On-Net: On-Policy Multi-Agent Reinforcement Learning for Adaptive Access Network Selection

On-Net: On-Policy Multi-Agent Reinforcement Learning for Adaptive Access Network Selection

Álvaro Solana *, Doriana Monaco †, Alessio Sacco †, Guido Marchetto †
* *ETSIT*, Universidad Politécnica de Madrid, Spain
† *Department of Control and Computer Engineering*, Politecnico di Torino, Italy

***Abstract*—The coexistence of multiple Radio Access Technologies (RATs) and wired access networks (e.g., Ethernet and fiber-optic broadband) provides flexible connectivity but also makes access network selection increasingly complex. Traditional optimization, game-theoretic, and heuristic methods require global information or incur high signaling overhead, limiting scalability in dynamic environments. While Reinforcement Learning (RL) offers adaptability, existing RL and Deep RL (DRL) approaches often rely on centralized coordination or network-assisted data, reducing their effectiveness in large-scale, mobile, and heterogeneous scenarios. We propose ON-policy multi-agent PPO for adaptive access NETwork selection (*On-Net*): a decentralized, user-centric DRL framework based on centralized training with decentralized execution (CTDE). In this design, users independently make network access selection decisions using only local observations, while benefiting from collaboration during training. We validate our approach in both simulation and a real-world testbed, demonstrating its effectiveness in selecting between various access technologies, whether wireless or wired. On-Net balances high throughput, fairness among users, and reduced handovers with controlled decision times; notably, real-world experiments demonstrated a 90% reduction in handovers.**

***Index Terms*—Access network selection, Reinforcement Learning, MAPPO**

I. Introduction

The proliferation of heterogeneous wireless networks has significantly reshaped modern communication systems, enabling improved Quality of Service (QoS) while simultaneously introducing new challenges in network management and resource optimization [1]. As communication systems evolve toward sixth-generation (6G) networks, the coexistence of multiple radio access technologies (RATs)–such as 5G New Radio (5G NR), Long Term Evolution (LTE), WiFi, and emerging platforms including satellite communications–has become a defining feature of next-generation network architectures [2], [3].

This growing heterogeneity is further compounded by the diversity of applications and devices that rely on seamless connectivity. In environments such as smart homes, Internet of Things (IoT) devices exhibit widely varying communication requirements, ranging from low-rate sensing traffic to latency-sensitive control and multimedia applications [4], [5]. Consequently, these systems demand adaptive access network selection mechanisms capable of responding to rapidly changing channel conditions, fluctuating traffic loads, and the dynamic availability of both wired and wireless access technologies.

To address these challenges, various approaches have been proposed [6], [7]. Early work relied on centralized optimization frameworks aimed at maximizing system utility under resource constraints [8], [9]. Although these methods can provide theoretically optimal decisions, they require perfect channel state information and substantial computational resources, which limits their practicality in large-scale and rapidly changing deployments. To alleviate these constraints, several heuristic strategies have been proposed [10], [11], including congestion-game formulations [12] and load-aware schemes [13]. While more scalable, these approaches generally assume stationary channels and offer limited adaptability to mobility or highly dynamic operating conditions. More recently, Reinforcement Learning (RL) has emerged as a promising paradigm for access network selection by modeling the problem as a sequential decision process [14]–[19]. Extensions based on Deep RL (DRL) further improve scalability by approximating value functions with neural networks, enabling efficient operation in high-dimensional and continuous state spaces. However, existing RL and DRL techniques often rely on shared global rewards or continuous information exchange among agents to ensure stable behavior in multi-user settings. In addition, many solutions are tailored to specific application domains or focus exclusively on wireless access technologies, limiting their applicability in heterogeneous environments that combine both wired and wireless access options [20]–[22].

