Reinforcement Learning based traffic steering in Open Radio Access Network (ORAN)

Arnab Gain
Department of Computer Science and Engineering
Cooch Behar Government Engineering College
Cooch Behar, India
arnabgaincgec@gmail.com

Shankar K. Ghosh
Department of Computer Science and Engineering
Shiv Nadar Institution of Eminence
Delhi, NCR
shankar.ghosh@snu.edu.in

Aaradhy Sharma
Department of Computer Science and Engineering
Shiv Nadar Institution of Eminence
Delhi, NCR
as783@snu.edu.in

***Abstract*—Open Radio Access Network (O-RAN) offers significant flexibility to overcome the existing challenges in traditional Radio Access Network-based 5G networks, such as vendor lock-in, inefficient load balancing among gNBs (Next Generation Node Bs), and lack of intelligent decision-making related to data transfer, with its vendor-neutral and interoperable components. In this work, we propose a joint optimization framework for traffic steering and resource allocation in an O-RAN environment. The novelty of our approach lies in its implementation within a RAN where each gNB has one or more link-beams, resource block (RB) sharing is allowed, and dual connectivity for user equipment (UE) is supported. We formulate the optimization problem with an Integer Linear Programming (ILP) model to maximize both the number of UEs achieving high throughput and the number of gNBs maintaining balanced loads. To efficiently solve this problem, we develop a Deep Reinforcement Learning (DRL)-based algorithm and evaluate its performance through comprehensive system-level simulations. The results demonstrate that our proposed method achieves a 94% performance gain and significantly outperforms existing approaches in terms of system throughput and load distribution.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The complexity of modern cellular networks is steadily increasing due to the integration of heterogeneous networks (HetNets) systems such as Long Term Evolution (LTE), Fifth Generation (5G) New Radio (NR), and Wireless Local Area Networks (WLANs). These networks are expected to support a wide range of services, including enhanced Mobile Broadband (eMBB), ultra-Reliable Low Latency Communication (uRLLC), and massive Machine-Type Communications (mMTC). The traditional radio access network (RAN) has been used for wireless communications till now; however, it is based on a single-vendor closed interface architecture. For this reason, it has disadvantages, including limited flexibility, poor interoperability among its components, lack of support for multi-vendor integration, and the inability to enable intelligent, service-adaptive RAN functionalities. As a result, it lacks the intelligence needed for efficient load balancing and resource allocation at the Next Generation Node B (gNB) level in the case of modern wireless communication. These disadvantages hinder the delivery of high-speed and low-latency services to rapidly increasing numbers of user equipment (UE) in dynamically changing network environments of wireless networks. In contrast to traditional RAN, Open Radio Access Network (O-RAN) allows components from different vendors to work together. It overcomes the limitations of traditional RAN due to its open, disaggregated, and interoperable architecture which is more flexible than traditional RAN. The main components of ORAN include the ORAN Central Unit (O-CU), ORAN Distributed Unit (O-DU), ORAN Radio Unit (O-RU), and the RAN Intelligent Controller (RIC) [1]. Among these components, the O-CU is responsible for data routing and management of Quality of Service (QoS). The O-DU handles functions such as scheduling and retransmissions. The O-RU focuses on digital signal processing tasks, including modulation, beamforming, and radio frequency (RF) operations. RIC can furthermore be divided into two sub-components: Non-Real-Time RIC (Non-RT RIC) and Near-Real-Time RIC (Near-RT RIC). The Non-RT RIC provides policy-based guidance, supports the training of machine learning (ML) models, and manages long-term optimization tasks (with time scales greater than one second). In contrast, the Near-RT RIC performs control and optimization tasks within time scales ranging from 10 milliseconds to 1 second, such as dynamic resource allocation and interference management. With the help of RIC, O-RAN takes intelligent decisions regarding efficient load balancing and resource allocation. In recent years, ML has shown great potential in addressing challenges in wireless communication, particularly in achieving high-speed and low-latency services. For this reason, our work focuses on reinforcement learning (RL), a type of ML that is particularly well-suited for dynamically changing environments. However, implementing ML-based algorithms in traditional RAN is difficult due to its single vendor-based closed interface architecture. In contrast,

the open and interoperable architecture of O-RAN is very helpful for implementing ML-based algorithms. Therefore, we adopt the O-RAN framework to effectively deploy our RL-based approach.

While delivering a wide range of services to users, O-RAN encounters several challenges. Among these, two of the most significant challenges are traffic steering (TS) and resource allocation (RA). RA is the process of assigning and distributing available network resources (e.g. bandwidth, spectrum, power, and computing resources) to different users, services, or applications to ensure optimal performance. TS [14], [17] is a process of intelligently directing or routing network traffic flows to specific network resources or paths on the basis of various factors such as network conditions, QoS requirements, user demands, loads on the base stations(BS)s or optimization objectives. The objective of our work is to ensure high throughput for all UEs under a RAN. Efficient TS plays a vital role in achieving this goal. ML-based algorithms — particularly RL are highly effective in optimizing TS performance. In this proposed work, we have implemented an RL-based algorithm for TS in an O-RAN environment, focusing on Use Case 5 as defined by O-RAN Working Group 1 in Release 15 [20], [21].

Modern cellular networks operate in a highly dynamic environment, with the values of RAN-related factors changing very frequently over time. As a result, maintaining optimized performance of TS in such an environment is very difficult. However, studies have shown that ML-based TS algorithms can significantly improve performance under these conditions. Moreover, ML can efficiently balance the load across O-DUs and reduce the frequency of unnecessary handovers during TS in a network. [2] addressed these objectives using a Support Vector Machine (SVM) combined with an ensemble Long Short-Term Memory (e-LSTM) model, while [3] employed a standard Long Short-Term Memory (LSTM) approach. The shortcoming of their works is the lack of support for heterogeneous and adaptable RAN environments. To address this, [5] proposed a Federated Meta Learning (FML) approach that enabled a locally adaptable model at distributed network nodes. In RAN environments, making optimal decision sequentially is very important. However, the approach taken by [5] lacks this capability. This shortcoming is mitigated in [6]–[8], which employed reinforcement learning-based models—namely Conservative Q-Learning (CQL), Deep Reinforcement Learning (DRL), and Markov Decision Processes(MDP), respectively—that performs well in this kind of task. With the rapid growth in user demand and service requirements, the use of beam-based network and dual-connectivity in O-RAN has become essential to ensure enhanced coverage, higher capacity, real-time adaptability, and better performance. It has been seen that none of [2], [3], [5]–[8] has explicitly considered beam-based network and dual connectivity in their works.

The main novelty of our work is that we have jointly optimized TS and RA. Specifically, we consider link-beams for each gNB, allow resource blocks (RBs) to be shared, and support dual connectivity. Our main contributions are:

- We have formulated the problem as an Integer Linear Programming (ILP) model, with the objective of jointly maximizing the number of UEs achieving high throughput and the number of gNBs maintaining balanced load distribution.
- Based on ILP we have proposed a DRL based traffic steering algorithm.
- We have evaluated the DRL using extensive system-level simulation and we have shown that our result is better than existing system.

The rest of the paper is organized as follows. Section II presents a survey of related work and Section III describes the system model. We have done problem formulation in section IV and demonstrated the proposed approach in section V. In section VI we have discussed about the experimental setup and results. Finally, the paper is concluded in section VII.

