OPEN-SOURCE EMULATION-BASED TEST ENVIRONMENT TO SETTLE O-RAN-COMPLIANT TRIALS

Ramon Fontes[1], Allan Martins[1], Vicente Sousa[1], Kaio Dantas[1], Lucas Medeiros[1], Pedro Alves[1], Marcelo Fernandes[1], Iago Rego[1], Eduardo Aranha[1], Vinícius Filho[1], Mateus Goldbarg[1], Wysterlanya Barros[1], Roger Immich[1], Augusto V. Neto[1]
[1]Federal University of Rio Grande do Norte (UFRN), Natal, Brazil

ramon.fontes@imd.ufrn.br, allan@dca.ufrn.br, vicente.sousa@ufrn.br, kaio.henrique@ifrn.edu.br, {lucas.medeiros.114, pedro.alves.700}@ufrn.edu.br, mfernandes@dca.ufrn.br, iago.diogenes.072@ufrn.edu.br, eduardo.aranha@ufrn.br, {vjmtbf, mateus.goldbarg, wysterlanya.barros.016}@ufrn.edu.br, roger@imd.ufrn.br, augusto@dimap.ufrn.br

Abstract – *Experimental tools are a key factor in both academic and industrial research communities to create design evaluations of new networking technologies that involve troubleshooting or changing the planning of deployed networks. Physical Software-Defined Radio (SDR) experimental platforms enable a design solution for the quick prototyping of wireless communication systems. However, SDR-based experimental platforms incur high costs, which leads to scalability limitations in the experimental settings. Having said this, network simulators, emulators, and new testbeds have attracted increasing attention. Emulation-based research prototyping can be distinguished from real communication networks and SDR-based platforms by allowing a tradeoff between cost and flexibility. This paper examines the Mininet-RAN emulation tool, which, as well as Radio Access Network (RAN) modeling, provides a way to test Open RAN Intelligent Controller (RIC) services without the need to deploy an entire RAN infrastructure. The Mininet-RAN creates virtual network elements, such as hosts, L2/L3 devices, controllers, and links, by combining some of the best emulator features, hardware testbeds, and simulators. By running the current code of standard practice Unix/Linux network applications and network stack, the Mininet-RAN enables real-world network data traffic patterns to be delivered to the RIC, regarding the most significant aspect of the dynamic generation of wireless system's KPIs. We provide the basic code of Mininet-RAN for the first two O-RAN Alliance-defined use cases involving V2X and UAV. The xApps are being implemented in O-RAN SC near-RT RIC, with Mininet-RAN which provides a closed-loop validation environment.*

Keywords – Emulation, O-RAN, RIC, UAV, V2X

1. INTRODUCTION

Fifth-generation (5G) mobile networks, and more advanced technologies, have established disruptive innovations to the wireless networking landscape, including vertical services from low-energy/high-dense IoT to delay/reliability-sensitive applications [1]. Apart from this, IEEE-based systems are evolving toward high-spectral efficient 802.11be (Wi-Fi 7), resulting in a significant enhancement of performance of the previous 802.11ax (Wi-Fi 6), as well as in terms of latency.

As a revolutionary and innovative measure, a set of important telecom players created the O-RAN Alliance to promote more democratic and permission-less telecom systems connected via open interfaces and multivendor interoperable components. For many vendors, operators, policy makers, and telecom ecosystem stakeholders, the O-RAN movement is crucial to the future of the 3GPP umbrella system, including 4G, 5G, and Wi-Fi [2]. These kinds of O-RAN stakeholders welcome the potential of this openness initiative to improve competition, network flexibility, cost effectiveness, and features through a centralized Radio Access Network (RAN) Intelligent Controller (RIC) and data-driven closed-loop control [3].

The conceptualization and adoption of new technologies constantly need new experimentation platforms to ensure trials can be carried out with enhanced and agile deployments. However, before researchers can move forward with full-scale projects to drive their commercial deployment, the project can be assessed upon pilot studies supported by experimental platforms, such as network simulation and emulation tools [4]. The reason for this is that the 5G/O-RAN technological system is not an off-the-shelf commercial product and involves the investigation of fundamental aspects (e.g., feasibility, cost, and potential problems).

Researchers have built testbeds that help to better understand and demonstrate the operational capabilities of targeting technologies. However, developing and maintaining a full-scale testbed is expensive and time-consuming. The management of these kinds of testbed settings is complex because of the large physical area required to set up any useful multihop topology. What is worse, it is also difficult to reconfigure such a testbed and even replicate it in other projects.

For this reason, both simulation and emulation tools have been welcomed by researchers in the academic community and in part of the industrial world. While the exact quantification of each characteristic and the degree of realism ultimately depend on the accuracy of the model implemented in each specific tool among other platform aspects that may affect each feature, Table 1 aims to illustrate the main strengths and shortcomings typically common to each type of experimentation approach as a first guide to choose the best type tool for a given set of research goals and constraints.

- *Total cost*: evaluates the cost of experiments, especially those related to hardware and software costs.
- *Overall fidelity*: capacity of transferability of the results, accuracy, conclusions, and the study environment into the real world.
- *Replay real traces*: capacity of replaying observed network behavior, such as, signal strength, throughput, latency, mobility, etc.
- *Real applications*: ability to run real applications without modifying the source code and with no additional effort by the user hand-side.
- *Traffic realism*: assesses the capability of generating, receiving, and processing real traffic.
- *Timing realism*: analyze whether the timing behavior of the system is close to the behavior of deployed hardware.
- *Scalability*: assesses the feasibility of large-scale experiments with respect to the number of nodes, the experiment duration, and the number of network connections during the experiment.
- *Maintainability*: describes the ability to maintain the evaluation environment. In other words, the effort necessary to keep the system runnable.
- *Flexibility*: describes the freedom in creating different experiment scenarios (e.g. network topology, number of nodes, etc).
- *Replication*: how straightforward the repetition of a given experiment in a specific study environment is.
- *Isolation*: assesses the degree to whose links, queues, and switches a network behaves.

The most appropriate approach that can be adopted in experimentally-driven research endeavors is to seek to evaluate the functionality and performance of a network, and this always involves a tradeoff [5]. On the one hand, a simulator enables fully controlled application testing-purpose software environments to be set up quickly and easily, with the hardware layer. On the other hand, an emulator moves further towards close-to-real scenarios by enabling both real-world software and hardware technologies to be set up in an integrated environment.

Even when its advantages are taken into account, designing an O-RAN-compliant emulator to perform RIC experimentation is not a trivial task. The fact is that, up to now, there has been a lack of open-source emulation tools that are accessible to the general public. In order

Table 1 – Ranking of simulators, emulators and testbeds (adapted from [6])

Characteristic	Sims	Emuls	Testbeds
Total Cost	●○○	●○○	●●●
Overall Fidelity	●○○	●●○	●●●
Replay Real Traces	●●○	●●○	●●●
Real Applications	●○○	●●●	●●●
Traffic Realism	●○○	●●●	●●●
Timing Realism	●●●	●●○	●●●
Scalability	●●●	●●○	●○○
Maintainability	●●●	●●●	●○○
Flexibility	●●●	●●●	●○○
Replication	●●●	●●●	●○○
Isolation	●●●	●●○	●●●

to fill this gap, we advance beyond the state of the art by designing a new open-source and easy-to-use emulation framework, which caters to end-to-end O-RAN close-to-real testing at a low cost, along with a moderate learning curve. This kind of proposal, called Mininet-RAN, takes precedence over other tools while an emulation platform, by enabling new O-RAN-compliant features to be implemented, evaluated, and validated atop a trustworthy O-RAN RIC environment. The research contributions of this paper are as follows:

- providing insights into existing simulation and emulation solutions used for a O-RAN RIC experimentation;
- designing an emulation tool capable of settling low-cost evaluation and validation trials of O-RAN RIC scenarios under real network traffic and wireless technology patterns;
- providing a closed-loop experimental use case scenario atop the O-RAN RIC approach;
- making available all the codes, of both Mininet-RAN and use case scenarios, in an open-source repository for global-community access.

