Multi-Agent Reinforcement Learning for Network Selection and Resource Allocation in Heterogeneous Multi-RAT Networks

Mhd Saria Allahham, *Student Member, IEEE*, Alaa Awad Abdellatif, *Member, IEEE*, Naram Mhaisen, Amr Mohamed, *Senior Member, IEEE*, Aiman Erbad, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—The rapid production of mobile devices along with the wireless applications boom is continuing to evolve daily. This motivates the exploitation of wireless spectrum using multiple Radio Access Technologies (multi-RAT) and developing innovative network selection techniques to cope with such intensive demand while improving Quality of Service (QoS). Thus, we propose a distributed framework for dynamic network selection at the edge level, and resource allocation at the Radio Access Network (RAN) level, while taking into consideration diverse applications' characteristics. In particular, our framework employs a deep Multi-Agent Reinforcement Learning (DMARL) algorithm, that aims to maximize the edge nodes' quality of experience while extending the battery lifetime of the nodes and leveraging adaptive compression schemes. Indeed, our framework enables data transfer from the network's edge nodes, with multi-RAT capabilities, to the cloud in a cost and energy-efficient manner, while maintaining QoS requirements of different supported applications. Our results depict that our solution outperforms state-of-the-art techniques of network selection in terms of energy consumption, latency, and cost.**

***Index Terms*—Heterogeneous networks, edge computing, wireless healthcare systems, multi-RAT architecture, deep reinforcement learning.**

Manuscript received August 14, 2021; revised January 4, 2022; accepted February 20, 2022. Date of publication March 2, 2022; date of current version June 9, 2022. This work was made possible by NPRP grant # NPRP12S-0305-190231 from the Qatar National Research Fund (a member of Qatar Foundation). The work of Amr Mohamed and Aiman Erbad was partially supported by NPRP grant # NPRP13S-0205-200265. The findings achieved herein are solely the responsibility of the authors. The associate editor coordinating the review of this article and approving it for publication was M. Chen. *(Corresponding author: Amr Mohamed.)*

Mhd Saria Allahham was with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar. He is now with the School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: ma1517219@qu.edu.qa).

Alaa Awad Abdellatif and Amr Mohamed are with the College of Engineering, Qatar University, Doha, Qatar (e-mail: alaa.abdellatif@ieee.org; amrm@qu.edu.qa).

Naram Mhaisen was with Qatar University, Doha, Qatar. He is now with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: nm1300940@qu.edu.qa).

Aiman Erbad is with the Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Ar-Rayyan, Qatar (e-mail: aerbad@hbku.edu.qa).

Mohsen Guizani is with the Machine Learning Department, MBZUAI, Abu Dhabi, UAE (e-mail: mguizani@gmail.com).

Digital Object Identifier 10.1109/TCCN.2022.3155727

I. Introduction

With the emerging of the Internet of Mobile Things (IoMT) and unprecedented 5G applications, several strict communication requirements have arisen. Such requirements demand wireless networks to be responsive while adapting to different applications' requirements. Specifically, for e-health applications, the ability of the healthcare systems to predict and instantly react to emergency situations is mandatory to reduce the risks of chronic diseases and the mortality rate [1]. However, the strict latency requirements for emergency conditions along with the enormous amount of generated data are still major challenges for e-health systems. Such demand for ultra-low latency and other Quality of Service (QoS) requirements has motivated us to leverage the evolution of the 5G network toward dense Heterogeneous networks (HetNets) [2]. 5G HetNets are expected to enhance users' QoS, enable diverse performance improvements, and fulfill various service requirements through increasing the opportunity of spatial resource reuse [3], [4]. Indeed, leveraging multi-Radio Access Technology (RAT) will enable a device/edge (e.g., mobile phones, edge gateways) to utilize the available radio resources across various spectral bands to connect with the network infrastructure. Hence, the edge devices that are equipped with multiple interfaces (e.g., Wi-Fi, 3G, 4G, Bluetooth) will be able to simultaneously access the available networks with different RATs. However, this calls for designing innovative and scalable networks selection schemes that consider e-health QoS requirements while maintaining high spectrum efficiency across diverse networks.

Artificial Intelligence (AI) has been a key feature in 5G networks recently, where introducing AI into HetNets can help in developing and executing efficient and intelligent network selection schemes [5]. Moreover, in AI-based user-centric network selection, different users can have customized and dynamic selection behaviors based on their own needs and requirements for different applications [6]. Specifically, the most efficient dynamic network selection techniques are based on Reinforcement Learning (RL). RL is an emerging field in AI that studies optimal sequential decision making for an agent in non-deterministic environments [7]. In single-agent RL, the agent tries to learn the optimal policy from the interactions with the environment, aiming to maximize its

reward. Nevertheless, in many environments of the real world, including self-driving cars, packet delivery, and others, there are multiple agents acting simultaneously on the environment, based on their own observation, with the aim of maximizing their own utility, which does not necessarily align with others. Single-agent RL modeling of such large-scale environments is not efficient, where the agent if deployed in a centralized manner (i.e., at the core network) it will lack the scalability, since it has to have complete knowledge about the environment including users' actions.

As such, the modeling framework that captures such complex interactions between multi-agents is game theory. However, due to the difficulty of classical solution methods for games, especially for environments with many agents, data-driven solutions based on reinforcement learning have been emerging as a promising solution known as Multi-Agent RL (MARL). MARL extends the RL framework to explicitly model the existence of multiple agents and the effect of their joint action on the environment. In this framework, the solution concept is to reach a set of policies (a policy for each agent) that form an equilibrium with the maximum reward (i.e., set of dominant policies). Of course, reaching this equilibrium, if it exists, is more challenging compared to single-agent RL. This is primarily due to the non-stationarity nature of the environment (i.e., for an agent, the same policy might result in a different performance according to what other agents are doing, which is not necessarily known by that agent).

MARL includes different sub-problems ranging from learning the cooperation between agents [8], [9], to learning the communication between different types of agents [10]. The recent advancements in deep learning along with its integration with the MARL lead to the new Deep Multi-agent Reinforcement Learning (DMARL) concept [11], [12], where DMARL has been widely utilized for various tasks in 5G networks such as distributed resource allocation [13], [14] and interference management [15]. Moreover, in edge computing, DMARL is being employed for numerous tasks such as computation offloading [16], resource allocation and cashing [17], [18] and control of network traffic [19]. Overall, there has been impressive progress at the algorithmic level in the DMARL community, which is yet to be customized, adapted, and utilized in practical scenarios. In this work, we utilize and adapt state-of-the-art DMARL algorithms to jointly tackle a practical and important issue of network selection and resource allocation.

II. Related Work

The related work in the literature exploits different methodologies for solving the network selection problem, including: game theory, Markov decision processes (MDPs), multi-attribute decision making, and optimization techniques [20]. Game theory approaches for network selection assume that the users aims to increase their rewards, while networks operators are trying to increase their revenues by increasing their number of users. However, it cannot be always guaranteed that the networks operators will act in a rational manner and the users will get the targeted QoS [21], [22]. Moreover, the proposed solutions, using game theory, are typically complexity-prohibitive, and their convergence to the optimal solution is not guaranteed. Even if they converge, it is not always guaranteed that they can converge to an optimal solution [20]. In other works, the network selection problem has been formulated as an MDP to investigate network switching between different RATs [23], [24]. However, obtaining an optimal solution using such approaches is again computationally intensive, especially in the case of large networks [25].

Network selection has been also studied using multi-attribute decision making. Such approaches assigned different weights for different factors affecting the network selection decision. Then, various techniques such as simple additive weighting [26], multiplicative exponent weighting [27], [28], and grey relational analysis [29], [30], have been considered for selecting the appropriate network. However, it is usually hard to prove that such approaches can obtain an optimal solution. Optimization techniques have been also investigated for solving network selection problem. Despite the guaranty of optimality using such approaches, typically, formulating the network selection problem as an optimization problem with reasonable complexity (to be run at the edge) is not an easy task. Obtaining the optimal solution, subject to different networks, applications, and power constraints may lead to an NP-hard problem [31]. Moreover, since network resources and characteristics can vary with time, as well as the edge resources, classical optimization approaches can be computationally expensive, where online optimization approaches such RL can be more appropriate.

RL-driven network selection has been used and studied in [32], [33], [34], where the authors showed that RL can achieve faster convergence and optimal behaviors. However, the existence of multiple users that employ network selection schemes along with resource-constrained RANs calls for the need for MARL and distributed optimization to decentralize the individual policies for individual users and RANs, which would scale efficiently in large scale systems. The authors in [35] have proposed a strategy for multi-RAT access based on MARL with the aim of maximizing the average system throughput while satisfying the users' QoS preferences. However, the authors did not address the mobile devices characteristics (e.g., energy budget) and the application requirements (e.g., the quality of the transmitted medical data).

Different from the aforementioned work in the literature, we aim to leverage the intelligence at the edge for optimizing networks selection decisions in the ultra-dense heterogeneous networks, and at the network side for optimizing the resource allocation for the edges. Specifically, we propose an approach that explicitly models the existence of two interacting groups of heterogeneous agents, to simultaneously learn and optimize the overall system performance. Namely, a group of autonomous end-users that aim to perform network selection and a group of autonomous RANs that seek to solve the problem of resource allocation. Note that this contrasts the previous studies cited in the paper as those focus on optimizing the task of one group only, while considering the other

as static respondents or assume a centralized control that optimizes the performance of the two groups. Our paper is the first to explicitly model the adaptive and decentralized behavior of these two groups, where each one is learning simultaneously and independently from the others. Moreover, the proposed framework is supported by a practical and essential application in smart health systems, which are perfect candidates to leverage the advantages brought by the framework. The entities of these systems dynamically change, e.g., the patients' health state, the wireless channel characteristics, and dynamics, the battery level of the mobile device$\cdots$ etc. Our main contributions can be summarized as follows.

