OpenAI dApp: An Open AI Platform for Distributed Federated Reinforcement Learning Apps in O-RAN

Mohammadreza Kouchaki, Aly Sabri Abdalla, Vuk Marojevic

Electrical and Computer Engineering, Mississippi State University, USA
mk1682@msstate.edu, asa298@msstate.edu, vm602@msstate.edu

***Abstract*—The evolution of open architectures for Radio Access Networks (RANs) is revolutionizing network management and optimization. This transformation, fostered by O-RAN, expedites data acquisition and examination by exploiting newly established open interfaces. Moreover, it has led to the rise of near real-time RAN Intelligent Controllers (RICs), instigating a wave of AI-driven applications, or xApps, that employ Artificial Intelligence (AI)/Machine Learning (ML) methods. Nevertheless, deploying xApps as centralized applications presents substantial challenges, such as handling vast data transactions, potential delays, and security vulnerabilities, which are notably prominent within the multifaceted, decentralized, multivendor, and trustless nature of open networks. To alleviate these predicaments, a transition from centralized apps operating in near real-time to distributed real-time apps is imperative for augmented security and efficiency. This paper addresses these complexities by introducing an open platform that integrates a federated reinforcement learning algorithm to operate as distributed Apps (dApps) within the next-generation O-RAN architecture. We present evaluation results in a specific test environment.**

Keywords: O-RAN, xApps, dApps, federated reinforcement learning, distributed applications, autonomous network management, optimization.

I. Introduction

The domain of network management and optimization has been significantly influenced by the emergence of software-based radio access network solutions and the introduction of intelligent controllers. This transformation stands as a key development in the telecommunications industry. The momentum of this shift has been further bolstered by Open RAN (O-RAN), an initiative that champions open and intelligent RANs, leading to a fundamental change in the way data acquisition and analysis are approached within cellular networks. O-RAN introduces the near real-time RAN intelligent controllers (near RT RICs) and inherently enables AI/ML-enhanced microservices in the form of applications called xApps. These applications promise a revolutionary overhaul in network optimization and operation.

Transitioning these technologies from theory to practice presents several technical challenges. One major challenge is the careful management of large-scale data transactions, potential latency bottlenecks, and security vulnerabilities that come with integrating AI-driven applications into the core infrastructure. These challenges become more pronounced within the open network setting, mainly due to its decentralized, multi-vendor framework. This framework increases complexity, especially when deploying xApps as centralized entities within the near-RT RIC.

Another critical aspect is the current reliance on centralized xApps operating in the near real-time domain. This reliance leaves a noticeable gap in the area of distributed applications capable of real-time operations. The space for real-time applications inside radio access networks such as dynamic traffic management, autonomous network troubleshooting, and adaptive quality of service provisioning is obvious [1]. These applications require immediate data processing and decision-making to maintain network performance and user experience, yet the existing centralized xApp solutions fall short in meeting these real-time demands. Existing solutions, such as intelligent O-RAN traffic steering for handover management [2], O-RAN resource allocation using actor-critic reinforcement learning (RL)[3], and connection management optimizing cell association using deep RL (DRL) with graph neural networks[4], all highlight the need for real-time applications within the radio access network. The wide variety of applications, machines, and programming languages in O-RAN adds complexity. Tasks such as managing Kubernetes nodes, handling Docker containers in pods, complex interface deployments, and socket communications become intricate [5]. Additionally, developing databases across multiple programming languages can deter AI/ML researchers and developers from O-RAN experiments.

Despite these challenges, the landscape lacks open platforms that can aid AI/ML developers in easy model development and integration. There's a pressing need for a platform that simplifies application development within the O-RAN architecture, allowing researchers to focus their efforts on the development, testing, and optimization of AI/ML solutions.