The paper is structured as follows. Section 2 provides an overview of the O-RAN RIC technology; Section 3 provides an analysis and ideas of key related work; Section 4 introduces the Mininet-RAN tool, along with the system's architecture; Section 5 features Mininet-RAN and employs two use cases; Section 6 discusses the current limitations of the research and ongoing work, and makes suggestions for future studies. Finally, Section 7 wraps up the paper with some final remarks and encourages the reader to follow the developments in the open-source code repository.

2. OVERVIEW OF O-RAN RIC

Aligned with 3GPP systems, the RIC is a central component of open RAN. The Open RAN has two main

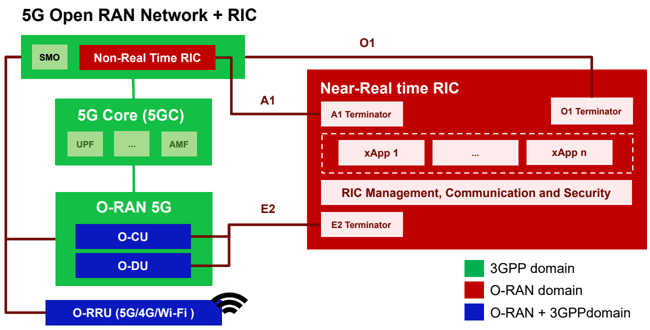


Fig. 1 – O-RAN RIC components
(adapted from https://www.o-ran.org/)

objectives: (i) to provide multivendor interoperability across different hardware and software components in a telecommunication system; and (ii) to create an open ecosystem for developing, deploying, and operating third-party applications (called xApps) to support Radio Resource Management (RRM), higher layer procedure optimization, policy optimization in RAN, and provide guidance, parameters, policies, and Artificial Intelligence/Machine Learning (AI/ML) models to cut costs, improve QoS, and generate new streams of revenue.

To unlock limitless innovation, the O-RAN Alliance, a worldwide community of mobile network operators, vendors, and R&D institutions, defines a reference architecture comprising of the near-Real-Time (near-RT) RIC, non-RT RIC, and A1, E2, and O1 interfaces. Fig. 1 illustrates the O-RAN capable 3GPP systems that exchange Key Performance Indicators (KPIs) with the RIC to evaluate and control 3GPP components.

These RAN control strategies can be carried out in a non-RT control loop (operating on a scale greater than 1000 ms) and near-RT control loop (performing operations between 10 ms and 1000 ms). The xApps are crucial elements of near-RT RIC. As they are a set of microservices, xApps can make intelligent decisions for adapting RAN parameters (e.g., subscriber positioning, handover to a cell, changes to different carrier frequencies) to optimize the subscriber experience and network performance. Thus, RIC xApps could request, via the E2 interface, KPIs from RAN, then send back control actions across the E2 interface to the RAN.

Unlike the traditional RAN evolutionary approach, a system architecture with the RIC is flexible enough to extend the RRM actions by implementing third-party microservices without changing the RAN implementation regardless of its brand. Thus, through the O-RAN set of protocols, a proprietary RAN that supports the RIC has a two-way communication channel through which the RIC can measure and control the RAN transmission functionality. The novelty is that anyone can provide those third-party implementations by xApps and not only the RAN proprietary companies.

With the supplementary operation of both near-RT and non-RT RIC loops, complete ML-powered management can be deployed, which covers optimization actions from lower to upper layers protocols. They may range from short-term MAC-layer procedures (e.g., scheduling) to long-term network decisions like network slice assurance to provide end-to-end connectivity and quality of service. While near-RT RIC monitors and rapidly regulates the RAN actions based on ML models, the non-RT RIC ensures the best-trained ML models and provides suitable policies to guide the operational performance of near-RT RIC functions. More specifically, the non-RT RIC takes advantage of the Service Management and Orchestration (SMO) framework to provide policy-based guidance, strategies for model management, and enrichment information to the near-RT RIC function in response to specialized applications called rApps.

3. RELATED WORK

This section gives an overview of the most significant footprints in network experimentation platforms, and highlights the main features to obtain a better understanding of the tradeoffs when seeking to support realistic O-RAN experimentation. It is worth noting that a few third-party tools are available for purchase, such as the VIAVI's TeraVM[1] and the Keysight's P8828S[2], which are solutions that emulate O-RAN Alliance WG3-standardized E2 node types for setting up RIC test trials in different scales. However, as these third-party solutions are very expensive, they are beyond the scope of this paper, which focuses on low-cost and open-source endeavors.

Table 2 depicts the list of the most relevant 5G and O-RAN-tailored open-source simulator and emulator initiatives we found in the literature. For each tool, we raise aspects ranging from physical layer modeling to End-to-End (E2E) evaluation, along with software licensing, the last code update, and the availability of the O-RAN E2-interface highlights.

The literature reveals a few footprints of open-source tools capable of emulating the O-RAN E2-interface for E2E evaluation. Three projects provide both functional software and an emulator of O-RAN-ready RAN: O-RAN SC (from O-RAN Alliance), SD-RAN (from Open Networking Foundation), and Open Air Interface (EURECOM).

Furthermore, the sim-e2-interface O-RAN SC denotes an E2-interface agent, which only seeks to test the E2AP protocol message exchange. Two improvement initiatives to feed RAN-related KPIs into the sim-e2-interface are ongoing. The first is a collaboration with VIAVE company (alternative to proprietary software), while the interface for the ns-3 simulator is the low-cost alternative that involves having a KPI-based closed-loop with the O-RAN SC emulator.

[1] http://rb.gy/qahxk6
[2] http://rb.gy/p56gft

Table 2 – Summary of open-source simulators/emulators for O-RAN end-to-end (E2E) performance evaluation

Experimentation Platform	Complete PHY abstraction (channel and error models)	E2E Evaluation	O-RAN Interfacing	Open Source	Last Update
OAI L1 RF Simulator [7]	only RF (radio channel simulator)	yes, with OAI infra	no	OAI Public License V1.1	Aug, 2021
OAI L2 nFAPI Simulator [7]	using OAI L1 RF simulation	yes, with OAI Infra	no	OAI Public License V1.1	Aug, 2022
simu5G [8]	yes	user plane only control plane not modeled [9]	no	LGPL	Oct, 2022
UERANSIM [10]	no radio protocols below the RRC layer	no	no	GPL-3.0	Oct, 2022
5G Lena ns-3 [11]	yes, without handover and mobility (NSA only) [12]	yes, simulation (no real time)	no	GNU GPLv2	Nov, 2022
FikoRE [13]	focus on application-level experimentation and prototyping	yes (VR/AR devices in real-time)	no	BSD-3-Clause-Clear license	Feb, 2023
O-RAN SC sim-e2-interface [14]	based on VIAVE's simulator dataset [15]	yes, with O-RAN SC 5G core	yes	Apache 2.0	May, 2022
SD-RAN RAN Simulator [16]	only an E2 agent using E2AP	yes, with USRP	yes	ONF Member-Only	Sep, 2022
mmWave ns-3 [17]	yes (NSA only)	yes, simulation (no real time)	yes	GNU GPLv2	Oct, 2022
Mininet-RAN	yes	yes	yes	GNU GPLv2	Feb, 2023

The SD-RAN provides another option which is to have an E2 agent for E2AP, but an E2E option is only possible by means of OAI tools. Although it includes the SDR-based code of 4G and 5G RANs, the OAI emulates a gNB with an nFAPI emulator. As introduced in [18], the nFAPI defines a network protocol that is used to connect a Physical Network Function (PNF) running a PHY layer (Layer 1) to a Virtual Network Function (VNF) running MAC (Layer 2) and above.