1) We formulate a multi-objective optimization problem that describes the whole system and aims at obtaining, for each user/patient edge node (PEN): (i) the optimal data compression ratio, (ii) the selected Radio Access Networks (RANs) for data transmission, and (iii) the allocated bandwidth at each selected RAN(s).
2) The problem is reformulated in terms of Multi-Agent theory, where the agents for each interacting group in the system (i.e., RANs and PENs) will be defined along with their observations, actions and their reward functions.
3) We propose a distributed DMARL algorithm, namely, Team-Based Multi-Agent Deep Deterministic Policy Gradient (TB-MADDPG), which handles the heterogeneity of the agents in the system, and looks for the optimal joint policy that maximizes the rewards for all agents.

The rest of the paper is organized as follows. Section III introduces the system model and the main Performance metrics considered in this study. Section IV presents the formulated optimization problem, along with the re-formulation in terms of multi-agent theory. In Section V, we present the proposed approach to solve the reformulated problem and learn the optimal policy. Section VI presents the performance evaluation of our approach and the comparisons against the state-of-the-art techniques, while Section VII presents the complexity analysis of the proposed method before concluding in Section VIII.

III. System Model and Performance Metrics

This section introduces the system model under study. Then, it presents the main application and network requirements that will be considered in the proposed framework.

A. System Model

In this work, we consider a mobile-health (m-health) network with ultra-dense heterogeneous network architecture, where multiple end-users can access multiple RANs as in Fig. 1. In the considered scenario, a combination of sensor nodes attached to the patients is used for monitoring the patient's health state (e.g., implantable or wearable sensors that measure various biosignals and vital signs). The acquired data from such sensors are forwarded to a patient edge node (PEN) that represents the data processing unit between the data sources and the RANs. Specifically, the PEN collects the medical data from various sources and applies an adaptive

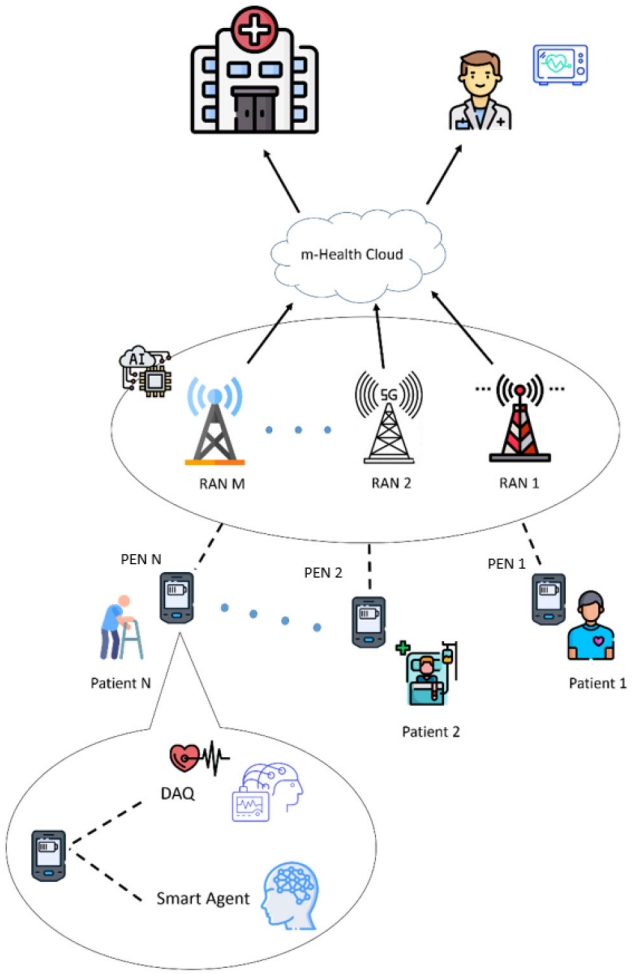

Fig. 1. System model under study.

compression scheme to control the data size, while considering the high-level application's requirements. Then, the PEN can forward the collected data to the m-health cloud (MHC) via the available RANs. Each RAN has different characteristics, such as data rate, energy consumption, monetary cost (i.e., requested payment for using network services), and transmission delay. Moreover, due to patient mobility, the level of quality of service offered by the available RANs may vary over time.

The PENs are considered to be battery-operated devices, where a PEN can be any mobile node (e.g., smartphone). However, it can collect an immense amount of data due to the continuous monitoring of diverse patient's conditions. Hence, it is important to transmit the collected data efficiently through the RANs without draining the PEN's battery, such that the PEN can be used for a long time without charging or replacement.

Lastly, we consider that the patient's conditions might change at any point in time in order to simulate a real-life scenario. More specifically, the patients in our system model can have abnormal conditions (such as seizure) for a period of time, and then their conditions can go back to normal after a certain time period. Such abnormal conditions influence how the vital signs should be processed locally at the edge on one

side, and what RANs will be selected to address the emergency case with ultra-low latency on the other side.

B. Performance Metrics

In the following, we assume on a time period *T*, each PEN *i* ($i = 1, \ldots, N$) has to transfer B_i bits of data towards the MHC, through *M* RANs. Moreover, each RAN *j* ($j = 1, \ldots, M$) has to allocate resources in terms of bandwidth to *N* PENs.

Adaptive Compression: The adaptive data compression at the PEN, is implemented using the discrete wavelet transform (DWT) compression scheme [36]. Hence, the generated data length, after compression, at PEN *i* is expressed as follows:

$$b_i = B_i(1 - \kappa_i) \tag{1}$$

where B_i being the length of the raw data sent by PEN *i* before compression, and κ_i is the compression ratio at that PEN. However, using lossy data compression, prior to the transmission, comes at the cost of introducing data distortion at the receiver side. In the proposed framework, we adopt the compression scheme in [37] for electroencephalogram (EEG) data compression, as an example of intensive medical data compression. Nevertheless, without loss of generality, the proposed framework can be easily extended to consider different compression schemes, medical or non-medical data (e.g., video) [38]. Using the obtained results in [37], the introduced distortion can be expressed as:

$$D_i = \frac{c_1 . \exp(1 - \kappa_i) + c_2 . (1 - \kappa_i)^{-c_3} + c_4 . F^{-c_5} - c_6}{100}, \tag{2}$$

where *F* is the wavelet filter length of the adopted DWT compression scheme, and $c_1 - c_6$ are the estimated parameters by the statistics of the EEG compression model.

Energy Consumption: First, the available data rate from RAN *j* to PEN *i* is given by:

$$r_{ij} = W_{ij} \log_2 \left(1 + \frac{P_t \, g_{ij}}{N_0 \, W_{ij}}\right) \tag{3}$$

where P_t is the PEN transmission power, W_{ij} is the allocated bandwidth, and is given by $W_{ij} = \theta_{ij} . W_j$, with θ_{ij} being the fraction of the RAN bandwidth W_j [39]. In (3), N_0 is the noise spectral density, while the channel gain g_{ij} is defined as:

$$g_{ij} = K \; . \; \sigma \; . \; \left|h_{ij}\right|^2 \tag{4}$$

where K = −1.5/(log(5BER)), σ is the path loss attenuation, and $|h_{ij}|$ is the fading channel magnitude for PEN *i* over RAN *j*. Then, the estimated energy consumption at PEN *i* to send b_i bits over RAN *j*, as defined in [40], is given by:

$$E_{ij} = \psi_j \, . \left(\frac{b_i N_0 \, W_{ij}}{r_{ij} \, g_{ij}} \left(2^{\frac{r_{ij}}{W_{ij}}} - 1\right)\right) + c_j, \tag{5}$$

where ψ_j and c_j are specific parameters that differ for each network interface [41].

Latency: The expected latency to send b_i bits from PEN *i* through RAN *j* can be defined as:

$$L_{ij} = \frac{b_i}{r_{ij}} + \xi_j \tag{6}$$

where ξ_j is the access channel delay over RAN *j*. More specifically, the expected latency represents the end-to-end delay when using a given technology [42]. Nonetheless, the data rate equation in (3) is generic and interference-free, and any other data rate model with interference can be simply integrated along with the data transmission energy consumption model.

Monetary Cost: the cost resulting from using RAN *j* by PEN *i* to send b_i bits is expressed in Euro's and can be defined as:

$$C_{ij} = b_i \varepsilon_j \tag{7}$$

where ε_j is the monetary cost per sent bit over RAN *j*. This monetary cost can be acquired through the use of, e.g., the IEEE 802.21 standard [43], which allows a user device to gather information about the available wireless networks. Such value can also be stored in the PENs in advance and updated if there are any changes in pricing.

IV. Problem Formulation

In this section, firstly, we formulate our problem for optimal Network selection and resource allocation as a multi-objective optimization problem under certain network constraints. Secondly, we motivate and introduce a necessary reformulation as a MARL framework by expressing the problem as a discrete time-variant system.

A. Multi-Objective Optimization Problem Formulation

The goal of the optimization problem is to minimize the transmission energy consumption, along with the data delivery latency and monetary cost, while meeting the medical data QoS requirements in terms of distortion. Therefore, we define an objective function that is a weighted sum aggregate of the aforementioned objectives.

