A Multi-Level Deep RL-Based Network Slicing and Resource Management for O-RAN-Based 6G Cell-Free Networks

Navideh Ghafouri, John S. Vardakas, *Senior Member, IEEE*, Kostas Ramantas, and Christos Verikoukis, *Senior Member, IEEE*

***Abstract*—With the deployment of the fifth generation (5G) of cellular networks, the focus of the information society has switched to the next era in which the limitations of 5G will be addressed, and the emerging services and applications will be satisfied. The sixth generation (6G) of wireless networks is envisioned to answer all demands of the next decade, which is only possible with advances in network design and management. This paper first presents a 6G-based network architecture that deploys emerging technologies, including Open-Radio Access Network (O-RAN) and Cell-Free massive Multiple-Input-Multiple-Output (CF mMIMO). Then, a hierarchical network slicing and resource management approach compatible with the presented architecture is defined. The proposed novel Reinforcement Learning (RL)-based scheme benefits from the openness of O-RAN to provide two levels of centralized multi-agent decision-making and decentralized single-agent execution for choosing proper service types by following the objective of maximizing the system capacity while guaranteeing the defined Quality of Service (QoS). To demonstrate the performance of the management method, Deep RL (DRL)-based algorithms for each level are proposed. Finally, the presented simulation results illustrate the effectiveness of the proposed solution in terms of peak data rate, user-experienced data rate, and latency.**

***Index Terms*—6G networks, open ran, cell-free networks, network slicing, reinforcement learning.**

I. Introduction

A. Background

STARTING from the era before 1980, known as the pre-cellphone era, no radio communication means existed in the zeroth-generation of mobile communication networks. The actual mobile telecommunication known as the first generation (1G) started with analog cellular systems in 1980, and so far, approximately every ten years, a new generation of mobile communication has emerged. In 2019, the fifth generation (5G) era started and gained significant attention in research works and among the information and communication society, due to the fact that 5G did not focus only on increasing the network's capacity but also on expanding mobile communication services from humans to things and from consumers to vertical industries. The result was a massive increase in the number of mobile subscriptions that are a combination of humans, machines, and things. It also provides novel and various services such as the Internet of Things (IoT), Virtual Reality (VR), and autonomous vehicles to traditional mobile broadband.

After evolving through five generations during the past four decades, mobile networks now deliver a peak data rate of up to 10 Gbps, with a delay of less than 1 ms, and network reliability and availability of more than 99.999 percent. However, it is envisioned that by 2030, this achievement will not be adequate to cover future demands [1]. Moreover, the evolution of smart-city ecosystems such as smart buildings, smart healthcare, and smart transportation, in addition to the emerging new services in the area of Robotics, Artificial Intelligence (AI), IoT, and Industry 4.0, require a more demanding level of communication capability [2], [3].

This evolution made academia and industry start shifting their attention to beyond 5G or to sixth generation (6G) systems, in order to satisfy the future demands for information and communications technology. The earliest article that discusses the topic of 6G was published in 2018 [4], and since then, many research works have considered 6G and described their visions, potential enabling techniques, use cases, and network parameters [5], [6]. 6G is expected to offer an entirely new service quality and enhance users' experience in current IoT systems [7], while the supported V2X communications will be an instrumental element of future connected autonomous vehicles [8]. In 6G networks, the peak data rate is targeted to be more than 1 Tbps and users will experience a 1 Gbps data rate, while for the emergency use-cases such as medical services, 6G targets to provide a latency between 10 to 100 μs. [9].

Since the 5G era, the concept of supporting dedicated use-cases and providing specific service types based on user demand

Manuscript received 5 January 2024; revised 12 May 2024; accepted 5 June 2024. Date of publication 18 June 2024; date of current version 7 November 2024. This work was supported in part by Research Program H2020 MARSAL under Grant GA 101017171 and in part by the H.F.R.I project ENABLE-6G, ID:16294. The review of this article was coordinated by Dr. Berk Canberk. *(Corresponding author: John S. Vardakas.)*

Navideh Ghafouri and Kostas Ramantas are with the Iquadrat Informatica, 08006 Barcelona, Spain (e-mail: n.ghafoori@iquadrat.com; kramantas@iquadrat.com).

John S. Vardakas is with the Iquadrat Informatica S. L., 08006 Barcelona, Spain, and also with the Department of Informatics, University of Western Macedonia, 50150 Kozani, Greece (e-mail: ivardakas@uowm.gr).

Christos Verikoukis is with the ISI/ATHINA, Greece, CEID, University of Patras, 26504 Patra, Greece, and also with the Iquadrat Informatica S. L., 08006 Barcelona, Spain (e-mail: cveri@upatras.gr).

Digital Object Identifier 10.1109/TVT.2024.3415656

has been considered in mobile networks. The previous generations used the one-fit-all strategy for all users and demands; however, the software-based architecture of 5G allowed the definition of different demand-based service types to be offered simultaneously and the shifting of 5G to a complete multi-tenant ecosystem. The 3 rd Generation Partnership Project (3GPP) has considered the three primary 5G use cases, including Ultra-Reliable and Low Latency Communication (uRLLC), Massive Machine Type Communication (mMTC), and Enhanced Mobile Broadband (eMBB) [10]. The idea of the Infrastructure as a Service (IaaS) cloud computing model, whereby different tenants share computing, networking, and storage resources in order to create different isolated, fully functional virtual networks on a common infrastructure, led to the concept of network slicing that has gained the attention of many researchers [11]. In the context of beyond 5G, dynamically creating cost-efficient, end-to-end network slices and dedicating them to diverse services is considered an important feature. Some of the research works present the challenges and requirements for 6G slices and service types [12]. To mention one, authors in [3] believe that 6G will be providing answers to many requirements through the definition of the Further enhanced Mobile Broadband (FeMBB), ultra-massive Machine-Type Communication (umMTC), Mobile Broadband and Low Latency (MBBLL), and massive Low-Latency Machine Type communication (mLLMT) services. Thus, it should be taken into consideration that the application of the network slicing concept in 6G networks can be considered as a complex procedure, since it requires the consideration of multi-level network structure with multiple controllers and the diversity in network resources in the access, front-haul, and mid-haul domains of the network [13].

B. Related Work

Even though the concept of 6G has been recently proposed, it has already gained significant research attention. In what follows, we provide an overview of recent works related to the network slicing concept in 6G networks, with a clear focus on the application of Machine Learning (ML) techniques. The following are the most similar works to help us clarify the advantages of our research in the next section.

In research work [14], a RAN slicing control mechanism has been proposed to improve the QoS in 6G networks. This mechanism is composed of assignments to the next generation NodeB (gNodeB) at the network level. The assignment is followed by adjusting the slice configuration at the gNodeB level and finally allocating resources to the users at the packet scheduling level. The authors in [15] propose a framework with a Global Resource Manager (GRM) and multiple Local Resource Managers (LRM). In this scheme, resources are firstly allocated to each tenant based on slice requirements, and then the allocated resources will be optimized and adjusted. This paper compares DQL, Q-learning, and Greedy algorithms to implement the proposed framework, while DQL shows the best results compared to the other solutions. In [16], authors consider a multi-band network with a radio base station and a THz base station located in the cell center. This study considers two FeMBB and eURLLC services and tries to jointly maximize FeMBB user data rate and eURLLC user reliability. This research work compares DQN, double DQN, and DuelDQN techniques and shows that DuelDQN has the best performance for its approach. The work in [17] considers a 6G core and a transport network in which the RAN consists of one cellular network. Then a tailored slice model based on that 6G network model is proposed. Finally, the tailored network slicing algorithms are presented and compared to the Greedy algorithm. Authors in [18] consider a network for providing autonomous driving services to vehicles on a highway that is 2 km long. This highway benefits from 2 base stations with a distance of 1 km. They propose an AI-native network slicing approach, in which AI exists in both Software-defined networking (SDN) and slices. In the result section, the system cost has been monitored for the evaluation procedure. In research [19], a data packet scheduling scheme has been proposed which is based on the distributed Deep Deterministic Policy Gradient (DDPG). The study also believes that using centralized training and distributed execution improves the stability of their algorithm. Results show that the proposed algorithm achieves fewer data overflow compared to the Greedy and Random algorithms.