In this work, we address this gap by proposing *On-Net*, an ON-policy multi-agent PPO framework for access NETwork selection in heterogeneous environments. On-Net introduces a fully decentralized, user-centric approach based on centralized training and decentralized execution (CTDE), enabling each device to act autonomously at runtime using only local observations, without requiring real-time coordination. Unlike prior solutions constrained to wireless RATs, On-Net natively supports both wired and wireless technologies. Through simulations and real-world testbed experiments, we demonstrate that On-Net maintains a robust balance between throughput, fairness, and handover minimization under mobility, with empirical results showing a 90% handover reduction, all within controlled decision times.

The remainder of this paper is organized as follows. We review existing approaches to the problem of access technology selection in Section II. Section III discusses in detail our proposed solution and the RL formulation, which is evaluated in Section IV. Finally, we draw our conclusion in Section V.

II. Related work

This section reviews the state-of-the-art approaches for access network selection in heterogeneous networks, broadly categorized into optimization-based, heuristic, and learning-driven methods.

A. Non-learning approaches

Optimization Approaches. Early research on RAT and network selection primarily relied on centralized optimization frameworks aimed at maximizing system utility under limited network resources. For instance, Wu et al. [8] propose a joint transmitter–receiver energy-efficient scheduling scheme for multi-radio networks using a divide-and-conquer algorithm. While effective under ideal conditions, their method assumes perfect and global channel state information, an assumption rarely met in practical deployments. Similarly, Nash-bargaining–based formulations such as [9] solve the user association problem as a global utility maximization task. Although these approaches offer theoretically optimal solutions, they depend on full network knowledge, centralized coordination, and computationally intensive procedures, which limit their scalability in large, dynamic networks.

Heuristic Approaches. To overcome these limitations, several heuristics have been explored. In [12], the network selection problem is formulated as a non-cooperative congestion game in which users independently select the RAT (e.g., Wi-Fi, LTE) that maximizes their estimated throughput using only local congestion observations. Through a best-response rule with hysteresis, users iteratively adjust their decisions until a stable equilibrium is reached. Likewise, the load-aware association mechanism of [13] employs the Range expansion (RE) heuristic, which applies per-tier SINR or rate biases to encourage users to associate with small cells even when their raw received power is lower. Both methods assume stationary channel conditions and do not account for user mobility.

B. Learning-based approaches

More recently, reinforcement learning (RL) has gained traction as a tool for network selection, enabling agents to learn effective policies from interaction data rather than solving static optimization problems. Early RL approaches favor lightweight, non-neural methods to avoid computational overhead [23]. Regret-based frameworks [18], [24] allow users to update action probabilities based on performance feedback, providing decentralized decision-making without requiring explicit modeling of network dynamics. Model-driven methods have also integrated tabular Q-learning with feature-engineered estimators to improve link quality predictions [17]. While these methods converge faster than naive Q-learning, they remain constrained by the "curse of dimensionality": maintaining explicit Q-tables becomes impractical as the state space grows.

To address these scaling issues, modern research increasingly adopts Deep Reinforcement Learning (DRL), which replaces discrete Q-tables with neural network function approximators that can handle continuous, high-dimensional state spaces. For instance, genetic algorithm–assisted RL has been used to evolve flexible RAT selection policies, allowing a single trained model to generalize across many agents during deployment [22]. Recent works in healthcare scenarios modeled this problem as a multi-agent setting. The TB-MADDPG framework [21] trains two teams of agents–patient's devices and RATs–through Multi-Agent Deep Deterministic Policy Gradient (MADDPG), while other approaches deploy Decentralised Partially Observable Markov Decision Process (DPOMDP) to satisfy Quality Of Experience (QoE) metrics [20].

Despite their progress, existing DRL solutions are mainly applied to domain-specific architectures, such as healthcare, and focus exclusively on wireless technologies, often assuming static environments. On the other hand, On-Net provides a decentralized Multi-agent PPO (MAPPO) framework for access network selection in heterogeneous environments that include both wireless and wired access options. Agents collaborate during training and use only local observations during deployment. Our approach also accounts for user mobility, offering a general-purpose solution that extends beyond the limited scenarios addressed in prior studies.