Given *N* PENs with *M* available RANs, the objective of our optimization problem is to minimize the PENs' transmission energy consumption E_{ij}, monetary cost C_{ij}, latency L_{ij}, and distortion D_i. This can be achieved through: (i) assigning the PENs to the optimal RAN(s), (ii) obtaining the optimal bandwidth allocation over different RANs, and (iii) setting the PENs' optimal compression ratio. Thus, the proposed optimization problem is formulated as:

$$\mathbf{P1}: \min_{\boldsymbol{\theta}, \mathbf{P}, \boldsymbol{\kappa}} \; \mathbb{E}\left[\sum_i \sum_j P_{ij} \, U_{ij} + \delta_i D_i\right] \tag{8}$$

$$s.t \; \sum_i \theta_{ij} = 1, \quad \forall j \in M \tag{9}$$

$$\sum_j P_{ij} = 1, \quad \forall i \in N \tag{10}$$

$$\frac{b_i P_{ij}}{r_{ij}} \leq T_{ij}, \quad \forall i \in N, \forall j \in M \tag{11}$$

$$0 \leq \theta_{ij}, P_{ij} \leq 1, \quad \forall i \in N, \forall j \in M \tag{12}$$

$$0 \leq \kappa_i < 1, \quad \forall i \in N \tag{13}$$

where $U_{ij} = \alpha_i E_{ij} + \beta_i C_{ij} + \lambda_i L_{ij}$ is the utility function of PEN *i* over RAN *j*. The weighting coefficients α, β, λ, and δ represent the relative significance of the four metrics in the

problem; where $\alpha_i + \beta_i + \lambda_i + \delta_i = 1$. Moreover, we denote the patient status as ζ_i, where:

$$\mathbf{P}(\zeta_i = 1) = \upsilon_i, \quad \forall i \in N \tag{14}$$

where the equation conveys that a patient *i* can have a seizure at any point in time with a probability of υ_i. The weighting coefficients α, β, λ, and δ are functions of ζ_i, where they can have different values according to the patient status, which allows us to optimize the RAN(s) selection and PENs' parameters based on the patient status. Since the weighting coefficients are a function of the random variable ζ_i, and hence, the objective function is stochastic, we optimize the expected value of the objective function. Moreover, It is worth noting that the metrics are normalized between 0 and 1 in order to sum the unitless values in the objective function (8).

In (8), we consider a network utilization indicator P_{ij} that represents the fraction of data that should be transmitted through RAN *j* by PEN *i*. We assume that the PENs have all information to compute the energy consumption and cost. The constraint in (9) ensures the full utilization of RANs' bandwidth by the PENs, while the constraint in (10) ensures that all the data that each PEN has to transfer to the MHC is actually sent through the RANs. The network capacity constraint is represented by (11), where T_{ij} is the maximum fraction of the time period *T* that can be used by PEN *i* over RAN *j* (i.e., resource share). T_{ij} depends on the number of PENs accessing the RAN, and we assume that it is notified by the RANs.

The decision variables in this problems are the θ_{ij}'s, P_{ij}'s and the κ_i, i.e., each RAN needs to determine the allocated bandwidth for each PEN, and each PEN needs to determine its compression ratio and the amount of data it should transfer through different RANs. However, the problem **P1** in (8) in its current form is not convex [44], since the second derivatives matrix of the utility function, i.e., the Hessian matrix, is not positive semi-definite. Moreover, an approach like transforming the problem into a Geometric Program (GP) would not work in this case due to the existence of the non-linearity of the distortion in the objective. However, the problem can be solved in a distributed manner where it can be divided into sub-problems and the variables can be separated, which the authors in [45] have done. Nevertheless, the obtained solution does not take into accounts the time variance in the system, and solving the problem **P1** by classical optimization approaches as in the previous work is computationally expensive, since the system parameters change dynamically with time (e.g., channel gain of each RAN, patients' conditions, PENs battery status$\cdots$ etc.). Also, with changing the environment, the objective function has to be re-optimized again considering the new changes. Thus, in what follows, we opt to reformulating the problem, leveraging multi-agent systems theory, to be solved using deep MARL (DMARL)-based approach.

B. Multi-Agent Reinforcement Learning Formulation

As mentioned earlier, we want to solve the problem in an online manner, adapting to the changes in real-time. However, since the problem is non-convex, solving it repeatedly at every variation of the system parameters is inefficient. Hence, we opt for learning-based optimization such as reinforcement learning (RL) to achieve this adaptation. To solve the problem through RL, a Partially-Observable Markov Decision Process (POMDPs) representation of the system has to be designed. In fact, due to the existence of more than one independent PEN in the system along with multiple RANs calls for the need of the multi-agent extension of POMDPs. In our new formulation, we consider multiple heterogeneous agents interacting simultaneously with a partially observable environment *V* in discrete time steps. The formulation is to be described by a partially observable Markov game (POMG) [46], which is a multi-agent extension for POMDPs. POMGs are usually represented by the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{N}$ is the set of all agents, $s_t \in \mathcal{S}$ is a single possible configuration of all the agents at time *t*, $a_t \in \mathcal{A}$ is a possible action for the agents where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$, $o_t \in \mathcal{O}$ is a possible observation of the agents where $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2 \times \cdots \times \mathcal{O}_N$, $\mathcal{T}$ is the state transition probability $\mathcal{T} : \mathcal{O} \times \mathcal{A} \mapsto \mathcal{O}$, $\mathcal{R}$ is the set of rewards for all the agents $r : \mathcal{O} \times \mathcal{A} \mapsto \mathbb{R}$, and γ is the reward discount factor. In what follows, we define each set in the tuple in the context of our network selection and resource allocation problem.

1) System Agents and Environment: In this work, we consider two types of agents, namely, the PEN agents and the RAN agents. The PEN agents are deployed at the PENs and can control the data transmission and compression, while the RAN agents are deployed in the RAN controller in order to optimize the RANs' bandwidth allocation to the PENs. Each agent *i* receives from the environment an observation, and take actions according to the joint policy $\boldsymbol{\pi} : \mathcal{O} \mapsto \mathcal{A}$, where $\boldsymbol{\pi} : \pi_1 \times \pi_2 \times \cdots \times \pi_N$, and it transits the agent to its next state according to the state transition probability $\mathcal{T}$, and the agent receives an immediate reward r_i. The environment *V* will be episodic, i.e., it has finite horizon, and it terminates when all the PENs run out of power. During an episode, the total cumulative discounted reward for an agent *i* is denoted by $R_i = \sum_{t=0}^{t'} \gamma^t r_i^t$, where $\gamma \in [0, 1)$ and t' is the episode time horizon. The state-action value function of a joint policy $\boldsymbol{\pi}$ for an agent *i* is expressed as $Q_i^{\boldsymbol{\pi}}(s, a) = \mathbb{E}[R_i^t | s^t = s, a^t = a]$. The goal of the agents is to find an optimal joint policy $\boldsymbol{\pi}^*$ that yields the optimal Q-function $Q_i^{\boldsymbol{\pi}^*}(s, a) = \max_{\boldsymbol{\pi}} Q_i(s, a)$ for each agent through direct interactions with the environment, without having an explicit pre-knowledge about it (the transition probability $\mathcal{T}$).

2) Agents Observations and Actions: In our system, we consider two types of agents, namely, the RAN agent type and the PEN agent type. Each type of agents has different observations and actions, and each agent from each type has his own observations and actions. We assume that agents from the same type are independent from each other, and they do not share any observation or any kind of information. In fact, this is the case in real-world scenarios, where PENs as actors do not share any information with each other in order to preserve their privacy, and each RAN does not have any information about other RANs and how much data have been flowed through them. However, agents from different type have to share some information between each other, including whether PENs are

willing to send any data to a RAN or not, and how much bandwidth is each RAN allocating to each PEN. We denote the RAN agent *j* observation and action at time *t* by $o^t_{\rho_j}$ and $a^t_{\rho_j}$ respectively, and the PEN agent *i* observation and action at time *t* by $o^t_{\varphi_i}$ and $a^t_{\varphi_i}$ respectively. The RAN agent *j* observation at time *t* is the status of the PENs if they have sent any data or not to RAN *j*, and given by:

$$o^t_{\rho_j} = \left\{ n^t_{ij} \right\}, \quad \forall i \in N \tag{15}$$

where,

$$n^t_{ij} = \begin{cases} 1 & P^t_{ij} > 0 \\ 0 & P^t_{ij} = 0, \end{cases} \tag{16}$$

whereas the RAN agent action is how much bandwidth it allocates to each PEN, or the fraction of the RAN bandwidth that each PEN is getting at time *t*, and given by:

$$a^t_{\rho_j} = \left\{ \theta^t_{ij} \right\}, \quad \forall i \in N \tag{17}$$

As for a PEN agent *i*, its observation at time *t* consists of the PEN energy consumption, latency, monetary cost and the channel state for each RAN that the PEN has sent any data through. Moreover, the PEN agent observes the patient seizure status if it is active or not, along with the medical data distortion and the PEN battery level. The full PEN agent observation can be expressed as the following:

$$o^t_{\varphi_i} = \left\{ \bar{E}^t_{ij}, \bar{C}^t_{ij}, \bar{L}^t_{ij}, \bar{D}^t_i, \zeta^t_i, \bar{\Gamma}^t_i \right\}, \quad \forall j \in M \tag{18}$$

where $\bar{E}^t_{ij}, \bar{C}^t_{ij}, \bar{L}^t_{ij}, \bar{D}^t_i$ and $\bar{\Gamma}^t_i$ are the normalized energy consumption, monetary cost, latency, distortion and battery level respectively, by max normalization (i.e., they fall in the range of $[0-1]$). Moreover, $\bar{E}^t_{ij}$, $\bar{C}^t_{ij}$ and $\bar{L}^t_{ij}$ at time *t* can be evaluated by plugging P^{t-1}_{ij} and θ^{t-1}_{ij} in (5), (6) and (7) respectively, whereas $\bar{D}^t_i$ by plugging κ^{t-1}_i in (2), and finally, $\bar{\Gamma}^t_i$ is evaluated as follows:

$$\bar{\Gamma}^t_i = \frac{\Gamma^{t-1}_i - \sum_j E^t_{ij}}{\Gamma^0_i} \tag{19}$$

It is worth pointing out that the transitions of $\bar{E}^t_{ij}$, $\bar{L}^t_{ij}$ and $\bar{\Gamma}^t_i$ are stochastic, due to the fact that these observations mainly depend on the channel gain estimation, which introduces the randomness in the system. Also, the patient status is not deterministic, as the patient is prone to abnormal conditions (i.e., seizures) at any point in time.