On the basis of the information in this section, none of the related solutions described above can cater for highly-accurate wraparound test settings for an O-RAN-compliant system workflow. This lack of solution is a motivation factor in our task in designing a low-cost and open-source tool to create a controllable testbed that can help to exercise and evaluate O-RAN-compliant features quickly, as well as seeking to simplify the task of validating the new innovations before they are introduced to the market.

Thus, this paper provides an O-RAN compatible experimentation platform which includes a complete wireless PHY abstraction (e.g., RF, modulation, channel coding, and antenna aspects), several MAC layer features (e.g., link adaptation, error detection, multiple access), as well as real-time end-to-end service delivery (e.g., video streaming).

4. MININET-RAN DESIGN AND WORKFLOW

This section provides a general description of the proposed solution, in terms of architecture, functionalities, and experimental testbed capabilities. As explained in the previous sections, the O-RAN RIC can run xApps that act as microservices with very specific tasks. In order to test an application comprising a set of xApps, it is necessary to have a means of providing network information (e.g., KPIs) and being able to react with actions (e.g., performing handover) and, more importantly, ensuring that those actions can be reflected in the new measurements. This closed-loop scenario is not trivial and needs some simulation/emulation in real time. For the purposes of this work, we propose an architecture together with a particular use case that fulfills this requirement. Fig. 2 shows the complete O-RAN RIC that is running with Mininet-RAN.

Fig. 2 shows the proposed near-RT RIC architecture. The gray area shows the near-RT RIC connected to both the Service Management and Orchestration (SMO), which includes the non-RT RIC, and the E2-related blocks. The near-RT RIC includes all xApps and the remaining infrastructure is shown only as a reference. From that infrastructure, the two main parts are the Messaging Infrastructure (RMR) and Shared Data Layer (SDL). The proposed architecture comprises four xApps that run on the near-RT RIC and two external applications.

The xApp highlighted in blue interfaces with the following external elements: (i) Mininet-RAN to collect RAN-related information; and (ii) the dashboard to display data to a browser using a Javascript web app.

As stated before, each xApp runs as a specialized microservice. Their goals and lifecycles are outlined as follows:

- **xApp KPI Monitor**:
  - goal: to constantly request KPIs from the xApp Interface and update the SDL database with them.

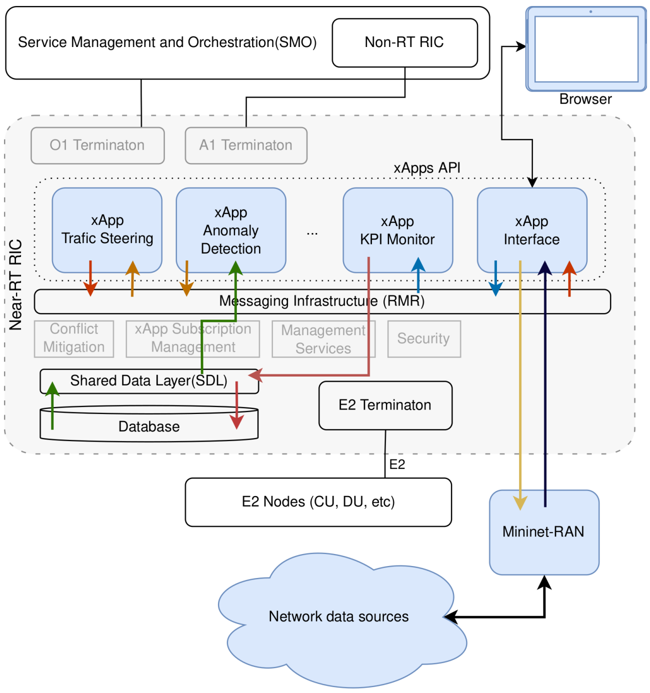


Fig. 2 – Proposed architecture for the testbed

- lifecycle:
    - ∗ Request KPIs via RMR from the xApp Interface;
    - ∗ serialize KPIs and store them in the KEY-VALUE pair in the SDL;
    - ∗ store the KPI data on SDL;
    - ∗ repeat.

- **xApp Anomaly Detection**:
  - goal: to evaluate the KPIs in the SDL based on the trained ML algorithm. The result of this evaluation should be reflected in some action to be sent via RMR.
  - lifecycle:
    - ∗ Read KPIs from the SDL;
    - ∗ process the KPI, generating an action according to the output of the ML algorithm;
    - ∗ send this action via the RMR to whoever is subscribed (normally the xApp Traffic Steering).

- **xApp Traffic Steering**
  - goal: to listen for actions that need to be taken and perform them if necessary via xApp Interface.
  - lifecycle:
    - ∗ Listen to the RMR for actions to be performed (normally from the xApp Anomaly Detection;
    - ∗ check or filter the action and decide its significance on the basis of an algorithm;
    - ∗ send the action to the xApp Interface via the RMR.

- **xApp Interface**
  - goal: to serve as an interface with the external software. It communicates with the Mininet-RAN to either ask for KPIs or it carries out actions on the emulator. It also supplies the external dashboard with network and RIC data.
  - lifecycle:
    - ∗ Listen for RMR calls to access Mininet-RAN KPI endpoints when requested;
    - ∗ listen for RMR calls to access Mininet-RAN actions endpoints when requested.

4.1 Architecture and implementation

Being a network emulator, Mininet-RAN is able to run with virtually all (if not all) tools, programming languages, and other resources supported by Linux systems. In our implementation, network KPIs are collected with the support of Scapy, a packet manipulation library written in Python, which listens for predefined KPIs by the user. The communication between Mininet-RAN and the near-RT RIC is performed by REST API. The API has basically two main endpoints: `/api/v1/getKPI` and `/api/v1/performHandover`. The former is performed via `GET` (with no parameters), and it returns the current KPI measurement, whereas the latter is performed through the `POST` method. The same approach was adopted to communicate near-RT RIC and the external dashboard.

4.2 Creating a network

The network topology script is one of the most important artifacts of Mininet-RAN. To start Mininet-RAN, a Python script is required to bring up the desired network topology. As a result, nodes may be created and, customized, while its services are instantiated. Currently, Mininet-RAN supports wireless technologies such as IEEE 802.11, particularly IEEE 802.11p, as well as IEEE 802.15.4. We have also started a prototype with the wwan_hwsim module[3], a Linux Kernel module that we plan to provide and expect that it will allow extensive experiments for LTE and 5G NR settings.

4.3 Network customization and user interaction

Mininet-RAN supports all the commands related to Linux system wireless tools such as `iw`, `iwpan`, and others,

[3]`https://github.com/torvalds/linux/blob/master/drivers/net/wwan/wwan_hwsim.c`

some of them well-known by network emulators such as Mininet [19] and Mininet-WiFi [20]. For example, the user may use `node1 iw dev node1-wlan0 scan` to scan for available base stations (BSs), and `node1 iw dev node1-wlan0 connect ssid` to connect to a selected base station with the same SSID. Since it is a runtime emulator Mininet-RAN allows users to add new network features and elements to ensure a more versatile experiment. To verify the connectivity between virtual devices such as nodes, the user can type the CLI command: `node1 ping node2`. Common Linux commands can be executed by the user at runtime, for instance, to check the available bandwidth between two nodes (node1 and node2): `node1 iperf -s & node2 iperf -c 10.0.0.1`. As we will see from the next section, Mininet-RAN's versatility makes it possible for this emulator to interoperate with well-known simulators in the community, such as SUMO and CoppeliaSim, bringing more realism to the development of case studies.