III. Proposed Solution

This paper presents a decentralized, user-centric strategy in which each user independently makes decisions based solely on locally observable information. To address this problem, we adopt Reinforcement Learning (RL) — specifically MAPPO — due to its ability to learn effective behaviors without requiring prior domain knowledge, relying instead on direct interaction with the environment. MAPPO [25] is an on-policy deep reinforcement learning algorithm for cooperative multi-agent systems built within the CTDE paradigm. The environment is modeled as a Dec-POMDP with agents $i \in \{1, \ldots, N\}$, each receiving a local observation o_t^i and selecting an action a_t^i, for which they receive an individual reward r_t^i. In our setting, each agent maintains its own decentralized actor $\pi_{\theta_i}(a_t^i \mid o_t^i)$. A centralized critic $V_\phi(s_t)$, where s_t denotes the concatenated observations, provides advantage estimates for all agents during training only to perform cooperation.

For each agent i, the PPO-style clipped objective is maximized:

$$\mathcal{L}_i^{\text{CLIP}}(\theta_i) \;=\; \mathbb{E}_t\left[\min\left(r_t^i(\theta_i)\,\hat{A}_t^i,\; \text{clip}\big(k_t^i(\theta_i),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_t^i\right)\right], \tag{1}$$

where the probability ratio $k_t^i(\theta_i)$ measures how much the new policy deviates from the old one:

$$k_t^i(\theta_i) = \frac{\pi_{\theta_i}(a_t^i \mid o_t^i)}{\pi_{\theta_i^{\text{old}}}(a_t^i \mid o_t^i)}, \tag{2}$$

and $\hat{A}_t^i$ denotes the generalized advantage estimate computed using the centralized critic. The global training objective aggregates all agent policy losses together with a value-function loss and an entropy regularizer:

$$\mathcal{L}(\{\theta_i\}, \phi) = -\sum_{i=1}^{N} \mathcal{L}_i^{\text{CLIP}}(\theta_i) \;+\; c_v\,\mathcal{L}^{\text{VF}}(\phi) \;-\; c_e\,\mathcal{L}^{\text{ENT}}(\theta_i), \tag{3}$$

Following the standard configuration in [26], we set the scalar coefficients to $c_v = 1$ and $c_e = 0.01$. The former equilibrates the gradient contributions between the actor and critic components, while the latter preserves policy stochasticity to mitigate premature convergence toward sub-optimal local extrema.

We model the MDP in the following way:

State space. The state space is a vector that typically includes the available medium, the current selection, the channel metrics for each station (for wireless medium), and cable (for wired technologies). In general, it includes all available information that indicates the potential final throughput. Each state vector contains information specific to a single user and is constructed solely from that user's local observations.

Action space. The action selected by each user corresponds to the choice of a technology medium (i.e., which RAT station or wired cable to connect to). The neural network outputs a vector of action logits that parameterize a categorical policy over the available access choices; an action is then selected by sampling from this distribution (during training) or by taking the most probable action (during evaluation), thereby maximizing expected long-term return via policy gradient optimization.

Reward. We define the reward function based on the throughput achieved by the user, which is computed differently for each technology. The following throughput models have been used in similar works [12], [17], [18]. For LTE, a proportional-fair model is used, i.e., users experience a throughput proportional to their perceived rate:

$$T_j^i = \frac{R_j^i}{n_i} \tag{4}$$

where R_j^i is the estimated rate for user j on base station i, and n_i is the number of users connected to that station. The rate estimation procedure is explained in the next subsection.

For WiFi, a throughput-fair model is used, i.e., users connected to the same base station experience the same throughput:

$$T_j^i = \left(\sum_{j=1}^{n_i} \frac{1}{R_j^i} \right)^{-1} \tag{5}$$

For wired technology, we use the Mathis formula [27] to obtain an upper bound on the throughput of user i.

$$T^i < \frac{MSS}{RTT \times \sqrt{p}} \tag{6}$$

where the standard MSS value is 1460 Bytes, the packet loss is 0.00001 for the LAN environment, and the RTT is measured.

