Adaptive Traffic Steering in Open RAN: Integrating Rule-Based Policies with Reinforcement Learning

Utkarsh Sharma*, Hua Wei†, Mingzhe Chen‡, Jie Xu§, and Yuchen Liu*
*North Carolina State University, USA; †Arizona State University, USA;
‡University of Miami, USA; §University of Florida, USA

Abstract—**Open Radio Access Networks (Open RANs) offer a flexible and interoperable wireless network infrastructure to address growing demands for scalability and resource optimization. Current traffic steering and resource allocation approaches use either static policies or machine learning (ML) workflows, each with significant drawbacks—static policies lack adaptability, while ML-based methods can be computationally intensive and inefficient under certain load conditions. To overcome these issues, we propose a hybrid traffic steering mechanism within the Open RAN framework, integrating an enhanced rule-based policy with a global-local reinforcement learning (RL) algorithm. The enhanced rule-based policy is triggered for efficient resource allocation under low network load, encompassing a broader range of User Equipment (UE) slices for finer-grained control. Under high-load variability, the system transitions to the RL-based strategy that learns optimal allocation policies in real-time, factoring in slice types and base station capacities. Evaluation results in a city-scale scenario demonstrate that the proposed adaptive approach significantly improves user satisfaction, reduces unmanaged UEs, and balances cell utilization across varying traffic conditions compared to existing schemes. These findings underscore the potential of combining rule-based policies with reinforcement learning workflows to advance the efficiency and adaptability of Open RANs.**

I. Introduction

As mobile networks become increasingly complex, the shift towards Open Radio Access Networks (RAN) is transforming the landscape of network flexibility and innovation. With its open interfaces and modular architecture, Open RAN aims to deliver enhanced interoperability and cost-efficiency [1]–[3]. However, this flexibility brings challenges, particularly in the dynamic management of network resources to ensure high-quality service, as its open and modular nature requires continuous coordination among multiple components from different vendors. This can lead to complexities in ensuring real-time responsiveness, resource optimization, and consistent performance across diverse network conditions.

One key capability brought by Open RAN's openness and intelligence is on-demand traffic steering, which supports flexible slicing for diverse users and use cases, ensuring that network resources are dynamically allocated where they are most needed [4]. This adaptability is essential for maintaining service quality amidst fluctuating traffic demands. However, existing traffic steering methods [5], which are largely static and rule-based, often fall short in the face of fluctuating network demands and struggle to adapt to rapid changes in user behavior and network load. Some other approaches [6] focus on leveraging Open RAN's RIC (RAN Intelligent Controller) with built-in machine learning (ML) workflows to manage resource allocation and traffic steering. However, these methods often face challenges such as high computational demands, limited generalizability across different network conditions, and delays in model convergence, which can impact the effectiveness of real-time network optimization.

Specifically, recent research in RAN traffic steering has explored various approaches, from reinforcement learning to specialized ML implementations. RL-based methods [7] implemented Random Ensemble Mixture with Conservative Q-learning, showing strong coverage but facing computational demands during initial learning. Similarly, hierarchical methods like h-DQN [8] used bi-level architecture for efficient exploration through meta-controller coordination between RICs, though reliability remained limited under low-load conditions. The O-RAN Alliance has explored UE-specific customization [4], optimizing cell association based on slice-level requirements, but static policies implemented via A1 interface limited adaptability to changing network conditions. Other research focused on ML-based schemes for advanced 5G applications [9], utilizing hierarchical processes combining heuristics and optimization. Approaches like [10] demonstrated significant improvements, surpassing benchmark systems by up to 45.5%, but typically focused on specific traffic types, limiting generalizability across diverse network scenarios. While each approach offers unique advantages, they all face limitations in balancing computational efficiency with adaptability across varying network conditions, highlighting the need for more comprehensive solutions.