In response to these challenges, our research in this paper introduces an open customizable platform designed to tackle the complex challenges inherent to O-RAN and to enable a variety of testing scenarios that reflect the intricate dynamics within O-RAN networks. Our platform initiates a shift towards distributed real-time applications (dApps), Leveraging advanced AI algorithms to amplify the scope and efficiency of O-RAN. The foundational element of this work is the Federated Reinforcement Learning (FRL) algorithm. This algorithm has demonstrated its applicability in areas such as dynamic spectrum access for IoT and wireless communications [6] as well as in enhancing spectrum utilization for D2D communication [7].

FRL is positioned to tackle the existing security concerns associated with expansive, data-intensive open network architectures like O-RAN. By implementing a federated learning approach, the platform ensures data privacy and security, providing a robust framework for the efficient operation of dApps in the decentralized, multi-vendor environment characteristic of O-RAN.

Furthermore, our platform provides a solid framework that supports the development, deployment, and real-time execution of distributed applications. It empowers traditionally centralized xApps to transition into decentralized, responsive dApps capable of real-time operation, thereby reducing latency and improving network optimization. Moreover, the platform offers an intuitive interface for AI/ML researchers and developers, aiding in overcoming the existing hurdles in O-RAN experimentation. This accelerates the research and deployment of AI/ML solutions within O-RAN, paving the way for exploring new optimization methods and advancing O-RAN technology.

The design of our platform reflects a thorough understanding of the O-RAN architecture, along with a vision to address the current technical challenges. It offers a robust, scalable, and secure framework anticipated to foster the transition towards a more intelligent and open RAN ecosystem. With the introduction of this platform, we aim to contribute significantly to the ongoing efforts to advance the O-RAN architecture, propelling the telecommunications industry towards a more open, intelligent, and efficient future.

The rest of this paper is organized as follows: Section II provides a brief introduction to the O-RAN architecture. In Section III, we present our FRL dApp design and its components. Section IV outlines the deployment and integration procedures for the proposed framework. Section V introduces our initial experiment scenario and obtained results, and Section VI is our concluding remarks.

II. O-RAN Background

O-RAN forms a sophisticated architecture through its near Real-Time RAN Intelligent Controller (near-RT RIC), non-RT RIC, and service management and orchestration (SMO), interconnected through interfaces such as E2, O1, and A1. The near-RT RIC offers control and optimization through the E2 interface for near-RT responses to rapidly changing radio and network environments, while the non-RT RIC provides policy and model management by interfacing with the near-RT RIC through the A1 interface for a more policy control perspective over the network. The SMO orchestrates the RAN's end-to-end operation and manages the shared data layer (SDL), a network-wide repository. It employs various interfaces, such as A1 for interacting with the non-RT RIC and O1 to communicate with the network management systems [8].

The dynamic radio environments can incite conflicts, which are managed by predefined strategies. Additionally, the O-RAN structure enforces various cycles: the near-RT cycles overseen by the near-RT RIC for high-speed but non-instantaneous data processing; and the non-RT cycles that are slower, strategic operations governed by the SMO and non-RT RIC. These cycles ensure optimal network performance by balancing efficiency and adaptability.

In the open architecture of O-RAN which is shown in Figure 1, the O-RAN Central Unit (O-CU), O-RAN Distributed

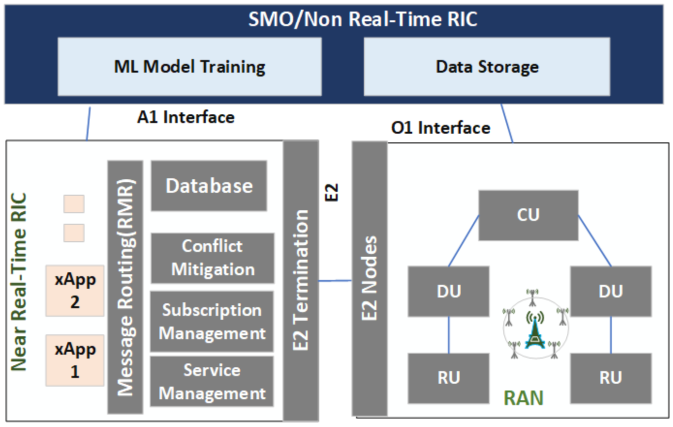

Fig. 1. O-RAN architecture.

