Deployable Hierarchical ML Traffic Steering for O-RAN RICs

Roberto Riggio
Università Politecnica delle Marche, Ancona, Italy
Email: r.riggio@univpm.it

***Abstract*—Current mobile networks often fail to achieve their full performance potential, as user devices commonly experience data rates well below their possible capabilities. This limitation arises from conventional cell selection mechanisms, which often fail to identify and utilize higher-performing candidate cells, resulting in suboptimal use of network resources. In this paper, we introduce a hierarchical machine learning–based traffic steering solution for O-RAN-enabled 5G and beyond networks. The proposed solution supports network-assisted cell selection, guiding user devices toward more optimal cells. The proposed solution is developed in accordance with O-RAN Alliance design principles and operates without any modifications to existing 3GPP signaling or to mobile devices. Experimental evaluations on a small-scale testbed demonstrate that the proposed approach can improve the median throughput by up to 75% in certain traffic scenarios while also improving overall network fairness.**

***Index Terms*—Machine learning, O-RAN, B5G, Traffic Steering, Cell Selection**

I. Introduction

Upcoming 5G Advanced and future 6G networks are anticipated to support a vast range of heterogeneous use cases at massive scale, demanding highly automated mechanisms for network control and automation. Despite this, today's cellular architectures do not offer fine-grained control over Radio Access Network (RAN) operations. Leveraging open interfaces and off-the-shelf hardware, O-RAN is rapidly emerging as the leading approach for building virtualized 5G and beyond RANs. This paradigm introduces an open architecture enabling closed-loop control, data-driven decision-making, and intelligent optimization of RAN operations.

In an O-RAN–compliant architecture the radio control framework is organized around two key components: the Near-Real-Time RAN Intelligent Controller (near-RT RIC) and the Non-Real-Time RAN Intelligent Controller (non-RT RIC). At the top of this hierarchy, the non-RT RIC hosts rApps responsible for executing high-level, long-term RAN optimization functions. It leverages comprehensive network information and provides AI-driven guidance to the near-RT RIC. In contrast, the near-RT RIC operates closer to the network edge, managing time-sensitive control and optimization of RAN components through modular applications known as xApps. This separation of concerns results in a flexible and programmable architecture, ideally suited for the integration of advanced AI-based modules that enhance network performance and adaptability through coordinated operation of rApps and xApps.

Traffic steering represents one of the key application areas within the O-RAN framework. Developing an effective traffic steering requires the RIC to handle the complexity of managing numerous cell configurations. In fact, in current mobile networks, it is common for user devices to remain connected to suboptimal cells [1], experiencing much lower data rates even when nearby cells could offer up to ten times better performance. This issue arises from legacy cell selection mechanisms, maintained for backward compatibility, which prioritize stable and “good enough” connectivity over optimal performance. Originally, this approach suited earlier, sparser, and more homogeneous networks where most cells had similar characteristics. However, today's networks are far denser and more heterogeneous, combining large macrocells with many small cells that vary widely in bandwidth (from 5 MHz to 100 MHz typically), leading to many more handoff opportunities and greater performance disparities between cells.

This paper introduces a hierarchical machine learning-based traffic steering approach designed to mitigate the underutilization of network resources described earlier. The proposed solution is guided by three key design principles. First, unlike previous approaches such as [1], it enhances cell association decisions by jointly considering radio quality, cell bandwidth, and user distribution. Second, it fully complies with existing 3GPP signaling standards while relying on gNBs that support the O-RAN–defined interfaces. Third, the approach is entirely network-driven, requiring no modifications to user equipment (UE), which continues to support the process by performing standard measurement tasks as instructed by the serving base station. Experimental evaluations on a small-scale 5G testbed show that the proposed approach can improve the median throughput by up to 75% in some specific traffic scenarios while also achieving better network fairness.

The structure of the paper is the following. The related work is discussed in Sec. II. The motivation behind this work is presented in Sec. III. Section IV discusses the system design and the proposed cell-selection algorithm while the implementation details and the results of the evaluation are the focus of Sec. V. Finally, Sec. VI concludes the paper and suggest some future research directions.

II. Related Work

The study in [1] presents iCellSpeed, a device-centric approach designed to enhance cell selection by addressing the inefficiencies of traditional network-controlled methods. In contrast, our work argues that relocating decision-making to the network side enables more effective coordination and resource management, ultimately unlocking greater and more consistent performance gains.

In [2] we address a similar problem by proposing a solution aiming at improving cell selection in 5G and beyond networks.

The proposed solution operates on the global network view maintained by the near-RT RIC to jointly optimize mobility management for channel quality and bitrate. The proposed solution is a rule-based heuristic and does not include context-aware adaptation. Moreover, the solution is not integrated with AI/ML-based xApps and the algorithm is static, not learning from context or history.

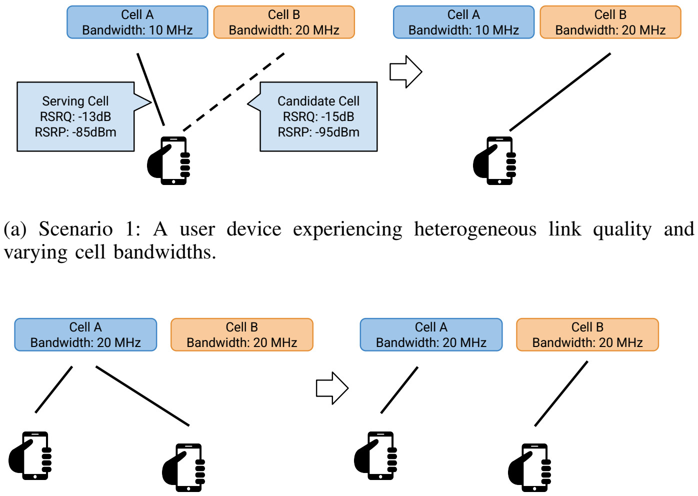
Recent research has increasingly leveraged data-driven and machine learning techniques to optimize cellular network performance. Examples include self-tuning algorithms that enhance Quality of Experience (QoE) in multi-carrier 4G systems [3], predictive models for large-scale handover management [4], and reinforcement learning–based approaches for dynamic load balancing [5]. Collectively, these works reflect a clear shift toward intelligent, adaptive optimization methods for user association and network management.