The research work [20] first explains how the current framework for network slicing in 5G is limited in various aspects. Later it proposes a new framework in which slices have embedded management support. This work believes the new approach will address the deficiencies of 3GPP approaches to network slicing. In addition, in [21], a hierarchical RAN slicing scheme was proposed in which a loose control is performed to ensure QoS performance. The proposed analysis targets the improvement of the Spectral Efficiency (SE) of the slices. The considered system in this research work is a typical downlink cellular network with one base station, and two slice types, eMBB and uRLLC are considered. The proposed algorithm used in this work is based on the DDPG. The authors in [22] propose an actor-critic DRL framework for network slicing in a 6G-like RAN. The 6G-like RAN considered in this study is a CF mMIMO network scheme with edge-cloud computing capabilities. The average latency of the scheme has been compared to some other ML-based algorithms, such as DDPG, and the results show a better performance. The research work [23] considers network slicing in both ground and satellite networks. In this proposed scheme, an ML-based technique is used to weigh passes in order to assign the proper path to each request. The user acceptance rate of the proposed framework is compared to the shortest delay and best-fit algorithm. Furthermore, the approach in [24] considers a number of slices on common network infrastructure as an undirected graph. This research work tries to address the issues of network slicing and routing jointly. This work uses an actor-critic DRL model in a packet-level experiment platform while the URRLC, VoLTE, audio, and the satisfaction ratio of the video services are presented. Authors in [25] discuss the scheduling of radio resources in C-RAN architecture, which is considered slice-aware radio resource calendaring. They propose heuristic algorithms and monitor the rate of request acceptance. In [26], a management framework for network slicing is presented that tries to minimize the management overhead with respect to

the considered SLA constraints. The system considered in this work consists of Central Units (CU) and Distributed Units (DU) as Virtual Network Functions (VNF) at the edge. Moreover, Monitoring Systems (MS), Decision Engines (DE), and Analytic Engines (AE) have been considered as entities of this system. Finally, the work in [27] uses DQN dynamically in order to allocate resource blocks to mobile units. This work integrates blockchain to record the allocated resources. The system in this work has been designed using the Gym library, and the proposed scheme has been compared to Duelling DQN and non-DQN approaches.

While each work in the state of the art has its own advantages, in this work, we try to go one step back in improving the network to meet the 6G requirements. In other words, our investigation starts from the system model and moves forward to the management of the new system model. The following section provides the details of how this work goes beyond the state of the art.

C. Motivation and Contributions

Even though the state-of-the-art review in this paper does not cover the total number of related works due to space limitations, the following novelties can be observed by having a thorough study of the related works:

- Considering new features for 5G and 6G system models, i.e., CU and DU entities, CF structures and 6G-like RAN configurations for the 6G systems [22], [26], specifically, studying resource management and network slicing in O-RAN [28] and a combination of O-RAN and CF mMIMO [29] (which is similar to our work),
- Considering the benefits of centralized training and distributed execution for the scalability and complexity of future networks [22],
- Proposing two levels of resource sharing and management [21], [30],
- Offering more than one service type in the network slicing and resource sharing [16], [21], [30].

However, in this research, we consider a combination of more extended novelties in the system model, service types, and the proposed approach. Firstly, we benefit from the ultimate programmability and openness offered by the newly emerged O-RAN, allowing the proposed approach to be later deployed in real scenarios contrary to all research works considering ML-based techniques in RAN system models that do not support entities hosting ML algorithms. Our system model also considers CF mMIMO to robust the 6G architecture by eliminating the cell boundaries and their related limitations. On the other hand, removing the boundaries creates more challenges when defining the slices. This is due to the fact that the RAN status is changing continuously since there is no pre-setting to have fixed groups of users being served by each network entity. Since in network slicing, slices should not share the same resource blocks, managing this delicate issue in a CF-based system model is very challenging. Thus, intelligent monitoring and online management are needed to support this dynamic and distributed network while not adding more complexity to it. Therefore, having a distributed intelligent approach can be what this system model requires. One of the key challenges in distributed management is synchronization and comprehensiveness. This means that while it is beneficial to have distributed resource management to improve the complexity, and scalability and be practical, the slice defining and resource assigning need to be reliable considering the availability of resources and overlaps of in-use resources. As a result, the proposed approach benefits from a Multi Agent RL (MARL)-based centralized decision-making level and a RL-based distributed execution level. Two different time slots for the decision-making and resource assigning are needed. In other words, in a level of MARL with longer time steps, the agents cooperate and make decisions, and in a nested RL level with local observations, single agents execute the decisions. It should be noted that this approach is different from Federated RL (FRL) since this work considers two hierarchical physical and performance levels along with their related time levels for an overal management of the network.

Secondly, this work also considers all envisioned service types for 6G in contrast to the existing works studying only one or two. Since different service types require contradictory demands, while it is more similar to real scenarios, it adds more difficulties to the management.

According to all mentioned above, the novelty of this work comes from its comprehensiveness. While the state-of-the-art focuses on solving one specific problem or considers less challenging system models, in this work we try to consider a more realistic scenario and propose a generic management approach. Even though we also propose the details of this approach, they can be changed or improved. Thus, to the best of our knowledge, this work goes beyond the state-of-the-art by considering a complex but robust system model, addressing the complexity of slice management and source sharing by deploying two levels of performance and time, being general, scalable, and able to consider any number of service types for 6G, and being practical by considering the entities in the RAN to place the intelligent algorithms. All those, as mentioned above, together create the novelty of this work.

The benefits of the proposed DRL-based two-level approach lie in its compatibility with the new 6G-related technologies and the complex system model of 6G networks, resulting from its imitation of human teamwork (e.g., like a soccer team). This means that our approach benefits from a set of agents in the network that observe the network thoroughly and make decisions cooperatively to handle the user requests and slices. After the centralized decision-making is realised, single agents, which are placed closer to the resources and only observe the resource blocks, try to execute those decisions without needing all the network status information and just by assigning resources optimally. In this way, the decision-making process is consistent, and the network is synchronized; the resource allocation process guarantees the QoS, and the only information transferring or added overload is sending decisions through the A1 interface (i.e., the interface between the near-real-time and non-real-time controllers of the O-RAN architecture). In addition, having two levels of performance results in reaching both the general goal of maximizing the network capacity and providing the QoS of users. Moreover, the complexity is distributed in two levels

and is decreased even further by selecting proper algorithms. Thus, the utilization of two levels of decision in our approach is compatible with our system model, and the incorporation of a centralized decision-making scheme is able to overcome any bottleneck and inconsistency in the system while having decentralized execution to improve scalability and reduce the complexity.

The rest of this paper is organized as follows: Section II describes the technologies considered in this work. While Section III introduces the proposed scheme's general idea and explains the implementation details, in Section IV, the simulation details and results of our implementation are provided. This structure is to emphasizes that the general approach can be implemented in other ways and with other algorithms too; however, we provide the implementation that we see best. Finally, Section V concludes the paper and reviews the possible future steps and improvements.

II. Preliminaries and Enabling Technologies

This section starts with a brief review of the deployed methodology. We have studied ML and, more specifically, RL techniques to find the most appropriate implementation for our proposed approach. The introduction continues with the presentation of technologies such as O-RAN and CF mMIMO that are used in the considered network architecture in this research work.

A. ML in Wireless Networks

ML is a form of AI in which machines learn to perform tasks based on data-driven decisions independently. ML has shown great potential in data analysis and interactive decision-making based on a set of parameters. Wireless communications is one of the areas that is benefiting from ML techniques in order to address various issues in network design and network management. Since ML can model systems that can not be represented mathematically, several research works believe that it will be a main component in 6G networks. ML techniques could generally be used for real-time monitoring, management, and control. This need is the result of the fact that 6G is inherently dynamic and will have a complex network architecture. Thus, to adapt to the system's constant changes and monitor it continuously while considering multi-tenancy and the required QoS, ML techniques have been considered as a promising enabling technology in 6G. It is envisioned that 6G will be transformative and will revolutionize the progress of wireless networks from the concept of connected things to the connected intelligence concept [31], [32].

All ML techniques can be categorized into three main types based on the source and type of data and the learning process. The first two groups, named Supervised and Unsupervised Learning, are commonly used for classification and regression, or finding the pattern and the structure of the data. While these two types learn from input data from a data set, the third ML technique group, RL, can learn from interacting with their environment. Based on our problem statement, RL is the most appropriate technique for managing the dynamic environment of our system model.