II. Related works

O-RAN Alliance Work Group 1 has stipulated some use cases [20], [21]. TS is one of them. TS is crucial in O-RAN because it enables intelligent, dynamic and programmable control over how user traffic is routed through the network. Most TS problems are classified as NP-hard, making them computationally challenging to solve in real-world scenarios. ML techniques have demonstrated significantly greater efficiency in addressing these complex problems as compared to traditional rule-based approaches.

Authors in [2] proposed a ML-based TS technique that achieved a uniform load distribution of UEs, with fewer handovers and improved throughput. Therein, SVM and e-LSTM have been used to identify UEs with low throughput and to predict cell throughput. Authors in [3] has decomposed the resource optimization problem into two sub-problems: a long-term sub-problem and a short-term sub-problem. To address the long-term sub-problem, they employ a LSTM model to predict dynamic traffic demands, enable efficient RAN slicing and make flow-splitting decisions. These predictions lead to a non-convex short-term sub-problem, which is subsequently transformed into a computationally tractable form using successive convex approximation (SCA) techniques. Authors in [5] propose a FML based algorithm that enables RICs to effectively manage Radio Access Technology (RAT) allocation in a distributed and adaptive manner. Authors in [6] have developed a cloud-native near-RT RIC and a custom xApp that leverages a data-driven methodology for TS optimization in 5G networks. Therein, a user-centric handover management technique has been proposed, which demonstrates substantial gains in both throughput and spectral efficiency relative as compared to conventional techniques. [7] proposed a three-step hierarchical process that integrates heuristic methods, DRL and Convex Optimization to jointly address flow-split distribution, congestion control and RA respectively. Authors in [8] have formulated the problem of Network Selection and TS as an MDP, aiming to enrich Quality of Experience (QoE), ensuring QoS and achieving effective load balancing. Therein, a Q-Learning-based control mechanism has been

proposed to achieve the aforesaid objectives. Authors in [15] has developed a TS framework and investigated its interaction with carrier aggregation (CA) in HetNet scenario. Authors in [10] have proposed four TS algorithms. Among them two algorithms, i.e., RA and proportional load balancing algorithm are categorized as static, as they rely on preconfigured network parameters. The remaining two algorithms, i.e., user load adaptive and user throughput adaptive are dynamic in nature, utilizing real-time network state information. Therein, dynamic TS strategies have been shown to outperform static counterparts. [11] introduced a joint flow-split distribution, congestion control and scheduling (JFCS) framework to enable intelligent TS in O-RAN, aiming fast convergence, long-term utility and to reduce latency. Despite utility and delay improvements, the JFCS framework faces challenges in real-time deployment due to high computational demands and limited adaptability to unpredictable traffic scenarios. Authors in [12] propose a data-driven self-tuning algorithm for TS aimed at enhancing overall QoE in multi-carrier LTE networks. The algorithm adjusts inter-frequency handover margins based on Reference Signal Received Quality measurements. Therein, the QoE indicator accounts connection traces, which captures the impact of handovers on user experience. While the algorithm improves QoE, it may not always balance QoE and efficient use of network resources in every situation. The fuzzy logic-based algorithm has been proposed in [13] to optimize network parameters for TS, enabling effective load balancing across different network layers while considering user-specific QoS requirements. This algorithm suffers from increased signalling costs associated with the higher number of handovers and ping-pong effect. Authors in [14] propose a joint resource allocation framework that maximizes throughput and minimizes latency by solving the optimization problem accounting QoS, power and front-haul limitations. To meet the stringent latency and reliability users connect to multiple BSs simultaneously. Network slicing is employed to dynamically allocate resources between bandwidth hungry and delay-stringent application. An iterative algorithm based on SCA is proposed to efficiently solve the optimization problem. However, this work assume ideal channel conditions, which may not always reflect real-world scenarios where interference and fading can significantly impact performance. Authors in [15] propose a user-specific priority adjustment scheme based on composite available capacity and radio conditions, communicated through RRC CONNECTION RELEASE signalling for precise TS. Further, it has been shown that introducing dedicated priority validity timers and UE behaviour prediction improve responsiveness and effectiveness during periods of user inactivity. The primary shortcoming of this work is its reliance on real-time signalling, which may introduce delays in priority updates during rapid load fluctuations. Authors in [16] optimizes eMBB user experience by selecting key performance metrics like RSRQ and Physical Resource Blocks, choosing the strongest component carriers based on RSRP, and distributing traffic heterogeneously according to network conditions. This approach dynamically allocates resources to each CC in real time to enhance throughput and maximize QoE, resulting in higher mean opinion scores (MOS) and improved user satisfaction. The shortcoming of this technique is its reliance on the quality metrics, which may not always be optimal for all types of services, particularly those with varying latency requirements. Authors in [17] highlight TS as a key method to intelligently manage user traffic across different network units and spectrum bands, helping improve resource usage and user satisfaction. To ensure LTE and Wi-Fi coexistence, techniques like carrier sensing adaptive transmission and listen before talk have been recommended. This approach demonstrate the benefits of implementing TS in LTE networks utilizing unlicensed bands. Shortcoming of this work is the the need for continuous information exchange regarding load and capacity among cells which may lead to increased overhead and potential inefficiencies. Authors in [18] developed the Energy Savings Management Control (ESMC) rApp, an AI-powered system that continuously monitors and analyses real-time and historical traffic data to make smart decisions about load distribution, to minimize operational cost. It also works with the TS xApp to smoothly offload users from cells marked for energy saving, and tests using Vodafone's dataset showed it could cut energy use significantly 25% without sacrificing network accessibility. A shortcoming of their work is that it mainly focuses on accessibility, potentially overlooking other important QoE metrics like latency and user satisfaction. Including these aspects could offer a more complete understanding of how energy-saving strategies truly affect the overall user experience. [19] proposed a TS algorithm namely compute aware distributed scheduling (CArDS), aiming to optimize service request distribution based on the compute capabilities of service instances. This approach uses a weighted round-robin approach to allocate more traffic to instances with higher processing power, ensuring better load balancing and resource utilization.

The optimal TS mechanism should answer the question that how to allocate a particular RB to an UE through a link beam of O-RU associated with a O-DU. The TS mechanism should also consider the downlink SINR which in turn depends on traffic load, user association and handover management. However,the existing TS mechanisms do not consider these aspects in a disaggregated architecture.

III. System model

In this proposed system, it is considered that there are n O-DUs (indexed by $j = 1, ..., n$) and o O-RUs (indexed by $k = 1, ..., o$) to serve m UEs (indexed by $i = 1, ..., m$), where $m > n$. Suppose that there are a pool of r number of RBs (indexed by $k = 1, ..., r$) shared by the n O-DUs. Let U denote the set of all UEs in the network. Based on their connectivity type, U is partitioned into two disjoint subsets: U_1 (indexed by $i_1 = 1, .., m_1$) represents m_1 UEs with single connectivity, and U_2 (indexed by $i_2 = 1, .., m_2$) represents m_2 UEs with dual connectivity, where $m = m_1 + m_2$. In our proposed system, there are one or more link-beams is governed by a O-RU. Suppose K_i

represents the set of RBs assigned to UE i, and L_i and D_i represents set of link-beams and O-DUs connected to an UE i.

