Synchronizing UAV Teams for Timely Data Collection and Energy Transfer by Deep Reinforcement Learning

Omar Sami Oubbati, Mohammed Atiquzzaman, Hyotaek Lim, Abderrezak Rachedi, Abderrahmane Lakas

▶ **To cite this version:**

Omar Sami Oubbati, Mohammed Atiquzzaman, Hyotaek Lim, Abderrezak Rachedi, Abderrahmane Lakas. Synchronizing UAV Teams for Timely Data Collection and Energy Transfer by Deep Reinforcement Learning. IEEE Transactions on Vehicular Technology, 2022, 71 (6), pp.6682-6697. 10.1109/TVT.2022.3165227 . hal-03819561

HAL Id: hal-03819561
https://hal.science/hal-03819561

Submitted on 18 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synchronizing UAV Teams for Timely Data Collection and Energy Transfer by Deep Reinforcement Learning

Omar Sami Oubbati, *Member, IEEE*, Mohammed Atiquzzaman, *Senior Member, IEEE*, Hyotaek Lim, Abderrezak Rachedi, *Senior Member, IEEE*, and Abderrahmane Lakas, *Senior Member, IEEE*

***Abstract*—Due to their promising applications and intriguing characteristics, Unmanned Aerial Vehicles (UAVs) can be dispatched as flying base stations to serve multiple energy-constrained Internet-of-Things (IoT) sensors. Moreover, to ensure fresh data collection while providing sustainable energy support to a large set of IoT devices, a required number of UAVs should be deployed to carry out these two tasks efficiently and promptly. Indeed, the data collection requires that UAVs first make Wireless Energy Transfer (WET) to supply IoT devices with the necessary energy in the downlink. Then, IoT devices perform Wireless Information Transmission (WIT) to UAVs in the uplink based on the harvested energy. However, it turns out that when the same UAV performs WIT and WET, its energy usage and the data collection time are severely penalized. Worse yet, it is difficult to efficiently coordinate between UAVs to improve the performance in terms of WET and WIT. This work proposes to divide UAVs into two teams to behave as data collectors and energy transmitters, respectively. A Multi-Agent Deep Reinforcement Learning (MADRL) method, called TEAM, is leveraged to jointly optimize both teams' trajectories, minimize the expected Age of Information (AoI), maximize the throughput of IoT devices, minimize the energy utilization of UAVs, and enhance the energy transfer. Simulation results depict that TEAM can effectively synchronize UAV teams and adapt their trajectories while serving a large-scale dynamic IoT environment.**

***Index Terms*—UAV, Wireless Powered Communication Network (WPCN), Trajectory optimization, Multi-Agent Deep Reinforcement Learning (MADRL).**

I. Introduction

As 5G mobile networks advance and become a reality, Unmanned Aerial Vehicles (UAVs) are more often deployed as aerial base stations to support such networks and build a communication bridge between distant and unconnected ground nodes [1], [2]. In another application, UAVs are able to gather data from IoT devices, and relay the collected data to a central controller for decision making. However, the freshness of gathered information, the energy utilization of both UAVs and IoT devices, and the optimization of UAV trajectories play important roles in increasing the central controller's decision quality [3]. The freshness of information generated by a given IoT device is measured by tracking the time passed since the last information intercepted at the UAV was created on the IoT device, which is well-known as the age of information (AoI) [4]. This metric is widely adopted in recent UAV-based data collection schemes, such as in [5]. However, most of these works did not consider the restricted energy of IoT devices.