We compute the final reward by normalizing throughput values between 0 and 1. If the throughput is 0, meaning the user chose a station to which it could not connect, the reward is set to a negative value of -0.1. To discourage unnecessary handovers, we apply a penalty when the user switches to a new RAT and obtains less than a 10% improvement in throughput.

IV. Evaluation

In this section, we evaluate the performance of On-Net with particular focus on user throughput, fairness, and handover. We compare our approach against two benchmarks:

- Heuristic [12]: a RAT switch is triggered with a fixed probability only when the expected throughput improvement exceeds a threshold, and a hysteresis mechanism is applied to avoid oscillations: if the device returns to a previously used RAT class, it must surpass the previously achieved throughput in that class.
- RLNF [18]: a regret-based reinforcement learning method that incorporates network-assisted feedback from base stations—specifically the computed per-user throughput and current load. Unlike DRL methods, RLNF maintains a probability vector over RAT choices rather than using a neural network.
- DQN: a centralized policy that outputs all agents' actions. This approach would require a central entity that performs the computation and then forwards the corresponding action to each agent. We adopt the same RL formulation as On-Net to compare the two learning approaches.

A. Performance in a Simulated Environment

We first evaluate our approach in a simulated environment for RAT selection. This scenario simulates a (150×150) meter area populated with 20 users randomly distributed throughout the space. Additionally, 4 LTE base stations are randomly positioned at the corners, and 9 Wi-Fi Access Points (APs) are randomly placed. The two Radio Access Technologies use separate carrier frequencies; therefore, no interference occurs between the RATs [12]. Moreover, since the serving nodes of the same RAT are positioned apart, interference among them is also considered negligible [12].

Each agent selects the RAT and the specific station to connect to based on the model's input, i.e., the state. The state includes the normalized estimated available rate per station, which is set to -0.1 if the station is out of reach. Additionally, a one-hot encoded vector indicates the current station selected, and a binary vector represents the RAT type (0 for LTE, 1 for WiFi) of each station. We estimate the transmission rate in the following way:

$$\text{Rate} = \text{Bw} \cdot \eta \tag{7}$$

where Bw is the user available bandwidth and the spectral efficiency η is obtained based on the Modulation and Code Rate used. These parameters can be chosen based on the Signal-to-noise ratio (SNR) using the different thresholds for LTE and WiFi proposed in [28] and [29], respectively. The values used for the mapping are shown in the tables I and II.

To compute the SNR for a given user–station pair, we consider the transmit power, path loss, fading, and noise characteristics. Specifically, the SNR is computed following Eq. 8:

$$\text{SNR} = \frac{P_{\text{tx}} \cdot G_{\text{fading}}}{P_{\text{noise}} \cdot P_L} \tag{8}$$

TABLE I: SNR values used for LTE

SNR (dB)	Modulation	Code Rate	η (bps/Hz)
-9.478	QPSK	78/1024	0.1523
-6.658	QPSK	120/1024	0.2344
-4.098	QPSK	193/1024	0.3770
-1.798	QPSK	308/1024	0.6016
0.399	QPSK	449/1024	0.8770
2.424	QPSK	602/1024	1.1758
4.489	16-QAM	378/1024	1.4766
6.367	16-QAM	490/1024	1.9141
8.456	16-QAM	616/1024	2.4063
10.266	16-QAM	466/1024	2.7305
12.218	16-QAM	567/1024	3.3223
14.122	16-QAM	666/1024	3.9023
15.849	16-QAM	772/1024	4.5234
17.786	16-QAM	873/1024	5.1152
19.809	16-QAM	948/1024	5.5547

TABLE II: SNR values used for WiFi

SNR (dB)	Modulation	Code Rate	η (bps/Hz)
-3.83	BPSK	1/2	0.5
0.00	QPSK	1/2	1.0
2.62	QPSK	3/4	1.5
4.77	16-QAM	1/2	2.0
8.45	16-QAM	3/4	3.0
11.67	64-QAM	2/3	4.0
13.35	64-QAM	3/4	4.5
14.91	64-QAM	5/6	5.0
17.99	256-QAM	3/4	6.0
–	256-QAM	5/6	–

where P_{tx} is the transmit power, G_{fading} models the Rayleigh fading gain, P_L is the path loss calculated using the COST 231 Walfisch-Ikegami model, and P_{noise} is derived from the noise power spectral density and the bandwidth.