Motivated by these gaps, this paper proposes a hybrid approach that combines the robustness of rule-based policies with the flexibility of machine learning workflows in Open RAN systems. This algorithm-learning-system harmony aims to strike a balance between computational efficiency and adaptability across diverse network conditions and traffic types, which aligns with current O-RAN development trends, emphasizing policy-driven network management while leveraging ML within RICs for advanced data processing. Specifically, we first design an enhanced rule-based strategy by incorporating a broader range of User Equipment (UE) slices and adding an additional layer of base station (BS) known as anchored cells, which connect to the internet core. This inclusion goes beyond typical pico and macro cells, providing more granular policy-level control over network resources within the non-RT RIC. Subsequently, we develop a novel RL-based approach within the near-RT RIC to accommodate specific traffic types and adjust decisions dynamically. With a novel global-local reward function, the network takes into account both system load balancing and user satisfaction, where the global aspect of the reward ensures that network resources are distributed to maintain overall balance across heterogeneous BSs. Meanwhile, the local aspect focuses on individual user satisfaction, ensuring that specific Quality of Service (QoS) requirements are met, such as obtaining their preferred cell type or ensuring minimal latency. As a result, the rule-based policy and the RL approach are finally integrated to form a hybrid traffic steering mechanism, ensuring a more holistic and responsive traffic steering strategy. Under low network load conditions, the enhanced rule-based policy efficiently allocates resources with minimal computational overhead. In medium to high load scenarios, the system switches to our dual-focus RL-based strategy that optimizes allocation policies on-the-fly by considering UE types and current BS capacities. Extensive simulation results demonstrate that our adaptive mechanism minimizes unmanaged UEs and maximizes the percentage of UEs obtaining their preferred cell types, while effectively balancing the load across the network.

II. Enhanced Rule-Based Policy for Traffic Steering

This section introduces an enhanced rule-based policy for traffic steering in Open RAN environments, extending beyond conventional approaches [1]. The proposed policy incorporates slice-aware allocation strategies with anchored cells, providing more granular control over network resources and serving as the foundation for the subsequent hybrid approach.

Unlike existing rule-based policies that typically consider only basic UE types and conventional BS infrastructure [5], [11], our approach incorporates a diverse range of network slices and a new layering of deployed BSs (macro, pico, and anchored cells). The proposed policy maps each network slice to an ordered preference list of BS types, tailored to optimize specific service requirements.

Specifically, for voice services, macro cells are prioritized for wide coverage (i.e. first preference), followed by pico and anchored cells. Enhanced Mobile Broadband (eMBB) slice favors pico cells for higher throughput, while Ultra-Reliable Low-Latency Communications (URLLC) slice prioritizes anchored cells for minimal latency. Massive Machine Type Communications (mMTC) slice optimizes dense device deployment by preferring pico cells, with a fallback to anchored and macro cells as needed. This structured preference-based allocation enables an effective alignment between UE service requirements and network resources. Based on this network structure and configuration, a *proximity-aware allocation algorithm* is developed for slice-aware traffic steering, taking into account both slice preferences and the geographical distribution of BSs [10]. Unlike traditional methods, this approach integrates distance-based sorting within each preference tier, optimizing resource utilization while adhering to service quality requirements.

Algorithm 1 presents the proximity-aware allocation mechanism. For each UE in the network, the algorithm begins by identifying its slice type and retrieving the ordered list of BS preferences (Lines 2-3). It then iterates over each preferred BS type, collecting all stations of that type and sorting them by proximity to the UE (Lines 6-7). This proximity-based sorting is crucial for minimizing potential handovers and reducing network overhead. For each sorted BS, the algorithm checks if there is sufficient capacity to accommodate the UE's throughput requirements (Lines 8-13). Upon matching a suitable BS, the UE is immediately associated, and the algorithm exits both inner loops to process the next UE (Lines 10-12). If no allocation is possible after all preference tiers are exhausted, the UE is marked as unmanaged (Lines 17-19).