RL, as a sub-field of ML, provides an agent that learns from interacting with the environment. The agent takes actions and, by observing the result of its actions, tries to learn an optimal sequence of actions to reach a specific goal. Since the performance of this technique highly depends on the representation of the input data, pre-processing the data is a critical step in using RL. Deep neural networks greatly help these techniques, thus providing high efficiency when solving real problems. To this end, this advanced intelligence paradigm is envisioned as a viable solution for the next generations of wireless networks [33], [34]. Several research works in different areas of communication networks, including Mobile Edge Computing (MEC), Vehicular Networks, and IoT, have considered RL or DRL in order to address their problems [35], [36], [37], [38].

The advances and the success of the single agent RL approach led to having more than one agent to meet the need for multiple decision-makers that simultaneously interact in real-world applications. Thus, MARL systems have gained attention with applications in various areas, targeting to deal with challenging problems with real-world complexity [39]. For example, authors in [40] used MARL to allocate resources in 6G in-X sub-networks; this approach confirms that MARL is entering the 6G wireless communication research area.

In current and future wireless communication paradigms, network slicing has become a challenge and, thus, a key research topic. 6G services have contradicting requirements in terms of data rate, latency, and reliability. As a result, the end-to-end QoS and the Quality of Experience (QoE) are vital aspects of 6G. Conventional network optimization methods without learning capabilities are not able to cope with the complex and dynamic nature of 6G. On the other hand, AI-based techniques are able to confront the challenges related to network management and resource utilization in 6G. Specifically, DRL provides an optimal control policy using agents that can interact with the constant-changing environment and handle the controlling issues [41], [42]. Several research works use various DRL techniques, including Deep Q-Learning (DQL) and actor-critic learning, for network slicing and resource allocation in 5G networks [43], [44], [45], [46], [47].

B. Open-RAN in Future Intelligent Networks

The application of ML in future networks requires its own infrastructure. There have been efforts to incorporate intelligence into the 5G core network, but since the 6th generation is still in its early stage of research and standardization activities, this procedure can take advantage of the unique features of the Open-RAN, which the O-RAN Alliance has introduced [48]. Together with the advances in network virtualization and programmability, O-RAN can reshape the network architecture and provide flexible network implementations. The concept of O-RAN facilitates the movement from a closed hardware-based ecosystem and conventional vertical stack towards an open modular cloud-based ecosystem in which interoperable software-based network entities can be implemented.

By evolving towards O-RAN, each base station will be changed to one O-RAN Centralized Unit (O-CU), one O-RAN Distributed Unit (O-DU), and one O-RAN Remote Unit (O-RU), which are adapted to the O-RAN Alliance definition [48]. Such a programmable and highly flexible structure will also provide the infrastructure for the presence of AI/ML in the network. This presence is realized through the RAN Intelligence Controller (RIC) module in O-RAN, which allows ML algorithms to lunch new service deployments. The RIC consists of two further modules named near-Real-Time (near-RT) RIC, which hosts the xApps that manage the users' QoS requirements and the non-Real-Time (non-RT) RIC that uses rApps with optimization and management functions in order to monitor RAN components and perform network orchestration. Using appropriate real-time engines and network interfaces for collecting necessary ML data will help perform the network control. [49], [50], [51], [52], [53]. It is worth mentioning that there exists research works that focus on implementing O-RAN specifications in 5G, and on incorporating ML models to configure and optimize the network [54], [55].

C. Cell-Free Massive MIMO in Future RAN Implementations

When following the 6G physical-layer-related research works, it is believed that 6G will remove the conventional cell structures by incorporating the unique features of the CF mMIMO concept. CF mMIMO has been proposed to overcome the boundary effects of cellular networks by distributing many access points in a geographic coverage area, which are able to serve multiple users in the same time-frequency resources coherently. In other words, CF mMIMO provides multi-user interference suppression and thus achieves stronger diversity gains with the help of scheduling, joint transmission, and coordinating beamforming. Some of the benefits of CF mMIMO, in comparison to the conventional cellular topologies, include improved interference management, high Spectral Efficiency (SE), high Energy Efficiency (EE), low latency, low power consumption, and high reliability. As a result, it has been considered as a promising solution to deliver huge amount of data to the end-users while covering the strict QoS requirements of 6G services. Moreover, IoT services can also benefit from a CF-based network architecture that is able to handle significantly massive connectivity [56], [57], [58], [59], [60]. It should be noted that while benefiting from CF mMIMO in RAN, the complexity that will be added to the network slicing and managing resource assignments is remarkable. This is due to contradict natures of network slicing and CF structure. While slices should be defined precisely so as not to share any common resources, CF aims to remove any boundaries, groups, and predefined and fixed allocations.

III. The Proposed Network Slicing and Resource Management Scheme in 6G Environment

A. Problem Statement and General Idea

Following the approach in [26], [29], [52], [61], [62], we consider a 6G-based network architecture that consists of O-RAN and CF mMIMO. The complexity of this system model is mostly the result of the incorporation of the CF technology since any RU and DU can provide service to each user. This challenge can be addressed by defining slices that do not share the same resource blocks, but can be passed through a common RU or DU and are reconfigured in every new time slot. Due to the unique features of the CF part, in addition to the robust dynamic nature of 6G, this problem can not be modeled and solved mathematically. Thus, we are motivated to use the openness and flexibility of O-RAN and apply a novel approach to the application of AI in the system in order to handle these issues intelligently. It is evident that 6G networks need to benefit from a dynamic slice configuration/reconfiguration and resource management system. This dynamic nature, along with the complex architecture of 6G, is a significant problem to the concept of network slicing, which offers separate portions of resources to each user. In addition, network slicing introduces additional complexity to the network; thus, all these features confirm the necessity of using ML techniques for 6G resource management, even though the complexity of such an ML-based management scheme can become significant. Furthermore, when considering the details of network slicing, we need to ensure that the scheme can achieve maximum utilization of the network capacity while providing the agreed QoS to the users.

These challenges motivated us to take into account both the intelligent O-RAN-based controllers and the benefit of having a two-level nested management model. This approach targets reducing the complexity of the scheme by distributing it to the two levels and splitting the desired duties and goals between these levels. As a result, while management and slice defining are synchronized and consistent, the users' QoS are guaranteed, and consequently, the probability of success and reliability increases. Moreover, the scalability and complexity of the management is improved significantly.

Selecting the most appropriate ML technique is an important issue for the performance of the proposed scheme. As mentioned before, the complex and dynamic nature of the stated problem omits the mathematical-based static solutions, and the only way to cope with this complex problem is by imitating human management. Among all ML techniques, RL is selected in this work in order to benefit from agents who are continuously interacting with the network while learning and making real-life-kind decisions.

By following the research related to RL, we notice that the most common application of this RL approach is in games and robotics, which result from human behavior for interacting with the environment, learning, and acting. However, the application of multi-agent RL in other research fields, including communication networks, is limited. We consider our problem as a game in which agents continuously provide what is needed for the requests they receive, and they win if they maximize the use of available resources in the network; the latter part highlights the need to assign the resources optimally. Thus, we were inspired to benefit from the cooperative multi-agent RL approach to address the delicate issue of consistency by not assigning the same resource for more than one user and having a smooth distributed management instead of having a bottleneck. We propose a set

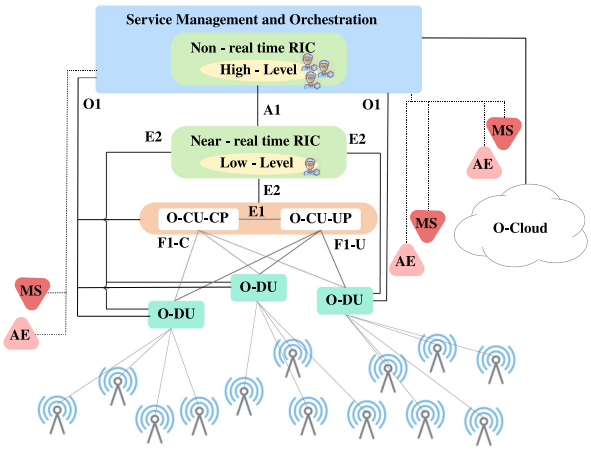

Fig. 1. System Model.

of agents that each chooses one service type for one request cooperatively. The cooperation should result in maximizing the network capacity. We propose to separate the execution level, so that not only can we set another goal of guaranteeing the user's QoS but also have a distributed execution by deploying single agents. At this level, each single agent must take a set of allocating actions to provide the predefined QoS for any specific slice type. Thus, in our network environment, we consider a set of agents who receive requests and jointly choose a slice type for each request, while then one agent assigns resources for each slice type until the requirements of that slice type are met. The details of this idea and its compatibility with our system model will be discussed in the next section.