Deep learning has been applied to traffic steering by leveraging multiple network state features to predict and optimize data flow, supporting objectives like load balancing and congestion reduction. For instance, [6] employs a Long Short-Term Memory (LSTM) model to learn traffic patterns and forecast incoming packets for proactive steering, while [7] uses a Convolutional Neural Network (CNN) to assign users to base stations dynamically. Despite their promise, these approaches rely heavily on large, labeled datasets that are often unavailable in real-world networks, leading to limited training accuracy and potentially suboptimal steering decisions.

Reinforcement Learning (RL) has become a key tool for addressing optimization challenges in Open RAN environments. Traffic steering in O-RAN must account for highly dynamic factors such as congestion levels, fluctuating traffic patterns, and varying user demands. RL techniques can effectively adapt to these changing conditions and learn optimal steering policies over time. Approaches like Q-learning, Deep Reinforcement Learning (DRL), and Hierarchical RL (HRL) have been applied in this context. While Q-learning offers conceptual simplicity [7], its slow convergence and inefficiency in large state-action spaces limit scalability. Deep Q-Networks (DQN) overcome these issues by using neural networks to estimate state-action values instead of large Q-tables, enabling faster and more robust learning, as demonstrated in [8]. However, they often operate as single-layer agents, learning only at one timescale, which limits scalability.

Federated learning offers a decentralized approach to traffic steering by harnessing the collective intelligence of user equipment (UEs) and edge devices. Each device trains locally and shares model parameters rather than raw data, enabling collaborative optimization of traffic management while preserving user privacy [9]. However, this approach faces key challenges, including high communication overhead from frequent model updates and significant heterogeneity in device capabilities—such as differences in computation, storage, and connectivity—which can hinder consistent training and deployment efficiency.

Hierarchical learning enables agents to simultaneously develop high-level strategies for overall objectives and low-level policies for specific actions. When applied to traffic steering, combining this approach with DQN can enhance performance by decomposing the complex optimization process into layered decision-making stages [10]. Hierarchical RL offers a way to separate strategic and tactical decision-making, but so far, it has mostly remained in simulation, not on real, O-RAN-compliant testbeds.



(a) Scenario 1: A user device experiencing heterogeneous link quality and varying cell bandwidths.

(b) Scenario 2: Two user devices with uniform link quality and identical cell bandwidths.

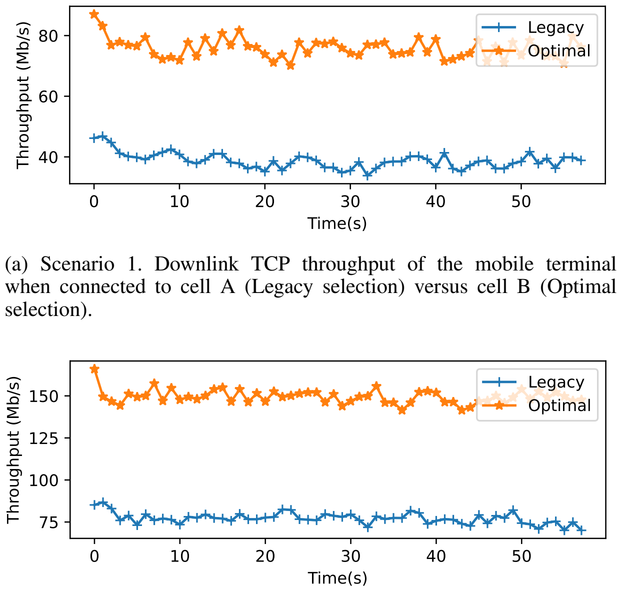
Fig. 1. The traditional cell selection mechanism may cause inefficient use of network resources because it fails to consider differences in cell bandwidth.

Building upon the insights of previous research addressing performance limitations in cellular networks, our work introduces a practical, deployable solution that leverages cutting-edge technologies. It distinguishes itself in two key aspects: first, it is explicitly designed for implementation in real-world network environments; and second, it represents the first application of a hierarchical DQN-based approach to cell selection for O-RAN RICs.

III. A Motivating Example

Cell selection relies on downlink RSRP and RSRQ measurements collected by the mobile device. At a high level, a handover is initiated from the serving cell to a target cell when the serving cell's RSRP remains below a defined threshold for a certain duration and the target cell's RSRQ exceeds that of the serving cell by a predefined margin. The detailed configuration of these parameters is beyond the scope of this paper and has been extensively studied in the literature [11], [12]. It is worth noting, however, that (i) a device generally remains connected to its current cell as long as the signal quality is acceptable, even if a better-performing cell exists, and (ii) traditional handover mechanisms do not consider factors such as cell bandwidth in their decisions.

An illustrative example is shown in Fig. 1a. Here, a mobile terminal is connected to serving cell A and continuously monitors its signal quality, which determines when to begin scanning neighboring cells, a process known as the look-around phase. For instance, measurement triggering may occur when conditions such as $RSRQ(A) < -15\,\text{dB}$ or $RSRP(A) < -122\,\text{dBm}$ are met. In the scenario depicted,



(a) Scenario 1. Downlink TCP throughput of the mobile terminal when connected to cell A (Legacy selection) versus cell B (Optimal selection).