The path loss P_L (in dB) is modeled as:

$$P_L(d, f_c) = -35.4 + 20\log_{10}(d) + 20\log_{10}\left(\frac{f_c}{10^6}\right) \tag{9}$$

where d is the distance in meters between the user and the station, and f_c is the carrier frequency in Hz.

The noise power P_{noise} is calculated as:

$$P_{\text{noise}} = N_0 \cdot \text{Bw} \tag{10}$$

where N_0 is the noise power spectral density and Bw is the user available bandwidth in Hz. The values for the parameters described above are reported in Table III.

TABLE III: Parameters used for SNR computation

Parameter	Value
Carrier frequency (LTE)	2.6 GHz
Carrier frequency (WiFi)	5.8 GHz
Bw (LTE)	20 MHz
Bw (WiFi)	80 MHz
Transmit power (User)	1 W
Noise power spectral density N_0	-174 dBm/Hz

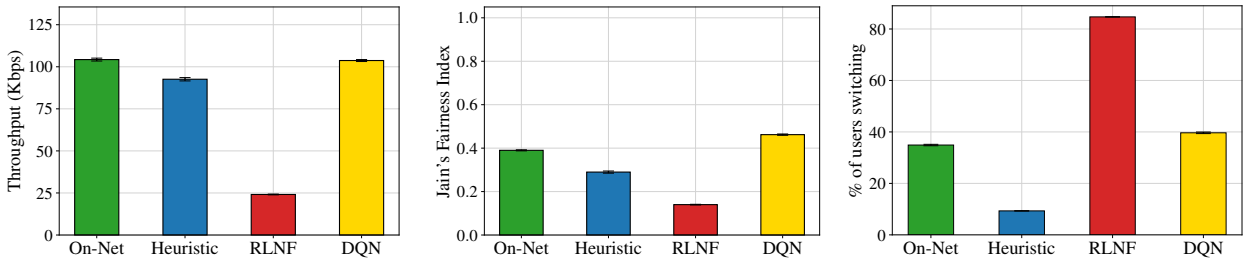
Each agent is equipped with its own policy network, implemented as a Multi-Layer Perceptron (MLP) with three hidden layers of 256 neurons each and ReLU activations. The critic is implemented as a 3-layer MLP with 256 hidden units and Tanh activations, producing a single global value estimate that is broadcast to all agents. Training follows the MAPPO framework using the Clipped PPO objective with Generalized Advantage Estimation (GAE) ($\gamma = 0.99, \lambda = 0.95$). We use Adam optimizer with a learning rate of $3.5e^{-5}$. Entropy regularization with a coefficient of 0.01 is used to sustain exploration during training. Training is performed over $10,000$ episodes, each lasting 10 steps. During an episode, each agent moves in the area following the Random Waypoint Mobility Model [30].

For evaluation, the trained model is tested in the same environment over $1,000$ iterations. To ensure statistical significance, results are aggregated over 30 runs with different random seeds, and 95% confidence intervals are reported. Fig. 1 summarizes the system's performance across throughput, fairness, and switching rate.

From Fig. 1c, we observe that the heuristic achieves the lowest RAT-switching rate, largely due to its built-in hysteresis mechanism, which suppresses frequent transitions unless a significant throughput gain is expected. While this behavior effectively reduces oscillations, the probability-based mechanism limits the system's ability to exploit more favorable RAT configurations. On the contrary, RLNF apparently does not converge to a stable solution, as users frequently change RAT without seeing a significant improvement in throughput. This instability stems from its reliance on probability vectors that do not scale well with larger action spaces, causing oscillatory behavior rather than deliberate switching decisions. In contrast, On-Net and DQN induce more user switches than the heuristic, but this additional flexibility translates into improved throughput, as shown in Fig. 1a. By learning when switching is genuinely beneficial—rather than relying on hand-crafted thresholds—our RL-based method more effectively adapts to network dynamics. Due to the larger action space, all solutions struggle to fairly choose the stations, but the learning-based approaches still perform better (Fig. 1b).