B. System Model and Compatibility

As mentioned in the previous section, the proposed system model follows the specification of the O-RAN, where the non-RT RIC controller is placed in the Service Management and Orchestration (SMO), as depicted in Fig. 1. In the RAN section, a CF network is considered, where users (UEs) are served by a set (or cluster) of O-RUs. In the mid-haul, the near-RT RIC is responsible for real-time operations related to radio resource management and O-RU cluster formation (based on the CSIs collected from the O-DUs). Furthermore, the non-RT equivalent of the near-RT RIC is located at the SMO and is responsible for the non-RT functions. This non-RT RIC is connected to the near-RT RIC through an A1 interface. The SMO is connected to the O-RAN cloud (O-cloud) as well as to the O-DUs using interfaces O2 and O1, respectively. Interface E2 connects O-CUs and O-DUs to the near-RT RIC, and O-CU can manage the O-DUs through interface F1. Last but not least, our system model benefits from the AEs and MSs as mentioned in [26], [61], which are distributed in different locations of the edge infrastructure, and they provide online information regarding the availability of the network resources to the SMO. The rest of the components and interfaces are all O-RAN-based, as specified by the ORAN Alliance.

In such a multi-layer network architecture, resources are distributed in various locations, including the edges of the network; the non-real-time RIC can benefit from all the connections from SMO to MSs, AEs, O-DUs, O-RUs, and O-cloud to gather information regarding the network status. As a result, the first level of our proposed scheme, referred to as the high-level part in the rest of the paper, will be located in non-RT RIC. At this level, a multi agent DRL set will observe the network system and the available resources, and then, based on their observations and cooperation, a joint high-level policy and action will be applied. These agents try to assign a proper slice type to each request based on system status and a global goal to optimize in longer time slots. To this end, we consider a team of agents who observe the network thoroughly, and based on their observations, the cooperation decides which slice should be assigned to each user so that the final result of their decisions maximizes resource usage. Since, at this level, agents can observe the entire network status while they are able to cooperate, the synchronization and consistency of assignments will be handled in this level; thus, if the high-level part assigns a slice to a request, the resources needed for that request are guaranteed.

The near-RT RIC is connected to the non-RT RIC through an A1 interface and hosts the second level of the proposed scheme, referred to as the low-level part. In this part, single agents perform the decentralized execution since the centralized decision-making in the high-level part guarantees the availability of the resources. Each agent receives a chosen slice type from the high-level part and tries to assign resource blocks to realize the slice type. Since near-RT RIC is also connected to O-CU and O-DUs, which are responsible for coping with O-RUs and users in the CF mMIMO, the low-level part can benefit from these connections to assign the resource blocks in the network entities.

By introducing two levels of operation, the need for defining two levels of scheduling seems inevitable. Thus, we consider a long time slot for the decision-making level and a short time slot for the execution level. Inevitably, the long time slot is within the control loop offered by the non-RT RIC (longer than 1 s), and the short one is limited by the near-RT RIC control loop (between 10 ms to 1 s). As a result, for every high-level decision to be addressed by the low-level agents, a minimum of 1 s time is assigned. Thus, it is guaranteed that the low-level agent finishes the assignment at some point between 10 ms to 1 s. As the proposed network architecture considers CF mMIMO, where O-RUs provide connectivity to the end users, assuming a fixed path to assign resources cannot be viewed as a realistic assumption. Thus, to address this complex issue, in every new longer time slot, requests will be addressed with new performance decided by agents. It is worth mentioning that the availability of guaranteed resources by the high-level is only valid if the low-level assignments are optimal and single agents hold on to both lower and upper limits. Algorithm 1 demonstrates the hierarchical structure of the proposed approach.

C. Implementation Technicalities

Following the description of how the proposed scheme fits into the system model, the technical details of the implementation

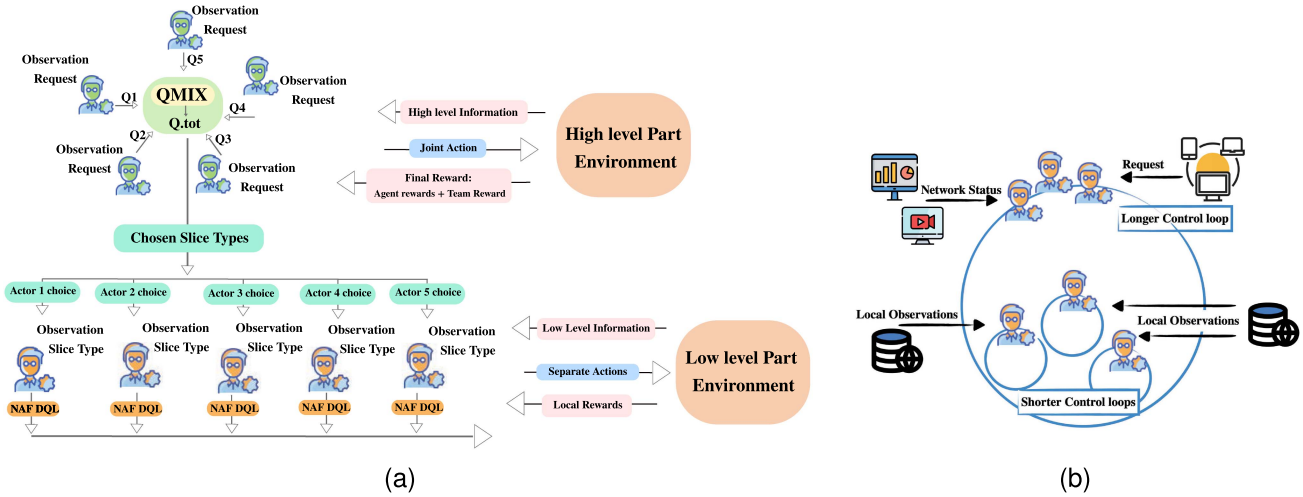

Fig. 2. The proposed approach: (a) Agents and environments communication. (b) Time slots and control loops.

Algorithm 1: The Structure of Management Approach.

LONGER TIME SLOT $(\mathbf{T_L})$:
  $\mathbf{Receiverequests}: \mathbf{R} = \{r_1, r_2, \ldots, r_n\}$
  $\mathbf{Slicetypes}: \mathbf{S} = \{s_1, s_2, \ldots, s_n\} \leftarrow$ Cooperation + Observation + joint policy
  $\mathbf{GeneralReward}: \mathbf{R_T} =$ Maximizing the resource in-
  used
  Low-level $\leftarrow \mathbf{Slicetypes}: \mathbf{S} = \{s_1, s_2, \ldots, s_n\}$
  SHORTER TIME SLOT $(\mathbf{T_S})$:
    $\mathbf{Realizing}\ S_1 \leftarrow$ Observation + policy
    $\mathbf{LocalReward}: \mathbf{R_1} =$ Providing the predefined QoS for S_1
  END OF $\mathbf{T_S}$
  SHORTER TIME SLOT$(\mathbf{T_S})$:
    $\mathbf{Realizing}\ S_2 \leftarrow$ Observation + policy
    $\mathbf{LocalReward}: \mathbf{R_2} =$ Providing the predefined QoSfor S_2
  END OF $\mathbf{T_S}$
    ...
  SHORTER TIME SLOT$(\mathbf{T_S})$:
    $\mathbf{Realizing}\ S_n \leftarrow$ Observation + policy
    $\mathbf{LocalReward}: \mathbf{R_n} =$ Providing the predefined QoS for S_n
  END OF $\mathbf{T_S}$
END OF $\mathbf{T_L}$

need to be taken care of. As shown in Fig. 2, both levels are described as follows:

High-level Part: As mentioned earlier, 6G networks will consider additional service types to the three main types already available in 5G. Thus, according to the service types suggested in [9], we consider 5 slice types for our 6G environment, including FeMBB: further-enhanced mobile broadband, umMTC: ultra-massive machine-type communications, ERLLC: extremely reliable and low-latency communications, LDHMC: long-distance and high-mobility communications, and ELPC: extremely low-power communications; corresponding use-cases and KPIs for each service type are listed in Table I.