5. CASE STUDIES

This section features Mininet-RAN and employs two use cases defined by the O-RAN Alliance: (i) the context-based dynamic handover management for V2X; (ii) and the flight path-based dynamic UAV resource allocation. In the context of 5G, we expect Mininet-RAN to offer more significant and complementary advantages than simulation or testbed-based experimental approaches. The source code repository including artifacts as well as reproducible results is available at https://github.com/mininet-ran/mininet-ran.

5.1 Use case 1: Context-based dynamic handover management for V2X

5G NR V2X has been introduced as a standard by 3GPP in Release 16 with advanced functionalities on top of the 5G NR air interface to support connected and autonomous driving use cases with stringent requirements. It promises numerous benefits such as increased road safety and, reduced emissions, as well as saving time by orchestrating the traffic and assisting individual user decisions based on real-time information on the road and traffic conditions, driver intentions, and other factors [21].

This case study explores context-based dynamic handover management for a V2X scenario. This problem becomes exceptionally challenging when vehicles experience frequent handovers. Owing to their high speed and the heterogeneous nature of the wireless environment, vehicles might be handed over frequently or in suboptimal ways, which may cause handover anomalies such as short stay, the ping-pong effect, and poor radio conditions of the remote cell.

This case study aims to show how versatile Mininet-RAN can be through its integration with O-RAN's near-RT RIC. We leveraged Mininet-RAN by introducing a communication path to the near-RT RIC, to generate and share a list of KPIs to drive ML-supported proactive smart handover decisions. More specifically, we use the Open Source Simulation of Urban Mobility (SUMO) [22], as illustrated in Fig. 3, to simulate the mobility of vehicles and components of a vehicular network, such as traffic lights and vehicle speed.

The context-based dynamic handover management for V2X proposed here works as follows:

- While vehicles move around, BSs continuously transmit messages to the near-RT RIC. The messages contain the Received Signal Strength Indication (RSSI) picked up by the base stations;
- the near-RT RIC receives and processes the messages sent by the BSs;
- the near-RT RIC sends control messages to the base stations which instruct vehicles when they have to roam to a BS that provides better signal quality.

We now showcase the first set of results and discussions, especially those related to ML methods applied to this use case, followed by the pattern of behavior of the monitored vehicle using a dashboard.

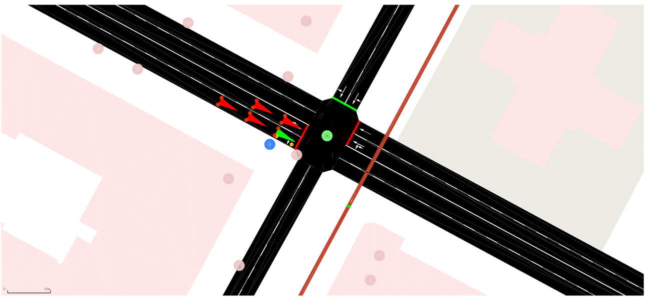


Fig. 3 – Vehicular simulation on SUMO

5.1.1 Machine learning of the use case 1

The dataset used to train the ML in this use case has 900 samples, each consisting of the Signal-to-Noise Ratio (SNR) values of three neighbor BSs (`BS1`, `BS2`, `BS3`), the current connected BS, and a timestamp. The dataset is divided into four columns: `SNR1`, with values referring to the SNR of `BS1`; `SNR2`, with values referring to the SNR of `BS2`; `SNR3`, with values referring to the SNR of `BS3`; and `Current_BS`, with the indication of the current connected BS. Since the goal is to predict the BS switching, the output of each sample should refer to the `Current_BS` of the following sample. This output is encoded using the One Hot Encoding technique. It is worth mentioning that the purpose of this use case is to demonstrate the versatility of Mininet-RAN for testing an ML-based near-RT RIC without presenting a deep performance study about the proposed handover.

Our ML approach is a Multilayer Perceptron (MLP) with four inputs (`SNR1`, `SNR2`, `SNR3`, and `Current_BS`) and three outputs (the probability of changing the connection for `BS1`, `BS2`, and `BS3`). The preprocessed dataset is split for training and testing the MLP model, with sets containing 70% and 30% of the data, respectively.

The MLP architecture is defined by the GridSearchCV function, which employs hyperparameter tuning to determine the optimal values for a given model. In addition, this technique evaluates the model for each combination using the k-fold cross-validation method. We analyzed the performance of the MLP architecture with one or two hidden layers containing 10 to 50 neurons in each layer. While the hidden layers adopt Relu as the activation function, the output layer has Softmax and shows the switching probability of each BS as the output.

The best result from GridSearchCV is for an MLP with two hidden layers of 40 and 20 neurons in the first and second layers, respectively. The k-fold cross-validation with five folds provides 99.84% of model accuracy as illustrated in the learning curve of Fig. 4. The figure also shows a similar degree of accuracy for the test data, which suggests that the model is not overfitting and can be generalized well for new data, i.e., it has learned to trace the underlying patterns in the data and can make accurate predictions about unseen data.

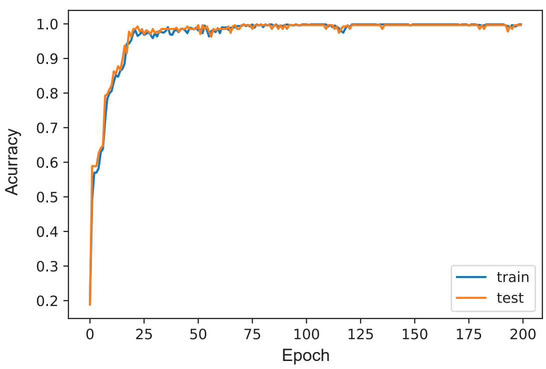
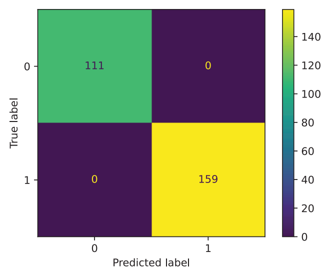
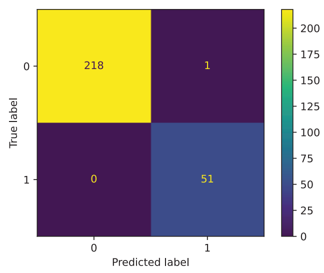


Fig. 4 – Learning curve

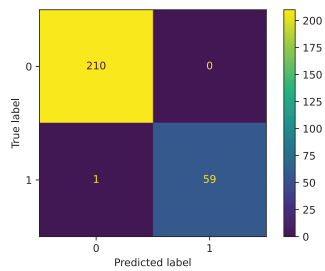
The confusion matrices of the trained MLP are shown in Fig. 5 for the three BSs. According to the results, ML made only two mistakes, which involved `BS2` when it was not the best option and fails to involve `BS3` when it was the best choice. Figures 5b and 5c illustrate those errors, respectively.



(a) BS1



(b) BS2

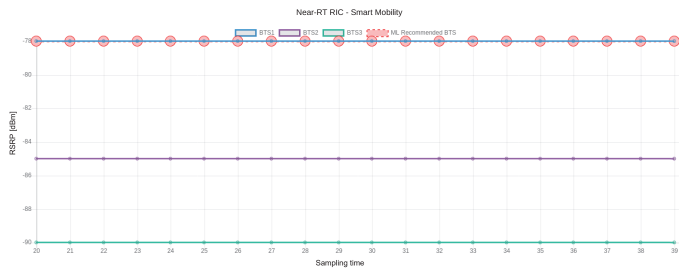


(c) BS3

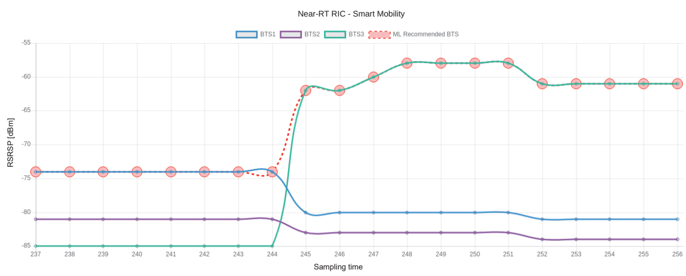
Fig. 5 – Confusion matrices

5.1.2 Results of use case 1

As illustrated in Fig. 6, the vehicle dashboard shows a history of the KPIs' measurement from the three nearest BSs of the monitored vehicle. The three solid lines represent a history for each BS SNR level sensed by the vehicle, while the dotted line with a circle marker makes clear which BS is recommended by the ML to best serve the vehicle.