Unit (O-DU), and O-RAN Radio Unit (O-RU) implement the functions of a conventional base station in a distributed manner. The O-CU, interfacing with the core network, performs the Packet Data Convergence Protocol (PDCP) and manages the Radio Link Control (RLC) for centralized control and session management. The O-DU communicates with the O-RU and the O-CU, carrying out real-time, low-latency functions such as scheduling and error correction. The O-RU implements the physical layer, data conversion, and radio frequency processing. These components work in harmony to render the network more flexible, scalable, and capable of supporting the evolving demands of modern wireless communications [9].

In the evolving domain of O-RAN, the number of established solutions devised for robust testing and experimentation remains limited. Three major open frameworks have been specifically developed and introduced to date to aid in the process of testing and exploring next-generation O-RAN solutions.

One such framework is Open AI Cellular (OAIC) [10]. It serves as an open-source software architecture, providing a comprehensive toolset that incorporates AI controllers (OAIC-C) and an AI testing framework (OAIC-T). OAIC's primary function lies in its capacity to incorporate AI controllers, initially developed as xApps within the near-RT RIC, for controlling 5G network processes. This integration fosters the expansion of research and development on AI-enabled RAN networks, capitalizing on open-source 5G software and software-defined radios (SDRs).

In tandem, OpenRAN Gym [11] represents another instrumental software-based platform. Designed primarily around the principles of the O-RAN architecture, it specializes in data collection within the near-RT RIC by leveraging the OpenAirInterface software and SDR hardware for RAN implementations. The data collected can subsequently be utilized to develop AI/ML control algorithms through xApps, thereby enhancing the network's management and optimization capabilities. POWDER softwarized testbed [12] is another open platform that enables advanced wireless and data-driven experimental research with SDRs.

III. Proposed FRL dApp Design

This paper advocates for the extension of the current OAIC framework and enables real-time dApps [13]. This augmented platform enables users to configure an array of AI/ML solutions and optimizations [14], extending the scope of the O-RAN architecture. This enhanced platform allows researchers to experiment with diverse testing scenarios and problems such as interference management, beamforming, mobility, vehicular communication, and resource management, leveraging SDRs and available datasets [15], or open-source RAN simulators. Designed for compatibility with a range of environments, this platform expands its applicability and utility for a variety of research purposes. We advocate for the implementation of FRL algorithms to meet the high-performance demands of real-time operation. These algorithms represent the forefront of AI modeling, integrating key aspects of efficiency, security, and swift responsiveness to environmental changes. In the subsequent section, we will delve into a more comprehensive discussion regarding the intricacies of the FRL algorithm, in addition to detailing the proposed approach for its deployment within the context of our system design.

A. Algorithm Design

FRL incorporates the concepts of federated learning (FL) and RL, aiming to train a model that can interact with its environment and learn the optimal strategy, while preserving data privacy and reducing communication overhead. In recognition of the sophistication and potential utility of these models, we are going to discuss them more in-depth here.

1) Federated Learning

FL is an ML technique where multiple edge devices (clients) collaboratively learn a shared prediction model while keeping all the training data on the original device, decoupling the ability to do ML from the need to store the data in the cloud. In other words, FL is a decentralized ML approach that enables model training on a large network of devices, or nodes, while keeping data on the original device.

The generic update rule in FL is

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{k=1}^{K} n_k \nabla F_k(\theta^{(t)}), \tag{1}$$

where θ represents the global model parameters, t the current iteration, η the learning rate, K the number of clients, n_k the number of data samples for client k, and F_k the loss function for client k.