Algorithm 1 Rule-Based Allocation with Proximity Sorting

```
 1: for each UE in UE_list do
 2:     $slice \leftarrow UE.slice\_type$
 3:     $preferences \leftarrow policy[slice]$
 4:     $allocated \leftarrow False$
 5:     for each $type$ in $preferences$ do
 6:         $stations \leftarrow get\_stations(type)$
 7:         $sorted\_stations \leftarrow sort\_by\_proximity(UE,$
                $stations)$
 8:         for each $bs$ in $sorted\_stations$ do
 9:             if $bs$ has capacity then
10:                 allocate $UE$ to $bs$
11:                 $allocated \leftarrow True$
12:                 break inner loops
13:             end if
14:         end for
15:         if $allocated$ then
16:             break
17:         end if
18:     end for
19:     if not $allocated$ then
20:         mark $UE$ as unmanaged
21:     end if
22: end for
```

Such an enhanced rule-based policy introduces significant advancements over traditional methods [9], [11] by incorporating several key features. First, it ensures slice-aware allocation, prioritizing service requirements across network slices for tailored resource distribution based on QoS needs. Second, the integration of anchored cells in the loop adds an extra resource tier optimized for latency-sensitive applications, while proximity-based optimization within each preference tier reduces unnecessary handovers and network overhead. Despite these benefits, the policy, like all rule-based approaches, experiences performance degradation under high load conditions. This limitation highlights the need for adaptive, learnable strategies, which will be explored in the subsequent section.

III. Joint Rule-Based and Reinforcement Learning Methodology

Building upon the enhanced rule-based policy, this section integrates a reinforcement learning (RL) workflow to develop a robust traffic steering solution. This joint methodology harnesses the strengths of both frameworks: the computational efficiency and reliability of rule-based mechanisms, and the adaptive learning capabilities of RL for managing complex, high-load situations.

A. Reinforcement Learning-Based Approach

Our RL approach enables the RIC to determine UE association strategies such that the network can jointly optimize the BS capacity and user experiences. The main components of the proposed RL and training process are detailed as follows.

State and Action Spaces The state space is designed to capture both network conditions (BS load) and UE characteristics (traffic demand and mobility). Each state s consists of a $(n+k)$-dimensional vector, where n is the number of BSs. The first n components represent normalized cell capacities (i.e. C_{cur}/C_{max}) for each BS, providing the agent (RIC) with current resource availability information. The remaining k components in the vector include the normalized traffic demands of the current UE (T_{UE}/T_{max}) and a one-hot encoded representation of the UE's slice type (e.g., Voice, eMBB, URLLC, and mMTC). This state representation ensures the agent has complete view of both network resources and UE requirements. The action space a is discrete, consisting of n possible actions, with each action representing the selection of a specific BS for the UE association. This formulation enables the agent to make timely allocation decisions while considering current network state and UE dynamics.

Global-Local Reward Function The reward function for the adaptive traffic steering combines local user-specific rewards with global system-level rewards:

$$R(s,a) = \underbrace{R_{alloc}(s,a) + R_{pref}(s,a)}_{\text{Local User Reward}} + \underbrace{R_{load}(s,a)}_{\text{Global System Reward}}, \quad (1)$$

where the allocation component R_{alloc} imposes a penalty γ when a UE remains unmanaged, incentivizing the agent to maximize successful resource allocations across the available BSs. The preference satisfaction component R_{pref} rewards successful allocations based on the UE satisfactory level:

$$R_{pref}(s,a) = \alpha_i, \quad i \in [1,m], \quad (2)$$

where m is the number of preference levels based on the rule-based policy in Sec. II, and $\alpha_i > \alpha_{i+1}$ for all $i \in [1, m-1]$, ensuring higher rewards for better preference satisfaction. From the global view, the load balancing component R_{load} considers network-wide utilization as:

$$R_{load}(s,a) = \begin{cases} \beta_1 & \text{if } \max_{b\in B}(L_b) > \tau, \quad \beta_1 < 0 \\ \beta_2 & \text{otherwise}, \quad \beta_2 > 0, \end{cases} \quad (3)$$

where L_b represents the normalized load of BS b, i.e.

$$L_b = \frac{C^b_{max} - C^b_{cur}}{C^b_{max}}. \quad (4)$$

In Eq. (3), τ represents the preferred maximum load threshold (typically set at 0.95 in practice), beyond which the system applies penalties to prevent BS overloading. This threshold helps maintain operational stability and ensures sufficient capacity buffer to handle unexpected traffic spikes.