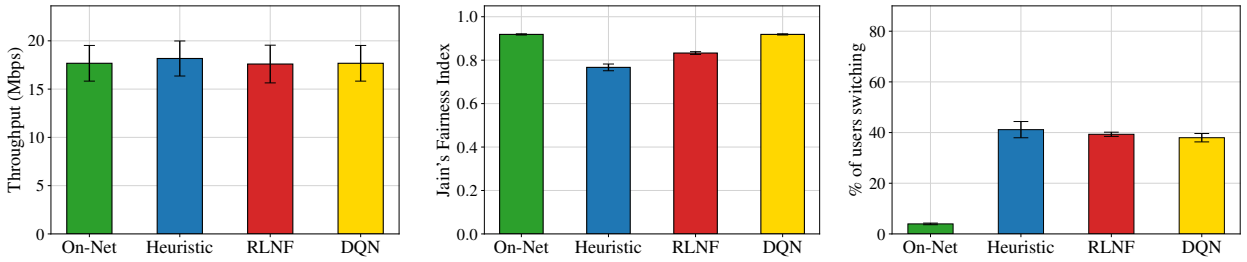
B. Performance in a Real-world Scenario

For this set of experiments, we collected real-world data from network equipment, which enabled us to test the effectiveness of our solution for both wireless and wired technologies. The scenario involves 10 routers from Tiesse SpA, spread across Italy, each capable of selecting between LTE and cabled connections. The dataset includes key network quality metrics, such as RSRP, RSRQ, RSSI, and SINR, to assess 4G connection quality, along with RTT measurements for both 4G and Ethernet links. These metrics were recorded every minute by each router, resulting in a dataset of over $1,200$ datapoints. This dataset is exclusively reserved for evaluation in the testbed and is not used during model training. Instead, the model is trained on a synthetic dataset including 1 million randomly generated datapoints designed to mirror the statistical properties of the real data described above. The training process takes $100,000$ iterations and uses the same hyperparameters described previously in the simulation setup. During training, the model receives as input normalized datapoints based on the real-world metrics described above, such as RSRP, RSRQ, RSSI, SINR, and RTT. This forms the



(a) Average Throughput over time (b) Average System Fairness over time (c) % of users that switch RAT each step

Fig. 1: Performance evaluation *in a simulated environment*: On-Net and DQN learned when switching technology is effective.



(a) Average Throughput over time (b) Average System Fairness over time (c) % of users that switch RAT each step

Fig. 2: Performance evaluation *in a real-world testbed*: On-Net has the lowest number of handovers per step and the highest fairness, while maintaining high throughput.

state space used in this setting. In the testbed, each of the 10 routers performs access network selection over 64 time steps, corresponding to approximately 20 hours of data collection. To ensure statistical significance, we conduct 30 runs with different seeds, and show 95% confidence intervals.

To assess the performance of the solution, we will consider the same metrics that we previously evaluated in the simulated scenario. These performance metrics are presented in Fig. 2.

As evident from Fig. 2a, all benchmarks have comparable performance in terms of average throughput with real data. However, most of them also result in frequent unnecessary handovers. When deploying On-Net, only 4% of the users switch technology at each step on average (90% less than others on average), which still results in high performance, while the other benchmarks incur substantial overhead due to the signaling procedures needed to maintain the service under technology changes (Fig. 2c). Thanks to the penalty term in the reward and the coordination learned during centralized training, On-Net learns to switch only when the gain is significant. As a result, it minimizes the energy overhead associated with frequent handovers [31]. Due to the limited amount of actions in these experiments, *i.e.,* choosing Wi-Fi or Ethernet, all methodologies easily reach a fair solution, but On-Net achieves the highest fairness index. This stems from its learned policy returning more consistent decisions across routers, whereas heuristic or regret-based approaches may still favor suboptimal oscillations or react too strongly to temporary fluctuations.