(b) Scenario 2. Aggregated downlink TCP throughput of two mobile terminals when both are connected to cell A (Legacy selection) versus when traffic is balanced across both cells (Optimal selection).

Fig. 2. Throughput over time for the two illustrative scenarios.

this means the device would not measure neighboring cell B until the serving cell's signal degrades below these thresholds. Once RSRQ(A) reaches $-15\,\mathrm{dB}$, the device starts evaluating nearby cells. A secondary reporting condition is also configured, typically defined as $RSRQ(A) < RSRQ(other) - 3\,\mathrm{dB}$, ensuring that a new cell is reported to the base station only if it offers at least a 3 dB improvement over the current one.

In this example, it would actually be advantageous to hand over the mobile terminal to cell B, even though its signal quality is slightly lower than that of cell A. This is because cell B offers a wider bandwidth (10 MHz vs. 5 MHz), which can deliver higher data rates. As illustrated in Fig. 2a, the downlink TCP throughput achieved using an optimized cell selection strategy that considers bandwidth outperforms that of the legacy approach. Despite the weaker signal on cell B, the terminal attains higher throughput, as the increased bandwidth compensates for the lower link quality.

Another illustrative scenario is presented in Fig. 1b. Here, two mobile terminals are connected to the same serving cell A under similar radio conditions, i.e., their RSRQ values differ by less than 3 dB. Although cell B offers slightly lower signal quality (not shown for clarity), both cells have the same channel bandwidth. In this context, handing over one terminal to cell B, despite its weaker link, would improve overall efficiency by distributing the load more evenly. This effect is confirmed by the measurements in Fig. 2b, which show the aggregated downlink TCP throughput for both terminals. When both devices remain attached to cell A (legacy case), throughput is limited, whereas balancing them across cells A and B (optimal case) can improve the median network throughput by up to 75%, with only a minor increase in variability.

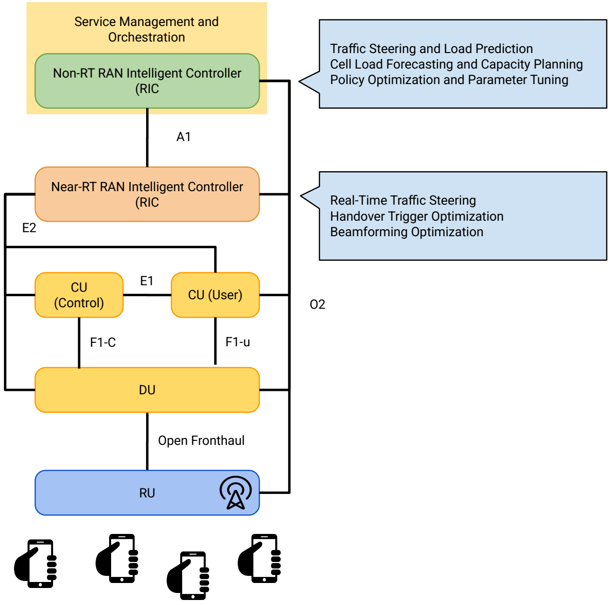

Service Management and Orchestration
Non-RT RAN Intelligent Controller (RIC
A1
Near-RT RAN Intelligent Controller (RIC
E2
CU (Control)
E1
CU (User)
F1-C
F1-u
DU
Open Fronthaul
RU
O2
Traffic Steering and Load Prediction
Cell Load Forecasting and Capacity Planning
Policy Optimization and Parameter Tuning
Real-Time Traffic Steering
Handover Trigger Optimization
Beamforming Optimization

Fig. 3. Hierarchical learning scheme for O-RAN.

While the scenarios discussed above may appear straightforward, such behavior cannot be achieved in current cellular networks for two main reasons: (i) measurement and reporting thresholds are conservatively configured by the serving cell, and (ii) even when superior candidate cells are detected, the serving cell lacks a comprehensive network-wide perspective to make handover decisions that also consider factors such as cell bandwidth. The first limitation could be mitigated by adopting more aggressive measurement and reporting criteria. Addressing the second challenge, however, represents the primary focus of this work.

IV. Hierarchical Machine Learning–Based Traffic Steering

A. *Hierarchical Learning Schemes in O-RAN*

Hierarchical learning algorithms are a class of machine learning methods that structure the learning process into multiple levels (or hierarchies), allowing an agent (or model) to handle complex tasks by breaking them down into simpler subtasks. As presented in Fig 3, In O-RAN traffic steering, a hierarchical learning approach could work as follows:

- Meta-agent (Non-RT RIC rApp): decides which general policy to apply (e.g., load balancing).
- Agent (Near-RT RIC xApp): execute actions such as adjusting transmission power or reassigning users.

For instance, in the context of traffic steering, a hierarchical learning approach can effectively combine the strengths of supervised learning and RL to manage decision-making across multiple levels of the network. At the higher level, supervised learning can be used to train models that predict optimal cell associations based on historical data—such as RSRP, RSRQ, bandwidth availability, and user density—thereby establishing an initial policy for traffic distribution. At the lower level, RL can then refine these policies in real time by interacting with the network environment, receiving performance feedback (e.g., throughput gains, reduced congestion), and adapting its decisions through methods like Q-learning or policy gradients. Furthermore, integrating unsupervised learning techniques such as clustering can enhance this process by identifying hidden patterns in user mobility or traffic demand. When combined with RL, this allows the system to both discover meaningful structures in network behavior and adapt dynamically to optimize resource allocation and user experience across cells.