Suppose that, Nj is represented as total number of link-beams in ORU_k. Each particular link-beam is represented as $l_{a,j}$ where value of a ranges from 1 to total number of link-beams for a particular O-DU j and j indicates that particular O-DU from which the link-beam a belongs to. Transmission power of O-DU j is divided into each link-beam such that, transmission power of each link-beam,

$$\hat{P}_{a,j} = \frac{P_j}{N_j} \tag{1}$$

In this proposed work, when an UE is in active state, it is connected to one (for single connectivity UE) or two (for dual connectivity UE) resource block(s), link-beam(s) and O-DU(s) for a particular time slot t. Hence, to indicate whether the link-beam $l_{a,j}$ in O-DU j is assigned to UE i, a variable is used that is named by the UE access indicator $u^i_{a,j}(t)$.
Here, $u^i_{a,j}(t) = 1$ indicates that, UE i is assigned to link-beam $l_{a,j}$ and otherwise $u^i_{a,j}(t) = 0$. Furthermore, one more variable is taken which is named by RB allocation indicator or $f_{i,j,k,a}(t)$. If RB k is assigned to UE i through link-beam a and O-DU j at time slot t, $f_{i,j,k,a}(t) = 1$ and $f_{i,j,k,a}(t) = 0$, otherwise. Major notations used in this paper is summarized in table I.

A. Channel Gain Calculation:

The proposed work has used the variable $g^k_{i,l_{a,j}}(t)$ to indicate channel gain between UE i, link-beam $l_{a,j}$ at O-DU j and in RB k at time slot t. It is affected by two kinds of fading components: large-scale fading component ($q_{i,j}(t)$) and small-scale fading component ($h^k_{i,j}(t)$).

Hence, according to [22] the channel gain is, $g^k_{i,l_{a,j}}(t)$=$q_{i,j}(t)|h^k_{i,j}(t)|^2$

B. Down-link Transmissions:

1) Achievable Data Rate Calculation For Singly-Connected UEs: We know in ORAN architecture one resource block can be shared with multiple UEs. In this case, UE i_1 is connected to one gNB. Hence, the Signal-to-Interference-plus-Noise Ratio (SINR) at time t for UE i_1 in RB k and link-beam $l_{a,j}$ at O-DU j is given by,

$$\gamma^k_{i_1,a,j}(t) = \frac{\hat{P}_{a,j} g^k_{i_1,l_{a,j}}(t)}{\sum\limits_{n_{l_k} \in L_k \setminus l_{a,j}} P_{n_{l_k}} f_{i_1,k}(t) g^k_{i_1,n_{l_k}}(t) + \sigma^2} \tag{2}$$

Where L_k is set of link-beams allocating to resource block k to other UEs, $\hat{P}_{a,j}$ and $P_{n_{l_k}}$ are the transmission power levels of link-beam $l_{a,j}$ and other link-beams respectively and σ^2 denote noise power. The achievable data rate of UE i_1 is,

$$c^k_{i_1,l_{a,j}}(t) = \hat{B} log_2(1 + \gamma^k_{i_1,a,j}(t)) \tag{3}$$

where B is the bandwidth of each RB. In the proposed framework, it is assumed that all resource blocks (RBs) have an identical time duration. Consequently, the receiving rate of UE i is defined as the sum of the achievable rates for all RBs assigned to it.

2) Achievable Data Rate Calculation For Dual-Connected UEs: In the case of dual connectivity, each User Equipment (UE) is connected to two (O-DUs) when in an active state. Suppose, UE i_2 is connected to BS j_1 and j_2 via link-beams l_{a,j_1} and l_{a,j_2} respectively. Although each Resource Block (RB) can be shared among all gNBs, in the dual connectivity scenario, the same RB must not be simultaneously used by both gNB j_1 and j_2 for the same UE. Hence, assume that RB k_1 is accessed via link-beam l_{a,j_1} and RB k_2 is accessed via l_{a,j_2}. The SINRs associated with the links between UE i_2 and BSs j_1 and j_2 respectively , during time slot t.
Now,

$$\gamma^{k_1}_{i_2,a_1,j_1}(t) = \frac{P_{a_1,j_1} g^{k_1}_{i_2,l_{a_1,j_1}}(t)}{\sum\limits_{n_{l_k} \in L_{k_1} \setminus l_{a_1,j_1}} P_{n_{l_k}} f_{i_2,k_1}(t) g^{k_1}_{i_2,n_{l_k}}(t) + \sigma_1^2} \tag{4}$$

$$\gamma^{k_2}_{i_2,a_2,j_2}(t) = \frac{P_{a_2,j_2} g^{k_2}_{i_2,l_{a_2,j_2}}(t)}{\sum\limits_{n_{l_k} \in L_{k_2} \setminus l_{a_2,j_2}} P_{n_{l_k}} f_{i_2,k_2}(t) g^{k_2}_{i_2,n_{l_k}}(t) + \sigma_2^2} \tag{5}$$

Here, σ_1 and σ_2 represent the noise power for the links between UE i_2 and gNBs j_1 and j_2, respectively. For the aforesaid links, during time slot t in RB k_1 and k_2. The achievable rates are calculated using the Shannon capacity formula [4]:

$$c^{k_1}_{i_2,l_{a_1,j_1}}(t) = \hat{B} log_2(1 + \gamma^{k_1}_{i_2,a_1,j_1}(t)) \tag{6}$$

$$c^{k_2}_{i_2,l_{a_2,j_2}}(t) = \hat{B} log_2(1 + \gamma^{k_2}_{i_2,a_2,j_2}(t)) \tag{7}$$

Here, B indicates bandwidth of each RB. Let K_1 and K_2 represent the sets of resource blocks (RBs) allocated to UE i_2 over the respective link-beams l_{a,j_1} and l_{a,j_2} corresponding to O-DUs j_1 and j_2. We assume that these sets are mutually exclusive, i.e., $K_1 \cap K_2 = \phi$.

3) Receiving Data Rate Calculation: Suppose K_i represents the set of resource blocks assigned to UE i, and L_i and D_i represents set of link-beams and O-DUs connected to an UE i. Hence, the receiving data rate of UE i in time slot t is

$$R_{i,j}(t) = \sum_{k \in K_i} \sum_{l \in L_i} \sum_{j \in D_i} f_{i,k}(t)\, c^k_{i,l_{a,j}}(t) \tag{8}$$

Here, cardinality of L_i and D_i is 1 for single-Connected UEs and 2 for Dual-Connected UEs.

IV. Problem formulation

In our proposed work, a constant downlink throughput rate T is considered for each UE to ensure high-speed internet access for each of the UEs in the network.