(a) Vehicle 1 connected to BTS1.



(b) Vehicle 1 connected to BTS3.

Fig. 6 – Near-RT RIC dashboard monitoring information

To illustrate how the dashboard of this use case works, the emulation started with a vehicle connected to `BS1`, as shown in Fig. 6a. The vehicle also obtains a stream of a high-resolution video. As the vehicle moves along the route established in the SUMO, the near-RT RIC keeps measuring the available KPIs and feeding the xApp Anomaly detection, resulting in a handover action sent to the vehicle. Fig. 6b illustrates the moment a handover is recommended as the BS3 SNR level gets better than the BS1 to which the vehicle is connected. The handover decision is not made immediately after a new BS KPI has improved, a necessary measure to avoid the ping-pong effect between BSs. A difference can be noticed between the experiment both with and without the near-RT RIC optimization in Fig. 7a, as illustrated in Fig. 7b.

(a) Without near-RT RIC solution enforcement

(b) With near-RT RIC solution enforcement

Fig. 7 – Quality of videos

5.2 Use case 2: Flight path-based dynamic UAV resource allocation

Unmanned Aerial Vehicles (UAVs) are expected to be an integral part of the upcoming wireless networks, by potentially facilitating wireless broadcasting and supporting high-speed transmissions. Compared to communications with fixed infrastructures, UAVs face new challenges because of their high altitude above the ground and great flexibility of movement in three-Dimensional (3D) space. Some critical issues include Line-of-Sight (LoS) dominant UAV-ground channels, the distinct communication Quality of Service (QoS) requirements for UAV control messages versus payload data, the stringent constraints imposed by the Size, Weight, and Power (SWAP) limitations of UAVs [23], to name a few.

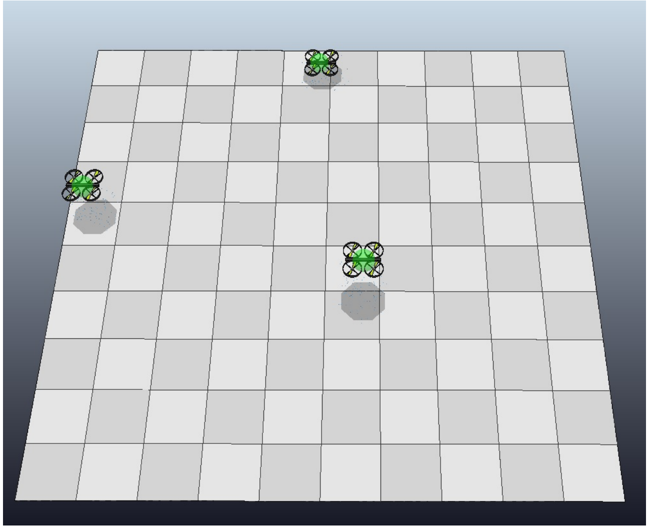


Fig. 8 – UAVs on CoppeliaSim

The application of UAV has played a significant role in civil applications and is rapidly expanding [24], including agricultural plant protection, power inspection, police enforcement, geological exploration, and environmental monitoring. In this case study, the near-RT RIC makes a radio resource allocation for on-demand coverage of UAVs, that takes account of the radio channel conditions, flight path information, and other information about applications.

For this second use case, we ran Mininet-RAN with the robotics simulator CoppeliaSim[4] in a UAV flight path scenario which is useful for setting the UAV paths (see Fig. 8). The network topology consists of four BSs on land and three UAVs in the air. Each BS on land has its own predefined and static channel. UAVs also act as the BS but with a capacity to dynamically adjust their channel by receiving near-RT RIC control commands, avoiding channel overlapping among them and BSs on land. Fig. 9 illustrates the simulation deployment scenario, which comprises a Mininet-RAN topology with four BSs transmitting signals on channels 1, 6, and 11, and three UAVs on channel 1.

During the experiment it is expected to observe considerable interference between BS1 and BS4 as well as among UAVs because they are transmitting signals on the same channel, as illustrated in Fig. 10. As a result of the occupation of the same channel by multiple BSs on land and UAVs in the air, the communication using this channel will be impaired owing to the high level of interference.

Our ML-powered xApps mitigate this problem by dynamically defining the UAV channels. We trained a model that can select the highest channel number

[4] https://www.coppeliarobotics.com/

distance between BSs and UAVs, dynamically allocating UAVs' channels while they fly around. It should be noted that this use case is not aiming to prove the ML algorithm efficiency but to demonstrate Mininet-RAN and near-RT RIC interaction capabilities.

From now on, we will concentrate on the second set of results, and discuss the ML methods applied to the second use case, followed by the behavior of the monitored UAVs by means of the implemented dashboard.

5.2.1 *Machine learning of use case 2*

The dataset used to train the ML for this use case has approximately 48000 samples, each consisting of the connection between one of three UAVs (`UAV1`, `UAV2` and `UAV3`) and one of four different sensors (`S1`, `S2`, `S3` and `S4`) attached to the BSs on land (`BS1`, `BS2`, `BS3` and `BS4`).

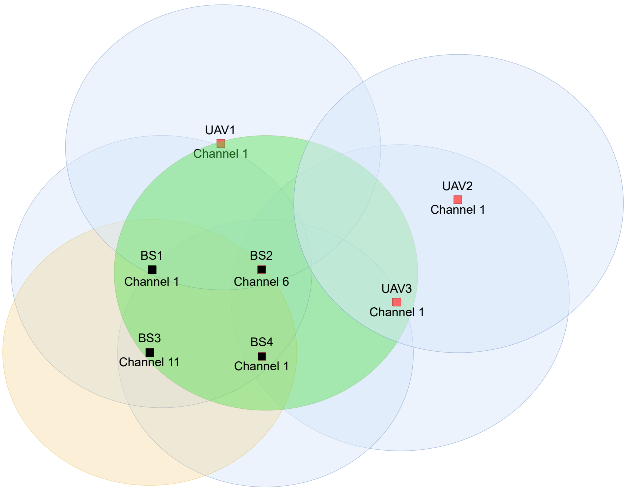


Fig. 9 – Network topology

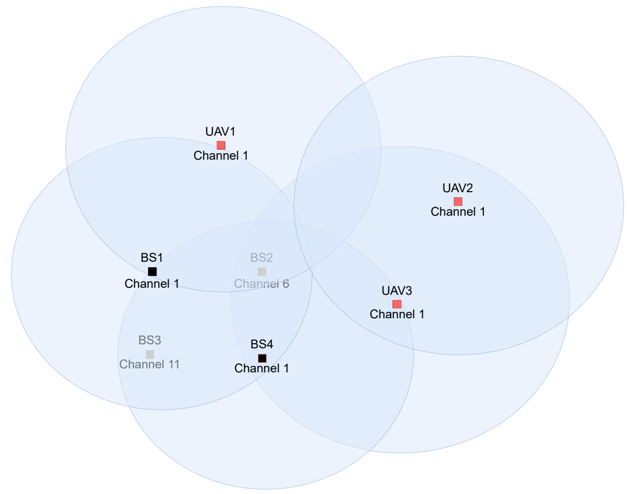


Fig. 10 – Signal overlapping illustration

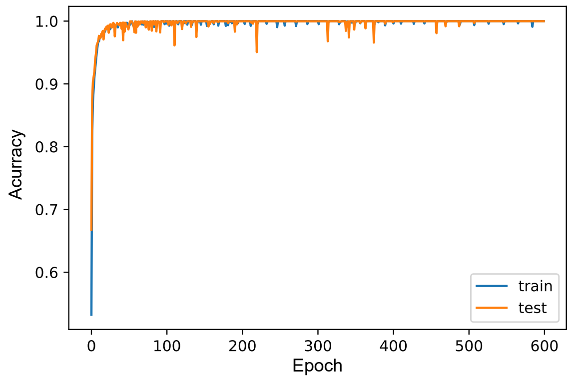


Fig. 11 – Learning curve

These sensors could measure the RSSI from UAVs and neighboring BSs.