TABLE I
THE 6G SERVICE TYPES CONSIDERED IN THIS WORK

	Service type	Use cases	Requirements
1	FeMBB	Holographic Verticals Full-Sensory Digital Reality (VR/AR) Tactile/Haptic Internet UHD/EHD Videos	Peak Data Rate: >1 Tb/s User-Experienced Data Rate: 1 Gb/s Area Traffic Capacity: 1 Gps/m^2 $SE: 5 - -10\mathring{A}\sim$
2	umMTC	IoE Smart City/Home	Latency: 10–100 µs Connectivity Density: 107 d/km^2 $EE: 10 - -100\mathring{A}\sim$
3	ERLLC	Fully Automated Driving Industrial Internet	Latency: 10–100 µs Mobility: >1,000 km/h Connectivity Density: 107 d/km^2
4	LDHMC	Deep-Sea Sightseeing Space Travel Hyper HSR	Mobility: >1,000 km/h
5	ELPC	Nanodevices/ Nanorobots/ Nanosensors e-Health	Connectivity Density: 107 d/km^2 $EE: 10 - -100\mathring{A}\sim$

The number of agents at the high-level part is relative to multiple criteria, including how many tenants at the same time should be supported in addition to the number of service types and their popularity in the network. This is because each agent receives one request in every longer time slot, thus, the total received requests equals the number of agents. In the following, we consider 5 agents in the MARL set to estimate equal requests for each service type in every longer time slot. In this way, we predict approximately one request for each service type. However, this number can change in case of having more service types or more number of users. The observation of each agent includes a request in addition to available service types. The request can be just one or a combination of defined KPIs, as

reported in Table I. For example, a request can correspond to a latency requirement of $10 - 100\,\mu$s, meaning the agent can assign one of umMTC or ERLLC. Following the goal of maximizing the use of network capacity, the agent will make an optimal choice that gives a better result in cooperation with other agents' requests and decisions. Each agent tries to find the proper slice type according to the request. This means that no assignment will happen if the resources for the requested slice type are unavailable in the system. As a result, if agents assign the proper slice type or wait in case of no availability, they will receive their individual rewards. However, the general joint goal is to maximize the use of available resources and thus maximize the network capacity. Consequently, the final reward of this level is a combination of each agent's reward and the reward the team receives based on how much resource have been assigned at the end of the long time slot. We consider the higher ratio of the final reward for the joint goal to encourage allocating more resources while guiding the agents to make decisions. Accordingly, The high-level reward is calculated as follows:

$$R_{tot-agent_i} = \frac{1}{3} * R_{agent_i} + \frac{2}{3} * R_{team} \tag{1}$$

in which R_{agent_i} is equal to 1 if the assigned resources are available. Otherwise, the agent's individual reward is 0. R_{team}, on the other hand, is the ratio of resources in-use to all resources, which means more resources in-use result in a better reward. The considered ratio is the final decision of various trials with different ratios.

One of the main problems in multi-agent settings is the exponential growth of joint action spaces with the number of agents. To overcome this complexity, we consider the QMIX algorithm [63] for our MARL setting, which lies between fully centralized and independent MARL. This means the algorithm learns decentralized policies in a centralized fashion and represents complex centralized action-value functions in a factored manner. Moreover, it does not require on-policy learning and, thus, remains practical even when deploying more agents [64]. It is worth mentioning that fully cooperative MARL is an active area of research. Still, there is a challenge of providing centralized training for agents to find optimum global policy while ensuring decentralized execution. QMIX is similar to Value Decomposition Networks (VDNs) [65] since they both can learn a centralized but factored *Q.total-value* by representing the *Q.total-value* as the sum of individual value functions that condition only on individual observations and actions. Thus, VDN also lies between independent Q-Learning and centralized Q-Learning, but in QMIX, the full factorization of VDN is not needed to extract decentralized but fully consistent policies. QMIX is made of agents'networks producing each agents' *Q-value* and a mixing network that combines them in a complex, non-linear way to ensure consistency between centralized and decentralized policies. It is worth emphasising that this MARL differs from FRL, which involves training a ML model across multiple decentralized edge devices. In this MARL set, multiple agents interact with a common environment simultaneously, so the existence of other agents affects the environment of each agent continuously. Also, the considered algorithm is cooperative but does not need a joint action space. Instead, each agent has a network producing a *Q-value* and a mixing network that receives the Q-values) and produces the *team Q-value*. Later, the *team Q-value* can be presented with each agent's *Q-value* again. Thus, The chosen *team Q-value* produces the individual *Q-values* that produced that highest value. This will be mapped to the related slice type.

Low-level Part: While at the high-level part the multi-agent set tries to maximize the network capacity by choosing proper slice types to the requests, the agents in the low-level part are responsible for providing the QoS of each slice by assigning the resource blocks during that longer time step. To achieve this objective, every agent tries allocating available resource blocks in shorter time steps. In other words, by defining a proper action space low limit and high limit, the agent will apply the assignment so that all KPIs defining that slice are in the proper range. As an example, if a single agent has been assigned to provide FeMBB service for a user, the sequence of assignments should result in a system peak data rate of at least 1 $Tbps$ and a user-experienced data rate of 1 $Gbps$. In the evaluation section, we explain that proper maximum and minimum limits for action space can keep the assignments efficient. Each single agent receives its local reward since at this level agents act independently. The local reward starts from zero and the highest amount is when all predefined KPIs are in their proper range. Meaning, the local reward starts from zero, and increases with every KPI being in the desired range.

Considering the complexity of assigning resource blocks in this architecture, the best way to address this difficulty is to use learning agents inspired by the psychology of human learning. DQL, as a model-free off-policy DRL algorithm that uses both Experience Replay and Network Cloning, has shown good sample efficiency and stable performance [21]. Since the DQL agent can collect information and train its policy in the background, the learned policy stored in the neural network can be easily transferred to the situations [43]. The benefits of DQL aside, our high-level algorithm is a Q-based algorithm with replay memory. Since the execution level is in a nested structure, having the same nature and procedure at both levels decreases the possibility of inconsistencies and improves the harmony of the general performance. DQL uses a discrete action space, which is improper for most real-world problems. To have a compatible method with our continuous action space in the low-level part, we use DQL with the help of Normalized Advantage Function (NAF DQL) [66]. NAF DQL is DQL compatible with continuous action space environments. While in regular DQNs, the output demonstrates all possible actions, and later, the highest value is chosen, in NAF, the neural network estimates the value function and the Advantage. Combining two streams produces the Q-*value*, and then the $argmax$ is taken. According to [66], NAF DQL outperforms DDPG in solving the majority of tasks. Having all the benefits of Q-learning and its superior performance in problems with continuous action space assures that NAF DQL is a compatible algorithm for the low-level part.

The selected algorithms in both levels benefit from the experience replay mechanism, which refers to the case where

experiences are stored in replay memories. At the high-level part, *Q.tot-values* in the memory help agents select the best service types based on previous similar states and observations. At the low-level part, each experience in the memory helps agents to know the best sequence of assigning actions to realize that service type. Although complexity and scalability were mentioned throughout the text, the next section provides a general view of how practical the proposed approach is.

D. Practicality, Complexity, and Scalability

Even though complexity is an inseparable part of intelligent networks, specific considerations can improve the system's complexity. Task distribution, deployment of different levels, and proper selection of algorithms are able to reduce the overall complexity effectively. Decision-making is a task that requires the agents to know all the available resources in the system to be synchronized and consistent. On the other hand, after guaranteeing the availability of required resources, the low-level agents can perform their tasks without any extra information transferring, which will result in unnecessary overload for the network. The proposed scheme prevents excessive data transmission and overload by choosing the right place to implement each level. The only communication between two levels is through A1 interface, which is not noticeable.

In addition to that, synchronization and consistency needed for the high-level require a cooperative MARL set, which generally is not scalable and practical. However, choosing the QMIX as the MARL algorithm not only helps to decrease the severity of the complex nature of the problem but also ensures the scalability of the scheme in case of increasing the number of agents. This is because by adding more agents, the action space does not increase exponentially. This means the proposed scheme can handle more requests and users by adding more agents or service types. However, this approach needs one neural network for each agent in both levels in addition to one mixing network for the high-level implementation. The complexity and time required to train are significantly relative to the network presented by the environment. The off-policy DQL algorithm with efficient sample numbers in the low-level part also helps the scheme to stay practical.