A key advantage of hierarchical learning in the context of traffic steering for O-RAN is its ability to distribute intelligence across different control layers—namely, the non-RT RIC and the near-RT RIC, each operating on distinct time scales and decision granularities. At the higher layer, the non-RT RIC can leverage powerful machine learning techniques to develop long-term optimization strategies, such as predicting network load trends, user mobility patterns, or bandwidth utilization. These insights are then passed down to the near-RT RIC, which operates on shorter time intervals and uses reinforcement learning or other adaptive algorithms to make rapid, fine-grained decisions, such as dynamically steering users between cells to balance traffic or improve throughput. Moreover, hierarchical learning enables reinforcement learning algorithms to effectively manage long-horizon objectives, for example, maintaining network stability and fairness over time, while still reacting swiftly to short-term variations in traffic or signal quality. Such a design not only facilitates seamless interaction between rApps (in the non-RT RIC) and xApps (in the near-RT RIC) but also makes the overall traffic steering system more adaptive, scalable, and suitable for real-world O-RAN deployments.

In the context of traffic steering, one of the main challenges in O-RAN is the limitation on performing online learning directly from untrained models, as this could disrupt live network operations. This issue can be effectively addressed through several complementary strategies. First, models can be pre-trained offline using historical network data or realistic simulated environments that capture key aspects of radio conditions, mobility patterns, and traffic behavior. This allows the hierarchical learning framework to start with well-informed policies rather than random initializations, ensuring stable and efficient early-stage performance. Second, implementing a model versioning and update pipeline can further enhance reliability. In this setup, models are periodically retrained and validated offline, using fresh network data, and then safely deployed into the operational environment. This cycle not only complies with O-RAN's restrictions on online learning from unverified models, but also enables continuous adaptation to changing network conditions, thus improving the responsiveness and robustness of the traffic steering system.

B. Hierarchical DQN-Based Traffic Steering Framework

We propose a *hierarchical deep Q-network* framework for intelligent traffic steering in O-RAN-based 5G and beyond networks. The proposed architecture follows the O-RAN control hierarchy and integrates two coordinated learning agents: a *meta-agent* deployed as an *rApp* in the *non-RT RIC*, and a *agent* deployed as an *xApp* in the *near-RT RIC*. Together, they enable adaptive, goal-oriented decision-making across different time scales.

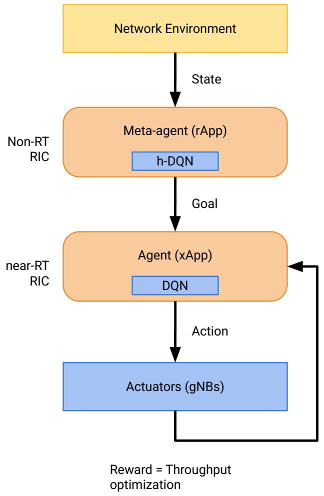


Fig. 4. Hierarchical Reinforcement Learning Architecture for Traffic Steering in O-RAN. The *meta-agent* (rApp) defines long-term goals while the *agent* (xApp) executes short-term steering actions through DQN-based learning.

As illustrated in Fig. 4, the *meta-agent* operates at a coarser timescale (seconds to hours) within the non-RT RIC. It observes aggregated network parameters such as cell load, interference, traffic type distribution, and QoE indicators, collected via the O1 interface and the Service Management and Orchestration (SMO) layer. Based on these observations, the *meta-agent* defines high-level *goals* for the *agent*, for instance, maintaining load-balancing thresholds, minimizing queuing delay, or achieving target throughput across multiple cells or RATs.

The *agent*, implemented as a DQN-based xApp within the near-RT RIC, operates at a finer timescale (10 ms–1 s). It receives network states directly from the RAN environment, such as RSRP/RSRQ, SINR, buffer occupancy, or UE mobility indicators, and performs short-term actions such as user handovers, load-threshold tuning, or RAT-based traffic steering. The *agent* interacts with the environment, receives intrinsic rewards based on throughput or delay improvements, and refines its Q-network parameters accordingly. The *meta-agent* monitors long-term goal progress and periodically updates objectives, forming a closed feedback loop between strategic and operational control layers.

The *meta-agent* uses historical or simulated data to pre-train its model offline and can query an AI/ML model hosted in the SMO for inference. It periodically updates the *agent* via the A1 interface, providing goal vectors and reward-weight parameters that shape the *agent*'s local decision-making process. The *agent* then maximizes the cumulative reward defined as:

$$r_t = \lambda_{\text{thr}} \cdot \Delta T_u + \lambda_{\text{fair}} \cdot \Delta J - \lambda_{\text{ho}} \cdot C_{\text{ho}}, \tag{1}$$

where λ_{thr}, λ_{fair}, and λ_{ho} are weights set by the *meta-agent* for throughput, fairness, and handover cost, respectively.

This setup allows the system to balance conflicting goals, for example, maintaining high buffer utilization while avoiding excessive delay from queue congestion.

Both the global and local agents rely on a DQN formulation. The local agent adopts a standard DQN [13] architecture to ensure computational simplicity and compatibility with resource-constrained nodes. While advanced variants such as Double-DQN or Dueling-DQN could further improve stability, we deliberately prioritize low inference complexity and fast convergence, which are critical for onboard execution in satellite and HAPS environments. At the global level, the Q-function is trained offline using Conservative Q-Learning (CQL) [14], enabling safe policy learning from simulated data while avoiding overestimation outside the support of the training distribution.

The proposed framework aligns with the O-RAN Alliance's AI/ML workflow. Data from RAN components are collected via the O1 interface and forwarded to the non-RT RIC. The rApp trains ML models and transmits updated policies to the xApp through the A1 interface. The xApp then performs near-real-time decisions through the E2 interface to base stations (i.e., gNBs), adjusting RAN parameters such as scheduling thresholds or RAT selection. Network elements act as actuators, enforcing the steering actions on active traffic flows.

C. Offline Training of the Meta-Agent Using Simulated Data

The meta-agent, operating at the non-RT RIC level, is responsible for setting long-horizon goals that guide the near-RT RIC xApp in performing short-term traffic steering. Its model can be effectively trained offline using simulated data, thereby avoiding the risks associated with online exploration in live networks. The training pipeline consists of several stages, from the construction of a calibrated simulator to policy distillation and deployment.