Wireless Energy Transfer (WET) process has recently emerged as an alternative solution for wirelessly powering IoT devices using Radio Frequency (RF) signals. Once IoT devices are charged, they will be able to transfer information to UAVs or establish what is known as the Wireless Information Transfer (WIT) process. A novel concept, called Wireless Powered Communication Network (WPCN), was proposed to support both WET and WIT processes under a single system based on UAVs [6]. The WPCN concept is adopted in a wide range of works in the literature, *e.g.,* in [7], where the UAV trajectories are optimized to maximize the available energy on a restricted number of IoT devices. Furthermore, in [8], the authors dispatch a single UAV-enabled WPCN at a fixed position to reduce the accomplishment time of gathering a given amount of bits per IoT device. Nevertheless, as far as we know, most UAV-assisted WPCN solutions did not investigate the high densities of IoT devices and their dynamics over a vast area. Moreover, three other main issues are frequently distinguished in the previously discussed works. First, it was noticed that the adopted RF-based WET process in most UAV-enabled WPCN schemes is impacted energy attenuation due to the path loss. Second, the WET process frequently causes the problem of collecting stale information, especially when it is jointly integrated with the WIT process in the same UAV. Finally, the constant mobility of UAVs could quickly deplete their batteries, and consequently, they could always stop working after a short period of time.

To address all these issues, it is required to provide an efficient energy supply and a punctual collection of fresh information over a large-scale dynamic IoT environment. For this purpose, in this work, it is assumed that the WIT and WET processes are allotted to two distinct teams of UAVs, as in the motivating scenario shown in Fig. 1. In addition, the trajectories of UAVs should be optimized and adequate technologies have to be exploited to supply IoT devices with exhausted batteries, to ensure a certain freshness of collected information, and to reduce the energy consumption

O.S. Oubbati is with LIGM, University Gustave Eiffel, Marne-la-Vallée, France. E-mail: omar-sami.oubbati@univ-eiffel.fr

M. Atiquzzaman is with the University of Oklahoma, Norman, OK USA. E-mail: atiq@ou.edu

H. Lim is with Department of Computer Engineering Dongseo University 47011 Busan, South Korea. Email: htlim@dongseo.ac.kr

A. Rachedi is with University of Paris-Est, France. E-mail: rachedi@u-pem.fr

A. Lakas is with College of Information Technology, United Arab Emirates University, United Arab Emirates. Email: alakas@uaeu.ac.ae

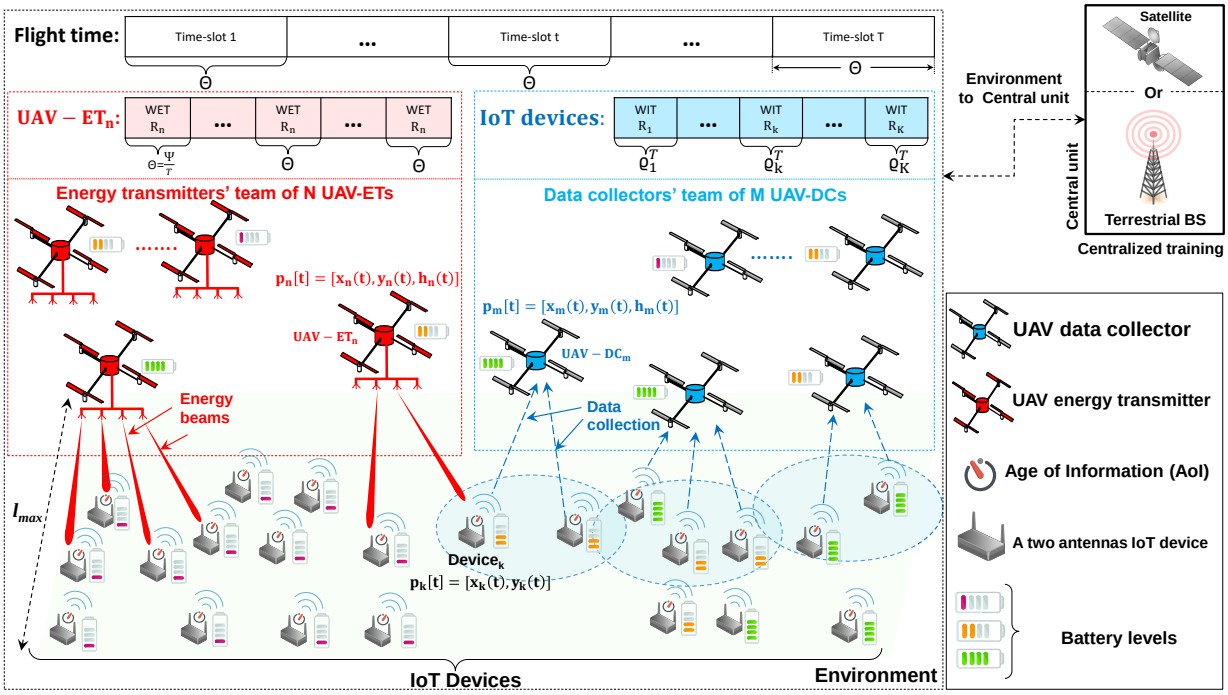


Fig. 1: Application scenario of TEAM framework.

