selections, higher bit error rates, and reduced throughput. These observations are consistent with radio link adaptation theory and support the functional validity of the monitoring setup, although no formal regression or inferential statistical model is claimed here.

Overall, this experiment shows that the monitoring xApp can capture controlled radio channel variations in near real time, supporting the suitability of this testbed for functional validation of O-RAN monitoring capabilities under reproducible laboratory conditions.

5.3 Use Case 2: O-RAN-based Handover Control in NS-3

Use Case 2 leverages Implementation 1, a fully simulated O-RAN environment based on an extended ns-3 framework. Unlike emulated or real deployments, this platform enables highly flexible experimentation with full control over mobility, topology, and protocol parameters, making it particularly suitable for evaluating RAN control logic.

The selected experiment focuses on handover (HO) decision mechanisms. In legacy LTE networks, HO decisions are based exclusively on radio measurements such as RSRP, RSRQ, or SINR (e.g., A3 event). In contrast, O-RAN enables the integration of external intelligence through xApps, allowing HO decisions to incorporate non-radio information.

The simulation duration was 120 s. UE mobility was generated through the trajectory-generation mechanisms available in the ns-3 framework, using seeded execution to ensure repeatability of the evaluated scenarios. The objective of this use case is to compare the qualitative behavior of the distance-based and A3-based handover logics under identical controlled conditions, rather than to derive statistically general mobility-performance bounds. In this use case, a custom O-RAN logic module performs handovers based solely on the geometric distance between UEs and base stations, while a standard A3-based HO mechanism runs in parallel for comparison. The complete configuration files, experimental scripts, and results are publicly available[4].

Simulation results show that, under the controlled scenarios considered, distance-based handovers occur near the geometric midpoints between cells, whereas A3-based handovers are triggered later due to measurement offsets and time-to-trigger constraints. This qualitative difference is coherent with LTE handover theory and illustrates how O-RAN-capable environments enable alternative control strategies that would be difficult to implement in traditional RAN architectures.

While this simulated environment provides exceptional flexibility and repeatability, it also presents limitations: the lack of direct portability of xApps to real RICs, and the intrinsic abstraction of simulation. Nevertheless, this testbed proves highly effective for rapid prototyping, algorithm validation, and early-stage feasibility studies.

5.4 Use Case 3: Hybrid O-RAN Deployment with Emulated and COTS UEs

Use Case 3 is based on Implementation 5 and represents the most realistic and versatile experimental setup among the evaluated platforms. It supports hybrid deployments combining emulated RAN components with commercial off-the-shelf (COTS) UEs and software-defined radios (SDRs), while maintaining compatibility with both OSC Near-RT RIC and FlexRIC.

Three complementary experiments were conducted. The first experiment deploys a fully emulated stack using srsRAN for the gNB and UE, Open5GS for the core, and the OSC Near-RT RIC. A control xApp simultaneously utilizes the E2SM-KPM and E2SM-RC service models to monitor traffic and enforce control actions. When a predefined traffic threshold is exceeded, the xApp dynamically limits the number of

[4] https://gitlab.fing.edu.uy/mobile-optical-networks/testbed/mobile-network

allocated PRBs, resulting in a measurable throughput reduction. Packet-level inspection confirms the correct exchange of E2AP messages, validating protocol compliance and control-loop operation.

The second experiment replaces the emulated UE with a COTS smartphone connected via an SDR-based gNB. In this configuration, FlexRIC is used as the Near-RT RIC, and a monitoring xApp collects real-time uplink and downlink throughput metrics. Measured values closely match those reported by application-level tools on the UE, confirming the consistency of E2-based measurements in a real radio environment.

The third experiment combines the previous scenarios by deploying multiple gNBs within the same network, some serving emulated UEs and others serving COTS UEs (see Fig. 9). This hybrid architecture demonstrates the ability of the testbed to support heterogeneous RAN deployments and multi-gNB monitoring through a unified RIC framework.

Overall, Use Case 3 shows that meaningful O-RAN experimentation can be achieved on a single commodity host and extended toward more distributed laboratory deployments. These results should be interpreted as evidence of practical feasibility for experimentation, rather than as proof of carrier-grade scalability. The availability of mature RIC implementations and reusable xApps further reinforces the suitability of this platform for advanced experimentation, education, and applied research.