Lastly, the PEN agent *i* actions at time *t* are the amount of data that the PEN is sending to each RAN, along with the compression ratio, and can be expressed as:

$$a^t_{\varphi_i} = \left\{ P^t_{ij}, \kappa_i \right\}, \quad \forall j \in M. \tag{20}$$

3) Agents Reward Functions: The reward functions have to be designed such that they describe the original optimization problem along with maximizing the PENs battery lifetime. The PEN agents aim is to minimize the energy consumption in order to guarantee a longer battery lifetime, while jointly minimizing the cost, latency and distortion. Therefore, the PEN agent reward function is defined as the following:

$$r^t_{\varphi_i} = \begin{cases} \left(1 - \sum_j P^t_{ij} \bar{U}^t_{ij}\right) + \delta_i \left(1 - \bar{D}^t_i\right) + \bar{\Gamma}^t_i & \zeta^t_i = 0 \\ \lambda_i \left(1 - \sum_j \bar{L}^t_{ij}\right) + \delta_i \left(0.1 - \bar{D}^t_i\right) + \bar{\Gamma}^t_i & \zeta^t_i = 1 \\ -1 & \text{violated constraints} \end{cases} \tag{21}$$

where $\bar{U}^t_{ij}$ is the normalized utility function (i.e., the sum of the normalized energy consumption, monetary cost and latency). The first and the second term represents the objective function (8), while the third term addressed the battery energy level, and it forces the agent to minimize the energy consumption in order to maintain a slow decrease in the reward over time, rather than a sharp decrease. However, when the patient has a seizure, the data has to be delivered with the minimum distortion and delay possible. Hence, the reward will disregard the cost factor, and care less about the energy consumption, and give more significance to the latency and distortion terms. Lastly, if the agent violates any constraints, it will receive a penalty of -1.

The RAN agents' main goal is to maximize the connected users QoE, which can be done by maximizing their rewards. Hence, the reward for the RAN agents is defined as the following:

$$r^t_{\rho_j} = \begin{cases} \frac{\sum_i n^t_{ij} r^t_{\varphi_i}}{\sum_i n^t_{ij}} & (9), (11) \text{ hold} \\ -1 & \text{otherwise} \end{cases} \tag{22}$$

where n^t_{ij} is the status of PEN *i* if it has sent any data to RAN *j* at time *t* and $r^t_{\varphi_i}$ is the reward of the PEN agent *i*. The aforementioned reward function represents the average rewards of the connected PENs to the RAN *j*. The agents at each RAN will try to maximize the rewards of its connected PENs by allocating the optimal bandwidth for each one, in order to guarantee the best QoE for the connected users.

V. Team-Based MADDPG Solution For Network Selection and Resource Allocation

MADDPG [47] is the multi-agent extension of the DDPG [48] algorithm, where each agent learns a policy based on its local observation only, and each agent has his own Q-function, which works as an indicator of how good is the agent policy. The MADDPG algorithm adopts the centralized training with decentralized execution framework. In the training phase, which is depicted in Fig. 2, each agent utilizes a Q-function (critic), and a policy function (actor), where the critics have access to the joint observations and actions (hence the name "centralized"), and guides the training of the actors. In the testing (execution) phase, shown in Fig. 3, the critic is discarded (i.e., the global observation/actions are not needed any more) and only the actor, which depends only on local observations, is utilized. The centralized training allows the policies to make use of additional information to ease and stabilize training, so long as this information is not used at execution time. This adopted framework in MADDPG leads to learned joint optimal policies that only

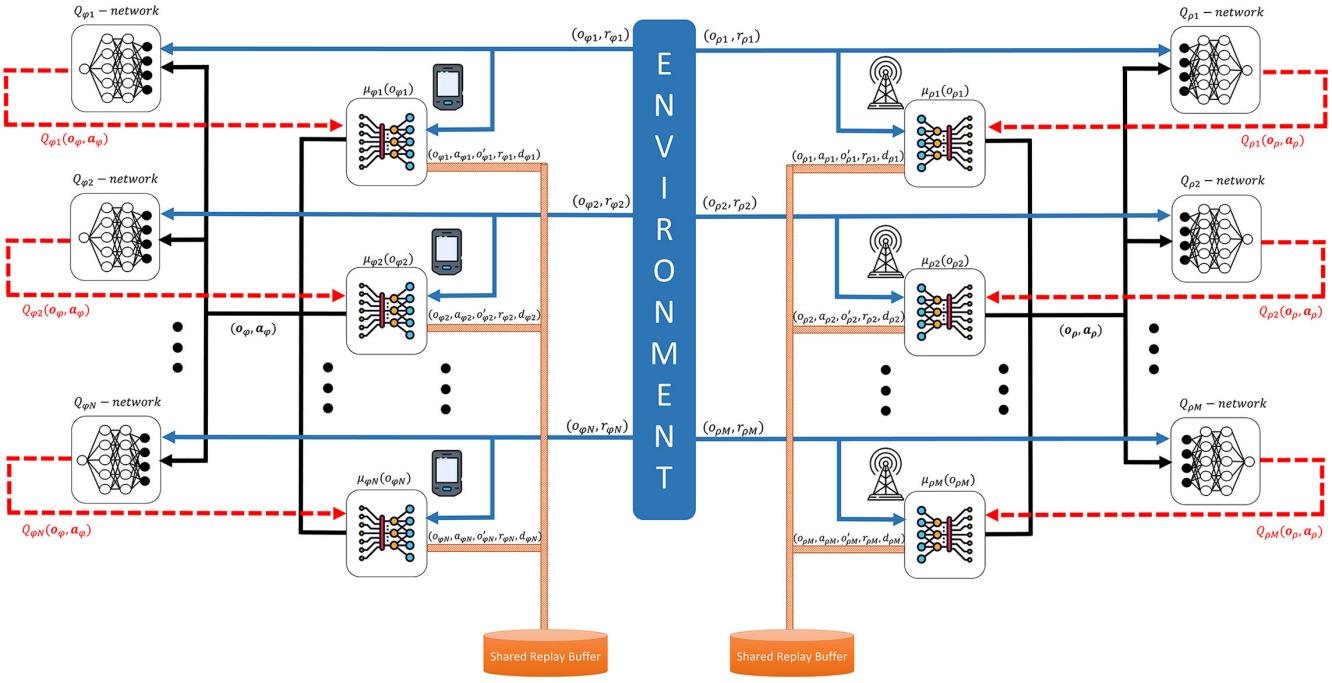

Fig. 2. TB-MADDPG training architecture.

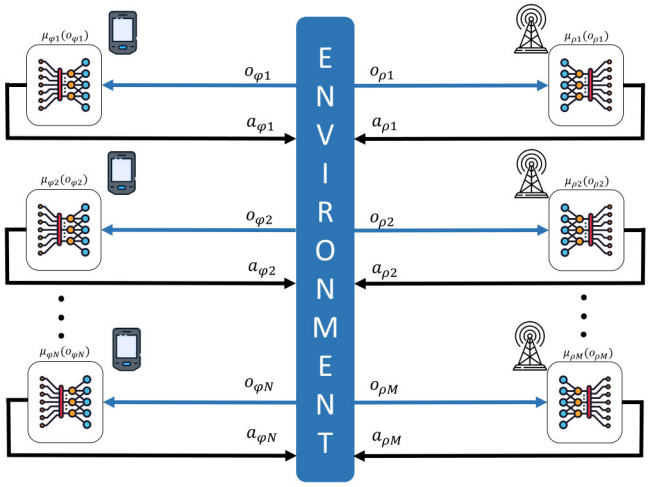

Fig. 3. TB-MADDPG execution architecture.

use local information (i.e., their own observation only) at execution time. Moreover, MADDPG is not only limited to a single type of interaction between the agents, but can use any type of interaction including cooperative, competitive, mixed or neither of these types.

However, the MADDPG assumes the agents are homogeneous (i.e., they have the same structure of observation and actions), which is not the case in our environment, where we have two types of agents, with different structure of observations and actions. Moreover, having heterogeneous agents in the environment makes the environment even less stationary, which hardens the agents-learning process from the environment interactions. Hence, we propose the Team-Based MADDPG (TB-MADDPG), which is essentially a dual MADDPG frameworks interacting with each other, and groups each type of agents in a team, and each team has a different joint policy, and therefore, a different Q-function to assess each joint policy. The first MADDPG has the PENs as its agents, and considers the other team as part of the environment. Similarly, the second MADDPG has the RANs as its agents, and considers the PENs as part of its environment. Furthermore, each team make use of a shared replay buffer that stores all the agents' experiences. An agent experience can be represented by the tuple (o, a, o', r, d), where d represents the done flag (i.e., when the agent stops interacting with the environment). The agents from the same team are not allowed to observe other agents experiences in the shared replay buffer. However, the critic networks make full use of that buffer in order to assess the joint actions given a joint observation, and hence, assessing the joint policy. Such modification makes the environment more stationary, and make the learning in each team as a normal homogeneous MADDPG training.