of UAVs. However, since it is the case of a time-varying and unknown environment (*i.e.,* dynamic IoT devices), it will be difficult to proceed using traditional optimization techniques. Hence, each UAV of both teams is controlled by its own Deep Reinforcement Learning-based (DRL-based) agent to intelligently control and manage its trajectories and resource allocation. To do so, it is supposed that both teams of UAVs are backhaul-connected either with a satellite or a fully working terrestrial BS, which is a popular hypothesis in different works [9]. This allows to ensure the centralized training of the different agents and to have a global vision and continuous observation of the environment (*i.e.,* locations of UAVs, energy usages, AoI, etc.) and to intelligently control each UAV movement towards achieving an intelligent synchronization with both teams and efficiently assist IoT devices. However, transmitting periodical updates from the environment to these central units causes significant energy consumption, without mentioning the different communications restrictions due to the existing obstacles. These constraints are out of the scope of this paper, and therefore the focus is to be placed on how to control and synchronize the movements of both UAV teams to adapt to the dynamics of the IoT environment and optimally serve it. To sum up, the following contributions are carried out:

- Designing a multi-UAV-enabled WPCN system consisting of two teams providing an optimal AoI-aware data collection over a large set of dynamic IoT devices.
- Leveraging a Multi-Agent DRL (MADRL) based strategy for minimizing the overall AoI of the system by jointly maximizing the throughput of IoT devices, optimizing the trajectories and energy consumption of UAVs, and the resource allocations of UAVs in each team.
- Conducting simulation experiments to study the effectiveness of the system and evaluate its numerical results.

The remaining organization of this paper can be outlined as follows. Section II presents an extensive review of the related work. Section III provides a detailed description of the system model and formulates the problem statement. Section IV briefly introduces the background of reinforcement learning and presents the proposed MADRL framework, namely TEAM. In Section V, the obtained numerical results are analyzed. Finally, the paper is concluded in Section VI.

II. Related Work

Due to their advantages and flexibility, UAVs are expected to be fully integrated into the future generation of mobile networks, such as B5G and even 6G, for a win-win situation [16]. This integration efficiently addresses the problems related to data collection, such as the energy consumption of mobile devices (*i.e.,* UAVs and served ground devices), and access protocols and security. Moreover, UAVs are an excellent choice to serve ground users and IoT devices. To avoid being out of the scope of this work, this section will be focused on three significant research challenges related to UAV-assisted IoT devices: (i) Data collection, (ii) AoI, and (iii) WPCN.

A. UAV-based data collection and AoI

Due to their flexibility, UAVs could be considered a promising solution to be deployed as data collectors, allowing IoT devices to upload their data in the uplink with low energy and in a reduced time. For instance, in [10], the transmission scheduling of terrestrial sensors and the UAV trajectory are jointly optimized to minimize the energy

TABLE I: Features comparison of the related approaches for UAV-assisted IoT networks.