To enable safe and reproducible experimentation, we employ an ns3-based simulator to act as a digital twin of the target O-RAN deployment. The simulator reproduces the essential dynamics of the system, including radio propagation, scheduler behavior, traffic models, and interference patterns. A few key parameters, such as pathloss exponents, noise figures, and scheduler efficiency, are fitted to match statistics extracted from O1 telemetry, ensuring that simulated network metrics (e.g., PRB utilization distributions and throughput/RSRQ curves) are consistent with those observed in reality in our small-scale testbed (see later). Notice how the simulation is computationally lightweight, permitting the generation of thousands of episodes, favoring abstracted link and queuing models over full physical-layer simulation (which are not supported by ns3).

Once the simulator is available, a broad distribution of synthetic flows is generated to expose the meta-agent to a wide variety of operating conditions. These tasks include variations in topology (i.e., distance of users from the serving cells), carrier bandwidth, and traffic load. Randomization is used to allow the meta-agent to generalize across unseen environments and mitigates the gap between simulations and reality.

Each simulated episode produces trajectories that describe the network evolution under a baseline policy, typically reflecting standard threshold-based handover and load-balancing rules. For every time step t, the simulator logs the tuple (s_t, a_t, r_t, s_{t+1}), where s_t encodes long-term network state features such as per-cell PRB utilization, RSRP/RSRQ distributions, and handover rates; a_t represents the meta-action, for example, updated steering quotas or PRB caps; and r_t is a scalar reward measuring the network performance according to the throughput. The reward is typically shaped as

$$r_t = \lambda_{\text{thr}} \cdot \text{Thr}_{50} - \lambda_{\text{ho}} \cdot H, \tag{2}$$

where Thr_{50} is the median throughput and H measures handover activity. The coefficients $\lambda_{.}$ control the trade-off between performance and stability.

The resulting dataset $\mathcal{D}$ is then used for offline reinforcement learning. Before policy optimization, a state encoder is trained using self-supervised learning, such as masked feature modeling or contrastive forecasting, on all available telemetry to obtain robust representations of network states. The meta-agent's policy is subsequently optimized on $\mathcal{D}$ using CQL. These methods constrain policy updates to the support of the logged dataset, preventing unsafe extrapolation. This leads to better generalization and safer policies.

The learned policy is evaluated offline through off-policy evaluation techniques, including weighted importance sampling or doubly-robust estimators, which provide unbiased estimates of the expected performance of the learned policy on held-out trajectories. Stress tests covering extreme cases such as cell failures, flash-crowd events, and spectrum reallocation could be performed during this phase to ensure the robustness of the learned behavior and to verify that SLA and safety constraints remain satisfied. They are, however, outside the scope of this paper and will thus be ignored.

To mitigate the gap between simulation and reality, several strategies are employed during training. Domain randomization exposes the model to a wide range of possible configurations, while feature normalization encourages reliance on causal signals such as load and bandwidth rather than spurious correlations. Furthermore, realistic noise and delays are injected into the simulated telemetry to mimic the latency and staleness of O1 data collection. Temporal smoothness regularization, such as the Smoothing Moving Average (SMA), is added to penalize abrupt goal changes, yielding more stable real-world behavior.

After convergence, the trained meta-policy is distilled into a compact model suitable for execution within the non-RT RIC environment. The final network is exported as a lightweight neural network or gradient-boosted model that outputs A1 policy targets for the near-RT RIC xApp. Together with each predicted goal vector, the rApp also produces a confidence score to signal the reliability of its recommendation. When confidence is low, the system can default to conservative baseline heuristics to ensure safe operation. This feature, however, has not been implemented in the system presented in this paper.

During the offline CQL training process, several quantitative metrics are used to characterize both learning efficiency and computational overhead. Convergence is measured in terms of the number of training iterations and epochs required for the CQL-based meta-agent to reach a stable policy, defined as less than a predefined relative improvement (e.g., ¡1–2%) in the median throughput–based reward over consecutive evaluation checkpoints. In practice, convergence is typically achieved after a few thousand simulated episodes, thanks to the use of abstracted link and queuing models that significantly reduce simulator complexity.

Computational overhead is quantified by the total wall-clock training time, GPU/CPU utilization, and memory footprint of the replay dataset. Using lightweight ns-3–based simulations and offline training, a full training cycle can be completed within a few hours on a single GPU-equipped workstation, while inference complexity remains negligible once the policy is distilled into a compact model for deployment in the non-RT RIC. Additional metrics such as dataset size (number of state–action–reward tuples), average training batch time, and model size after distillation are also monitored to ensure that the resulting rApp complies with non-RT RIC resource constraints and supports periodic retraining without impacting operational timelines

V. Implementation and Evaluation

A. Implementation details

The proposed framework has been realized by integrating the reference O-RAN near-RT RIC with a custom-developed non-RT RIC. The latter is responsible for maintaining a comprehensive, system-wide view of the network. It oversees the generation and delivery of control and management policies to the near-RT RIC via the standardized A1 interface. In the current implementation, the non-RT RIC layer aggregates the following network information:

- *RSSI/RSRP/RSRQ*. The Carrier Received Signal Strength Indicator (RSSI) represents the total power measured across all received reference signals. Unlike the Reference Signal Received Power (RSRP), which provides a narrowband measurement focused on specific reference signals, RSSI captures the overall power across the entire system bandwidth. The Reference Signal Received Quality (RSRQ) further refines this evaluation by incorporating the number of Physical Resource Blocks (PRBs) utilized in the measurement.
- *Traffic Matrix*. The system monitors, for each UE, the total number of packets and bytes transmitted and received over the network. Both the cumulative values (representing the absolute traffic volume) and the instantaneous throughput are computed. The bitrate is calculated within the most recent observation window, allowing applications to assess short-term traffic dynamics, detect variations in user activity, and make informed decisions regarding resource allocation and traffic steering.
- *PRB Utilization*. The system continuously monitors the number of PRBs allocated and utilized at each gNB. This metric provides a direct indication of the cell's occupancy and spectral efficiency. By analyzing PRB utilization over time, it becomes possible to identify congestion trends, assess load distribution among neighboring cells, and optimize scheduling or handover decisions.