IV. Evaluation and Results

In this section, we evaluate the proposed solution through numerical simulations. These simulations use Python 3.9 with PyTorch and PyTorch Lightening libraries. Py-Charm IDE, the Anaconda platform, and Google Colab have been used for coding. Since each level of the proposed scheme has access to different representations of the network, two different environments have been developed with the help of OpenAI gym. The environment for the high-level part interacts with the QMIX algorithm, and the environment for the low-level part interacts with single agents using NAF DQL [63], [67]. This simulation does not implement all structures of the O-RAN. We consider O-RAN as an enabling technology for 6G networks to be able to use our RL-based approach in the RICs. instead, each environment simulates the data that algorithms receive from interacting with an envisioned 6G network. Having CF mMIMO gives the environment RUs and DUs that can be assigned to multiple users as long as they do not share one resource block with multiple users. In the implementation of our proposed approach, we refer to the high-level episodes and their time steps as longer episodes and longer time steps, respectively. The same rule applies to the shorter episodes and time steps in the low-level part. It is worth mentioning that this simulation is one implementation of the general idea proposed in this paper. The environments and algorithms can be changed or improved.

In our simulation, in the high-level part, as mentioned in Section III, the state space consists of the general information of the network status, which are the service types as a group of KPIs mentioned in Table I. At the same time, the observation of each agent includes a request. Without loss of generality, each request asks only for one KPI in our simulation. So sates and observations are represented as:

$$state\ space = \{Service\ types\} \tag{2}$$

$$observation\ space = \{r_i, (Service\ types)\} \tag{3}$$

in which $Service\ types$ are sets of KPIs and r_i is also one KPI. Action is choosing one of the predefined service types, which is a discrete number between 0 and 4. The next state is all the KPIs affected by removing the chosen service type and its related resources. Since the reward system was discussed in Section III, we avoid it here and follow this section with the last remaining element of the 4-tuple Markov Decision Process $P_a(S, S')$, which is the probability that action a in state S at time t will lead to state S' at time $t + 1$. Similar to most reinforcement learning cases, it is challenging to represent the transition probability distributions; instead, an episodic simulator can be used. As a result, both levels of this simulation benefit from episodic environment simulators that can be started from an initial state and yield a subsequent state and reward every time they receive an action input.

There exist five agents in our simulation; each one's network produces a *Q-value* regarding the slice type it chooses for the request received in its observation. According to the QMIX algorithm, the *Q-values* will be fed to the mixing network. The weights and biases of the mixing network are produced by hyper-networks, which use the state and generate the layers. Since the weights should be non-negative, the leaner layer is followed by an absolute activation function. The final bias is also followed by a $ReLU$ non-linearity. The mixing network and each agent's network have been created according to [63].

After running a set of random trials, we chose the set of values 0.00001, 0.90, and 0.70 for $learning\ rate$, $gamma$, and $epsilon$ which gave us the best training result. In the training phase, every episode consists of 4 cycles, each ending whenever all the resources are in use, and no free resource is available. The stored experience at this level consists of current and next observations and states, in addition to actions and rewards. Fig. 3 illustrates the loss and reward plots of the multi-agent set in a 7500-episode run. The loss plot converged after around 5000 episodes, though. The descending loss plot and ascending reward plot show that our ML algorithm works properly in the simulated

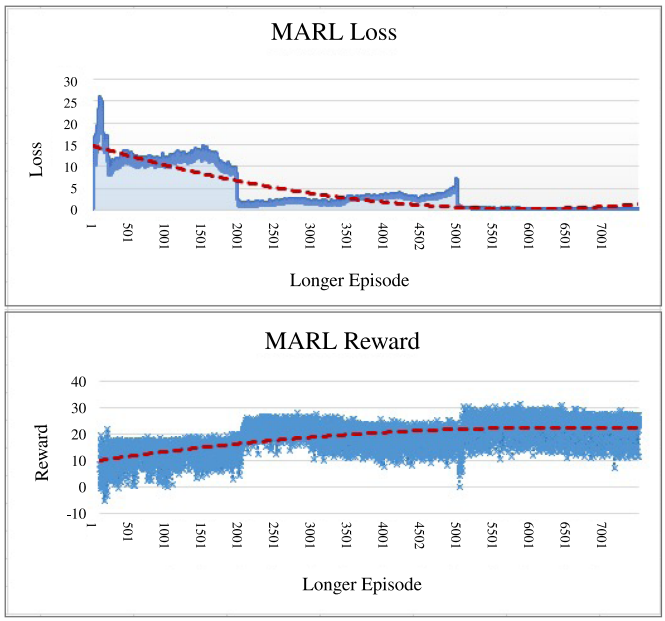

Fig. 3. Loss and Reward at the high-level part.

environment. In addition, the reward plot's importance comes from the fact that this level's reward system presents both agent and team rewards. The general goal for the mixing network is to maximize the number of assignments, which means each assignment should be optimal so that more users are served. Thus, the ascending plot shows the assigned resources corresponding to the network capacity increase. As mentioned before, increasing the number of accepted assignments means increasing network capacity in this research work. It should be noted that in this simulation if the requests repeat asking for the same service type, which can be unavailable after a while, agents will receive their rewards for not assigning the wrong slice type. Still, the total reward may decrease since there are no free resources. This explains the multiple rises and falls of the reward in Fig. 3. The red linear trend-line has an increasing pattern, though.

While the MARL algorithm at the high-level part selects the slice types in longer time steps, at the low-level part, the environment provides information related to resources that can be assigned. The state and observation spaces are the same at the beginning of each episode and present continuous values representing accessible resource blocks. Each agent has access to a limited number of resource blocks. The resources accessed by each agent are the ones located closer to the user. Contrary to the first level, action space is continuous and consists of eight (equal to the number of KPIs) positive and negative numbers. each array of actions changes the state values until each value is in the proper range for the service type. Similar to the first environment, at this level, the environment will produce the next state according to the input action and its effects on the available resources. Using the proper action space limits after multiple trials, and checking constraints in the reward system assign the optimum amount of resources to each request. In other words, since the high-level part monitors the availability and makes decisions, and the execution is decentralized by single agents at

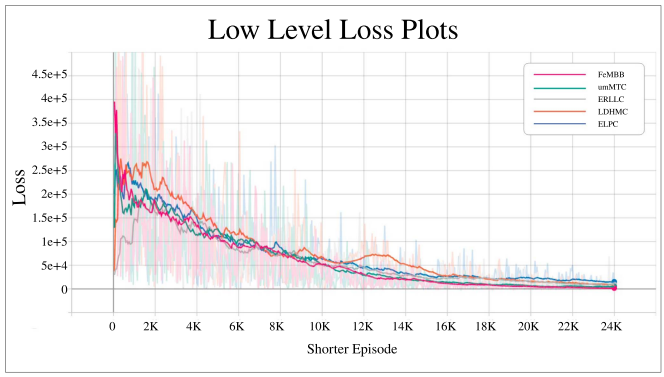

Fig. 4. Losses at low-level part.

the low-level part, the agents will not just guarantee a lower limit but also an upper limit. Thus, as defined in the reward system of the low-level part environment, for an agent to receive its reward, the KPIs should be between a maximum and a minimum level. In our simulation, all single agents are the same, but each one is trained to realised one of service types.

In the training phase, each episode consists of a maximum of 4 steps. At the beginning of the training process, in the trade-off between exploration and exploitation, each agent mostly uses random actions to explore and sample more data. Thus, the value of epsilon starts from 1 and reduces over passing epochs until the agent mostly uses *Q-values* to take actions from replay memory. Each experience in the replay memory consists of the current and next observation, action, and reward. In order to do that, we reduce the initial random sample probability to over 100 epochs. In other words, in the $training - epoch - end$ method of PyTorch Lightening, epsilon will be chosen according to the:

$$\max\left\{epsilonMin, epsilonStart - \frac{currentEpoch}{100}\right\} \quad (4)$$

Based on (4), the random sampling probability will start at its highest value and decrease during 100 epochs until it reaches the minimum value considered for epsilon. We used the Optuna library along with PyTorch Lightening to perform 20 trials to find the best value for $learning\ rate$ (0.00015193) and $gamma$ (0.011332).