As mentioned before, there will be two teams in our solution design, namely, the RANs and the PENs teams. We denote the joint observation and joint action for the RAN team by $\boldsymbol{o_\rho}$ and $\boldsymbol{a_\rho}$ respectively, where a single action for an agent $a_{\rho_j} = \mu_{\rho_j}(o_{\rho_j}; \vartheta_{\rho_j})$, with $\mu_{\rho_j}(.)$ denoting the output of the RAN actor network j, and ϑ_{ρ_j} is the set of parameters of the actor network. Similarly for the PENs team, we denote the joint observation and joint action by $\boldsymbol{o_\varphi}$ and $\boldsymbol{a_\varphi}$ respectively, and single action for the PEN agent i is $a_{\varphi_i} = \mu_{\varphi_i}(o_{\varphi_i}; \vartheta_{\varphi_i})$. Also, the joint policy for the RANs team and the PENs team are denoted by $\boldsymbol{\pi_\rho}$ and $\boldsymbol{\pi_\varphi}$. Finally, a single Q-function in the RANs team and the PENs team is denoted by $Q_{\rho_j}(\boldsymbol{o_\rho}, \boldsymbol{a_\rho}; \phi_{\rho_j})$ and $Q_{\varphi_i}(\boldsymbol{o_\varphi}, \boldsymbol{a_\varphi}; \phi_{\varphi_i})$, where ϕ_{ρ_j} and ϕ_{φ_i} are the set of parameters for the critic network j and i in the RANs and the PENs team respectively.

Learning in TB-MADDPG consists of Q-learning for the critics, and policy learning for the actors. The concept of Q-learning is that if the optimal Q-function is known, then the optimal action to take is the one which maximizes the Q-function. Since each critic parameterize the Q-function, the goal of the critic is to be as close as possible to the optimal Q-function. Therefore, for each critic, we can set up an indicator of how close the critic to the optimal Q-function is, which is the Mean-Squared Bellman Error (MSBE), and defined as the following:

$$\mathcal{L}(\phi_x, \mathcal{B}_x) = \underset{\mathcal{B}_x \sim \mathcal{D}_x}{\mathrm{E}} \Big[\big(Q_x(\boldsymbol{o}_x, \boldsymbol{a}_x; \phi_x) - y_x(\boldsymbol{o'}_x, r_x, d_x) \big)^2 \Big] \tag{23}$$

where $x \in \{\rho, \varphi\}$, $\mathcal{D}$ is the shared replay buffer in the team, and $\mathcal{B}$ is a sampled batch of joint experiences from the buffer. The term $y_x(.)$ is known as the Temporal Difference (TD)-target, and is given by:

$$y_x\big(\boldsymbol{o'}_x, r_x, d_x\big) = \big(\ r_x + \gamma(1 - d_x) Q_x\big(\boldsymbol{o'}_x, \boldsymbol{a'}_x; \phi_x\big)\big) \tag{24}$$

where $\boldsymbol{a}'$ is the joint action from the actors given the next state $\boldsymbol{o}'$. The aim of each critic is to minimize the MSBE, which can be done by gradient descent on its parameters ϕ:

$$\phi_x \leftarrow \phi_x - \eta_{\phi_x} \nabla_{\phi_x} \mathcal{L}(\phi_x, \mathcal{D}_x) \tag{25}$$

where η_ϕ is the learning rate of the critic network.

Each actor in TB-MADDPG learns independently, relying on its local observations and its critic only, where the actor tries to learn an optimal policy that maximizes the Q-function. Hence, the objective of each actor from both teams can be expressed as:

$$\max_{\vartheta_x} \underset{o_x \sim \mathcal{D}_x}{\mathrm{E}} \left[Q_x(o_x, \mu_x(o_x; \vartheta_x); \phi_x) \right] \tag{26}$$

which can be maximized by simple performing gradient ascent on the actor parameters φ as follows:

$$\vartheta_x \leftarrow \vartheta_x + \eta_{\vartheta_x} \nabla_{\vartheta_x} \underset{o_x \sim \mathcal{D}_x}{\mathrm{E}} \left[Q_x(o_x, \mu_x(o_x; \vartheta_x); \phi_x) \right] \tag{27}$$

where η_ϑ is the learning rate of the actor network. Moreover, to stabilize the learning process [49], each actor and critic network have a time-delayed copy of itself, and are called the target networks. The parameters of the target networks are updated as follows:

$$\phi_{targ} \leftarrow (1 - \epsilon)\phi_{targ} + \epsilon\phi \tag{28}$$

$$\vartheta_{targ} \leftarrow (1 - \epsilon)\vartheta_{targ} + \epsilon\vartheta \tag{29}$$

where ϵ is the soft update parameters, and $\epsilon << 1$. The TB-MADDPG algorithm is summarized in Algorithm 1.

Lastly, it is worth recalling that the TB-MADDPG algorithm is done with centralized training (Fig. 2) and decentralized execution (Fig. 3). In other words, during training, the agents' critics can make use of extra knowledge about other agents' interaction with the environment in order to help the actors to learn the optimal joint policy. However, at inference time, the actors only receive their local observation and do not share or exchange any private information with other agents. The online training can be held in a trusted virtualized environment, where the replay buffers of each team can be stored and can receive the experiences from the agents, and deploy the models on the PENs and the RANs every once in a while.

Algorithm 1: Team-Based-Multi Agent DDPG

```
Initialize actors and critics networks parameters
Initialize a random process N for action exploration
for episode e=1 : Episodes do
    Receive initial states for both teams o_ρ, o_φ
    for time step t=1 : Steps do
        At the PENs side:
        for PEN agent i = 1 : N do
            Select and execute action
            a_φj = μ_φj(o_φj) + N following policy π_φ
        end
        Observe PENs next states o'_φ and rewards r_φ
        Store (o_φ, a_φ, o'_φ, r_φ) in PENs team buffer D_φ
        At the RANs side:
        for RAN agent j = 1 : M do
            Select and execute action
            a_ρj = μ_ρj(o_ρj) + N following policy π_ρ
        end
        Observe RANs next states o'_ρ and rewards r_ρ
        Store (o_ρ, a_ρ, o'_ρ, r_ρ) in RANs team buffer D_ρ
        At both sides:
        if time to train then
            Sample a mini-batch B_x from each D_x
            for each agent do
                Compute targets y_x(o'_x, r_x, d_x)
                according to (24)
                Compute loss function L(φ_x, B_x)
                according to (23)
                Update critic parameters:
                φ_x ← φ_x − η_φx ∇_φx L(φ_x, B_x)
                Update actor parameters:
                ϑ_x ← ϑ_x +
                η_ϑx ∇_ϑx E_{o_x∼D_x} [ Q_x(o_x, μ_x(o_x; ϑ_x); φ_x) ]
                Update target networks parameters
                according to (28), (29)
            end
        end
    end
end
```

VI. Simulation Results

In this section, we first present the environmental setup. Second, we evaluate the agents' behavior in terms convergence and the learned policy. Lastly, we compare our proposed approach to existing state-of-art techniques for network selection and resource allocation.

A. Environmental Setup

The simulations were conducted using the parameters as in Table I. Moreover, to adjust the trade-off between the metrics according to the patient status, the hyperparameters α, β, λ and δ will have similar values when the patient has no seizure.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Number of RANs M	3
Number of PENs N	5
Probability of seizure υ	0.1
Data length B	1 Mb
Distortion parameters $c1 - c6$	as in [37]
RAN types	[5G, 4G, 3G]
Data rates r_1, r_2, r_3	[40, 25, 15] Mbps
Monetary costs $\varepsilon_1, \varepsilon_2, \varepsilon_3$	[6, 3, 0.1]$*10^{-6}$ Euros
Bandwidth W	20 Mhz
Noise spectrum density N_0	-174 dBm
Path loss attenuation σ	$3.6 * 10^{-6}$
$Episodes$	8000
$Steps$	200
Discount factor γ	0.95
Replay buffer size $\mathcal{D}$	10000
Batch size $\mathcal{B}$	128
Optimizer	Adam[50]
Critic learning rate η_ϕ	0.0003
Actor learning rate η_ϑ	0.0001

However, λ and δ, which indicate the significance of the latency and distortion respectively, will have higher values than the others when the patient has a seizure. As for the seizure, we assume that the seizure can happen at any point in time with a probability υ_i, where $\mathbf{P}(\zeta_i^t = 1) = \upsilon_i$.

B. Rewards Convergence and Policy Evaluation

In the training process of the agents, we ran Algorithm 1 for 6000 episodes, where in the first 500 episodes the agents' actions are fully exploratory. After that, the exploration starts decaying until the agents actions become fully exploitary. Fig. 4 shows the achieved rewards for the agents during training. In Fig. 4(a), it can be seen that the RAN agents obtained the convergence around episode 4000. Similarly, in Fig. 4(b), most of the PEN agents obtained the convergence at episode 4000, except for *PEN 2* agent, it converged around episode 5200.

We show the learned policy for the agents from both teams, the *PEN* and the *RAN* teams during one episode of evaluation. The learned policy for the RAN team after training is depicted in Fig. 5. Firstly, in Fig. 5(a), we can see that *RAN* 1, has the highest data rate and has allocated equal share of the bandwidth for *PEN*s 1, 4 and 5 and a smaller bandwidth for *PEN* 3, while *PEN* 2 got neglected completely and has not been allocated anything by *RAN* 1. Secondly, the *RAN* 2 bandwidth allocation is shown in Fig. 5(b). It can be seen that *PEN*s 1, 2, 3 have been allocated equal share of the bandwidth, while *PEN* 5 has a slightly larger bandwidth, and *PEN* 4 got neglected completely. Lastly, the bandwidth allocation for *RAN* 3, which has the smallest data rate, is shown in Fig. 5(c). In fact, *RAN* 3 could only allocate for only two *PEN*s, which are *PEN* 2 and 4, where allocating to more *PEN*s will make the data rate for each one very low, and will not be sufficient to deliver the data on time.