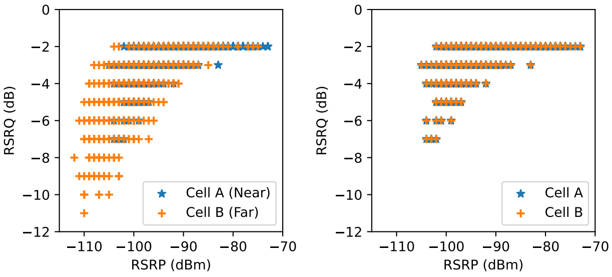
The gNB in the testbed has been implemented using srsRAN [15] and the Ettus X310 Software Defined Radio (SDR) platform [16], while the Open5GS [17] platform has been employed to implement the 5G Core Network. The Ettus X310 SDR platform has been configured with two RF daughterboards, each configured to operate as a separate cell, resulting in a dual-cell setup. Each cell has been equipped with a pair of omnidirectional 3 dB antennas. The srsRAN framework, specifically its gNB application, has been enhanced through the integration of a software agent designed to enable seamless interaction with the near-RT RIC. This agent handles the exchange of control and monitoring messages over the E2 interface, ensuring full compliance with the O-RAN specifications[1]. The agent is tasked with collecting detailed network telemetry from the gNB, including PHY and MAC layer statistics, resource utilization metrics, and performance indicators. These measurements are periodically reported to the near-RT RIC, providing it with a real-time view of the network conditions and enabling intelligent, data-driven decision-making for functions such as traffic steering, load balancing, and interference management. Standard 3GPP intra-gNB handovers are used and influenced by the near-RT RIC via the E2 interface, without UE modifications.

The proposed framework has been realized as a distributed software architecture comprising two main components: an xApp deployed within the near-RT RIC and an rApp hosted in the non-RT RIC. Together, these applications cooperate to implement adaptive and intelligent cell selection policies, leveraging the comprehensive, cross-layer network view maintained by the RICs. To facilitate this functionality, the non-RT RIC exposes a Python-based Application Programming Interface (API) that enables the rApp to interact seamlessly with the underlying RIC platform. Through this design, the rApp can focus on decision logic and reinforcement learning tasks, relying on the API to handle communication, data retrieval, and policy enforcement across the O-RAN infrastructure. The resulting modular and technology-agnostic approach simplifies experimentation, promotes code reusability, and allows the proposed solution to be easily integrated or extended to different network scenarios and radio configurations.

B. Evaluation Methodology

Our evaluation methodology has been designed to demonstrate the ability of the proposed framework to perform cell selection decisions that jointly consider both radio signal quality and cell capacity, expressed as the number of available PRBs. The evaluation focuses on two representative test scenarios: a single-UE setup (illustrated in Fig. 1a) and a dual-UE setup (illustrated in Fig. 1b). In both configurations, a single TCP flow is generated in the downlink direction, from the 5G Core to each active UE, to emulate a continuous data session. In the single-UE scenario, the UE is positioned approximately 1 meter from Cell A and 3 meters from Cell B (an extension cable has been used in this case). Given the limited transmit power of the Ettus X310 SDR used in the testbed, which provide an overall coverage range of roughly 5 meters, this

[1] Note how platforms such as the Ettus X310 lack Open Fronthaul support with 7.2 split which is typically used in O-RAN deployment. However, the focus of this work is on RIC-level intelligence and control-loop validation, which is largely orthogonal to the fronthaul split.



(a) Scenario 1

(b) Scenario 2

Fig. 5. RSRP/RSRQ measurements of the two cells in the two reference scenarios. Notice how for scenario 2 the measurements of one single UE are reported (the other were essentially the same).

setup realistically reflects a user positioned at the cell center for Cell A and at a mid-cell region for Cell B in a real-world deployment. The cells are configured heterogeneously: Cell A operates with 10 MHz of bandwidth, while Cell B offers 20 MHz of bandwidth. In the dual-UE scenario, both UEs are co-located at an equal distance from the two cells, thus experiencing identical channel conditions. In this case, both cells are configured with 20 MHz of bandwidth. The UEs are configured to report RSRP/RSRQ measurements every 240 ms, providing timely feedback for adaptive decision-making.

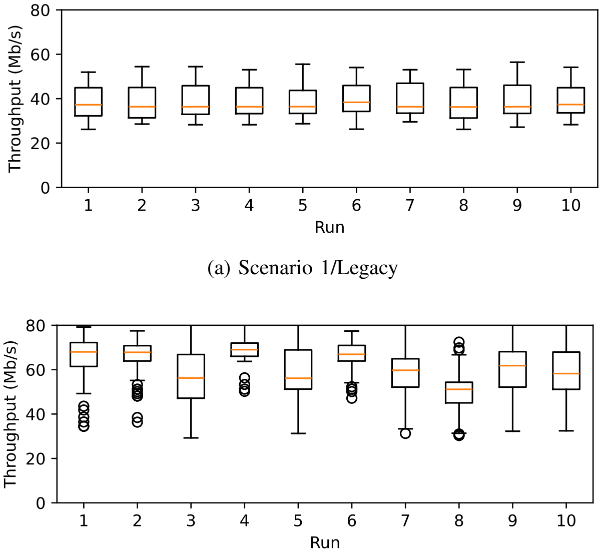
C. Results

In this subsection, we analyze and discuss the experimental results obtained from the measurement campaign carried out on our small-scale O-RAN testbed.