Fig. 3 shows the average time needed by each agent to perform the action selection based on the adopted methodology. The heuristic is the fastest approach due to its simple mechanism, while applying the learned probability vector from RLNF takes additional time but remains very fast. The two RL approaches are slower due to the forward pass of the neural network, however, the decentralized On-Net requires only 0.06 ms per agent, making it almost 5× faster than the centralized DQN, which averages 0.28 ms. Overall, On-Net offers an effective balance between reducing unnecessary handovers, maintaining high throughput, and keeping decision latency low, while also avoiding the additional signaling overhead of centralized DQN, which requires each agent to request and receive actions from a local controller.

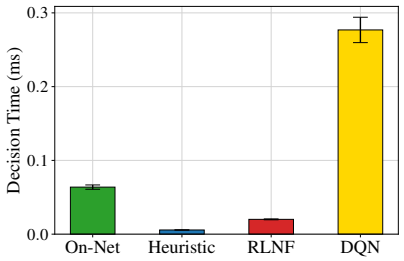


Fig. 3: Decentralization in On-Net makes it decision-making faster than a centralized DQN solution that does not scale well with the number of agents.

V. Conclusion

In this paper, we propose On-Net, a learning-based decentralized solution to solve the access network selection problem. To reduce the signaling overhead that is impractical for large-

scale deployments, we model the problem as a DPOMDP process and adopt the MAPPO framework. We train multiple agents in a collaborative way through the CTDE paradigm, obtaining independent policies to apply at runtime. In this way, decisions are based solely on local observations, which include channel quality metrics and perceived rate or RTT. We validate our solution through simulations and a real testbed, demonstrating that it delivers high throughput, fair resource allocation, and 90% reduction in handovers. The decentralized approach ensures minimal decision latency, operating almost $5\times$ faster than a centralized approach.

ACKWNOWLEDGEMENT

The work of Álvaro Solana was conducted while visiting Politecnico di Torino.

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"). A special thanks to Tiesse SpA, and in particular to Francesco Lucrezia, for the precious support in the realization of real-world experiments.

REFERENCES

[1] A. Alwarafy, B. S. Ciftler, M. Abdallah, M. Hamdi, and N. Al-Dhahir, "Hierarchical multi-agent drl-based framework for joint multi-rat assignment and dynamic resource allocation in next-generation hetnets," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2481–2494, 2022.

[2] H. Shahid, C. Amatetti, R. Campana *et al.*, "Emerging advancements in 6 g ntn radio access technologies: An overview," in *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2024, pp. 593–598.

[3] M. Sarker, M. Z. Hassan, and G. Kaddoum, "Joint user association and bandwidth assignment for digital twin-assisted multi-rat networks," *arXiv preprint arXiv:2505.04829*, 2025.

[4] K. Zia, A. Chiumento, and P. J. Havinga, "Ai-enabled reliable qos in multi-rat wireless iot networks: Prospects, challenges, and future directions," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1906–1929, 2022.

[5] D. Monaco, A. Sacco, D. Spina, F. Strada, A. Bottino, T. Cerquitelli, and G. Marchetto, "Real-time latency prediction for cloud gaming applications," *Computer Networks*, vol. 264, p. 111235, 2025.

[6] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "Rat selection games in hetnets," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 998–1006.

[7] Z. H. Mir, J. Toutouh, F. Filali, and E. Alba, "Qos-aware radio access technology (rat) selection in hybrid vehicular networks," in *International Workshop on Communication Technologies for Vehicles*. Springer, 2015, pp. 117–128.

[8] Q. Wu, M. Tao, and W. Chen, "Joint tx/rx energy-efficient scheduling in multi-radio wireless networks: A divide-and-conquer approach," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2727–2740, 2016.

[9] D. Liu, Y. Chen, K. K. Chai, and T. Zhang, "Nash bargaining solution based user association optimization in hetnets," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 587–592.