Using the aforementioned values of $learning\ rate$ and $gamma$, Fig. 4 shows the Loss plots for all five single agents in 24000 epochs, each learning to realize one slice type. At the beginning of training, the plots for each service type increase due to taking random actions. However, as the training continues, after approximately 2000 epochs, the loss for each agent starts a descending pattern, and finally, after 6000 episodes, all plots converge to the minimum amount.

By jointly considering Figs. 3 and 4, we notice that both levels' training processes have succeeded. However, since our proposed ML techniques works in a network environment, we need to ensure that the agents can manage and allocate network resources in the system. Due to the fact that there are no similar research works in the literature to use as the baseline and compare the results, we used the trained model in the same interactive environment to test some KPIs and their value. This

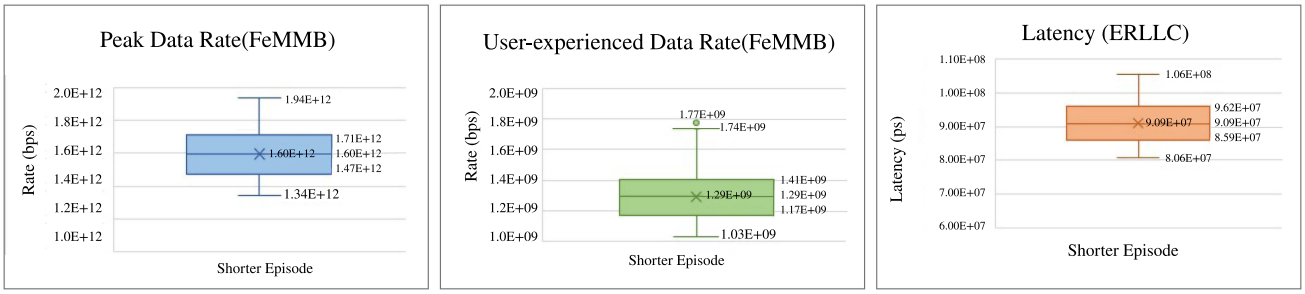

Fig. 5. Network KPIs: (a) Peak data rate in FeMMB. (b) User-experienced data rate in FeMMB. (c) Latency in ERLLC.

test monitors three of the network KPIs in 1000 episodes to assure that KPIs are kept in the predefined range by the agents. If so, the purpose of this work has been fulfilled. As the system level and user-experienced data rates are the two critical KPIs in the FeMMB service type, we chose these two KPIs to monitor in the environment using the trained agents. In addition, latency was observed as a critical KPI in the ERLLC service type. According to Fig. 5, the system peak data rate has been kept at more than 1 $Tbps$ in all 1000 episodes. Based on the user-experienced data rate, a minimum of 1 $Gbps$ data rate has been provided. The uniform distribution without having any sparse large value among the numerical results show that the upper limitations to control unnecessary assignments in the decentralized part have guaranteed the optimal assignments. Finally, Fig. 5 shows that latency in the ERLLC service type is less than 10 μs in all episodes, confirming the success of both levels' agents' performance. Having the desired value for the network KPIs, as shown in Fig. 5, affirms that agents in both levels perform the expected duties, and the overall solution works smoothly and can be a promising technique for the 6G complex system model.

V. Conclusion and Future Steps

The intelligent information society and emerging applications in the early future demand the next generation of wireless networks to overcome current limitations and provide different quality and service levels. New technologies that introduce openness, flexibility, and intelligence to the network are needed to address these demands. This paper first details candidate technologies for 6G system models such as O-RAN and then proposes a two-level DRL-based network slicing and resource assignment, which is compatible with these new system models and aims to maximize the network capacity while providing QoS of each service type. The proposed scheme benefits from imitating human teamwork and offers the flexibility and intelligence of agents that can observe, learn, and take actions online. Having two levels of performance and time not only facilitates optimal assignments but also helps the complex nature of the problem to become practical and scalable. Moreover, to study the performance of the scheme, Deep ML algorithms are proposed for the 6G environment at both levels and implementation details are discussed. The choice of algorithms at each level has been based on compatibility and complexity concerns. Since this work has considered a new system model, the compatible environment was simulated using the OpenAI Gym library; the general idea of managing the slices and assigning resources can be used with other proper algorithms and network environments.

Even though one of the objectives of this approach is to address the complex problem of network slicing and resource management in a system model consisting of CF in RAN, in this research work, low-level agents only try to provide the required KPIs by assigning resource blocks. On the other hand, KPIs such as mobility and latency are affected significantly by clustering strategies for communication links and physical layer-related resource management. The lower part can be improved to consider both communication links and resources, in addition to creating clusters for users in the CF RAN. Moreover, in our current simulations, each agent is trained for one service type. Having agents that can be trained for all types can robust the network management. Adopting other algorithms to each level and simulating different environments may result in exciting results.

References

[1] Z. Chen et al., "Terahertz wireless communications for 2030 and beyond: A cutting-edge frontier," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 66–72, Nov. 2021.
[2] J. R. Bhat and S. A. Alqahtani, "6G ecosystem: Current status and future perspective," *IEEE Access*, vol. 9, pp. 43134–43167, 2021.
[3] C. De Alwis et al., "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.
[4] K. David and H. Berndt, "6G vision and requirements: Is there any need for beyond 5G?," *IEEE Trans. Veh. Technol.*, vol. 13, no. 3, pp. 72–80, Sep. 2018.
[5] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.
[6] C. Christophorou et al., "Adroit6G DAI-driven open and programmable architecture for 6G networks," in *Proc. IEEE Globecom Workshops*, 2023, pp. 744–750.
[7] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2021.
[8] M. Noor-A-Rahim et al., "6G for vehicle-to-everything (V2X) communications: Enabling technologies, challenges, and opportunities," *Proc. IEEE*, vol. 110, no. 6, pp. 712–734, Jun. 2022.
[9] Z. Zhang et al., "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, Sep. 2019.
[10] N. M. Nasir et al., "Evolution towards 6G intelligent wireless networks: The motivations and challenges on the enabling technologies," in *Proc. IEEE 19th Student Conf. Res. Develop.*, 2021, pp. 305–310.

[11] M. Maule, J. Vardakas, and C. Verikoukis, "5G RAN slicing: Dynamic single tenant radio resource orchestration for eMBB traffic within a multi-slice scenario," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 110–116, Mar. 2021.

[12] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2019.

[13] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, 2020.

[14] J. Mei, X. Wang, and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," *IEEE Intell. Converged Netw.*, vol. 1, no. 3, pp. 281–294, Dec. 2020.

[15] W. Guan, H. Zhang, and V. C. M. Leung, "Customized slicing for 6G: Enforcing artificial intelligence on resource management," *IEEE Netw.*, vol. 35, no. 5, pp. 264–271, Sep./Oct. 2021.

[16] M. Rasti et al., "Evolution toward 6G wireless networks: A resource management perspective," 2021, *arXiv:2108.06527*.

[17] H. Cao et al., "Toward tailored resource allocation of slices in 6G networks with softwarization and virtualization," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6623–6637, May 2022.

[18] W. Wu et al., "AI-native network slicing for 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.

[19] Z. Shi, X. Xie, S. Garg, H. Lu, H. Yang, and Z. Xiong, "Deep reinforcement learning based Big Data resource management for 5G/6G communications," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[20] S. Kukliński, L. Tomaszewski, R. Kołakowski, and P. Chemouil, "6G-LEGO: A framework for 6G network slices," *IEEE J. Commun. Netw.*, vol. 23, no. 6, pp. 442–453, Dec. 2021.

[21] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6063–6078, Sep. 2021.

[22] F. Rezazadeh, H. Chergui, L. Blanco, L. Alonso, and C. Verikoukis, "A collaborative statistical actor-critic learning approach for 6G network slicing control," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[23] T. K. Rodrigues and N. Kato, "Network slicing with centralized and distributed reinforcement learning for combined satellite/ground networks in a 6G environment," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 104–110, Feb. 2022.

[24] T. Dong et al., "Intelligent joint network slicing and routing via GCN-powered multi-task deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1269–1286, Jun. 2022.

[25] Z. Sasan and S. Khorsandi, "Slice-aware resource calendaring in cloud-based radio access networks," in *Proc. 30th ICEE*, Tehran, Islamic Republic of Iran, 2022.