Figure 5a presents the RSRP/RSRQ measurements collected by the UEs in the two reference scenarios. The scatter plots clearly illustrate the distinct radio conditions experienced in each setup. In Scenario 2, where both UEs are positioned close to the gNBs, the vast majority of samples cluster in the upper-right region of the graph, corresponding to high RSRP and RSRQ values. This indicates that both UEs operated under favorable channel conditions with minimal interference and strong received signal power.

In contrast, Scenario 1 exhibits a more heterogeneous distribution. Measurements for Cell A remain concentrated in the upper-right corner, confirming a stable and strong link. However, data points corresponding to Cell B spread noticeably toward the lower-left area, which reflects degraded signal quality and fluctuating radio conditions. This behavior aligns with the UE's greater distance from Cell B and the limited transmission power of the Ettus X310-based gNBs, resulting in variable channel quality.

It is also important to note that the results shown in Figure 5b refer to only one UE, since both UEs in Scenario 2 experienced virtually identical measurements due to their symmetric positioning relative to the gNBs. Overall, the results in Figure 5 confirm that the network environment accurately matched the expected experimental conditions, thereby validating the reliability of the test setup used to evaluate the proposed scheme.



(a) Scenario 1/Legacy

(b) Scenario 1/Optimal

Fig. 6. Aggregated network throughput for scenario 1 across 10 runs.

Figure 6 presents the boxplots of the aggregated downlink throughput observed in Scenario 1 for both the legacy baseline and the proposed approach.

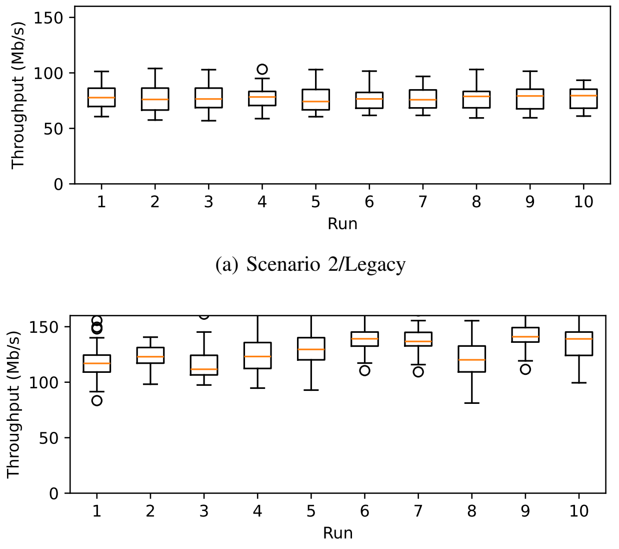
In the legacy configuration (Fig. 6a), the UE predominantly remains connected to Cell A. This behavior aligns with the standard 3GPP cell selection and handover procedures, which prioritize signal strength and link stability over overall network performance. Consequently, the UE is not handed over to Cell B, even in instances where the radio conditions between the two cells become comparable. Moreover, the legacy algorithm does not account for cell bandwidth availability and therefore cannot exploit cases where a cell with slightly poorer signal quality might still deliver a higher effective throughput thanks to a larger number of available PRBs.

By contrast, the proposed scheme introduces a more holistic decision-making process, considering both signal quality (RSRP/RSRQ) and cell capacity. It is configured to tolerate a signal degradation of up to –3 dB if this trade-off leads to a connection with a cell offering greater bandwidth or improved overall performance. As shown in Fig. 6b, this strategy yields a notably higher average downlink throughput, confirming the effectiveness of the reinforcement learning–based approach.

A minor trade-off is observed, however: the throughput distribution under the proposed approach exhibits slightly increased variability, reflecting the system's more dynamic adaptation to changing channel and load conditions. Nonetheless, the net performance gain significantly outweighs this marginal instability, validating the benefit of intelligent, bandwidth-aware cell selection.

Figure 7 illustrates the boxplots of the aggregated downlink throughput obtained in Scenario 2 for both the legacy configuration and the proposed approach.

In the legacy case (Fig. 7a), both UEs remain connected to Cell A. This situation does not necessarily result from a deterministic configuration but may arise either from previous



(a) Scenario 2/Legacy

(b) Scenario 2/Optimal

Fig. 7. Aggregated network throughput for scenario 2 across 10 runs.

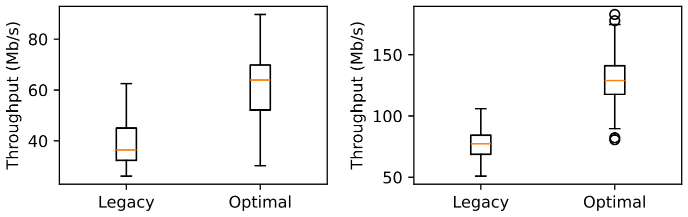
network attachment events or due to random initial conditions. Even though both UEs experience comparable channel quality toward the two gNBs, the standard 3GPP handover and cell selection logic does not intervene to optimize the load distribution. This behavior stems from the 3GPP framework's focus on association stability, i.e., as long as the received signal remains within acceptable thresholds, the network refrains from initiating handovers that could disrupt ongoing sessions. As a result, Cell A becomes overloaded, while Cell B remains underutilized, leading to suboptimal use of network resources.

Conversely, the proposed framework leverages its global network awareness and learning-based optimization to identify load-balancing opportunities in real time. By analyzing both the channel conditions and the resource utilization across gNBs, we see that the proposed approach dynamically distributes the two UEs between the two cells, ensuring a more balanced allocation of PRBs and reducing cell congestion.