Definition 1: The comparison function $\mathcal{C}_i(t)$ is used to check

Serial Number	Notation	Definition
1	n,B	Number and Set of DUs
2	m,U	Number and Set of UEs
3	r,K	Number and Set of RBs
4	U_1 and U_2	Set of singly connected and dual connected UEs
5	m_1 and m_2	Number of singly connected and dual connected UEs
6	$g^k_{i,l_{a,j}}(t)$	The channel gain when the UE i is served by link-beam $l_{a,j}$ of gNB j on RB k during time frame t
7	L	Set of link-beams
8	P_j	Transmission power of gNB j
9	$u_{i,l_{a,j}}(t)$	UE access indicator
10	$f_{i,k}(t)$	RB allocation indicator for UE i in RB k at time slot t
11	$q_{i,j}$	large scale fading component
12	$h^k_{i,j}(t)$	small scale fading component
13	$P_{l_{a,j}}$	Transmission power of link-beam $l_{a,j}$
14	$\gamma^k_{i_1,l_{a,j}}(t)$	Signal-to-Interference-plus- Noise Ratio (SINR) at time t for a single-connected UE i_1 in RB k and link-beam $l_{a,j}$ at BS j
15	$R_{i_2}(t)$	receiving rate of UE i_2 during time slot t
16	$\Lambda_j(t)$	Load Balancing factor for gNB j at time slot t

TABLE I
LIST OF SYMBOLS.

whether a particular UE i receives data at the speed of T in the time slot t or not.

$$\mathcal{C}_i(t) = \begin{cases} 1 & if \quad R_i(t) \geq T \\ 0 & otherwise \end{cases} \tag{9}$$

Where, $R_i(t)$ is the receiving rate of UE i at the time slot t.
Optimization 1: The optimization of all the UEs is:

$$\begin{aligned} \max_{U} \quad & \sum_{i\in\mathcal{U}}\sum_{l\in L} u_{i,l}(t)\mathcal{C}_i(t) \\ \text{s.t.} \quad & \sum_{l\in\mathcal{L}} u_{i,l}(t) = 1, \quad \forall i \in \mathcal{U}, \\ & u_{i,l}(t) \in \{0,1\}, \quad \forall (i,l) \in \mathcal{U}\times\mathcal{L}, \end{aligned} \tag{10}$$

where, U=$[u_{i,l_{a,j}}(t)\ \forall i, l_{a,j}, t]$ are vectors of $u_{i,l_{a,j}}(t)$
Definition 2: In this work we have considered $\Lambda_j(t)$ as Load Balancing factor for BS j at time slot t. This function is defined as:

$$\Lambda_j(t) = \begin{cases} 1 & if \quad T = \frac{\Theta_j(t)}{U^m_j(t)} \\ 0 & otherwise \end{cases} \tag{11}$$

Where, $\Theta_j(t)$is the downlink capacity of the gNB j and $U^m_j(t)$ is the total number of UEs connected to the gNB j and t is the time duration.
Optimization 2: Optimization of all the gNBs is:

$$\max_{\mathcal{B}} \quad \sum_{j\in\mathcal{B}} \Lambda_j(t) \tag{12}$$

Where, B is set of gNBs indexed by $j = 1$ to n.

V. Proposed RL based Traffic steering mechanism

In this work, we address the critical challenge of dynamic traffic steering and resource allocation within the Open Radio Access Network (O-RAN) architecture through the Reinforcement Learning (RL) application. Our proposed mechanism aims to optimize network performance by allowing User Equipments (UEs) to intelligently select serving Base Stations (BSs) and acquire appropriate Resource Blocks (RBs) for enhanced throughput and balanced network load. The core idea is to train an autonomous agent that learns optimal steering policies through interaction with the simulated O-RAN environment.

A. Agent Design

The proposed system utilises a single centralised RL agent that orchestrates all UEs' traffic steering and resource allocation decisions. This agent interacts with the O-RAN environment, including a dynamic layout of gNBs (BSs in the simulation for simplicity) and UEs, shared Resource Block (RB) pools, and a realistic channel model. The agent's decision-making process is framed as a Markov Decision Process (MDP), characterised by states, actions, and rewards.

B. State Space

The state presented to the RL agent for each UE at a given time step is a numerical representation of the relevant network conditions affecting that UE. For tabular Q-learning variants (Tabular Q-Learning, SARSA, Expected SARSA, N-Step SARSA), a coarse, categorised state is used to manage the discrete state space, consisting of:

- **UE Throughput Satisfaction Category:** Indicates whether the UE's current data rate meets or exceeds a predefined target throughput. This is typically a binary categorisation (satisfied/not satisfied).
- **Primary Serving BS Load Category:** Reflects the UE's primary serving BS load levels (e.g., low,medium,high), indicating potential congestion or under-utilisation.
- **Best Neighbour RSRP Difference Category:** Compares the Reference Signal Received Power (RSRP) of the best neighboring BS to that of the current serving BS. This relative measure helps identify potential handover opportunities based on signal strength.

For the Deep Q-Learning (DQL) agent, a more granular, continuous state representation is employed, providing richer input to the neural network:

- **Top-K RSRP Values:** The RSRP values from the top-K strongest neighbouring BSs (including the serving BS) are normalised and provided. This offers a detailed view of the radio environment.
- **Current Total UE Throughput:** The UE's aggregate data rate (in Mbps), normalized, to assess its current performance.
- **Allocated RBs from Primary BS:** The number of RBs currently allocated to the UE from its primary serving BS is normalised by the maximum allowed.
- **Allocated RBs from Secondary BS:** The number of RBs currently allocated to the UE from its secondary serving BS (if dual-connected), normalised.
- **Primary Serving BS Load Factor:** The continuous load factor of the primary serving BS.
- **Secondary Serving BS Load Factor:** The continuous load factor of the secondary serving BS (if applicable).

C. Action Space

The actions available to the RL agent for each UE govern its connection configuration and resource block allocation. For tabular methods, the action space is discrete and represents high-level steering decisions:

- **Stay:** The UE maintains its current primary and secondary BS connections and attempts to keep the existing RB allocations.
- **Switch Single Connectivity:** The UE attempts to perform a handover to the currently best-measured neighbouring BS, aiming for single connectivity. Existing secondary connections are dropped.
- **Try Dual Connectivity:** The UE attempts to establish dual connectivity, potentially utilising its current primary BS and the best available neighbour (or second-best if the primary is already the best neighbour). Resource allocation is then split.

For the DQL agent, the action space is more complex, allowing the agent to select a combination of primary BS, secondary BS, and the number of RBs requested from each. This action is linearised into a single integer index. The agent maps this index back to:

- **Target Primary BS:** The intended primary serving BS (can be 'None' for no primary connection).
- **Target Secondary BS:** The intended secondary serving BS (can be 'None' for no secondary connection and must be different from the primary).
- **Number of RBs for Primary BS:** The desired number of RBs to request from the primary BS (categorised: 0, half of max, or max).
- **Number of RBs for Secondary BS:** The desired number of RBs to request from the secondary BS (categorised: 0, half of max, or max).

The simulation logic then translates these target configurations into actual RB allocations, respecting the total available RBs and the maximum RBs per UE/BS.

D. Reward Function

A critical aspect of RL is the reward function, which guides the agent's learning towards desired network behaviours. Our reward function is designed to be dynamic and comprehensive, encouraging:

- **High UE Throughput Satisfaction:** A positive, continuously scaled reward component is provided based on the proportion of UEs that achieve their target throughput. Partial satisfaction is also rewarded.
- **Balanced BS Load:** A reward component that penalises the deviation of BS load factors from an ideal balanced state (e.g., 1.0). This uses a squared error term, where a lower deviation results in a higher reward, encouraging efficient resource utilisation across the network.
- **Minimised Handovers:** A small negative penalty is applied for each handover initiated in a time step. This discourages unnecessary network reconfigurations and promotes connection stability, which is crucial for maintaining the quality of service.