This reward structure will guide the following Q function to optimize network performance by minimizing allocation failures through the penalty component γ, prioritizing preference satisfaction via the hierarchical rewards α_i in Eq. (2), and maintaining balanced cell utilization through the load-based rewards β_1 and β_2. Such a combination ensures that the agent learns policies that simultaneously satisfy individual UE requirements while maintaining optimal system-wide resource distribution, effectively preventing both underutilization and overloading of network resources.

Dueling DQN Architecture We introduce a Dueling Deep Q-Network (DQN) architecture to overcome limitations observed in standard DQN models. Traditional DQNs use a single stream to estimate the Q-value, representing the expected future rewards of an action taken in a given state. This unified approach can contribute to overestimation bias and unstable training dynamics. The Dueling DQN architecture mitigates these issues by separating the estimation of the state value from the action advantages. This separation allows for more precise value function estimation, improving stability and more efficient learning, especially in our considered dynamic wireless environments. The Dueling DQN architecture introduces two separate streams within the neural network:

- *Value Stream ($V(s)$)*: Estimates the value of being in a particular state s.
- *Advantage Stream ($A(s,a)$)*: Computes the advantage of each action a relative to others in state s.

As a result, the Q-value is then combined as:

$$Q(s,a) = V(s) + A(s,a). \quad (5)$$

This separation allows the model to stabilize the learning process by reducing variance in the estimations of $V(s)$ and $A(s,a)$ while minimizing overestimation. By independently assessing state values and action advantages, the Dueling DQN enhances the agent's ability to discern available cell capacity from advantageous actions, ensuring more effective resource allocation and optimized traffic distribution.

Training and Convergence The training process for the global-local RL model starts by capturing both cell resources and UE requirements, the discrete action space for selecting BS, and the reward mechanism in Eq. (1) to incentivize balanced network load and efficient resource usage. The agent is trained in a dynamic wireless environment, allowing it to explore and learn traffic steering strategies in a controlled setting. The training leverages experience replay, which stores past experiences in a replay buffer and samples mini-batches for training to break the temporal correlation between consecutive samples and stabilize training. The loss function used in training is the Mean Squared Error (MSE) between the predicted and target Q-values:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \eta \max_{a'} Q'(s',a';\theta^-) - Q(s,a;\theta)\right)^2\right]$$

where r is the immediate reward, η is the discount factor, s' is the next state, and Q' represents the target network's action-value function, θ are the parameters of the primary network, and θ^- are the parameters of the target network, updated periodically for stability.

To demonstrate the effectiveness of our training process, the agent needs to be trained over multiple episodes, with the training loss and average Q-values monitored to evaluate learning progress. Fig. 1 shows the stabilization of our average Q-values as training advances, indicating that the learned policy is converging. The steady increase and eventual plateau in Q-values suggest that the agent has effectively learned optimal strategies, validating the choice of our Dueling DQN architecture and the designed reward function in driving policy learning to enhance both user-level satisfaction and network-level performance. However, compared to the rule-based policy, we observe that it requires substantial computational resources and introduces additional overhead for the model to converge to an effective policy, making it less efficient in certain scenarios.

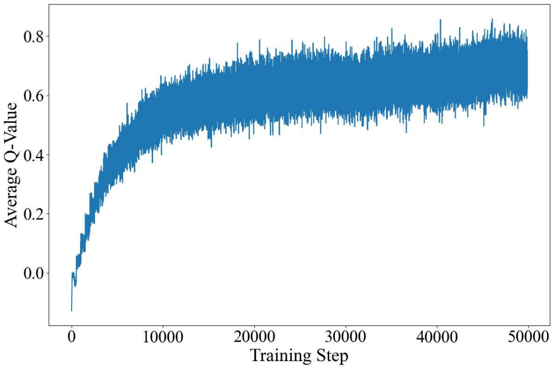

Fig. 1: Average Q-values of the RL agent during training.