Features	Data Collection		AoI-aware Data Collection		WPCN		TEAM framework
	Ref. [10]	Ref. [11]	Ref. [12]	Ref. [13]	Ref. [14]	Ref. [15]	
Main topic	Energy consumption minimization of IoT devices.	Completion time minimization of data collection	Minimizing weight sum AoI while leveraging a DRL strategy	Trajectory optimization of multiple UAVs to minimize AoI	Trajectory optimization of multiple UAVs to maximize throughput	Optimization of resource allocation in UAV-enabled WPCN	Intelligent trajectories of multiple UAVs enabled WPCN
Optimization	Differential Evolution (DE)	Min-max multiple Travelling Salesman	DRL		MADRL	Bellman dynamic programming	MADRL
IoT deployment	Uniform stationary positions		Stationary on grids	Moving on Roads	Stationary on clusters	Categorized with stationary positions	Randomly moving
Major advantage	Minimization of energy consumption of data collection.	Minimization of data collection completion and flying times	Minimization of AoI while considering the energy consumption of UAV	Minimization of AoI while considering the mobility of IoT devices	Enhancing the trajectories in each cluster and maximizing the minimum throughput	Achieving optimal power and price control of UAV	Minimizing the average AoI and energy consumption of UAVs and maximizing the throughput while considering the IoT devices' mobility
Major Limitation	Energy consumption of UAV is omitted	Energy consumption and collisions avoidance are not considered	Increasing of AoI when the number of IoT devices increases	Energy of UAVs is neglected	The freshness of collected information is not considered and it not cost-effective	Energy consumption of UAVs in overlooked	Complexity of the system

consumption of sensors, while ensuring the required amounts of collected data. In [17], multiple UAVs are deployed to collect data from time-constrained vehicles while maximizing the vehicular network throughput. Orfanus *et al.* [18] adopted the promising paradigm of self-organization to deploy a set of UAVs as wireless relays to serve ground sensors in the context of military scenarios. In [19],an energy-efficient self-organization model with two-level data aggregations was adopted for cluster-based communication UAV networks. Guan *et al.* [20] proposed a novel distributed algorithm for controlling self-organizing UAVs with massive Multiple Input Multiple Output (MIMO) network capacities. The authors of [11] minimized the completion time of data collection by employing multiple UAVs under the constraint of a defined amount of data to be collected. The authors of [12] applied a DRL strategy to optimize the trajectory and energy consumption of the UAV to minimize the AoI of IoT devices. Finally, in [13], multiple UAVs are dispatched to gather data from vehicles moving on a highway while leveraging a DRL method to reduce their average AoI.

B. UAV-enabled WPCN

Recently, UAVs have been considered as adequate objects for playing the role of energy sources, powering IoT devices, and collecting data. This concept is known as WPCN, which is widely investigated in the literature. For example, Park *et al.* [21] deployed a UAV-enabled WPCN to maximize the minimum throughput of sensors by jointly optimizing the movement and energy of UAV and resource allocation based on linear and non-linear energy harvesting models. The authors of [14] adopted a MADRL strategy to maximize the minimum throughput in UAV-enabled WPCN system by jointly optimizing the 3D trajectories of UAVs and resource allocation. In [15], the authors designed UAV-assisted wireless powered IoT network and addressed the resource allocation problem between IoT devices and the UAV, which is formulated as a dynamic game theory.

Table I selects a set of crucial features to make a comprehensive comparative study between the discussed schemes and TEAM framework. From this study, it comes that the majority of UAV-assisted network schemes adopt three main MADRL based methods. First, as in [14], MA-Deep Q Network (MADQN) based methods are mainly deployed to face small-scale discrete space problems with discrete action space of UAVs. However, as a drawback, MADQN-based methods could suffer from the problem of over-fitting, where the values of different actions get overestimated under certain situations, and thus the system converges more slowly. Second, MA-Double Deep Q Network (MADDQN) based methods are based on neural networks and extra layers to overcome the problem of over-fitting, which could slow down the learning speed of the system. Finally, as in TEAM framework, MA-Deep Deterministic Policy Gradient (MADDPG)-based methods could address all the issues mentioned above by facing complex and dynamic environments generating high-dimensional states and learning continuous control policies. Moreover, it was distinguished that there are four main challenges in most UAV-enabled data collection contributions to overcome. First, as already mentioned, most of these contributions are mainly based on straightforward scenarios involving a small set of IoT devices or sensors assisted with single or multiple UAVs, while omitting the scenarios where IoT devices are dynamic and consume a significant energy amount in both communication and movement. Second, the deployed UAVs in these contributions perform WET and WIT in an integrated way where UAVs should first perform Wireless Energy Transfer (WET) to supply IoT devices with the necessary energy, and then IoT devices transmit their sensed data based on the harvested energy. This technique causes a significant delay in data collection, and the energy usage of UAVs is severely penalized. Third, the WET process in these contributions is mainly based on RF-signals transmitted in an omnidirectional manner, which suffers from attenuation due to path loss and interference, thus significantly decreasing its performance. Finally, the relevant DRL-based contributions mainly consider a single agent and discretized UAV trajectory in their architectures, which substantially increases the error rate of obtained policies and considerably limits the deployment and adaptation of UAVs in a dynamic real-world and large-scale IoT environment.