The total reward at each step is a weighted sum of these components:

$$R_t = w_{ue} \cdot R_{ue\_satisfaction} + w_{bs} \cdot R_{bs\_load} + w_{ho} \cdot R_{ho\_penalty}$$

Where $R_{ue\_satisfaction}$ ranges from 0 to 1, R_{bs_load} ranges from 0 to 1 (1 is ideal), and $R_{ho\_penalty}$ is negative for handovers. The weights w_{ue}, w_{bs}, w_{ho} are tunable parameters, emphasising different aspects of network performance (e.g., higher weight on throughput, balanced load, or handover stability).

E. Learning Algorithms

Our framework supports multiple RL algorithms, allowing a comparative analysis of their effectiveness in traffic steering.

- **Baseline Agent:** This acts as a non-RL benchmark. It implements a classical handover mechanism based on RSRP hysteresis [24] and Time-to-Trigger (TTT), combined with a greedy resource allocation strategy for single connectivity.
- **Tabular Q-Learning:** A fundamental value-based RL algorithm where a Q-table explicitly stores the learned Q-values for each (state, action) pair. The agent updates these values using the Bellman equation to converge to an optimal policy [25, Section 6.5]. It directly approximates the optimal action-value function, $Q^*(s, a)$, using the update rule:

  $$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

  This update is performed off-policy, meaning it learns about the optimal policy while following a different, more exploratory policy.
- **SARSA Agent:** Similar to Q-learning, it is an on-policy control algorithm [25, Section 6.4]. It learns the Q-value

of the state-action pair (S_t, A_t) using the Q-value of the next state-action pair (S_{t+1}, A_{t+1}) where A_{t+1} is chosen by the *current* policy. The update rule is:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

- **Expected SARSA Agent:** An off-policy variation of SARSA that uses the *expected* Q-value of the next state S_{t+1}, averaged over all possible actions in S_{t+1} according to the current policy, rather than the Q-value of a single chosen action A_{t+1} [26]. The update rule is given by:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

- **N-Step SARSA Agent:** An extension considering a return from multiple future steps (N-steps) rather than just one. This provides a more comprehensive lookahead for updates, potentially accelerating learning by bridging the gap between Monte Carlo and one-step TD learning [25, Section 7.2]. The update is based on the n-step return, $G_{t:t+n}$, defined as:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$$

  The action-value function is then updated towards this target [25, Chapter 7]:

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha\Big[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)\Big]$$

- **Deep Q-Network (DQN) Agent:** For larger state and action spaces, a Deep Neural Network (DNN) approximates the Q-function [27]. The DQN implementation includes sophisticated features such as residual connections in the network architecture, double Q-learning for more stable training, and prioritized experience replay for better sample efficiency. It uses an experience replay buffer to store past transitions $(s, a, r, s', \text{done})$ and samples minibatches from it for training, breaking correlations. A separate target network stabilises training by providing stable targets for Q-value updates. The implementation also includes learning rate scheduling with exponential decay, proper gradient clipping, and L2 regularization for improved training stability.

F. Training Process

The RL agents are trained iteratively over multiple simulation time steps. At each step:

1) All UEs move according to their mobility model and perform RSRP measurements from all available BSs.
2) Each UE's current state is determined based on the observed network conditions.
3) The RL agent (or baseline logic) chooses actions for each UE based on its current policy (or rules). For RL agents, an ϵ-greedy strategy is used, balancing exploration (random actions) and exploitation (choosing the action with the highest Q-value). The epsilon parameter decays over time to favour exploitation as training progresses.
4) The chosen actions (connection changes and RB allocations) are applied to the UEs.
5) UE rates and BS load factors are recalculated based on the new configurations.
6) A global reward is computed using the dynamic reward function, reflecting the overall network performance.
7) For RL agents, the experience (current state, action, reward, next state, and a 'done' flag) is used to update the agent's knowledge (Q-table for tabular, neural network weights for DQL). DQL agents perform batch training from their replay buffer and periodically update their target network to stabilise learning.

This iterative process allows the RL agent to learn which actions lead to higher cumulative rewards, ultimately optimising the traffic steering and resource allocation policies for a dynamic O-RAN environment.

The designed simulation software [23] provides the implementation details for the simulation environment used in this work.

VI. Results and Discussion

This section comprehensively evaluates the proposed Reinforcement Learning (RL) based traffic steering and resource allocation mechanisms within the O-RAN simulation environment. We compare the performance of various RL agents, including Tabular Q-Learning, SARSA, Expected SARSA, N-Step SARSA, Deep Q-Network (DQN), and a modified DQN with constant exploration (NO_DECAY_DQN), against a traditional Baseline agent. The analysis focuses on key performance indicators (KPIs) such as UE throughput, UE satisfaction, BS load distribution, and handover frequency.

A. Experimental Setup

The simulations were conducted using a custom-built Python-based O-RAN simulator developed to model a dynamic cellular network environment accurately. The core components of the simulator include a realistic channel model (incorporating path loss, log-normal shadowing, and small-scale fading), UE mobility (random walk model), BS placement (Uniform Random or Poisson Point Process), Resource Block (RB) management, and support for UE dual connectivity.

1) Simulation Parameters: The network configurations and RL-specific parameters used for the experiments are summarised in Table II. These parameters were carefully selected to represent a plausible cellular deployment scenario while ensuring the simulation is computationally tractable for iterative RL training.

B. Performance Evaluation and Discussion

The performance of the various RL agents was evaluated based on their ability to optimize UE throughput, balance BS loads, and manage handover frequency, as reflected by the defined reward function. The analysis below references the results from the FINAL experiment set, summarised in Table III, comparing all agents including the NO_DECAY_DQN variant.

TABLE II
SIMULATION AND REINFORCEMENT LEARNING PARAMETERS FOR DEFAULT CONFIGURATION.