6 Conclusions and Future Work

This article presents a structured theoretical and experimental study of Open Radio Access Networks aimed at lowering the entry barrier to academic and small-scale O-RAN research. By consolidating dispersed knowledge on O-RAN motivations, architecture, components, and interfaces, and by comparatively evaluating a set of representative testbeds, this work provides a bridge from conceptual understanding to hands-on experimentation. This work does not aim to provide a carrier-grade benchmark of O-RAN performance. Instead, it shows that meaningful and reproducible O-RAN experimentation—covering simulated, hybrid, automated, and partially real deployments—can be conducted using open-source tools and modest laboratory hardware. The main outcome of the study is therefore methodological and practical: a structured and experimentally grounded guide to representative O-RAN platforms, their trade-offs, and their suitability for different research needs. The demonstrative use cases further show that Near-RT RIC-based monitoring and control can be exercised in reproducible laboratory conditions, while also highlighting practical limitations related to software version sensitivity, implementation divergence, and the operational instability observed in some optional components.

Looking forward, the platforms and configurations presented in this work open several directions for future research. A natural continuation lies in the design, implementation, and evaluation of advanced xApps and rApps, including closed-loop control mechanisms and AI/ML-based optimization strategies. Further efforts are also needed to integrate and validate SMO and Non-RT RIC functionalities as these components evolve. Beyond algorithmic development, future work may explore larger-scale and more heterogeneous laboratory deployments, the interaction between O-RAN and

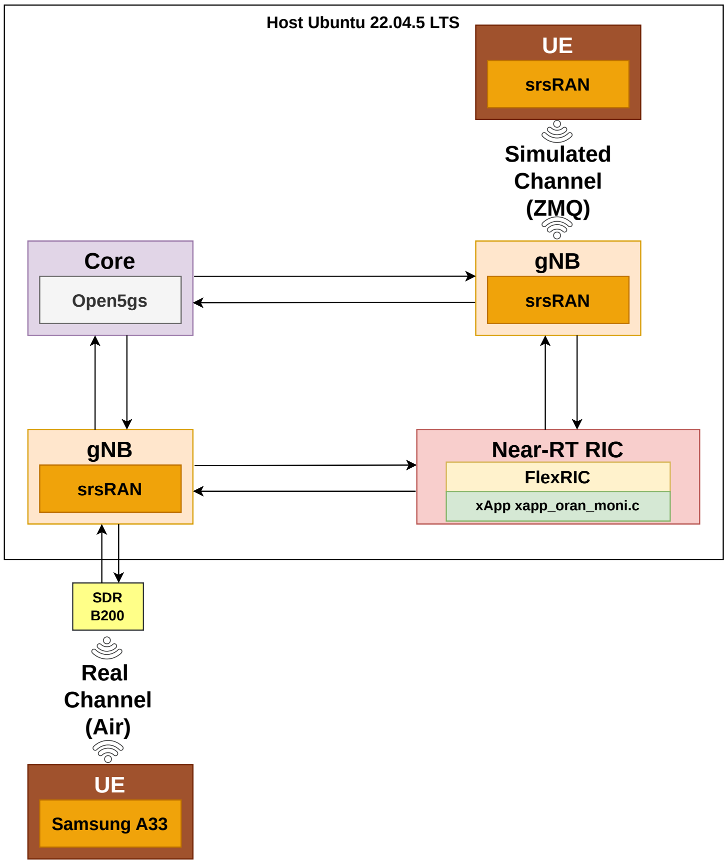


Fig. 9: Architecture of Experiment 3, featuring two gNBs: one serving COTS UEs and the other serving emulated UEs.

optical/transport networks, and the incorporation of these platforms into academic training and experimental testbeds. In this sense, the present study aims to serve as a practical reference for subsequent experimental, educational, and applied O-RAN research.

Acknowledgements. This work was funded by the "Intelligent Management of Optical-Mobile Networks through Disaggregated Architectures and Digital Twins" research project (ANII FMV_1_2025_1_186855) and the "L'Oréal–UNESCO" WIS program.