B. Hybrid Rule-based and RL Approach

As discussed in Sec. II and Sec. III.A, recognizing that neither method alone can optimally address all network conditions, a hybrid approach is proposed. This approach leverages the strengths of both methods: utilizing the rule-based policy's efficiency and stability under low network load and the RL-based policy's adaptability under high-load scenarios. By dynamically switching between the two based on real-time network conditions, the hybrid approach aims to optimize user-level and network-wide performance across various scenarios.

The hybrid approach monitors several key network metrics, such as current network load and BS utilization, to determine which policy to apply at any given time. Then, a threshold-based mechanism is employed to decide when to switch between the rule-based and RL-Based policies. To avoid frequent switching due to minor fluctuations in network load. The detailed decision-making process is outlined in Algorithm 2.

Specifically, the algorithm begins by initializing the current policy to our rule-based scheme from Sec. II and defining two thresholds, high load threshold Th_{high} and low load threshold Th_{low}, where $Th_{low} < Th_{high}$ to create a hysteresis band. It then enters a continuous monitoring loop where the network load is regularly assessed. If operating under the rule-based policy and the current network load exceeds Th_{high}, the system switches to the RL-model; conversely, if the RL-model is adopted and the network load drops below Th_{low}, it switches back to the rule-based policy. Upon selecting the policy, the algorithm applies the corresponding scheme for traffic steering and waits for a predefined monitoring interval before re-evaluating, thereby reducing unnecessary computational overhead. Such a load-aware mechanism can dynamically adjust to changing network conditions, balancing adaptability, efficiency, and stability across varying scenarios.

Algorithm 2 Load-Aware Traffic Steering Algorithm

```
Initialize $current\_policy \leftarrow$ Rule-Based
Initialize $Th_{high}, Th_{low}$
while network is operational do
    $network\_load \leftarrow$ calculate_network_load()
    if $current\_policy$ is Rule-Based then
        if $network\_load > Th_{high}$ then
            $current\_policy \leftarrow$ RL-Based
        end if
    else if $current\_policy$ is RL-Based then
        if $network\_load < Th_{low}$ then
            $current\_policy \leftarrow$ Rule-Based
        end if
    end if
    apply_$current\_policy$()
    wait for monitoring interval
end while
```

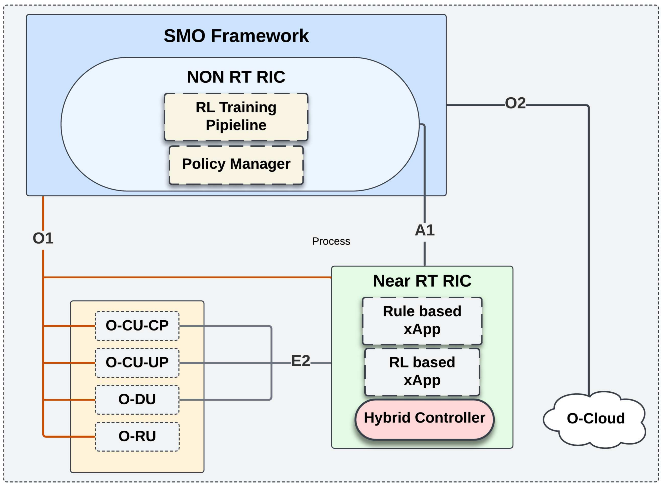

Fig. 2: Implementation of the proposed hybrid traffic steering solution in Open RAN Architecture. The Non-RT RIC houses the RL training pipeline and a policy manager, while the Near-RT RIC contains the rule-based xApp (Algorithm 1), RL-based xAPP (Sec. III.A), and a hybrid controller (Algorithm 2). The three-tiered control loop structure ensures optimal response times across different network conditions.