2) Reinforcement Learning

RL agents learn to make decisions by interacting with the environment through performing actions that enhance the model's capacity to adapt to changing conditions and optimize its actions based on reward feedback [16]. This will allow the system to rapidly adjust and respond to variations in the network environment, thereby improving network performance and reliability.

The core concept in RL is the action-value function Q for policy π,

$$Q^{\pi}(s,a) = E\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s, A_t = a\big], \tag{2}$$

where R_t is the reward at time t, S_t and A_t represent the state and action at time t respectively, and γ is the discount factor.

3) Challenges in Federated Learning and Reinforcement Learning

Despite their potential, both FL and RL come with unique sets of challenges. One of the major challenges of FL is dealing with non-independent and identically distributed (IID) data across clients and the possibility of unbalanced data. For RL, the challenges often stem from the need for a significant amount of interaction data, the risk of getting stuck in local optima, and difficulty in designing an appropriate reward function.

Algorithm 1 Federated Reinforcement Learning Algorithm Pseudocode.

Initialize global model parameters θ^0 at the server.

for *each round* $t = 0, 1, 2, ...$ **do**
    The server broadcasts θ^t *to all clients.*
    for *each client* k *in parallel* **do**
        Each client collects experiences by interacting with the environment using policy π_{θ^t}.
        Each client updates their local model parameters θ_k *based on these experiences using RL techniques.*
    end
    The server aggregates the updated model parameters from all clients to update the global model parameters.
    Updated parameters: $\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{k=1}^{K} n_k \nabla F_k(\theta_k^{(t)})$
end
Result: Optimal global model parameters θ

4) Federated Reinforcement Learning

FRL is an approach that combines FL and RL to address their respective challenges. The RL model parameters are updated locally on each client based on their own experiences. These updated parameters are then sent back to the server and aggregated to form an updated global model. Algorithm 1 provides the pseudocode of the FRL algorithm.

Parameter $\theta^{(t)}$ represents the global model parameters at iteration t, n_k the number of experiences collected by client k, F_k the RL loss function (such as the time temporal difference, or TD-error, for Q-learning or the objective function for policy gradient methods) for client k, and η the learning rate.

In the RL context, the gradients are calculated based on the TD-error for value-based methods, such as Q-learning, and based on the policy gradient theorem for policy-based methods. The TD-error for Q-learning is calculated by as

$$\delta = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a'; \theta_k^{(t)}) - Q(S_t, A_t; \theta_k^{(t)}). \tag{3}$$

The gradient of the objective function for policy gradient methods is obtained as

$$\nabla J(\theta) = E\big[\nabla_\theta \log \pi_\theta(A_t|S_t)(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...)\big]. \tag{4}$$