The *PEN*s agents learned policy is depicted in Fig. 6. The policy is represented by the network utilization indicators (i.e., P_{ij}'s) and the compression ratio. In Fig. 6(a), we can see that

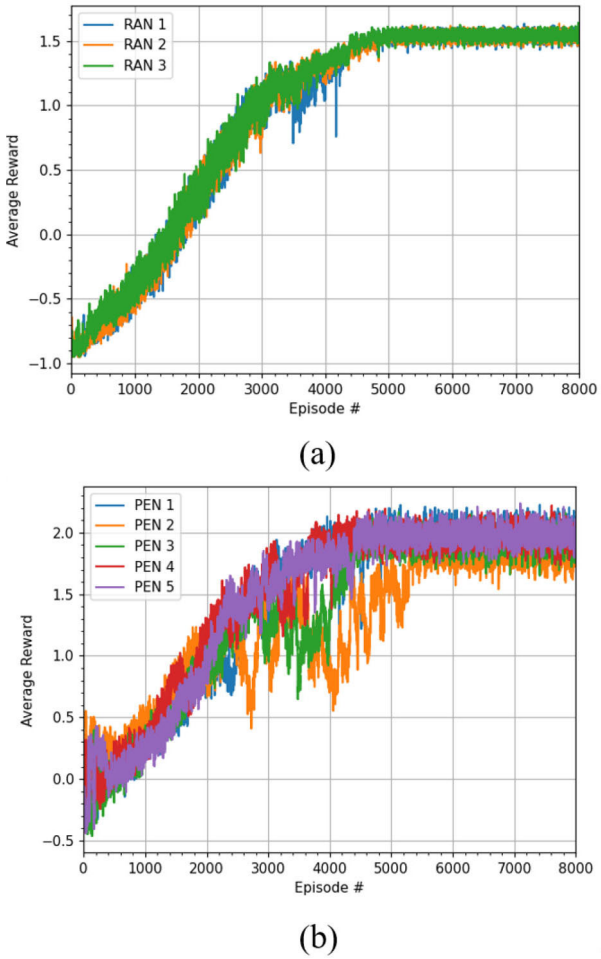

Fig. 4. The achieved rewards during training for agents from (a) RAN team and (b) PEN team.

PEN 1 has utilized *RAN* 1 the most, along with *RAN* 2, while it did not send any data to *RAN* 3 since it has no allocated bandwidth on it. As for *PEN* 2, which its allocation is shown in Fig. 6(b), it had utilized *RAN* 2 and 3 and ignored *RAN* 1 as it has no bandwidth share on it. Similar to *PEN* 1, *PEN*s 3, and 5 have the same behavior as they utilize *RAN* 1 and 2, while not sending any data to *RAN* 3. Lastly, *PEN* 4 had utilized *RAN* 1 along with *RAN* 3, and ignored *RAN* 2. However, a noticeable pattern in the *PEN*s policies exist, which is when the seizure happens, the *PEN*s tend to utilize the *RAN*s more on which the *PEN*s have more available bandwidth. Finally, the compression ratio for all the *PEN*s is shown in Fig. 6(f). Interestingly, all the *PEN*s maintain a high compression ratio, while at the seizure time, the compression drops down sharply to a very low value so that the medical data have the minimal possible distortion.

C. Performance Comparison

In order to illustrate our proposed algorithm's performance, we compare it to multiple techniques, namely, a heuristic, Autonomous and adaptive Network Selection (AANSC) [51] and the Optimal Network Selection and Resource Allocation (ONSRA) algorithms [45]. The heuristic algorithm will be

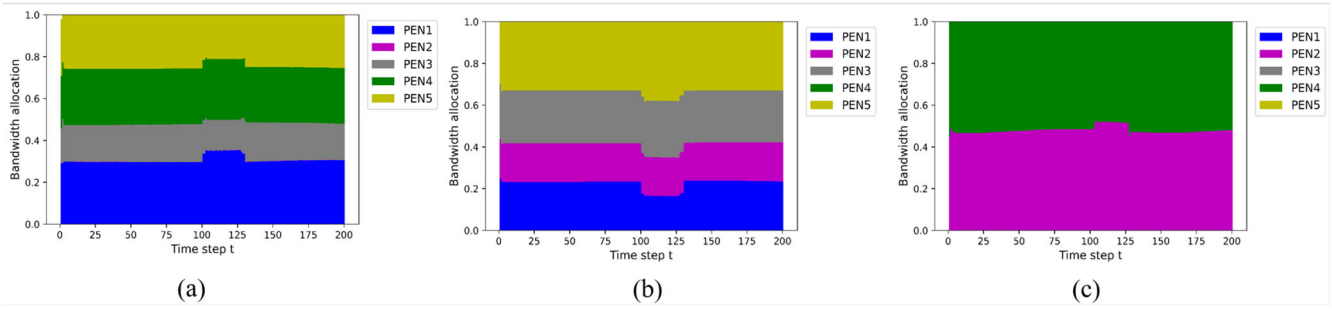

Fig. 5. The learned policy for (a) *RAN* 1 , (b) *RAN* 2 and (c) *RAN* 3 agents.

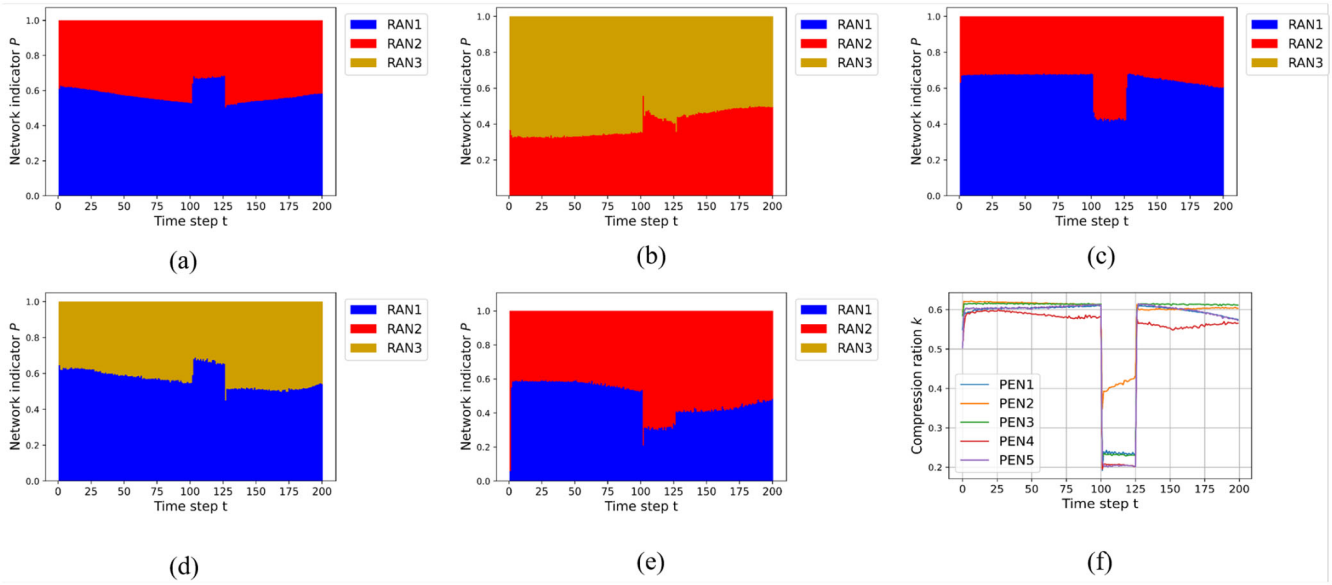

Fig. 6. The learned policy for *PEN* agents in terms of (a)-(e) network utilization, and (f) compression ratio.

the naive approach, which is the RANs allocate equal shares of the bandwidth to all the PENs, and the PENs to utilize all the RANs available equally, while setting the compression ratio to the minimum possible according to (11). The AANSC algorithm assumes that all the PENs have equal shares of the bandwidth on all the RANs, while tries to optimize the energy consumption, monetary cost, latency and distortion at the PEN side. As for the ONSRA algorithm, it simplifies the problem **P1** by sub-dividing it into two sub-problems. The first sub-problem is to optimize the resource allocation at the RANs side, given the network utilization indicators as constants from the PENs. Whereas the second sub-problem is the optimization of the energy consumption, monetary cost, latency and distortion at the PEN side after receiving the allocated bandwidth from the RANs, and it keeps exchanging the solutions of the two sub-problems between the RANs and the PENs until convergence. The heuristic approach will be the baseline, while the ONSRA will be the benchmark of the comparison. We show a sample for the comparison for one PEN in a setting, where we keep the PEN running for hours until the PEN runs out of battery. The comparison will be in terms of the average rewards achieved during the PEN lifetime according to 21, the battery lifetime, average energy consumption, latency, monetary cost and distortion, where Fig. 7 depicts all the comparisons between our proposed approach and the aforementioned algorithms.