C. Implementation in O-RAN Architecture

The proposed hybrid solution integrates seamlessly within the O-RAN architecture, primarily operating through the Near-RT RIC and Non-RT RIC components, as illustrated in Figure 2. First, the rule-based component operates as a specialized xApp within the Near-RT RIC, leveraging direct access to RAN metrics through the E2 interface for real-time monitoring of cell loads and UE distributions. This placement enables immediate execution of traffic steering decisions, ensuring low-latency response during normal load conditions. On the other hand, our RL-based component implements a distributed architecture across both RIC layers, with offline model training executed in the Non-RT RIC to leverage its inherent ML workflow capabilities, while policy execution is deployed as another xApp in the Near-RT RIC. This design enables efficient state collection and action implementation through the E2 interface, maintaining the system's ability to adapt to changing network conditions while adhering to O-RAN's control loop specifications.

Specifically, the implementation leverages O-RAN's key interfaces effectively, with the E2 interface serving as the primary channel for collecting the network metrics (e.g., user mobility, traffic demands, cell load) and enabling the hybrid controller to make informed decisions about which policy to apply based on current conditions. The O1 interface handles performance management and policy switching, while the A1 interface enables seamless policy updates (either rule-based or RL-based) from Non-RT RIC to Near-RT RIC. This comprehensive interface utilization ensures robust communication between different architectural components while maintaining the flexibility required for our adaptive traffic steering.

IV. Evaluation Results

This section presents the evaluation of the our traffic steering approaches using defined key performance indicators (KPIs), including the percentage of unmanaged UEs [12], user satisfaction rate [13], and cell utilization. The performance is analyzed under varying network load scenarios to assess the effectiveness of the proposed approach. Computational analysis is also included to demonstrate the feasibility of the proposed mechanism in a real-world operating RAN scenario.

A. Experimental Setup

Simulations with varying numbers of UEs were conducted, each with specific data demands, to create different load scenarios: Small Load System (SLS) with 50 UEs, Medium Load System 1 (MLS-1) with 250 UEs, Medium Load System 2 (MLS-2) with 350 UEs, and Large Load System (LLS) with 1,000 UEs. The BS capacities and UE preferences are configured to reflect realistic network conditions.

According to Algorithm 2, the thresholds for the hybrid approach are determined based on the observed performance crossover points between rule-based and rL-Bbsed approaches. At 50 UEs (20% of cell capacity) and 250 UEs (100% cell capacity), the rule-Based Approach performed optimally with minimal unmanaged UEs and high preference satisfaction. At 350 UEs (over 140% cell capacity), a performance crossover is observed where the RL-based approach starts to outperform the rule-based approach due to the high dynamics and traffic uncertainty, showing improved UE management and preference satisfaction. Based on these initial observations, the thresholds are set at 140% (Th_{low}) and 160% (Th_{high}) of cell capacity, with the hysteresis band preventing oscillation between policies during load fluctuations.

B. Performance Results and Analysis

Table I provides a comprehensive comparison of the three approaches across various load scenarios. The highlighted cells indicate the policy adopted by the hybrid approach in each scenario, with rule-based values emphasized for SLS and MLS-1, and RL-based values emphasized for MLS-2 and LLS. The dotted line represents the transition point between the two policies, marking where the hybrid approach switches from rule-based to RL-based control to optimize performance.

Table I: Performance comparison of traffic steering approaches.

Metric	Load Scenario	Rule-Based	RL-Based	Hybrid
Unmanaged UEs (%)	SLS (50 UEs)	0%	0%	0%
	MLS-1 (250 UEs)	0%	0.8%	0%
	MLS-2 (350 UEs)	9.1%	8.6%	8.6%
	LLS (1000 UEs)	65%	40.2%	40.2%
First Preference Satisfaction (%)	SLS	100%	98%	100%
	MLS-1	92.8%	94.4%	92.8%
	MLS-2	77.1%	88%	88%
	LLS	27%	58.6%	58.6%
Average Cell Utilization (%)	SLS	19.6%	19.3%	19.6%
	MLS-1	92.4%	87.9%	92.4%
	MLS-2	99.8%	92.7%	92.7%
	LLS	100%	96.1%	96.1%