S.N.	Category	Parameter	Value
1	General (*8)	Placement Method	Uniform/PPP
		Number of UEs (Uniform)	10 UEs
		Number of BSs (Uniform)	3 BSs
		Lambda BS (PPP)	$0.5\,\mathrm{BSs/km^2}$
		Lambda UE (PPP)	$2.0\,\mathrm{UEs/km^2}$
		Simulation Area (X, Y)	$1\,\mathrm{km}$ x $1\,\mathrm{km}$
		Time Step Duration	$0.2\,\mathrm{s}$
8		Total Simulation Steps	200 Steps
9	O-DU Parameters (*3)	O-DU Transmit Power	$38.0\,\mathrm{dB}$
		BS Link Beam Gain	$10.0\,\mathrm{dB}$
11		BS Access Beam Gain	$3.0\,\mathrm{dB}$
12	UE Parameters (*5)	UE Speed	$5.0\,\mathrm{m\,s^{-1}}$
		UE Noise Figure	$7.0\,\mathrm{dB}$
		Target UE Throughput	$1.0\,\mathrm{Mbit\,s^{-1}}$
		Max RBs per UE (Total)	4 RBs
16		Max RBs per UE per BS (Dual)	2 RBs
17	RB Parameters (*2)	Total Resource Blocks	20 RBs
18		RB Bandwidth	$0.5\,\mathrm{MHz}$
19	Channel Model (*4)	Path Loss Exponent	3.7
		Reference Distance	$1.0\,\mathrm{m}$
		Reference Loss	$32.0\,\mathrm{dB}$
22		Shadowing Std. Dev.	$4.0\,\mathrm{dB}$
23	HO (Baseline) (*3)	HO Hysteresis	$3.0\,\mathrm{dB}$
		HO Time to Trigger	$0.4\,\mathrm{s}$
25		Min Acquisition RSRP	$-115.0\,\mathrm{dB}$
26	RL Parameters (*17)	Learning Algorithms (see Sec. V-E)	Baseline/TabularQL/SARSA/ ExpectedSARSA/NStepSARSA/DQN/ NO_DECAY_DQN
		RL Gamma (γ)	0.95
		RL Learning Rate (α)	0.001
		RL Epsilon Start (ϵ_{start})	0.1
		RL Epsilon End (ϵ_{end})	0.1
		RL Epsilon Decay Steps	160 Steps
		RL Batch Size (DQL)	32 Samples
		RL Target Update Freq (DQL)	5 Steps
		RL Replay Buffer Size (DQL)	2000 Samples
		RL N-step (NStepSARSA)	3 Steps
		DQL Learning Rate	0.001
		DQL Gradient Clip Norm	1.0
		DQL L2 Regularization	0.0001
		DQL Hidden Layers	2
		DQL Hidden Units	64
		DQL Dropout Rate	0.1
42		DQL Soft Updates (τ)	0.01
43	No-Decay DQN (*4)	Epsilon Value	0.1
		Target Update Frequency	5 Steps
		Learning Rate	0.001
46		Soft Update Tau	0.01

*The notation '(*X)' in the Category column indicates that the category spans X rows (parameters).*

TABLE III
SUMMARY OF KEY METRICS FOR ALL AGENTS FROM THE FINAL EXPERIMENT SET

Agent	Throughput (Mbps)	Satisfied UEs (%)	BS Load Factor	Handovers (/Step)	SINR (dB)	RBs/UE	Reward
Baseline	4.38	99.15	0.74	1.68	-83.82	1.00	N/A
TabularQLearning	16.37	49.67	0.74	0.41	-85.09	4.00	0.88–0.94
SARSA	18.07	50.00	0.69	0.57	-86.19	4.00	0.88–0.94
ExpectedSARSA	16.19	49.95	0.69	0.51	-87.10	4.00	0.88–0.94
NStepSARSA	17.88	49.89	0.74	0.43	-81.76	4.00	0.88–0.94
DQN	6.11	87.47	0.65	2.77	-86.87	1.51	0.70
NO_DECAY_DQN	4.44	99.35	0.75	1.57	-79.62	1.00	0.78

1) Throughput and User Satisfaction: Figure 1 shows that the tabular RL agents (SARSA, NStepSARSA, TabularQLearning, ExpectedSARSA) achieve the highest average UE throughput (16.19–18.07 Mbps), with SARSA being the

best. DQN and NO_DECAY_DQN perform much worse (6.11 and 4.44 Mbps, respectively), only slightly better than the Baseline (4.38 Mbps). Interestingly, the percentage of satisfied UEs (Fig. 2) is highest for Baseline and NO_DECAY_DQN (99.15% and 99.35%), while DQN achieves 87.47%. All tabular agents are around 50%, indicating that their strategy of aggressively maximizing throughput does not necessarily translate to a higher proportion of satisfied UEs under the current satisfaction definition.

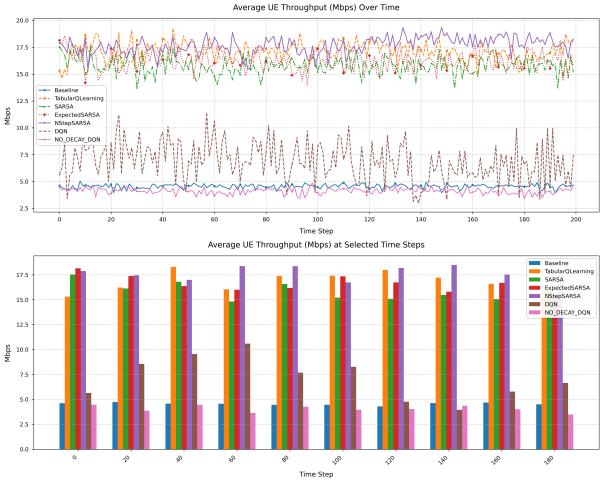

Fig. 1. Average UE Throughput (Mbps) over time and as a bar graph for each agent.

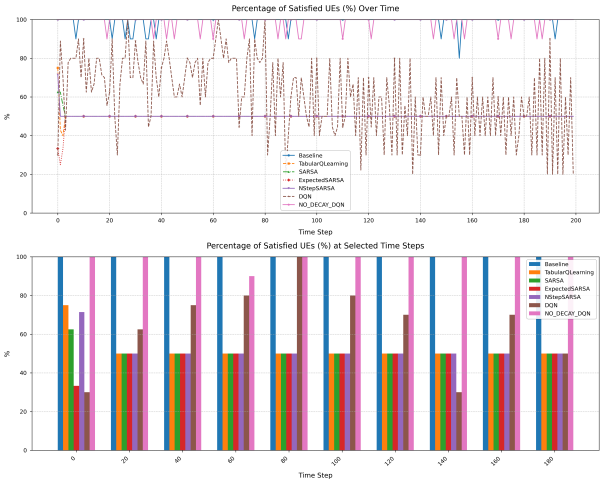

Fig. 2. Percentage of satisfied UEs for each agent.

2) Load Balancing and Resource Utilization: Figure 3 shows that all agents except DQN maintain similar average BS load factors (0.69–0.75), with NO_DECAY_DQN being the highest (0.75). DQN is lower (0.65), indicating less efficient network-wide utilization. This is correlated with resource allocation, shown in Fig. 4. All tabular agents allocate the maximum possible RBs per UE (4.00) to drive high throughput. In contrast, DQN and NO_DECAY_DQN allocate far fewer (1.51 and 1.00, respectively), similar to the Baseline agent.

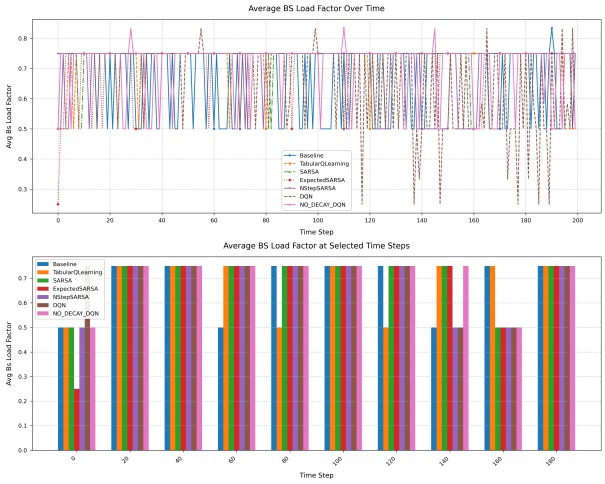

Fig. 3. Average BS Load Factor for each agent.