In Fig. 7(a), we can see the average obtained rewards during the PEN lifetime. The reward at first declines slowly as the battery level decreases over time. However, our proposed algorithm has a slower decline than the other approaches, and the heuristic has the fastest decline since it is the naive approach and does not take into consideration any optimization perspective. After two hours approximately, we can see the sharp drop in the rewards. This is due to the fact that the patient had a seizure at that time, and the reward function changed to signify the latency and the distortion importance while disregarding the energy consumption and the monetary cost. Moreover, the proposed approach had a lower drop than the other two approaches, the AANSC and the ONSRA, where the heuristic drained the battery before the seizure. The battery lifetime achieved by the algorithms is shown in Fig. 7(b). The heuristic approach has the worst battery lifetime, which is around 1.8 hours, while the AANSC and the ONSRA had achieved better PEN lifetimes with 2.5 and 2.8 hours respectively. Indeed, our proposed algorithm achieved the highest battery lifetime with 3.8 hours, where it achieved a lifetime which is better by 100% more

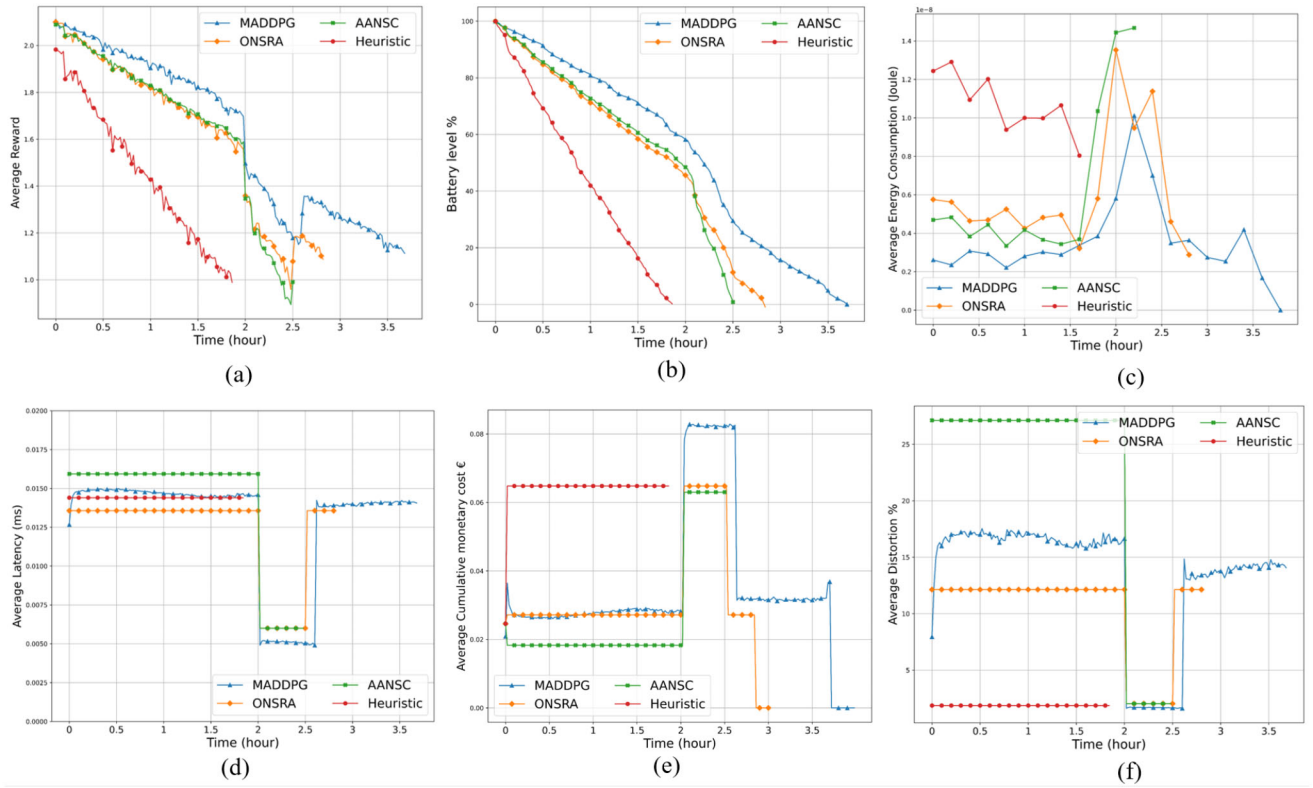

Fig. 7. Comparison between our proposed and other algorithms in terms of average (a) obtained rewards, (b) battery lifetime, (c) energy consumption, (d) latency, (e) monetary cost and (f) distortion.

than the heuristic, and by 45% and 30% more than the AANSC and the ONSRA respectively. In fact, the energy consumption of the algorithms, which is shown in 7(c) justifies the aforementioned results, where our proposed approach had the lowest energy consumption on average. Moreover, when the patient had a seizure, we can notice a spike in the energy consumption, where it resulted in increasing the declining slop in the battery lifetime in the previous figure.

As for the latency, which is depicted in Fig. 7(d), the algorithms achieved similar results, with a slight advantage to the ONSRA algorithm. However, at seizure time, our algorithm achieves a slightly better latency than the other algorithms. Afterwards, the cumulative monetary cost is shown in Fig. 7(e). While the heuristic had the highest cost, the other algorithms had similar overall cost. However, during the seizure, our algorithm had the highest overall cost. In fact, as mentioned before, the proposed algorithm disregards the trade-off between energy consumption, latency, cost and distortion, and focuses only on the latency and the distortion in order to minimize the medical data transmission time along with the minimum distortion possible. Finally, the distortion is shown Fig. 7(f). When the patient is not having a seizure, we can notice that the algorithms other than the heuristic allow some levels of distortion in order to minimize the energy consumption, latency and cost. Nevertheless, all the algorithms tend to have negligible distortion at seizure time.

VII. TB-MADDPG Complexity Analysis

The TB-MADDPG algorithm employs neural networks to facilitate the agents' and critics' training, specifically, the multi-layer perceptron architecture (MLP). First, for a single-agent RL that employs the MLP architecture with three layers, it has been shown in [52] that the training complexity is given by $O(e \times t(S \times \eta + \eta \times A))$, where e is the number of episodes of training, t can be either the number of iteration per episode or the lifetime of a PEN. The S denotes the input layer size, which also represents the agent's observations set size, η is the hidden layer size, and A denotes the output layer, which also represents the agent's actions set size. As for the critic, recall that in our algorithm, each critic in each team evaluate the joint actions and observations of the whole team with a single value. Thus, its training complexity is given by $O(e \times t((N \times (A+S)) \times \eta + \eta \times 1))$, where N is the number of agents in the team. Hence, the training complexity of one agent with its critic can be expressed as $O(\Omega)$, where Ω is given by:

$$\Omega = e \times t((S \times \eta + \eta \times A) + (N \times (A+S)) \times \eta + \eta \times 1). \tag{30}$$

Consequently, the training complexity of one team with N agents can by given by $\mathrm{O(N\,\Omega)}$. Finally, since we have two teams, namely, the RANs and the PENs teams, with M and N agents in each team respectively, the TB-MADDPG training complexity is given by $O(\Omega\,(N+M))$. As for the execution complexity, since each agents acts on its own without the

need for critic and other agents interactions, the complexity for taking one action at any given time step for each agent is the complexity of a single feed forward pass in a neural network, which is given by $O(S \times \eta + \eta \times A)$.

VIII. Conclusion

In this paper, we presented a novel approach for network selection along with adaptive compression at the PEN side, and resource allocation for the PENs at the RANs side. More precisely, we have proposed a DMARL algorithm, namely, the TB-MADDPG, which accounts for the heterogeneity of the agents. The TB-MADDPG groups each type of the agents in each team, and tries to find the optimal joint policy for the PENs team in terms of network utilization and adaptive compression, while finding the optimal policy for the RANs team in terms of bandwidth allocation to maximize the PENs QoE. The presented results in this paper shows that our proposed approach significantly outperforms the existing state-of-art techniques for network selection and resource allocation.

References

[1] A. A. Abdellatif, A. Z. Al-Marridi, A. Mohamed, A. Erbad, C. F. Chiasserini, and A. Refaey, "ssHealth: Toward secure, blockchain-enabled healthcare systems," *IEEE Netw.*, vol. 34, no. 4, pp. 312–319, Jul./Aug. 2020.

[2] A. Awad, A. Mohamed, C. Chiasserini, and T. Elfouly, "Network association with dynamic pricing over D2D-enabled heterogeneous networks," in *Proc. IEEE WCNC*, 2017, pp. 1–6.

[3] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, "A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 668–695, 2nd Quart., 2021.

[4] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security, and intelligence," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 126–132, Oct. 2020.

[5] X. Wang, X. Li, and V. C. Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," *IEEE Access*, vol. 3, pp. 1379–1391, 2015.

[6] X. Wang, J. Li, L. Wang, C. Yang, and Z. Han, "Intelligent user-centric network selection: A model-driven reinforcement learning framework," *IEEE Access*, vol. 7, pp. 21645–21661, 2019.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[8] M. Tan, "Multi-agent reinforcement learning: Independent versus cooperative agents," in *Proc. Tenth Int. Conf. Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[9] F. Fischer, M. Rovatsos, and G. Weiss, "Hierarchical reinforcement learning in communication-mediated multiagent coordination," in *Proc. 3rd Int. Joint Conf. Auton. Agents Multiagent Syst.*, vol. 3, 2004, pp. 1334–1335.

[10] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2016, pp. 2137–2145.

[11] J. W. Crandall and M. A. Goodrich, "Learning to compete, compromise, and cooperate in repeated general-sum games," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 161–168.

[12] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," 2015, *arXiv:1507.06527*.

[13] A. Gündoğan, H. M. Gürsu, V. Pauli, and W. Kellerer, "Distributed resource allocation with multi-agent deep reinforcement learning for 5G-V2V communication," in *Proc. 21st Int. Symp. Theory Algorithmic Found. Protocol Design Mobile Netw. Mobile Comput.*, 2020, pp. 357–362.

[14] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13124–13138, Nov. 2020.

[15] U. Challita, W. Saad, and C. Bettstetter, "Cellular-connected uavs over 5G: Deep reinforcement learning for interference management," 2018, *arXiv:1801.05500*.

[16] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 2, pp. 411–425, Feb. 2021.

[17] X. Liu, J. Yu, Z. Feng, and Y. Gao, "Multi-agent reinforcement learning for resource allocation in IoT networks with edge computing," *China Commun.*, vol. 17, no. 9, pp. 220–236, Sep. 2020.

[18] R. Wang, M. Li, L. Peng, Y. Hu, M. M. Hassan, and A. Alelaiwi, "Cognitive multi-agent empowering mobile edge computing for resource caching and collaboration," *Future Gener. Comput. Syst.*, vol. 102, pp. 66–74, Jan. 2020.

[19] Y. Zhang, Y. Zhou, H. Lu, and H. Fujita, "Cooperative multi-agent actor–critic control of traffic network flow based on edge computing," *Future Gener. Comput. Syst.*, vol. 123, pp. 128–141, Oct. 2021.