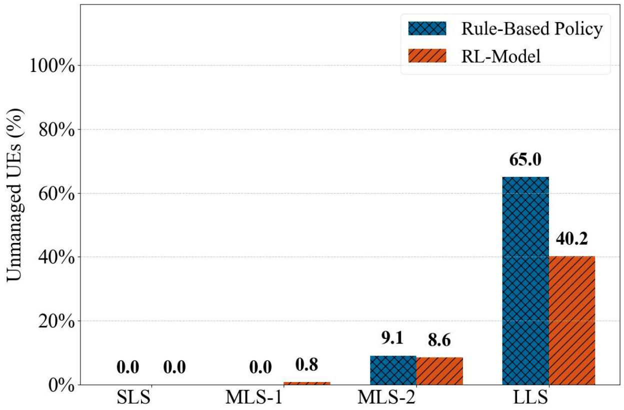

Fig. 3: Comparison of unmanaged UEs across different scenarios.

Specifically, Fig. 3 illustrates the percentage of unmanaged UEs across different load scenarios. The results indicate that while both approaches perform similarly under low-load conditions, the RL-based approach demonstrates significantly better UE management under high-load conditions. This is particularly evident in the LLS scenario, where the RL-based approach reduces unmanaged UEs by 24.8% compared to the rule-based method. It is important to note that the LLS case serves to evaluate the worst-case scenario with an extremely high load relative to the available cell capacity.

Next, Fig. 4 demonstrates the UE first preference satisfaction rates (as defined in Sec. II) across different load scenarios. As expected, the rule-based approach performs better in SLS scenarios, but the RL-based approach maintains substantially higher satisfaction rates as the network load increases, more than doubling the satisfaction rate in the LLS scenario (58.6% compared to 27% for the rule-based approach). In addition to user performance, Fig. 5 evaluates cell utilization patterns from a network-wide perspective. The results show that the RL-based approach achieves a more balanced resource distribution across BSs, particularly in high-load (LLS) scenarios, where

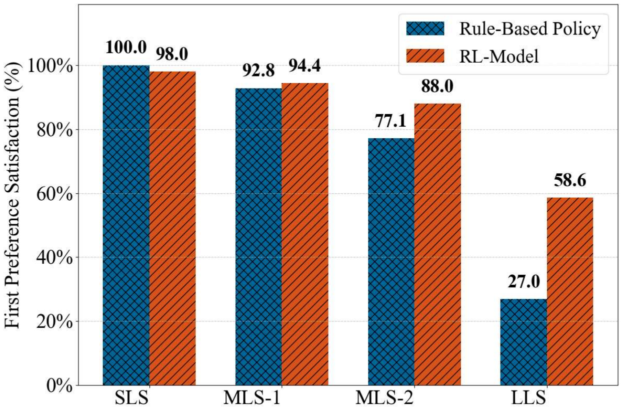

Fig. 4: Comparison of user's first preference satisfaction rate across approaches.

it maintains utilization levels below 97%, preventing complete saturation while effectively distributing the load. The balanced cell utilization also confirms the effectiveness of our reward function design in promoting efficient resource distribution while maintaining high user satisfaction rates. By harnessing the strength of both schemes as shown in Table I, our hybrid approach can strategically switch between the two methods, providing adaptive user and system performance tailored to varying network conditions.

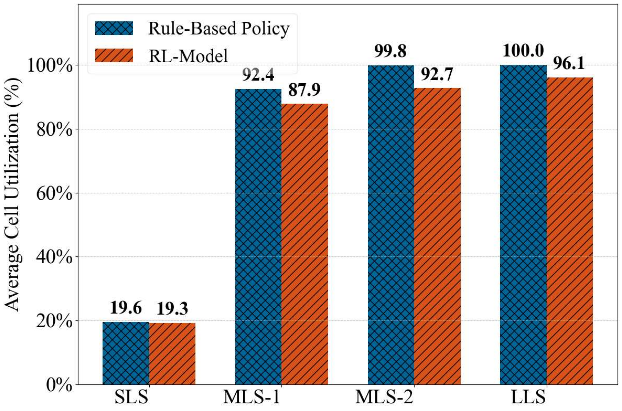

Fig. 5: Average cell utilization across different load scenarios.