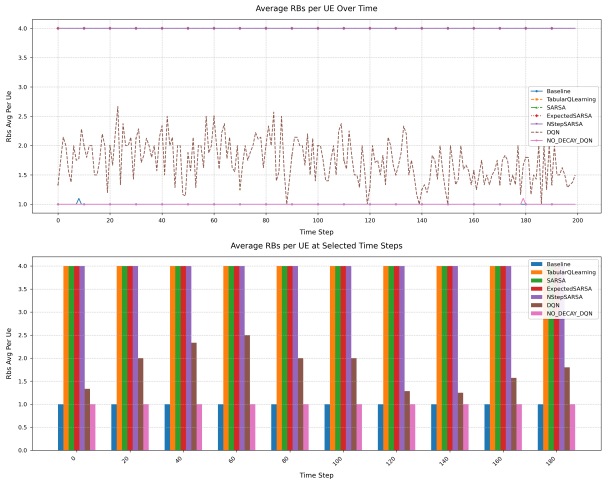

Fig. 4. Average RBs per UE for each agent.

3) Handover Management: Figures 5 and 6 show that DQN has the highest handover rate (2.77 per step), indicating significant instability. The Baseline and NO_DECAY_DQN agents have moderate handover rates (1.68 and 1.57, respectively). The tabular agents are by far the most stable, with much lower handover rates (0.41–0.57), suggesting they learn more robust UE-BS associations.

4) SINR and Signal Quality: Figure 7 shows that NO_DECAY_DQN achieves the best average SINR (-79.62 dB), followed by NStepSARSA (-81.76 dB) and Baseline (-83.82 dB). DQN and the other tabular agents have lower SINR values (-85 to -87 dB), suggesting that their strategies for maximizing throughput and load can come at the expense of signal quality.

5) Reward: Figure 8 demonstrates that the tabular agents achieve the highest average rewards (0.88–0.94). NO_DECAY_DQN achieves a higher reward (0.78) than the standard DQN (0.70). This confirms that while constant exploration helps NO_DECAY_DQN learn a better policy than

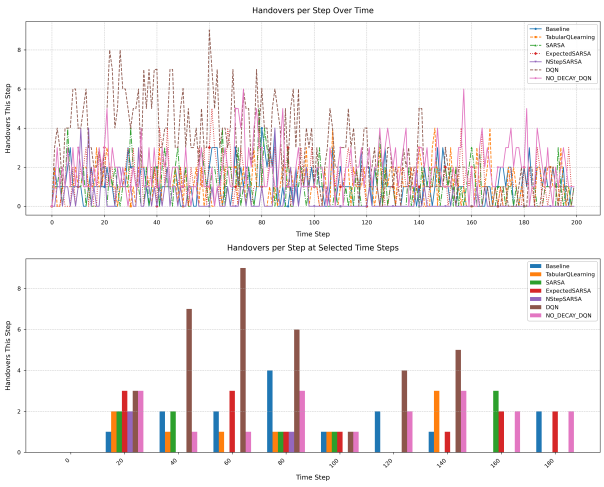

Fig. 5. Handovers per step for each agent.

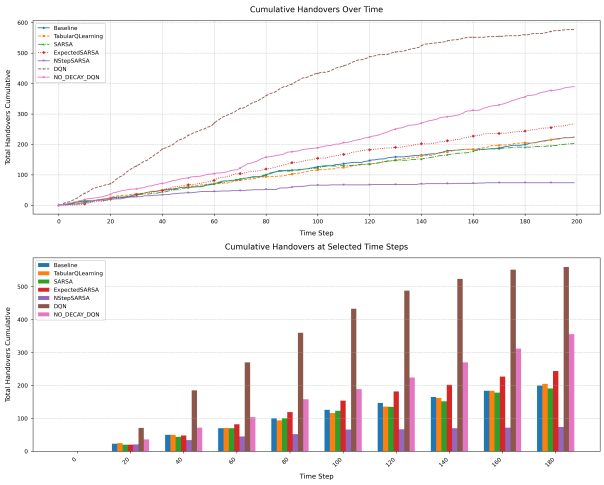

Fig. 6. Cumulative handovers for each agent.

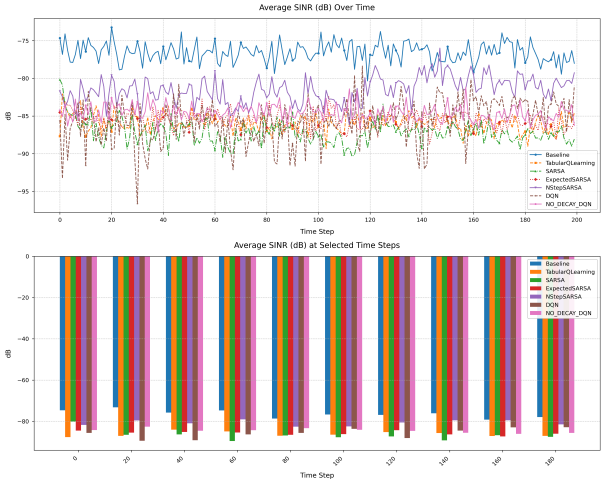

Fig. 7. Average SINR (dB) for each agent.

standard DQN, neither is as effective as the tabular methods in this environment according to the defined reward function.

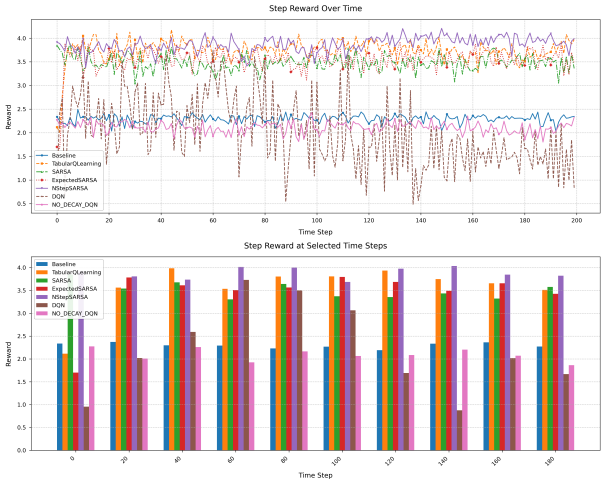

Fig. 8. Step reward for each agent.

6) Learning Dynamics: Figure 9 confirms that the exploration rate (epsilon) for the NO_DECAY_DQN agent remains constant at 0.1 throughout the simulation, while standard DQN and tabular agents decay to 0.43. Baseline does not use exploration (epsilon=1.0).

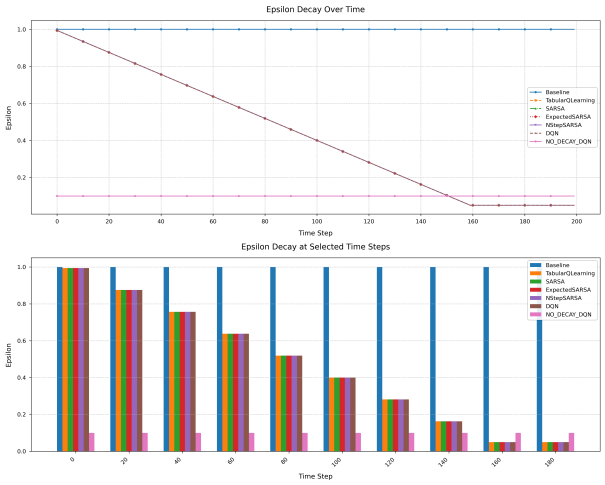

Fig. 9. Epsilon value over time for DQN agents.