[20] A. Awad, A. Mohamed, and C.-F. Chiasserini, "Dynamic network selection in heterogeneous wireless networks: A user-centric scheme for improved delivery," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 53–60, Jan. 2017.

[21] Y. Zhu, J. Li, Q. Huang, and D. Wu, "Game theoretic approach for network access control in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9856–9866, Oct. 2018.

[22] J. Cui, D. Wu, and Z. Qin, "Caching AP selection and channel allocation in wireless caching networks: A binary concurrent interference minimizing game solution," *IEEE Access*, vol. 6, pp. 54516–54526, 2018.

[23] E. Stevens-Navarro, Y. Lin, and V. W. S. Wong, "An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 1243–1254, Mar. 2008.

[24] M. El Helou, M. Ibrahim, S. Lahoud, K. Khawam, D. Mezher, and B. Cousin, "A network-assisted approach for RAT selection in heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1055–1067, Jun. 2015.

[25] J. G. Andrews, S. Singh, Q. Ye, X. Lin, and H. S. Dhillon, "An overview of load balancing in hetnets: Old myths and open problems," *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 18–25, Apr. 2014.

[26] M. Drissi, M. Oumsis, and D. Aboutajdine, "A fuzzy ahp approach to network selection improvement in heterogeneous wireless networks," in *Proc. Int. Conf. Netw. Syst.*, 2016, pp. 169–182.

[27] E. Stevens-Navarro and V. W. S. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks," in *Proc. IEEE 63rd Veh. Technol. Conf.*, vol. 2, 2006, pp. 947–951.

[28] C. Desogus, M. Anedda, and M. Murroni, "Real-time load optimization for multimedia delivery content over heterogeneuos wireless network using a mew approach," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2017, pp. 1–4.

[29] I. Joe, W. Kim, and S. Hong, "A network selection algorithm considering power consumption in hybrid wireless networks," in *Proc. Int. Conf. Comput. Commun. Netw.*, 2007, pp. 1240–1243.

[30] Q. Song and A. Jamalipour, "Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques," *IEEE Wireless Commun.*, vol. 12, no. 3, pp. 42–48, Jun. 2005.

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

[32] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.

[33] D. D. Nguyen, H. X. Nguyen, and L. B. White, "Reinforcement learning with network-assisted feedback for heterogeneous RAT selection," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6062–6076, Sep. 2017.

[34] Z. Chkirbene, A. Awad, A. Mohamed, A. Erbad, and M. Guizani, "Deep reinforcement learning for network selection over heterogeneous health systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 258–270, Jan./Feb. 2022.

[35] M. Yan, G. Feng, J. Zhou, and S. Qin, "Smart multi-RAT access based on multiagent reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4539–4551, May 2018.

[36] A. Awad, M. H. M. Elsayed, and A. Mohamed, "Encoding distortion modeling for DWT-based wireless EEG monitoring system," 2016, *arXiv:1602.04974*.
[37] A. Awad, M. Hamdy, A. Mohamed, and H. Alnuweiri, "Real-time implementation and evaluation of an adaptive energy-aware data compression for wireless EEG monitoring systems," in *Proc. 10th Int. Conf. Heterogeneous Netw. Qual. Reliabil. Security Robustness*, 2014, pp. 108–114.
[38] "Chapter 3—Edge computing for energy-efficient smart health systems: Data and application-specific approaches," in *Energy Efficiency of Medical Devices and Healthcare Applications*, A. Mohamed, Ed. London, U.K.: Academic, 2020, pp. 53–67.
[39] A. Awad, O. A. Nasr, and M. M. Khairy, "Energy-aware routing for delay-sensitive applications over wireless multihop mesh networks," in *Proc. 7th Int. Wireless Commun. Mobile Comput. Conf.*, 2011, pp. 1075–1080.
[40] A. Awad, A. Mohamed, A. A. El-Sherif, and O. A. Nasr, "Interference-aware energy-efficient cross-layer design for healthcare monitoring applications," *Comput. Netw.*, vol. 74, pp. 64–77, Dec. 2014.
[41] K. Mahmud, M. Inoue, H. Murakami, M. Hasegawa, and H. Morikawa, "Measurement and usage of power consumption parameters of wireless interfaces in energy-aware multi-service mobile terminals," in *Proc. IEEE Int. Symp. Personal Indoor Mobile Radio Commun.*, vol. 2, 2004, pp. 1090–1094.
[42] Y. Wang, M. C. Vuran, and S. Goddard, "Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 305–318, Feb. 2012.
[43] *IEEE Standard for Local and Metropolitan Area Networks—Part 21: Media Independent Handover*, IEEE Standard 802.21-2008, 2009.
[44] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[45] A. A. Abdellatif, M. S. Allahham, A. Mohamed, A. Erbad, and M. Guizani, "ONSRA: An optimal network selection and resource allocation framework in multi-RAT systems," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.
[46] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *Proc. AAAI*, vol. 4, 2004, pp. 709–715.
[47] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2020, *arXiv:1706.02275*.
[48] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2019, *arXiv:1509.02971*.
[49] S. Zhang and R. S. Sutton, "A deeper look at experience replay," 2017, *arXiv:1712.01275*.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
[51] A. A. Abdellatif, A. Mohamed, and C. Chiasserini, "User-centric networks selection with adaptive data compression for smart health," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3618–3628, Dec. 2018.
[52] M. S. Allahham, A. A. Abdellatif, A. Mohamed, A. Erbad, E. Yaacoub, and M. Guizani, "I-SEE: Intelligent, secure, and energy-efficient techniques for medical data transmission using deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6454–6468, Apr. 2021.

Mhd Saria Allahham (Student Member, IEEE) received the B.Sc. degree in computer engineering, with excellence, from Qatar University. He is currently pursuing the master's degree with Queen's University, Canada. His research interests include reinforcement learning and edge computing.

Alaa Awad Abdellatif (Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in electronics and electrical communications engineering from Cairo University, in 2009 and 2012, respectively, and the Ph.D. degree from Politecnico di Torino, in 2018, where he is currently a Postdoctoral Researcher. He also worked as a Senior Research Assistant and Research Assistant with Qatar University, from 2013 to 2015, and with Cairo University, from 2009 to 2012, respectively. He has authored or coauthored over 35 refereed journal, magazine, and conference papers in reputable international journals and conferences. He has served as a technical reviewer for many international journals and magazines. His research interests include edge computing, blockchain, machine learning and resources optimization for wireless heterogeneous networks, smart-health, IoT applications, and vehicular networks. He was a recipient of the Graduate Student Research Award from Qatar National Research Fund, and the Best Paper Award from Wireless Telecommunications Symposium 2018, in USA, in addition to the Quality Award from Politecnico di Torino in 2018.

Naram Mhaisen received the B.Sc. degree in computer engineering, with excellence, and the M.Sc. degree in computing from Qatar University in 2017 and 2020, respectively, where he was a Research Assistant with the Department of Computer Science and Engineering, College of Engineering, from 2017 to 2020. He is currently with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. His research interests include the design and optimization of distributed and multi-agent (learning) systems with applications to IoT.

Amr Mohamed (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of British Columbia, Vancouver, Canada, in 2001, and 2006, respectively. He has worked as an advisory IT specialist in IBM Innovation Center, Vancouver, from 1998 to 2007, taking a leadership role in systems development for vertical industries. He is currently a Professor with the College of Engineering, Qatar University and the Director of the Cisco Regional Academy. He has over 25 years of experience in wireless networking research and industrial systems development. He has authored or coauthored over 200 refereed journal and conference papers, textbooks, and book chapters in reputable international journals, and conferences. His research interests include wireless networking and edge computing for IoT applications. He holds 3 awards from IBM Canada for his achievements and leadership, and the four best paper awards from IEEE conferences. He is serving as a technical editor for two international journals and has served as a technical program committee co-chair for many IEEE conferences and workshops.

Aiman Erbad (Senior Member, IEEE) received the B.Sc. degree in computer engineering from the University of Washington, Seattle, in 2004, the Master of Computer Science degree in embedded systems and robotics from the University of Essex, U.K., in 2005, and the Ph.D. degree in computer science from the University of British Columbia, Canada, in 2012. He is an Associate Professor and ICT Division Head with the College of Science and Engineering, Hamad Bin Khalifa University. Prior to this, he was an Associate Professor with the Computer Science and Engineering Department and the Director of Research Planning and Development with Qatar University until May 2020. He also served as the Director of Research Support responsible for all grants and contracts from 2016 to 2018, and as the Computer Engineering Program Coordinator from 2014 to 2016. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. His research interests span cloud computing, edge intelligence, Internet of Things, private and secure networks, and multimedia systems. He received the Platinum award from the H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category), the 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. He is an Editor for *KSII Transactions on Internet and Information Systems* and *the International Journal of Sensor Networks* and a Guest Editor for IEEE NETWORK. He also served as the Program Chair of the International Wireless Communications Mobile Computing Conference (IWCMC 2019), the Publicity Chair of the ACM MoVid Workshop 2015, the Local Arrangement Chair of NOSSDAV 2011, and the Technical Program Committee Member in various IEEE and ACM international conferences (GlobeCom, NOSSDAV, MMSys, ACMMM, IC2E, and ICNC). He is a Senior Member of ACM.

Mohsen Guizani (Fellow, IEEE) received the B.S. (with distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently with the Department of Machine Learning, MBZUAI, and acts as an Associate Provost for Faculty Affairs and Institutional Advancement. Previously, he served in different academic and administrative positions with the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. He is the author of nine books and more than 500 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is currently the Editor-in-Chief of the *IEEE Network Magazine* serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of *Wireless Communications and Mobile Computing Journal* (Wiley). He also served as a member, the chair, and the general chair of a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He is a Senior Member of ACM.