As shown in Fig. 7b, this adaptive decision-making results in a significant increase in the aggregated network throughput. The outcome highlights the proposed approach's ability to autonomously optimize performance across multiple UEs and cells, achieving a more efficient utilization of radio resources without compromising link reliability or stability.

Figure 8 provides a consolidated view of the experimental results across ten independent runs of the evaluation campaign. The comparison highlights the consistent performance gains achieved by the proposed approach over the legacy baseline in both test scenarios.

The proposed approach demonstrates a remarkable throughput improvement of approximately 75% in Scenario 1, where cell bandwidth asymmetry and varying channel conditions offer greater optimization opportunities. In Scenario 2, where the UEs experience similar radio conditions and both cells have equal bandwidth configurations, the proposed approach still achieves a substantial 65% increase in median downlink throughput compared to the standard 3GPP-based selection mechanism. These results validate the system's ability to learn and exploit multi-dimensional network information, combining signal quality metrics and available cell capacity, to enhance overall spectral efficiency and user experience.



(a) Scenario 1

(b) Scenario 2

Fig. 8. Aggregated network throughput for scenario 1 and scenario 2.

The performance improvements introduced by the proposed approach come with a minor trade-off. As observed in both scenarios, the throughput measurements exhibit a slightly higher variance. This variability can be attributed to the dynamic exploration behavior of the reinforcement learning agent, which continuously adapts cell association decisions based on real-time feedback and fluctuating network states. While this results in marginally less stable instantaneous throughput, it is a natural byproduct of adaptive learning and does not compromise the system's overall advantage.

In summary, the results confirm that the proposed approach successfully balances adaptability and performance, achieving substantial throughput gains while maintaining acceptable stability across repeated experimental trials.

VI. Conclusions

In this paper, we have introduced a novel O-RAN–compliant traffic steering solution that performs bandwidth-aware cell selection for beyond-5G networks. Unlike traditional approaches that rely solely on RSRP/RSRQ-based handover decisions, the proposed solution leverages the hierarchical intelligence of the O-RAN architecture—specifically, the non-RT and near-RT RICs—to incorporate both radio signal quality and cell capacity information into its decision-making process. This enables a more holistic and adaptive form of traffic steering, where users are dynamically associated with the most suitable cell not only based on signal strength but also considering the available spectrum resources and load conditions.

The proposed solution was experimentally validated on a real-world O-RAN testbed, considering both homogeneous and heterogeneous radio conditions. Results demonstrate that the proposed framework can deliver up to a 75% improvement in median throughput in certain traffic configurations, while also enhancing network fairness and resource utilization efficiency across cells.

As part of future work, we plan to extend the decision space by incorporating additional traffic steering parameters, such as QoS requirements, traffic classes, and service-level metrics, thereby moving toward a multi-objective, context-aware cell selection framework. We also intend to take into consideration multi-RATs scenarios. Furthermore, we intend to scale up the testbed to include a larger number of UEs and gNBs.

References

[1] H. Deng, Q. Li, J. Huang, and C. Peng, "ICellSpeed: Increasing Cellular Data Speed with Device-Assisted Cell Selection," in *Proc. of ACM MobiCom*, London, United Kingdom, 2020.

[2] E. Coronado, S. Siddiqui, and R. Riggio, "Roadrunner: O-ran-based cell selection in beyond 5g networks," in *Proc. of NOMS*, Budapest, Hungary, 2022.

[3] C. Gijón, M. Toril, S. Luna-Ramírez, and M. Luisa Marí-Altozano, "A data-driven traffic steering algorithm for optimizing user experience in multi-tier lte networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414–9424, 2019.

[4] L. L. Vy, L.-P. Tung, and B.-S. P. Lin, "Big data and machine learning driven handover management and forecasting," in *Proc. of IEEE CSCN*, Helsinki, Finland, 2017.

[5] M. M. Hasan, S. Kwon, and J.-H. Na, "Adaptive mobility load balancing algorithm for lte small-cell networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2205–2217, 2018.

[6] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5g open ran based on lstm traffic prediction," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7727–7742, 2023.

[7] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures," *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 2882–2897, 2024.

[8] M. A. Habib, H. Zhou, P. E. Iturria-Rivera, M. Elsayed, M. Bavand, R. Gaigalas, S. Furr, and M. Erol-Kantarci, "Traffic steering for 5g multi-rat deployments using deep reinforcement learning," in *Proc. of CCNC*, Las Vegas, NV, USA, 2023.

[9] H. Erdol, X. Wang, P. Li, J. D. Thomas, R. Piechocki, G. Oikonomou, R. Inacio, A. Ahmad, K. Briggs, and S. Kapoor, "Federated meta-learning for traffic steering in o-ran," in *Proc. of VTC2022-Fall*, London, UK, 2022.

[10] M. A. Habib, H. Zhou, P. E. Iturria-Rivera, M. Elsayed, M. Bavand, R. Gaigalas, Y. Ozcan, and M. Erol-Kantarci, "Hierarchical Reinforcement Learning Based Traffic Steering in Multi-RAT 5G Deployments," in *Proc. of ICC 2023*, Rome, Italy, 2023.

[11] D. Castro-Hernandez and R. Paranjape, "Optimization of handover parameters for lte/lte-a in-building systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5260–5273, 2018.

[12] A. Awada, B. Wegmann, I. Viering, and A. Klein, "A son-based algorithm for the optimization of inter-rat handover parameters," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1906–1923, 2013.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[14] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1179–1191.

[15] srsRAN. Access on October, 2025. [Online]. Available: https://srsran.com/

[16] Ettus. Access on October, 2025. [Online]. Available: https://www.ettus.com/

[17] Open5GS. Access on October, 2025. [Online]. Available: https://open5gs.org/