Declarations

- Funding This work was supported by the Research Project *Intelligent Management of Optical-Mobile Networks through Disaggregated Architectures and Digital Twins* (ANII FMV_1_2025_1_186855) and by the L'Oréal–UNESCO program.
- Conflict of interest/Competing interests The authors declare that they have no competing interests.
- Ethics approval and consent to participate Not applicable
- Consent for publication Not applicable
- Data availability All configuration files, experimental scripts, and results generated during this study are publicly available at: https://gitlab.fing.edu.uy/mobile-optical-networks/testbed/mobile-network.
- Materials availability Not applicable
- Code availability All configuration files, experimental scripts, and results generated during this study are publicly available at: https://gitlab.fing.edu.uy/mobile-optical-networks/testbed/mobile-network.

References

[1] Navarro, J., Castro, A., Rattaro, C.: O-ran testbeds: From experimental deployments to simulation frameworks. In: 2025 13th Wireless Days Conference (WD), pp. 1–7 (2025). https://doi.org/10.1109/WD67713.2025.11302581

[2] O-RAN Alliance: Especificaciones de los grupos de trabajo de la O-RAN Alliance. https://specifications.o-ran.org/specifications. Accessed: 2024-05-20 (2024)

[3] Rouwet, W.: Open Radio Access Network (O-RAN) Systems Architecture and Design, 1st edn. Academic Press, Elsevier, Londres, UK (2022)

[4] Wong, I.C., Chopra, A., Rajagopal, S., Jana, R.: Open RAN: The Definitive Guide, 1st edn. Wiley, New Jersey, USA (2024)

[5] Polese, M., Bonati, L., D'Oro, S., Basagni, S., Melodia, T.: Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys and Tutorials **25**(2), 1376–1411 (2023)

[6] Simona Marinova, A.L.-G.: Intelligent o-ran beyond 5g: Architecture, use cases, challenges, and opportunities. IEEE Access **12**, 27088–27114 (2024)

[7] Huang, Y., Sun, Q., Z, N.L., Huang, J., H. Ding, C.L.: Validation of current o-ran technologies and insights on the future evolution. IEEE **42**(2), 487–505 (2024)

[8] Krasniqi, X., Hajrizi, E., Qehaja, B.: Challenges and lessons learned during private 5g open ran deployments. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), pp. 1–6 (2023). https://doi.org/10.1109/ICECCME57830.2023.10252662

[9] Villa, D., Khan, I., Kaltenberger, F., Hedberg, N., Da Silva, R.S., Kelkar, A., Dick, C., Basagni, S., Jornet, J.M., Melodia, T., Polese, M., Koutsonikolas, D.: An open, programmable, multi-vendor 5g o-ran testbed with nvidia arc and openairinterface. In: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6 (2024). https://doi.org/10.1109/INFOCOMWKSHPS61880.2024.10620908

[10] Bahl, P., Balkwill, M., Foukas, X., Kalia, A., Kim, D., Kotaru, M., Lai, Z., Mehrotra, S., Radunovic, B., Saroiu, S., Settle, C., Verma, A., Wolman, A., Yan, F.Y., Zhang, Y.: Accelerating open ran research through an enterprise-scale 5g testbed. In: Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. ACM MobiCom '23. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3570361.3615745 . https://doi.org/10.1145/3570361.3615745

[11] Ferguson, A.E., Pawar, U., Wang, T., Marina, M.K.: Campus5G: A Campus Scale Private 5G Open RAN Testbed. arXiv preprint arXiv:2506.23740 (2025)

[12] PC, A., Vignesh, N., Pei, Y., Venkatarayalu, N., Rezende, P.H.A., Mahato, S.B., Sun, S.: FCT O-RAN: Design and Deployment of a Multi-Vendor End-to-End Private 5G Testbed. arXiv preprint arXiv:2509.01891 (2025)

[13] Upadhyaya, P.S., Tripathi, N., Gaeddert, J., Reed, J.H.: Open ai cellular (oaic): An open source 5g o-ran testbed for design and testing of ai-based ran management algorithms. IEEE Network **37**(5), 7–15 (2023) https://doi.org/10.1109/MNET.2023.3320933

[14] Upadhyaya, P.S., Abdalla, A.S., Marojevic, V., Reed, J.H., Shah, V.K.: Prototyping Next-Generation O-RAN Research Testbeds with SDRs (2022). https://arxiv.org/abs/2205.13178