The main differences that distinguish our work from other

schemes discussed above are threefold:

- Two teams of UAVs are deployed for supporting a scalable number of dynamic and energy-constrained IoT devices that intermittently transmit amounts of data towards a central controller for decision making.
- The WET and WIT processes are performed separately by two distinct UAV teams to efficiently decrease the AoI of collected data and make IoT devices always sufficiently charged and ready for transmitting their data towards UAV-DCs. Moreover, the WET process is supported by the energy beamforming to increase the energy transfer efficiency and avoid the interference problem.
- A multi-agent DRL method is leveraged to optimally control and synchronize the movements of UAVs in both teams to optimally adapt them to the scalability and dynamics of the IoT environment while jointly maximizing the throughput of IoT devices, reducing the energy consumption of UAVs, and optimizing the energy charging of IoT devices to avoid unsuccessful data collection due to the insufficient energy of IoT devices.

III. System Model

As illustrated in Fig. 1, a multi-UAV-enabled WPCN system is deployed, which is consisting of two teams of UAVs acting separately as data collectors and energy transmitters. The team of UAV-ETs is indicated as $\mathcal{N}$, and the team of UAV-DCs is denoted as $\mathcal{M}$, where $\mathcal{N} \bigcap \mathcal{M} = \emptyset$. An important number of moving terrestrial energy-constrained IoT devices are uniformly distributed over a harsh region to sense various physical phenomena, which are denoted as $\mathcal{K} \triangleq \{k = 1, 2, \ldots, K\}$. A typical example of these IoT devices could be small dynamic robots deployed in areas where the human intervention is constraining. The system is analyzed during a predefined flight duration of UAVs, represented as $t \in [0, \psi]$. To simplify the analysis, the flight period is discretize into T time-slots, where $\Theta = \frac{\psi}{T}$ is the length of each time-slot. T is supposed to be sufficiently large such that UAVs in both teams appear to be approximately stationary at each time-slot. The locations of each IoT device $k \in \mathcal{K}$ is represented by $p_k[t] = [x_k(t), y_k(t)]^T$ at each time-slot $t \in \mathcal{T} \triangleq \{t = 1, 2, \ldots, T\}$, which are supposed to move only inside the target square area of width l_{max}. All UAVs from both teams are constantly flying around IoT devices to supply them with energy and gathering fresh information, all in a timely manner. At each time-slot $t \in \mathcal{T}$, the instantaneous trajectories of UAV-ET$_n$ and UAV-DC$_m$ projected onto the horizontal plane are denoted as $p_n[t] = [x_n(t), y_n(t)]^T$ and $p_m[t] = [x_m(t), y_m(t)]^T$, respectively, $\forall n \in \mathcal{N}, \forall m \in \mathcal{M}$. The altitudes of both UAV-ET$_n$ and UAV-DC$_m$ are denoted as $h_n[t]$ and $h_m[t] \in [h_{min}, h_{max}]$, respectively. The distance between each UAV i and device k are given by:

$$d_k^i[t] = \sqrt{\|p_i[t] - p_k[t]\|^2 + h_i^2[t]}, \tag{1}$$

where $i \in \mathcal{M} \cup \mathcal{N}$. It is assumed that all UAVs are backhaul-connected to a satellite or a fully working terrestrial BS in which the centralized training of each DRL agent is operated (*see* Section IV). For the sake of clarity, Table II defines the different notations used in this paper.

TABLE II: List of notations.