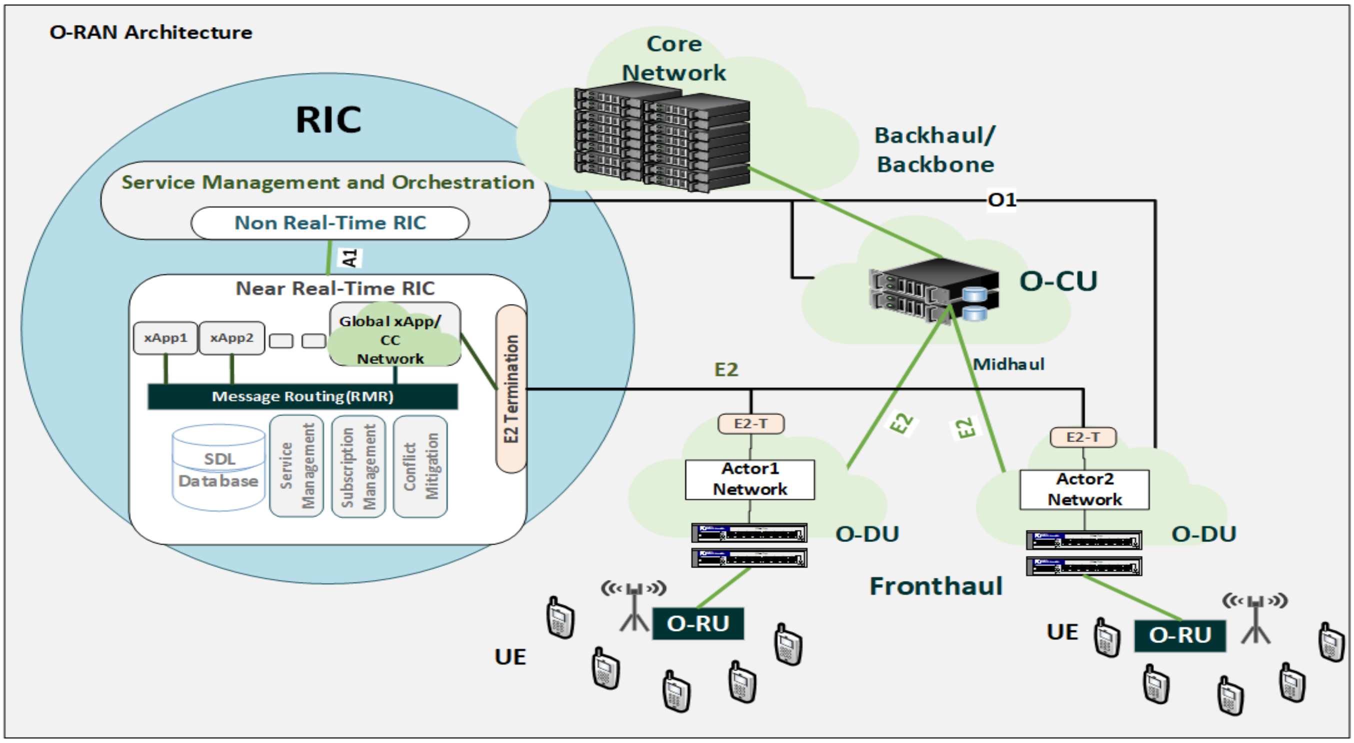

Fig. 2. Proposed O-RAN platform with FRL-based dApps.

By bringing together FL and RL into FRL, we can leverage the strengths of both paradigms, such as privacy preservation and minimized communication overhead from FL, and efficient exploration-exploitation trade-off from RL. The FRL approach can mitigate some of the challenges mentioned earlier, including the non-IID issue in FL and the data efficiency issue in RL, but also introduces some new challenges and complexities, such as dealing with non-stationary and potentially adversarial multi-agent environments. These challenges will create new opportunities for researchers to explore and experiment with a diverse range of optimization techniques in a new architecture enabled by the integration of open AI dApps into O-RAN, fueling innovation in the field.

B. Gym Environment

Delineating observations, reward function, state, and action spaces comprise the bedrock of the FRL algorithm. These parameters intrinsically relate to the environmental setup contingent on specific experimental scenarios. As outlined in the system design and introduction section, srsRAN and ns-3 are prominent, open-source RAN simulators that have been extensively evaluated and utilized in this sphere of study, supplemented by well-crafted frameworks. Complementing these simulators, data-driven platforms, such as OpenRAN Gym and Colosseum, serve as resources for providing extensive datasets that represent a wealth of information extracted from large-scale wireless networks.

We employ OpenAI Gym to cater to diverse RAN simulator types and data collection methodologies, thereby enhancing the research potential with our OpenAI dApp, [17]. OpenAI Gym is a comprehensive, open-source Python library, specifically designed to facilitate the development, comparison, and reproducibility of RL algorithms. It provides a standardized, highly extensible API that promotes smooth interaction between learning algorithms and an array of different environments. This standardized interface not only abstracts the complexities associated with environment setup but also fosters interoperability among diverse RL implementations. In our proposed model, Gym acts as a consistent intermediary between the AI algorithm and any RAN environment, thereby streamlining the definition of essential parameters such as observation, reward function, states, action space, and additional functions.