[26] H. Chergui, A. Ksentini, L. Blanco, and C. Verikoukis, "Toward zero-touch management and orchestration of massive deployment of network slices in 6G," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 86–93, Feb. 2022.

[27] P. Bhattacharya et al., "A deep-Q learning scheme for secure spectrum allocation and resource management in 6G environment," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4989–5005, Dec. 2022.

[28] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation O-RAN resource slicing for URLLC services using deep reinforcement learning," *IEEE Commun. Standards Mag.*, vol. 7, no. 1, pp. 66–73, Mar. 2023.

[29] Y. Cao et al., "From oran to cell-free ran: Architecture, performance analysis, testbeds and trials," 2023, *arXiv:2301.12804*.

[30] R. Ou, G. Sun, D. Ayepah-Mensah, G. O. Boateng, and G. Liu, "Two-tier resource allocation for multi-tenant network slicing: A federated deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20174–20187, Nov. 2023.

[31] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[32] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. Emad Ul Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021.

[33] A. Moubayed et al., "On end-to-end intelligent automation of 6G networks," *Future Internet*, vol. 14, no. 6, 2022, Art. no. 165.

[34] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.

[35] P. Wei et al., "Reinforcement learning-empowered mobile edge computing for 6G edge intelligence," *IEEE Access*, vol. 10, pp. 65156–65192, 2022.

[36] A. Mekrache et al., "Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G," *Veh. Commun.*, vol. 33, 2021, Art. no. 100398.

[37] R. Ali, I. Ashraf, A. K. Bashir, and Y. B. Zikria, "Reinforcement-learning-enabled massive Internet of Things for 6G wireless communications," *IEEE Commun. Standsrds Mag.*, vol. 5, no. 2, pp. 126–131, Jun. 2021.

[38] A. Salh et al., "A survey on deep learning for ultra-reliable and low-latency communications challenges on 6G wireless systems," *IEEE Access*, vol. 9, pp. 55098–55131, 2021.

[39] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, 2022.

[40] X. Du et al., "Multi-agent reinforcement learning for dynamic resource management in 6G in-X subnetworks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 3, pp. 1900–1914, Mar. 2023.

[41] F. Tang, B. Mao, Y. Kawamoto, and N. Kato, "Survey on machine learning for intelligent end-to-end communication toward 6G: From network access, routing to traffic control and streaming adaption," *IEEE Commun. Surv. Tuts.*, vol. 23, no. 3, pp. 1578–1598, Thirdquarter, 2021.

[42] F. Debbabi, R. Jmal, L. Chaari, R. L. Aguiar, R. Gnichi, and S. Taleb, "Overview of AI-based algorithms for network slicing resource management in B5G and 6G," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. Conf.*, 2022, pp. 330–335.

[43] R. Li et al., "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[44] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.

[45] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, "Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 68183–68198, 2020.

[46] B. Khodapanah, A. Awada, I. Viering, A. N. Barreto, M. Simsek, and G. Fettweis, "Slice management in radio access network via deep reinforcement learning," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–6.

[47] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[48] "O-RAN ALLIANCE," Accessed: Jun. 24, 2024. [Online]. Available: https://www.o-ran.org/

[49] A. Chaoub et al., "Self-organizing networks in the 6G era: State-of-the-art, opportunities, challenges, and future trends," 2021, *arXiv:2112.09769*.

[50] C. Yeh et al., "Perspectives on 6G wireless communications," *ICT Express*, vol. 9, no. 1, pp. 82–91, 2022.

[51] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network automation through orchestrated intelligence in the open RAN," in *Proc. IEEE Infocom*, 2022, pp. 270–279.

[52] M. Dryjański et al., "Toward modular and flexible open RAN implementations in 6G networks: Traffic steering use case and o-RAN xApps," *Sensors*, vol. 21, no. 24, 2021, Art. no. 8173.

[53] K. Ramezanpour et al., "Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of o-RAN," *Comp. Netw.*, vol. 17, 2022, Art. no. 109358.

[54] T. Karamplias et al., "Towards closed-loop automation in 5G open RAN: Coupling an open-source simulator with xApps," in *Proc. IEEE Joint Eur. Conf. Netw. Commun. & 6G Summit*, 2022, pp. 232–237.

[55] N. Kazemifard et al., "Minimum delay function placement and resource allocation for open RAN (o-RAN) 5G networks," *Comp. Netw.*, vol. 188, 2021, Art. no. 107809.

[56] M. Matthaiou, O. Yurduseven, H. Q. Ngo, D. Morales-Jimenez, S. L. Cotton, and V. F. Fusco, "The road to 6G: Ten physical layer challenges for communications engineers," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 64–69, Jan. 2021.

[57] S. Chen et al., "Wireless powered IoE for 6G: Massive access meets scalable cell-free massive MIMO," *IEEE China Commun.*, vol. 17, no. 12, pp. 92–109, Dec. 2020.

[58] H. I. Obakhena et al., "Application of cell-free massive MIMO in 5G and beyond 5G wireless networks: A survey," *J. Eng. Appl. Sci.*, vol. 68, no. 1, 2021, Art. .no. 13.

[59] R. Chataut and R. Akl, "Massive MIMO systems for 5G and beyond networks—overview, recent trends, challenges, and future research direction," *Sensors*, vol. 20, no. 10, 2020, Art. no. 2753.

[60] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99878–99888, 2019.
[61] J. S. Vardakas et al., "Towards machine-learning-based 5G and beyond intelligent networks: The MARSAL project vision," in *Proc. IEEE Int. Mediterranean Conf. Commun. Netw.*, 2021, pp. 488–493.
[62] J. S. Vardakas et al., "Machine learning-based cell-free support in the o-ran architecture: An innovative converged optical-wireless solution toward 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 20–26, pp. 20–26, Oct. 2022.
[63] T. Rashid et al., "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1–51.
[64] J. Yang, I. Borovikov, and H. Zha, "Hierarchical cooperative multi-agent reinforcement learning with skill discovery," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2020.
[65] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. 17th AAMAS*, 2018.
[66] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2829–2838.
[67] J. K. Terry et al., "Pettingzoo: Gym for multi-agent reinforcement learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 15032–15043, 2021.

Navideh Ghafouri received the B.Sc. degree in computer science and the M.Sc. degree in computer engineering with the main focus on networking from University of Esfahan, Isfahan, Iran. She is currently working toward the Ph.D. degree with the Department of Signal Theory and Communications, Polytechnic University of Catalunya (UPC), Barcelona, Spain. In 2022, she joined Iquadrat and is participating in research projects as a junior Researcher. She is currently working on network management and orchestration, slicing, and resource allocation for future wireless networks by deploying reinforcement learning techniques.

John S. Vardakas (Senior Member, IEEE) received the Dipl.-Eng. degree in electrical computer engineering from the Democritus University of Thrace, Komotini, Greece, in 2004, and the Ph.D. degree from the Electrical Computer Engineering Department, University of Patras, Patras, Greece, in 2012. He has authored or coauthored more than 45 journal articles and 90 conference articles, while he has participated in more than 20 competitive research programs, having served as a PC, and as a TM. His research interests include teletraffic engineering, performance analysis, and simulation of communication networks and smart grids.

Kostas Ramantas received the Diploma in computer engineering, the M.Sc. degree in computer science, and the Ph.D. degree from the University of Patras, Patras, Greece, in 2006, 2008, and 2012, respectively. In 2013, he joined IQUADRAT as a senior Researcher and has co-supervised many Ph.Ds. in the framework of RISE projects (e.g., CASPER, WATER4CITIES) and ITN projects (e.g., Spotlight, 5GAura). He was involved in multiple E.C. funded projects (e.g. 5GMediaHUB), having served as a PC, and as a TM. He has authored or coauthored more than 35 journal and conference papers. His research interests include modelling and simulation of network protocols, and scheduling algorithms for QoS provisioning. He was the recipient of two national scholarships.

Christos Verikoukis (Senior Member, IEEE) received the Ph.D. degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2000. He is currently an Associate Professor with the University of Patras, Patras, Greece. He has authored or coauthored 165 journal papers and more than 200 conference papers. He is also a co-author of three books, 14 chapters in other books, and two patents. He has participated in more than 40 competitive projects, and was a project coordinator of several funded projects from the European Commission and of national projects in Greece and Spain. He is currently the IEEE ComSoc GITC Member and the Editor-in-Chief of the IEEE NETWORKING LETTERS.