Notation(s)	Description
$\mathcal{M}, M, m$	Set, number, and index of UAV-DCs
$\mathcal{N}, N, n$	Set, number, and index of UAV-ETs
$\mathcal{K}, K, k$	Set, number, and index of IoT devices
$\mathcal{T}, T, t$	Set, number, and index of time-slots
$d_i^j[t]$	Distance between nodes i and j
$p_i[t]$	Coordinates of node i
$R_i, E_i[t]$	Transmission power and residual energy of node i
$A_k^t, O_k^m[t], \Gamma_{k,m}^t[k]$	AoI, rate, and data harvesting decision of device k
Ω_k^t, ϖ_k^t	Status of unsuccessful data upload and unserved device k at time-slot t
F_i^{max}, E_i^{max}	Maximum speed and energy capacity of UAV i
R_i^t, P_i^t	Reward and penalty of UAV i
o_i^t, a_i^t	Observation and action of UAV i
s_x^t, a_x^t	States and actions of the set $x \in \{ET, DC\}$
$\pi^i(.), Q^i(.)$	Actor and critic networks
$\pi^{i'}(.), Q^{i'}(.)$	Target actor and critic networks
η^{Q^i}, η^{π^i}	Parameters of critic and actor networks
$\eta^{Q^{i'}}, \eta^{\pi^{i'}}$	Parameters of target critic and actor networks
$\mathcal{B}_i, \Delta, \delta$	Replay buffer of UAV i, Mini-batch size and index
ε, ζ	Action noise and discount factor

It is noteworthy that the trajectory optimization of UAVs in both teams using the DRL method exploits different technologies to support WET and WIT processes. Therefore, in the following subsections, each technology will be defined along with a set of parameters that the DRL algorithm will fully exploit to synchronize the UAV teams to timely serve the IoT environment.

A. Channel Modeling

Each UAV-ET is mounted with an $U \times V$ uniform planar array (UPA), while each UAV-DC is equipped with a single antenna for collecting information from IoT devices. Furthermore, each IoT device k is mounted with two antennas operating over different orthogonal frequency bands, each of which is devoted to either energy harvesting or data transmission to avoid interference. Therefore, each UAV-ET could generate multiple narrow beams to simultaneously transmit RF signals to IoT devices. After receiving energy beams, the IoT devices convert the RF energy and satisfy the energy supply through RF-energy gathering devices. For effortlessness, the dynamicity of beam angles is disregarded because of the mechanical vibration of UAVs and wind flow. In the WET process, it is expected that the line-of-sight (LoS) dominates the channel between IoT devices and UAVs. According to [22], [23], the channel CH_k^n between UAV-ET$_n$ and device k can be expressed as follows:

$$CH_k^n = \sqrt{\eta_0 (d_k^n)^{-\alpha}} r(\varphi, \omega), \tag{2}$$

where $\alpha \geq 2$ represents the path loss factor. η_0 represents the median of the power gain at the reference distance $d_0 = 1$m. $r(\varphi, \omega)$ denotes the steering vector of the LoS path with input

parameters φ and ω as the azimuth and elevation angles, which are estimated by:

$$\begin{aligned} r(\varphi,\omega) = [1,\ldots \\ &, exp(j2\pi/\lambda a_{array}\sin(\varphi)[(u-1)\cos(\omega)+(v-1)\sin(\omega)]), \\ &\ldots, \\ &exp(j2\pi/\lambda a_{array}\sin(\varphi)[(U-1)\cos(\omega) \\ &+(V-1)\sin(\omega)])]^T, \end{aligned} \tag{3}$$

where a_{array} denotes the spacing between antenna elements. $\lambda = \frac{c}{f_c}$ represent the wavelength, c is the light speed, and f_c is the carrier frequency. u and v represents the coordinate of antenna elements. $[.]^T$ denotes the conjugate transpose. The time-varying channel gain between UAV-ET$_n$ and device k is calculated by:

$$G_k^n[t] = \frac{\eta_0}{(d_k^n[t])^{\frac{\alpha}{2}}}|r(\varphi,\omega)B|^2, \tag{4}$$