Interaction with srsRAN, a C++-based software library, is facilitated through socket programming. Meanwhile, ns3-gym, an extension of ns3-ai [18], enables seamless integration with the ns-3 RAN simulator owing to its design foundation on OpenAI Gym. For datasets, Gym is flexible in integrating various data reading techniques. At present, we leverage Pandas to extract data from diverse data sources such as .txt or CSV files, with an open-ended option to incorporate other methods, such as MySQL. All these methods are encapsulated as distinct functions within Gym, enabling them to be invoked on demand.

IV. Proposed FRL DAP Deployment

A. O-RAN Integration

In the context of federated models, there exists a server or global model along with multiple clients. In our proposed model which integrates with reinforcement learning within the O-RAN architecture, the clients are referred to as dApps or federated actors. These dApps are specifically designed to be integrated with the Distributed Units (O-DUs), enabling operations on connected RUs and user equipment (UEs). The global model is designed to reside inside the near real-time RIC and we refer to it as the global xApp. Communication, data transition, and messaging between the server and clients are facilitated through the E2 interface and E2 messages. Figure 2 depicts the architecture of our OpenAI dApp platform.

It is imperative to incorporate the essential libraries, frameworks, and functions of the O-RAN platform, as established by O-RAN developers [19], to integrate a AI/ML controller. These requirements are often streamlined when utilizing Python for app development. Furthermore, considering that the RIC cluster is built on the Kubernetes platform, apps must be developed as containerized applications for deployment on Kubernetes machines. This includes utilizing Kubernetes-native APIs, such as the deployment and service APIs, to define the desired state of the app and expose its functionality to other components within the cluster.

We build the system model in a modular approach to increase the flexibility and adaptability for further changes by users, researchers, and developers. This entails simplifying the main components of the system, including the environment intermediary, training loops, FRL models, network layers, and other critical elements. To ensure flexibility, these components were implemented in Python using Tensorflow2, a widely adopted framework. By leveraging Tensorflow2's capabilities, we facilitate ease of experimentation and future advancements.

B. Installation and Setup

The installation requires the OAIC platform, the model components including Global xApp, dApps, data layers, and environments, and also some essential configurations such as IP and port exposure settings. The open-source model implementation, documentation, and installation instructions will be made available through the OAIC repository [20].

V. Experimental Deployment and Results

In the evolving landscape of cellular networks, one of the most pressing challenges is the user-cell association. This pivotal concern plays a central role for resource allocation. The advent of 5G technology has substantially accelerated the increase in the diversity and quantity of connected devices. This influx of devices, coupled with inherent limitations in network resources, emphasizes the urgency for efficient and equitable distribution of these resources.

In this scenario, our test centers on an operational environment with multiple O-DUs. Each iteration gathers environmental observations encompassing the data rate, the numbers of connected and active UEs, and the Channel Quality Indicator (CQI) as states for the learning agent. The system aims to elevate the throughput while considering long-term fairness. So, these metrics are integrated into the reward function to encourage equitable resource distribution and promote long-term network robustness. We introduce c_{ijk} to represent the decision of assigning channel k to user j served by the i^{th} Du. The total data rate for channel k assigned to user j served by the i^{th} O-Du is d_{ijk} and F_i is the fairness function of the i^{th} O-Du. The reward function R for i^{th} O-Du then formulated as

$$R_i = \sum_{j=1}^{L} \sum_{k=1}^{M} c_{ijk}(d_{ijk}) - F_i. \quad (5)$$

TABLE I
TEST PARAMETERS.