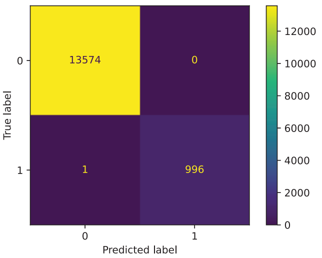
Since the primary goal of the ML for this use case is to select the best channel for each UAV, it is necessary to measure the distance between the channel number used by the BS on land and the UAVs. The channel number ranges from one to twelve, and the higher the channel number distance between them, the lower the interference is, hence, the better the UAVs' connection. All of this information is arranged in the dataset for each instant of time.

The dataset is divided into 12 columns as follows: `Timestamp`: responsible for indicating the instant of time that the data is collected; `Tx_BS_S`: refers to the BSs; `Rx_BS_S`: represents the sensor at BSs; `Channel_BS_S`: indicates the channel used by that BS; `Tx_DR_S`: refers to the UAV under analysis; `Rx_DR_S`: represents the UAV channel; `Power_DR_S`: this is the RSSI sensed by the BS sensor from that UAV; `Old_channel_DR_S`: indicates the current UAVs' channel; `Old_channel_DR_1`, `Old_channel_DR_2` and `Old_channel_DR_3`: represent, respectively, the channel where `DR1`, `DR2` and `DR3` are currently transmitting except for the UAV under analysis; `New_channel_DR_S`: represents the output of each sample, which refers to the best channel for the UAV under analysis. This output is encoded using the one hot encoding technique.

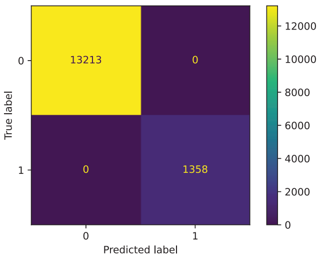
As in use case 1, we also used an MLP with the following configuration: (i) four inputs, values of `Channel_BS_S`, `Old_channel_DR_1`, `Old_channel_DR_2` and `Old_channel_DR_3`; and (ii) seven outputs: the probability of changing the UAVs' channel for `Channel_1`, `Channel_2`, `Channel_3`, `Channel_5`, `Channel_6`, `Channel_11` and `Channel_12`.

The preprocessed dataset is split for training and the testing sets contain 70% and 30% of the data, respectively. Fig. 11 depicts the learning curve of the produced model, which refers to the performance of the training and testing data. As we achieved a similar degree of accuracy for the training and testing sets, the ML model has learned how to adapt to the training data and is likely to perform well with unseen data.

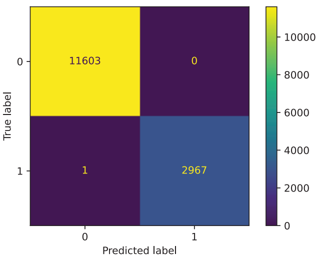
The confusion matrices of the trained MLP are displayed in Fig. 12 for `Channel_1`, `Channel_2`, `Channel_3`, `Channel_5`, `Channel_6`, `Channel_11` and `Channel_12`. According to the results, ML made four mistakes, indicating `Channel_5` and `Channel_12` when they were not the best option and failed to indicate `Channel_1` and `Channel_3` when they were the best choice. Figures 12d, 12g, 12a and 12c depict those errors, respectively.



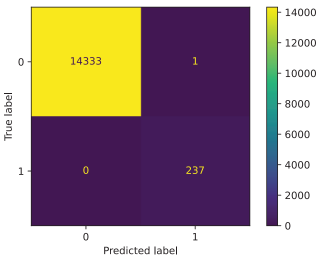
(a) Channel 1



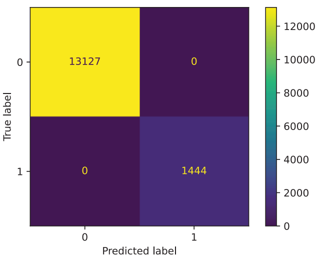
(b) Channel 2



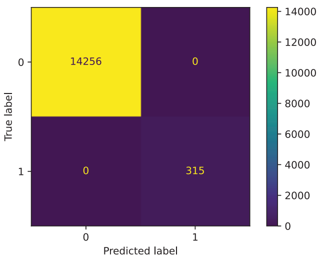
(c) Channel 3



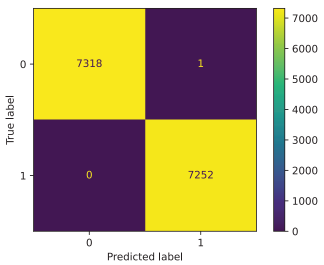
(d) Channel 5



(e) Channel 6



(f) Channel 11



(g) Channel 12

Fig. 12 – Confusion matrices

5.2.2 Results of use case 2

The dashboard designed for this use case shows the actual and the last measured power levels, as well as the channel of each BS and UAV in the emulated scenario.

Table 3 – Near-RT RIC dashboard monitoring information

TX	RX	Old Power	Old Channel	New Power	New Channel
BS1	S1	-80	1	-75	1
UAV1	S1	-50	7	-30	12
BS2	S2	-40	7	-40	7
UAV2	S2	-90	10	-90	10
UAV3	S2	-80	8	-80	8

As the UAVs move along their flight path which was previously defined in the CoppeliaSim, the near-RT RIC keeps measuring KPIs to supply ML-powered xApps. Here, the algorithm can recommend a better distance of the channel number for a UAV, triggering an action to change its channel by the near-RT RIC. As power levels and channel numbers change, the table of the implemented dashboard updates, keeping track of a brief record of the UAV channels and powers. We also could check the BSs power to confirm the interference level changes because of a better channel selection by the UAV. Table 3 displays this information.

6. LIMITATIONS OF THE STUDY AND SUGGESTION FOR FUTURE WORK

Mininet-RAN inherits all the limitations of the Mininet and Mininet-WiFi lightweight virtualization architecture, certainly adding a couple more related to our current approach. It is notorious that Mininet-RAN does not support 5G-related technologies, such as gNodeBs and the typical signaling signals for this network element. The E2 signaling, defined by O-RAN to communicate Mininet-RAN as a suitable O-RAN E2 node, is on our roadmap and will be added to Mininet-RAN soon.

The future of 5G in Mininet-RAN will certainly involve going through the wwan_hwsim Linux Kernel module as well as initiatives such as linux-wpan[5] where some updates are already being made in the Kernel of the Linux operating system, although they are not available in the Kernel main track yet.