[10] F. Moety, M. Ibrahim, S. Lahoud, and K. Khawam, "Distributed heuristic algorithms for rat selection in wireless heterogeneous networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 2220–2224.

[11] K.-L. Tsai, H.-Y. Liu, and Y.-W. Liu, "Using fuzzy logic to reduce ping-pong handover effects in lte networks," *Soft Computing*, vol. 20, no. 5, pp. 1683–1694, 2016.

[12] A. Keshavarz-Haddad, E. Aryafar, M. Wang, and M. Chiang, "Hetnets selection by clients: convergence, efficiency, and practicality," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 406–419, 2016.

[13] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, 2013.

[14] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for ai-enabled wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1226–1252, 2021.

[15] A. Alwarafy, M. Abdallah, B. S. Ciftler, A. Al-Fuqaha, and M. Hamdi, "Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey," *arXiv preprint arXiv:2106.00574*, 2021.

[16] B. Y. Yacheur, T. Ahmed, and M. Mosbah, "Drl-based rat selection in a hybrid vehicular communication network," in *IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. IEEE, 2023, pp. 1–5.

[17] X. Wang, J. Li, L. Wang, C. Yang, and Z. Han, "Intelligent user-centric network selection: A model-driven reinforcement learning framework," *IEEE Access*, vol. 7, pp. 21 645–21 661, 2019.

[18] D. D. Nguyen, H. X. Nguyen, and L. B. White, "Reinforcement learning with network-assisted feedback for heterogeneous rat selection," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6062–6076, 2017.

[19] D. Monaco, A. Sacco, C. Casetti, and G. Marchetto, "Scheduling latency-sensitive tasks in the cloud continuum with hierarchical reinforcement learning," in *NOMS 2025-2025 IEEE Network Operations and Management Symposium*. IEEE, 2025, pp. 01–09.

[20] B. Priya and J. Malhotra, "5ghnet: an intelligent qoe aware rat selection framework for 5g-enabled healthcare network," *Journal of ambient intelligence and humanized computing*, vol. 14, no. 7, pp. 8387–8408, 2023.

[21] M. S. Allahham, A. A. Abdellatif, N. Mhaisen, A. Mohamed, A. Erbad, and M. Guizani, "Multi-agent reinforcement learning for network selection and resource allocation in heterogeneous multi-rat networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1287–1300, 2022.

[22] R. M. Sandoval, S. Canovas-Carrasco, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "A reinforcement learning-based framework for the exploitation of multiple rats in the iot," *IEEe Access*, vol. 7, pp. 123 341–123 354, 2019.

[23] L. Pappone, A. Sacco, and F. Esposito, "Mutant: Learning congestion control from existing protocols via online reinforcement learning," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025, pp. 1507–1522.

[24] S. Hart and A. Mas-Colell, "A reinforcement procedure leading to correlated equilibrium," in *Economics Essays: A Festschrift for Werner Hildenbrand*. Springer, 2001, pp. 181–200.

[25] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in neural information processing systems*, vol. 35, pp. 24 611–24 624, 2022.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, 1997.

[28] R. Kovalchukov, D. Moltchanov, Y. Gaidamaka, and E. Bobrikova, "An accurate approximation of resource request distributions in millimeter wave 3gpp new radio systems," in *International Conference on Next Generation Wired/Wireless Networking*. Springer, 2019, pp. 572–585.

[29] H. Yu and T. Kim, "Beamforming transmission in ieee 802.11ac under time-varying channels," *The Scientific World Journal*, vol. 2014, no. 1, p. 920937, 2014.

[30] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless networks*, vol. 10, no. 5, pp. 555–567, 2004.

[31] R. M. Abdullah, I. Al-Surmi, G. R. Qaid, and A. A. Alwan, "Energy-efficient handover algorithm for sustainable mobile networks: Balancing connectivity and power consumption," *Journal of Sensor and Actuator Networks*, vol. 13, no. 5, p. 51, 2024.