Network Parameters	Value
Actor NN Model Hidden Layers	5
Number of Hidden Layer Neurons	256 -1024
Actor NN Model Parameters	176,418
Number of gNB	2
Number of UEs	14
Maximum Traffic per UE(DL)	1 Mbps
Frequency	6 GHz
Bandwidth	10 MHz
Tx	20 dBm
Discount factor	0.9
learning-rate	0.01
Optimizer	Adam
Number of Timeslots per Episode	1000
Number of Episodes	500-1000

The fairness function F_i is crucial in ensuring that the resource allocation does not disproportionately favor certain users or channels, maintaining balanced network performance over time. It encompasses metrics that assess the equity in resource distribution among different users, acting as a balance to the optimization of data rate. This emphasis on both data rate enhancement and fairness highlights our approach's focus on developing a robust and equitable network infrastructure capable of handling the increased device diversity and data traffic demands characteristic of 5G networks.

The action in this context refers to the cell association, specifically determining which UEs connect to which O-DUs. This association is represented by c_{ijk}, where channel k is assigned to user j. The decisions on cell association are crucial as they impact the overall data rate, fairness, and long-term robustness of the network, which are the primary objectives our system aims to optimize through the defined reward function R_i.

We use srsRAN with SDRs to implement the RAN. Table I provides the testing network parameters and federated dApp network configuration. For testing and evaluation, we are especially looking for two main results to prove the goal of decentralized RAN control. First, we evaluate if the

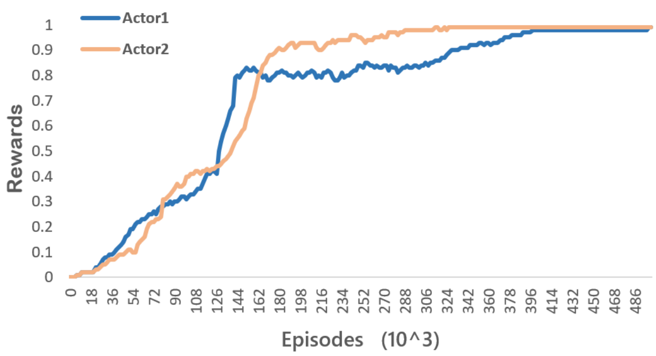

Fig. 3. Average rewards of dApps.

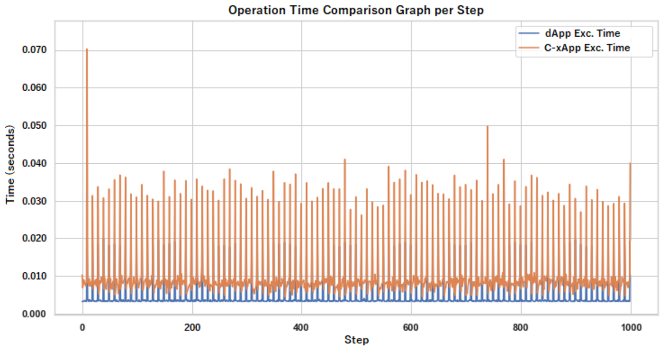

Fig. 4. Operation Time Comparison.

proposed FRL-based dApp can reach the network's target and convergence. Second, we verify real-time operation. For the first evaluation, the trend of the network to reach the maximum rewards is the best evaluation. Figure 3 plots the reward results for two federated actors. It shows that both actors reach the target and converge well. For the second evaluation, we compare the proposed dApps to a central xApp employing the same algorithm and the same actions. The execution time is gathered for a period of one thousand steps for both models. The results shown in Figure 4 show a considerable processing time improvement achieved by the proposed dApp control model. Operating at a time scale between 10 ms to 1 s, the results in Figure 4 show that the xApp operates at 40 ms. Moving to dApps the time improves to below 10 ms. Better results can be achieved by precisely tuning the model and optimization parameters.