7) Comparative Analysis: In summary, the results in Table III highlight critical trade-offs in RL-based RAN management. Tabular RL agents excel in maximizing throughput and resource allocation but, paradoxically, do not maximize the percentage of satisfied UEs. Conversely, NO_DECAY_DQN achieves the best SINR and highest user satisfaction but at the cost of throughput that is only marginally better than the Baseline. These findings suggest that the definition of ”satisfaction” and the reward function’s weights are crucial in determining the learned policy’s behavior and final performance characteristics.

REFERENCES

[1] Polese, M., Bonati, L., D'oro, S., Basagni, S., Melodia, T. (2023). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys Tutorials, 25(2), 1376-1411.

[2] Ntassah, R., Dell'area, G. M., Granelli, F. (2024). User Classification and Traffic Steering in O-RAN. IEEE Open Journal of the Communications Society.

[3] Kavehmadavani, F., Nguyen, V. D., Vu, T. X., Chatzinotas, S. (2023). Intelligent traffic steering in beyond 5G open RAN based on LSTM traffic prediction. IEEE Transactions on Wireless Communications, 22(11), 7727-7742.

[4] Shannon, C. E. (1948). A mathematical theory of communication. The Bell system technical journal, 27(3), 379-423.

[5] Erdol, H., Wang, X., Li, P., Thomas, J. D., Piechocki, R., Oikonomou, G., Kapoor, S. (2022, September). Federated meta-learning for traffic steering in o-ran. In 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall) (pp. 1-7). IEEE.

[6] Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B. S., Singh, T., Melodia, T. (2023). Programmable and customized intelligence for traffic steering in 5G networks using open RAN architectures. IEEE Transactions on Mobile Computing, 23(4), 2882-2897.

[7] Kavehmadavani, F., Nguyen, V. D., Vu, T. X., Chatzinotas, S. (2023, December). On Deep Reinforcement Learning for Traffic Steering Intelligent ORAN. In 2023 IEEE Globecom Workshops (GC Wkshps) (pp. 565-570). IEEE.

[8] Priscoli, F. D., Giuseppi, A., Liberati, F., Pietrabissa, A. (2020, May). Traffic steering and network selection in 5G networks based on reinforcement learning. In 2020 European control conference (ECC) (pp. 595-601). IEEE.

[9] Fotiadis, P., Polignano, M., Viering, I., Zanier, P. (2014, May). On the Potentials of Traffic Steering in HetNet Deployments with Carrier Aggregation. In 2014 IEEE 79th Vehicular Technology Conference (VTC Spring) (pp. 1-5). IEEE.

[10] Jorgensen, N. T. K., Laselva, D., Wigard, J. (2011, May). On the potentials of traffic steering techniques between HSDPA and LTE. In 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring) (pp. 1-5). IEEE.

[11] Nguyen, V. D., Vu, T. X., Nguyen, N. T., Nguyen, D. C., Juntti, M., Luong, N. C., ... Chatzinotas, S. (2023). Network-aided intelligent traffic steering in 6G O-RAN: A multi-layer optimization framework. IEEE Journal on Selected Areas in Communications, 42(2), 389-405.

[12] Gijón, C., Toril, M., Luna-Ramirez, S., Marí-Altozano, M. L. (2019). A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks. IEEE Transactions on Vehicular Technology, 68(10), 9414-9424.

[13] Munoz, P., Barco, R., Laselva, D., Mogensen, P. (2013). Mobility-based strategies for traffic steering in heterogeneous networks. IEEE Communications Magazine, 51(5), 54-62.

[14] Kavehmadavani, F., Nguyen, V. D., Vu, T. X., Chatzinotas, S. (2022, May). Traffic steering for eMBB and uRLLC coexistence in open radio access networks. In 2022 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 242-247). IEEE.

[15] Fotiadis, P., Polignano, M., Chavarria, L., Viering, I., Sartori, C., Lobinger, A., Pedersen, K. (2013, June). Multi-Layer Traffic Steering: RRC Idle Absolute Priorities Potential Enhancements. In 2013 IEEE 77th Vehicular Technology Conference (VTC Spring) (pp. 1-5). IEEE.

[16] Burgueño, J., de-la-Bandera, I., Palacios, D., Barco, R. (2020). Traffic steering for eMBB in multi-connectivity scenarios. Electronics, 9(12), 2063.

[17] Zhang, N., Zhang, S., Wu, S., Ren, J., Mark, J. W., Shen, X. (2016). Beyond coexistence: Traffic steering in LTE networks with unlicensed bands. IEEE wireless communications, 23(6), 40-46.

[18] Akman, A., Oliver, P., Jones, M., Tehrani, P., Hoffmann, M., Li, J. (2024, October). Energy Saving and Traffic Steering Use Case and Testing by O-RAN RIC xApp/rApp Multi-vendor Interoperability. In 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) (pp. 1-6). IEEE.

[19] Khandaker, K., Trossen, D., Yang, J., Despotovic, Z., Carle, G. (2022, August). On-path vs off-path traffic steering, that is the question. In Proceedings of the ACM SIGCOMM Workshop on Future of Internet Routing Addressing (pp. 37-42).

[20] O-RAN Alliance. (2024). O-RAN.WG1.TR.Use-Cases-Analysis-Report-R004-v15.00: Use Cases Analysis Report. O-RAN Alliance.

[21] O-RAN Alliance. (2024). O-RAN.WG1.TS.Use-Cases-Detailed-Specification-R004-v15.00: Use Cases Detailed Specification. O-RAN Alliance.

[22] Cao, Y., Lien, S. Y., Liang, Y. C., Chen, K. C., Shen, X. (2021). User access control in open radio access networks: A federated deep reinforcement learning approach. IEEE Transactions on Wireless Communications, 21(6), 3721-3736.

[23] Sharma, Aaradhy (2025). Reinforcement Learning based traffic steering inOpen Radio Access Network (ORAN)- oran-ts GitHub Repository. figshare. Software. https://doi.org/10.6084/m9.figshare.29262791.

[24] B. Herman, D. Petrov, J. Puttonen, and J. Kurjenniemi, 'A3-Based Measurements and Handover Model for NS-3 LTE', Nov. 2013.

[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second edition, The MIT Press, Cambridge, MA, 2018.

[26] H. van Seijen, H. van Hasselt, S. Whiteson, and M. Wiering, "A theoretical and empirical analysis of Expected Sarsa," in *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Nashville, TN, USA, 2009, pp. 177-184, doi: 10.1109/ADPRL.2009.4927542.

[27] M. Sewak, "Deep Q Network (DQN), Double DQN, and Dueling DQN: A Step Towards General Artificial Intelligence," $\in$ *Foundations of Reinforcement Learning with Applications in Finance*, $Singapore$: $SpringerSingapore, 2019, ch.\ 8, pp.\ 111-127$.