7. CONCLUSION

This work presents Mininet-RAN as an emulation platform in support of low-cost 5G and O-RAN open-source experimentation facilities. We evaluated Mininet-RAN from the standpoint of two O-RAN Alliance-defined use cases involving V2X and UAV. In particular, we explored the question of context-based dynamic handover management scenarios for V2X and flight path-based dynamic UAV resource allocation. Although our focus was not on exploring the efficiency of machine learning algorithms nor on proposing new efficient techniques that can provide new results to the scenarios explored in the use cases, the Mininet-RAN

[5]https://linux-wpan.org/documentation.html

proved it was a low-cost and viable alternative for experimenting with new O-RAN-compliant algorithms and techniques. Unlike the related work, we found in the literature that Mininet-RAN is the only solution that achieves a complete PHY abstraction, and allows an E2E evaluation with O-RAN interfacing.

ACKNOWLEDGMENT

This research was partially funded by Lenovo, as part of its R&D investment under Brazilian Informatics Law, and by the *Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior - Brasil* (CAPES) - Finance Code 001.

REFERENCES

[1] Stuart Clayman, Augusto Neto, Fábio Verdi, Sand Correa, Silvio Sampaio, Ilias Sakelariou, Lefteris Mamatas, Rafael Pasquini, Kleber Cardoso, Francesco Tusa, Christian Rothenberg, and Joan Serrat. "The NECOS Approach to End-to-End Cloud-Network Slicing as a Service". In: *IEEE Communications Magazine* 59.3 (2021), pp. 91–97. DOI: 10.1109/MCOM.001.2000702.

[2] Parallel Wireless and Intel Corporation. "Parallel Wireless Creates OpenRAN "ALL G" Radio Access Network Architecture - WhitePaper". In: *Communications Service Providers Radio Access Network* (2022).

[3] Iago Rego, Lucas Medeiros, Pedro Alves, Mateus Goldbarg, Vitor Lopes, Daniel Flor, Wysterlanya Barros, Vinícius Filho, Vicente Sousa, Eduardo Aranha, Allan Martins, Marcelo Fernandes, Ramon Fontes, and Augusto Neto. "Prototyping near-real time RIC O-RAN xApps for Flexible ML-based Spectrum Sensing". In: *2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. 2022, pp. 137–142. DOI: 10.1109/NFV-SDN56302.2022.9974940.

[4] Iago Diogenes Rego, Lucas Medeiros, Vitor Lopes, Vinicius Filho, Vicente Sousa Jr., Eduardo Aranha, Allan Medeiros, Marcelo Fernandes, Ramon dos Reis Fontes, Augusto Venancio Neto, and Guilherme Pupio. "Frequency Scanning xApps: O-RAN RIC and GNU Radio with RTL-SDR use case". In: *XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2022)*. 2022. DOI: 10.14209/sbrt.2022.1570817297.

[5] José Castillo-Lema, Augusto Venâncio Neto, Flávio de Oliveira, and Sergio Takeo Kofuji. "Mininet-NFV: Evolving Mininet with OASIS TOSCA NVF profiles Towards Reproducible NFV Prototyping". In: *2019 IEEE Conference on Network Softwarization (NetSoft)*. 2019, pp. 506–512. DOI: 10.1109/NETSOFT.2019.8806686.

[6] Alexander Zimmermann, Mesut Günes, Martin Wenig, Jan Ritzerfeld, and Ulrich Meis. *Architecture of the Hybrid MCG-mesh Testbed*. Los Angeles, CA, USA, 2006. DOI: 10.1145/1160987.1161004. URL: http://doi.acm.org/10.1145/1160987.1161004.

[7] Open Air Interface 5G RAN PROJECT GROUP. *OAI 5G RAN PROJECT GROUP web site*. URL: https://openairinterface.org/oai-5g-ran-project/.

[8] Intel and University of Pisa. *Simu5G: simulator for 5G New Radio networks*. URL: https://docs.sd-ran.org/master/ran-simulator/README.html.

[9] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. "Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks". In: *IEEE Access* 8 (Sept. 2020), pp. 181176–181191. DOI: 10.1109/ACCESS.2020.3028550.

[10] Ali Gungor. *UERANSIM - GitHub*. URL: https://github.com/aligungr/UERANSIM.

[11] Mobile Networks group of a public research institute, CTTC (Centre Tecnològic de Telecomunicacions de Catalunya). *5G-LENA simulator*. URL: https://5g-lena.cttc.es/.

[12] Natale Patriciello, Sandra Lagén, Biljana Bojović, and Lorenza Giupponi. "An E2E simulator for 5G NR networks". In: *Simul. Model. Pract. Theory* 96 (2019).

[13] Diego Gonzalez Morin, Manuel-Jose Lopez Morales, Pablo Pérez, Alvaro Villegas, and Ana García-Armada. *FikoRE: 5G and Beyond RAN Emulator for Application Level Experimentation and Prototyping*. 2022. DOI: 10.48550/ARXIV.2204.04290. URL: https://arxiv.org/abs/2204.04290.

[14] O-RAN. *E2 Simulator*. URL: https://wiki.o-ran-sc.org/display/ORANSDK/E2+Simulator.

[15] VIAVE. *5G Solutions*. URL: https://www.viavisolutions.com/en-us/5g.

[16] SD-RAN ONOS project. *SD-RAN RAN Simulator*. URL: https://docs.sd-ran.org/master/ran-simulator/README.html.

[17] Marco Mezzavilla, Menglei Zhang, Michele Polese, Russell Ford, Sourjya Dutta, Sundeep Rangan, and Michele Zorzi. "End-to-End Simulation of 5G mmWave Networks". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2237–2263. DOI: 10.1109/COMST.2018.2828880.

[18] CISCO. *open-nFAPI*. URL: https://github.com/cisco/open-nFAPI/blob/master/README.md.

[19] Bob Lantz, Brandon Heller, and Nick McKeown. "A network in a laptop: rapid prototyping for software-defined networks". In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. 2010, pp. 1–6.

[20] Ramon R Fontes, Samira Afzal, Samuel HB Brito, Mateus AS Santos, and Christian Esteve Rothenberg. "Mininet-WiFi: Emulating software-defined wireless networks". In: *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE. 2015, pp. 384–389.

[21] Karim Boutiba, Adlen Ksentini, Bouziane Brik, Yacine Challal, and Amar Balla. "NRflex: Enforcing network slicing in 5G new radio". In: *Computer Communications* 181 (2022), pp. 284–292.

[22] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. "SUMO–simulation of urban mobility: an overview". In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. 2011.

[23] Yongs Zeng, Qingqing Wu, and Rui Zhang. "Accessing from the sky: A tutorial on UAV communications for 5G and beyond". In: *Proceedings of the IEEE* 107.12 (2019), pp. 2327–2375.

[24] ORAN Alliance. "O-RAN Use Cases and Deployment Scenarios WhitePaper". In: *O-RAN ALLIANCE, Tech. Rep* (2020).

AUTHORS

Ramon dos Reis Fontes is professor in computer science at Federal University of Rio Grande do Norte (UFRN), Natal/Brazil. Before his current appointment, he received a M.Sc degree from University of Salvador (UNIFACS), Salvador, Brazil, in 2014, and a Ph.D. degree in electrical and computer engineering from the State University of Campinas, SP, Brazil, in 2018. During his Ph.D., he developed a Wireless Network Emulator that has been widely used by researchers worldwide, and was a visiting PhD student at the DIANA team, in Sophia Antipolis/France in 2016. From Jul/2011 to Jun/2011, he was a professor at the Federal Institute of Science and Technology of Bahia (IFBA), where he taught subjects related to computer networks including network operating system, routers, application servers, information security, and programming. His main research interests are in the field of software defined wireless networks (SDWN), vehicular networking, future internet architecture and cyber security.