VI. Conclusion

This paper has detailed the design, development, and testing of the proposed OAIC dApp, underpinned by a FRL algorithm. The model's efficacy has been assessed in the context of the user-cell association problem in an open-source RAN environment, encompassing two gNodeBs and fourteen UEs. The results indicate a robust convergence of the federated dApps, thereby validating our approach. We have further compared the latency of the deployed dApp with a central xApp under identical conditions. The comparative analysis highlights the efficiency of our designed model, exhibiting its potential to respond in real time in a live environment. We envisage expanding the scope of our work by including a larger and RAN in our testbed. We anticipate that this expansion will pave the way for wider research possibilities, contributing to a more comprehensive understanding of the performance of federated dApps in complex network environments. We remain committed to exploring new frontiers in this domain, continually striving to enhance our model and contribute to the evolution of O-RAN.

Acknowledgment

This work was supported in part by NSF award CNS-2120442.

References

[1] A. S. Abdalla and V. Marojevic, "Multi-Agent Learning for Secure Wireless Access from UAVs with Limited Energy Resources," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[2] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2023.

[3] M. Kouchaki and V. Marojevic, "Actor-critic network for o-ran resource allocation: xapp design, deployment, and analysis," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 968–973.

[4] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. Talwar, "Connection management xapp for o-ran ric: A graph neural network and reinforcement learning approach," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 936–941.

[5] A. S. Abdalla and V. Marojevic, "End-to-End O-RAN Security Architecture, Threat Surface, Coverage, and the Case of the Open Fronthaul," *arXiv preprint arXiv:2304.05513*, 2023.

[6] H.-H. Chang, Y. Song, T. T. Doan, and L. Liu, "Federated multi-agent deep reinforcement learning (fed-madrl) for dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5337–5348, 2023.

[7] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation in d2d-enabled 6g," *IEEE Network*, pp. 1–7, 2022.

[8] O.-R. Alliance, "O-RAN Architecture Description v2.00," https://www.o-ran.org/blog/23-new-o-ran-specifications, 2020, [Online; accessed, O-RAN WG 1, 2021].

[9] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward next generation open radio access networks: What O-RAN can and cannot do!" *IEEE Network*, vol. 36, no. 6, pp. 206–213, 2022.

[10] P. S. Upadhyaya, A. S. Abdalla, V. Marojevic, J. H. Reed, and V. K. Shah, "Prototyping next-generation o-ran research testbeds with sdrs," *arXiv preprint arXiv:2205.13178*, 2022.

[11] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Openran gym: An open toolbox for data collection and experimentation with ai in o-ran," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 518–523.

[12] D. Johnson, D. Maas, and J. Merwe, "Nexran: Closed-loop ran slicing in powder -a top-to-bottom open-source open-ran use case," 01 2022, pp. 17–23.

[13] S. D'Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, "dApps: Distributed Applications for Real-Time Inference and Control in O-RAN," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 52–58, 2022.

[14] A. S. Abdalla, A. Behfarnia, and V. Marojevic, "UAV Trajectory and Multi-User Beamforming Optimization for Clustered Users Against Passive Eavesdropping Attacks With Unknown CSI," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[15] L. Bonati, P. Johari, M. Polese, S. D'Oro, S. Mohanti, M. Tehrani-Moayyed, D. Villa, S. Shrivastava, C. Tassie, K. Yoder, A. Bagga, P. Patel, V. Petkov, M. Seltser, F. Restuccia, A. Gosain, K. R. Chowdhury, S. Basagni, and T. Melodia, "Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation," in *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2021, pp. 105–113.

[16] A. S. Abdalla and V. Marojevic, "DDPG Learning for Aerial RIS-Assisted MU-MISO Communications," in *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2022, pp. 701–706.

[17] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[18] P. Gawlowicz and A. Zubow, "ns-3 meets openai gym: The playground for machine learning in networking research," 11 2019, pp. 113–120.

[19] O-RAN Software Community, "O-RAN Software Community F Release," https://wiki.o-ran-sc.org/display/REL/F+Release, 2022, [Online; accessed, October 2022].

[20] Open AI Cellular, "Open AI Cellular Software Repository," https://github.com/openaicellular, 2022, [Online; accessed, September 2023].