C. Computational Analysis

The computational testing demonstrates the feasibility of our hybrid approach within O-RAN's operational time windows. The training phase is planned to be executed in an r-App in the non-RT RIC (>1s timescale) as shown in Fig. 2, completed in 160.74 seconds experimentally. On the other hand, the inference that will happen in the traffic steering x-App averaged approximately 180ms per traffic steering task, well within the operational window of x-Apps in the near-RT RIC (10ms-1s timescale). This timing profile ensures our hybrid policy can make real-time decisions for each UE requesting network access without violating the timing constraints of RIC workflows.

V. Conclusion

This paper presents a novel approach for traffic steering in Open RAN that effectively combines the reliability of rule-based policies with the adaptability of RL. The enhanced rule-based component goes beyond conventional methods by incorporating diverse UE slices and introducing anchored cells, enabling more precise resource allocation. Meanwhile, the RL component, implemented using a Dueling DQN architecture, demonstrates superior performance in high-load scenarios. The separation of training in non-RT RIC and inference in near-RT RIC demonstrates that our hybrid approach is both architecturally sound and computationally feasible within operational timing constraints. The proposed hybrid mechanism successfully leverages the complementary strengths of both approaches, ensuring optimal resource utilization across varying network conditions. This is evidenced by balanced cell utilization patterns and consistent performance metrics across all load scenarios. Future work will focus on implementing and validating this approach in practical network environments using a testbed like AERPAW at North Carolina State University, which will provide deeper insights into real-world performance and scalability characteristics.

Acknowledgement

This research was supported by the National Science Foundation through Award CNS–2312138, CNS–2312139, CNS-2332834 and IIS-2421839.

References

[1] O-RAN Alliance, "O-RAN architecture description," O-RAN Alliance, Technical Specification O-RAN.WG1.O-RAN-Architecture-Description-v05.00, 2023.

[2] M. Mushi, Y. Liu, S. Sreenivasa, O. Ozdemir, I. Guvenc, M. Sichitiu, R. Dutta, and R. Gyurek, "Open ran testbeds with controlled air mobility," *Computer Communications*, vol. 228, p. 107955, 2024.

[3] M. S. Wani, M. Kretschmer, B. Schröder, A. Grebe, and M. Rademacher, "Open ran: A concise overview," *IEEE Open Journal of the Communications Society*, pp. 1–1, 2024.

[4] O-RAN Alliance, "O-RAN use cases and deployment scenarios: Towards open and smart RAN," O-RAN Alliance, Technical Report, 2023.

[5] P. Kumar, Y. Wang *et al.*, "Load balancing and resource allocation in Open RAN: A deep learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2076–2090, 2021.

[6] R. Singh, S. Prakash *et al.*, "A comprehensive survey of reinforcement learning applications in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1050–1091, 2022.

[7] M. Lacava, R. Soos *et al.*, "A random ensemble mixture approach for traffic steering in Open RAN networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4228–4241, 2023.

[8] X. Chen, W. Li *et al.*, "Hierarchical deep Q-learning for traffic steering optimization in 5G networks," in *Proc. IEEE INFOCOM*, 2023, pp. 1–9.

[9] K. Zhang, K. Yang *et al.*, "URLLC-aware traffic steering using naive Bayes and deep Q-learning in Open RAN," *IEEE Communications Letters*, vol. 25, no. 3, pp. 891–895, 2023.

[10] M. Wilson, R. Brown *et al.*, "Traffic steering mechanisms for massive machine-type communications in 5G networks," *IEEE Access*, vol. 9, pp. 45 261–45 275, 2021.

[11] A. Martinez, T. Johnson *et al.*, "Preference-aware user association in heterogeneous cellular networks," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1089–1092, 2021.

[12] A. Hegde, S. Lobo, and A. Festag, "Cellular-v2x for vulnerable road user protection in cooperative its," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2022, pp. 118–123.

[13] J. Zhang and X. Li, "User optimum locations in cellular networks for tradeoff of throughput and user satisfaction," in *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2023, pp. 1005–1010.