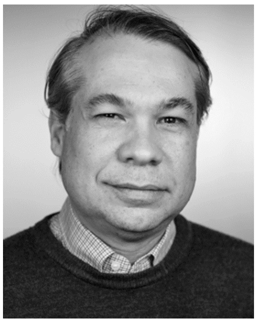
Allan de Medeiros Martins received a B.Sc. degree in electrical engineering, an M.E.Sc. degree in sciences, and a Ph.D. degree in electrical engineering from the Federal University of Rio Grande do Norte, Brazil. He is associated with the Department of Electrical Engineering, Federal University of Rio Grande do Norte. He does research in data mining, computing in mathematics, natural science, engineering and medicine and artificial neural networks.

Vicente Angelo de Sousa Junior received his B.S., M.S and Ph.D. degrees in electrical engineering from the Federal University of Ceará (UFC), Fortaleza, CE, Brazil, in 2001, 2002 and 2009, respectively. Between 2001 and 2006, he developed solutions to UMTS/WLAN interworking for UFC and Ericsson of Brazil. Between 2006 and 2010, he contributed to WIMAX standardization and Nokia's product as a researcher at the Institute of Technological Development (INdT). Dr Sousa is now a professor and the head of the GppCom Research Group at UFRN. He has been contributing to 5G Open RAN projects supported by Lenovo and to NIR measurements and evaluation projects supported by ANATEL Agency.

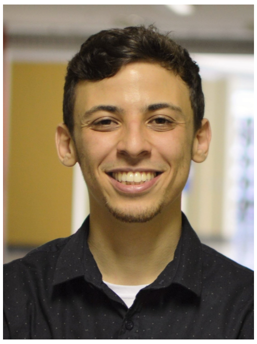
Kaio Henrique Fonseca Dantas received his A.S. in computer networks from the Federal Institute of Education, Science, and Technology of Rio Grande do Norte (IFRN) in 2013 and an MBA degree in information technology management from Anhanguera University (UNIDERP) in 2019. Currently, he serves as a professor of computer networks at IFRN - Campus Pau dos Ferros. Additionally, he holds the position of coordinator of IT infrastructure at the Center for Data Analysis and Computational Intelligence (NADIC) of IFRN.

Lucas Ismael Campos Medeiros received his B.S. degree in telecommunications engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil, in 2021. BSc. Medeiros is now pursuing a master's degree and conducts research with GppCom Research Group at the Federal University of Rio Grande do Norte (UFRN), Brazil. He has been contributing to 5G Open RAN projects supported by Lenovo and to NIR measurements and evaluation projects supported by ANATEL Agency.

Pedro Victor Andrade Alves received his B.S. degree in computer engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, in 2021, where he is currently pursuing a M.Sc. degree in electrical and computer engineering and where he is a team member of the Research Group on Embedded Systems and Reconfigurable Computing (RESRC). He has been contributing to 5G Open RAN projects supported by Lenovo. His main research topic is tactile internet. His research interests include artificial intelligence, embedded systems, and tactile internet.



Marcelo Augusto Costa Fernandes Fernandes was born in Natal, Brazil. He received his B.S. and M.S. degrees in electrical engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, in 1997 and 1999, respectively, and a Ph.D. degree in electrical engineering from the University of Campinas, Campinas, State of São Paulo, Brazil, in 2010. From 2015 to 2016, he worked as a visiting researcher with the Centre Telecommunication Research (CTR), King's College London, London, U.K. From 2019 to 2021, he worked as a visiting scholar with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, USA. He is currently an associate professor with the Department of Computer Engineering and Automation, Federal University of Rio Grande do Norte. He is the leader of the Research Group on Embedded Systems and Reconfigurable Computing (RESRC), and a coordinator of the Laboratory of Machine Learning and Intelligent System (LMLIS). His research interests include artificial intelligence, digital signal processing, embedded systems, reconfigurable hardware, and tactile internet.



Iago Diogenes do Rego received his B.Sc. in electrical engineering and his M.Sc. in electrical and computing engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, in 2018 and 2020, respectively. He is currently pursuing a joint Ph.D. degree in Paris at the Engineering School of Information and Digital Technologies (EFREI Paris) and at UFRN. He has been contributing to 5G Open RAN projects supported by Lenovo and his main research interests are in the field of mobile communications, 5G, resource allocation and machine learning.

Eduardo Aranha received his B.S. (2000), M.S. (2002) and Ph.D. (2009) degree in Computer Science from the Federal University of Pernambuco, Brazil. He is currently an associate professor at the Informatics and Applied Mathematics Department (DIMAp), Federal University of Rio Grande do Norte (UFRN), Brazil. He worked as a visiting researcher at the School of Computer Science, University of Adelaide, Australia. His research interests include computer science education, software engineering, 5G and artificial intelligence applications.

Vinícius José Miranda Toscano de Brito Filho received his B.Sc. (1999) degree in computer science from the Federal University of Pernambuco, Brazil. He is specialized in corporate network management by Santa Maria School (FSM) (2009), Pernambuco, Brazil, and in Information Technology by Federal University of Rio Grande do Norte (UFRN) (2021), Rio Grande do Norte, Brazil. He also received an M.B.A. in Business Management by the Getúlio Vargas Foundation (FGV) (2017), Brazil. He has been contributing to 5G Open RAN projects supported by Lenovo and his main research interests are in the field of mobile communications, 5G, resource allocation and machine learning.

Mateus Arnaud Santos de Sousa Goldbarg received a B.S. degree in Computer Engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, RN, Brazil in 2021. As a member of the Laboratory of Machine Learning and Intelligent Instrumentation, since 2020 he has been doing researches on techniques for compressing deep learning models for microservices and embedded systems. Currently, he is pursuing the M.Sc. degree in electrical and computer engineering at UFRN and has been contributing to the 5G Open RAN project supported by Lenovo.

Wysterlânya Kyury Pereira Barros received her B.Sc. degree in computer engineering (2018) and M.Sc. degree in electrical and computing engineering (2021) from the Federal University of Rio Grande do Norte (UFRN), Brazil. She is pursuing a Ph.D. in electrical and computing engineering at UFRN and is a researcher of the Research Group on Embedded Systems and Reconfigurable Computing (RESRC). She has been contributing to 5G Open RAN projects supported by Lenovo, and her research interests include artificial intelligence, 5G networks, computer vision, embedded systems, and reconfigurable hardware.

Roger Immich is a professor at the Digital Metropolis Institute (IMD) of the Federal University of Rio Grande do Norte (UFRN). He earned his Ph.D. in informatics engineering from the University of Coimbra, Portugal (2017). He was a visiting researcher at the University of California at Los Angeles, United States (UCLA) in 2016/2017, a postdoctoral researcher at the Institute of Computing of the University of Campinas (UNICAMP) in 2018/2019, and a visiting professor at the University of Málaga, Spain, in 2021. Roger Immich is author and co-author of over 75 articles published in highly reputable conferences and journals, both nationally and internationally. Additionally, he plays an active role in organizing conferences and workshops. His research interests encompass various areas, including smart cities, IoT, 5G, quality of experience, as well as cloud and fog computing.

Augusto Venancio Neto Augusto Venâncio Neto (Senior Member, IEEE) is an associate professor at the Department of Informatics and Applied Mathematics of the Federal University of Rio Grande do Norte (UFRN), Natal/RN, Brazil; a productivity fellow at the National Council for Scientific and Technological Development (CNPq); and a member of the Instituto de Telecomunicações, Portugal. He is the leader of the Research Group in Future Internet Service and Applications (REGINA), with background in cutting-edge technologies in computer networks and telecommunications, through which he manages large research and teams. This allowed him training tens of PhDs and masters' researchers, as well as authoring/coauthoring more than 180 papers, including patents and several best-paper awards. He is involved in the organization of several international conferences and workshops, and he also serves as guest editor for special issues of peer-reviewed scholarly journals. His research interests fall in the fields of 5G/6G networks, AI-supported networking, mobile computing, smart spaces, SDN, NFV and cloud computing.