[15] Chu, Y., Rahmani, M., Shackleton, J., Grace, D., Cumanan, K., Ahmadi, H., Burr, A.: Testbed development: An intelligent o-ran-based cell-free mimo network. IEEE Communications Magazine **63**(6), 74–81 (2025) https://doi.org/10.1109/MCOM.001.2400574 . Licensed under CC BY 4.0

[16] Parada, R., Vilajosana, X., Alfayoumi, S., Serra, J., Font-Bach, O., Dini, P.: An open testbed for o-ran experimentation with ai-enabled control and monitoring. Internet of Things **33**, 101729 (2025) https://doi.org/10.1016/j.iot.2025.101729

[17] Liu, P., Lee, K., Cintron, F., Wuthier, S., Savaliya, B., Montgomery, D., Rouil, R.: Blueprint for Deploying 5G O-RAN Testbeds: A Guide to Using Diverse O-RAN Software Stacks. Technical Note (NIST TN), National Institute of Standards and Technology, Gaithersburg, MD (2024). https://doi.org/10.6028/NIST.TN.2311 . https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=958753

[18] Maxenti, S., Shirkhani, R., Elkael, M., Bonati, L., D'Oro, S., Melodia, T., Polese,

M.: AutoRAN: Automated and Zero-Touch Open RAN Systems. arXiv preprint arXiv:2504.11233 (2025)

[19] Garey, W., Rouil, R.A., Black, E., Ropitault, T., Gao, W.: O-RAN with Machine Learning in ns-3. Workshop on ns-3 (WNS3) 2023, Ballston, VA, US (2023). https://doi.org/10.1145/3592149.3592157 . https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936291

[20] Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B.S., Singh, T., Zugno, T., Cuomo, F., Melodia, T.: Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures. IEEE Transactions on Mobile Computing, 1–16 (2023) https://doi.org/10.1109/TMC.2023.3266642

[21] Moore, J., Abdalla, A.S., Ueltschey, C., Marojevic, V.: Prototyping O-RAN Enabled UAV Experimentation for the AERPAW Testbed (2024). https://arxiv.org/abs/2411.04027

[22] Do, T.N.: AI-Open-RAN for Non-Terrestrial Networks. arXiv preprint arXiv:2511.11947 (2025)

[23] Lacatus, F.-F., Siller, M., Mastri, L., Gay, N., Vargas-Schaffer, G., Kaddoum, G.: Development of Open Radio Access Networks (O-RAN) for Real-Time Robotic Teleoperation. arXiv preprint arXiv:2502.14728 (2025)

[24] Luis Herrera, J., Montebugnoli, S., Scotece, D., Foschini, L., Bellavista, P.: A tutorial on o-ran deployment solutions for 5g: From simulation to emulated and real testbeds. IEEE Communications Surveys & Tutorials **28**, 1709–1748 (2026) https://doi.org/10.1109/COMST.2025.3577687

[25] Guemdani, M., Gu, P., Phung, C.-D., Secci, S.: Comparative e2e performance analysis of o-ran designs in a 5g standalone testbed. In: 2025 IEEE 26th International Conference on High Performance Switching and Routing (HPSR), pp. 1–6 (2025). https://doi.org/10.1109/HPSR64165.2025.11038873

[26] Polese, M., Bonati, L., D'Oro, S., Basagni, S., Melodia, T.: ColO-RAN: Developing Machine Learning-based xApps for Open RAN Closed-loop Control on Programmable Experimental Platforms. IEEE Transactions on Mobile Computing, 1–14 (2022)

[27] Shin, J., Yoo, H.-M., Choi, H., Kwon, H.S., Hong, E.-K.: Implementation of o-ran smo and ai/ml framework for closed-loop control via o1 interface. In: 2026 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 1288–1293 (2026). https://doi.org/10.1109/ICAIIC68212.2026.11454193

[28] Schmidt, R., Irazabal, M., Nikaein, N.: Flexric: an sdk for next-generation sd-rans. In: Proceedings of the 17th International Conference on Emerging Networking

EXperiments and Technologies. CoNEXT '21, pp. 411–425. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3485983.3494870 . https://doi.org/10.1145/3485983.3494870

[29] Pedreira, F., Tió, B.: PortaRAN: Implementación de un demostrador portable de red móvil 5G basado en srsRAN. Tesis de grado, Universidad de la República (Uruguay), Facultad de Ingeniería. En español (2025)