where $B = [b_{1v},\ldots,b_{uv},\ldots,b_{UV}]$ denotes the beamforming vector describing the phase and amplitude excitation of each array element $b_{uv} = c_{uv}(\varphi,\omega)\times I_{uv}\times exp(\rho_{uv})$, where $c_{uv}(\varphi,\omega)$ is the active pattern, I_{uv} is the amplitude excitation of $(u,v)-th$ array element, ρ_{uv} denotes the progressive phase shift. There is a scenario when multiple UAV-ETs cooperatively serve a given IoT device. In this case, a distributed energy beamforming protocol as proposed in [24] is adopted to achieve optimal energy transfer towards the IoT device. In the case when the batteries embedded on IoT devices have unlimited capacity, the total harvested energy at time-slot t from all UAV-ETs can be expressed using the following linear energy harvesting model:

$$E_k^{ET}[t] = \sum_{n=1}^{N} E_k^n[t] = \sum_{n=1}^{N} \xi_k G_k^n[t] R_n \Theta, \tag{5}$$

where R_n and $E_k^n[t]$ denote the transmit power and the transmitted energy of UAV-ET$_n$, respectively. $0 < \xi_k < 1$ represents the energy conversion efficiency of device k and it is set to a certain value for all IoT devices. We are aware that exploiting linear energy harvesting is not accurate enough because as the input RF power progressively increases in the practical energy harvesting circuits, the output direct current (DC) power eventually gets saturated. However, the energy harvester in this work is supposed to not operate in the saturation region, and the received power from UAV-ETs is not high. Therefore, a linear harvester may roughly represent the energy harvesting process. To be more realistic, the batteries of IoT devices are supposed to have a capacity restricted to E_k^{max}. Thus, the total residual energy of device k at time-slot $t+1$ (*i.e.,* after energy harvesting) is calculated as follows:

$$E_k[t+1] = \begin{cases} E_k^{max}, & \text{if } E_k[t] + E_k^{ET}[t] \geq E_k^{max}, \\ E_k[t] + E_k^{ET}[t], & \text{Otherwise}, \end{cases} \tag{6}$$

where $E_k[t]$ is the energy level (*i.e.,* residual energy) of device k at time-slot t. According to [25], the adopted WET technology could suffer from three major issues. First, IoT devices' limited energy capacity should suffice not only for data transmission, but also for sensing and idle listening made by the IoT devices. To address this issue, an RF-based wake-up mechanism can be deployed to activate IoT devices for data transmission and avoid maintaining them activated all the time while enhancing the energy consumption of sensing and idle listening. This mechanism is out of the scope of this framework, but it can be considered in it. Second, WET technology is sensitive to the UAV's restricted powering range, limiting the energy harvested at the IoT devices. In TEAM framework, the mobility and altitude of UAV-ETs are flexible and optimized through time to ensure an acceptable level of transferred energy. Finally, IoT devices generally use the result of spectrum sensing as a basis for energy harvesting or data transmission. Therefore, channel fading is not considered in this framework, and it is assumed to be constant during spectrum sensing.

As for the WIT process, since our framework is LoS dominant, it is appropriate to adopt a Rician fading model [26]. However, for the sake of simplicity, we adopted a pure LOS to model the used channel due to the neglect of small-scale fading in this work. Therefore, the effect of small-scale fading is considered as insignificant [27], [28]. Thus, the time-varying uplink channel gain between UAV-DC$_m$ and device k at time-slot t can be expressed as follows:

$$G_k^m[t] = \frac{\eta_0}{\|p_m[t] - p_k[t]\|^2 + h_m^2[t]}. \tag{7}$$

As depicted in Fig. 1, a given UAV-DC$_m$ can simultaneously serve multiple IoT devices in a single time-slot t. Thus, TDMA is adopted during the WIT process, where each time-slot t is divided into K sub-slots. Each k-th sub-slot of a duration ϱ_k^t is dedicated to a covered device k. For the proper functionality of the data collection, the following constraints should be satisfied:

$$\|p_m[t] - p_m[t-1]\|^2 \leq (F_m^{max}\Theta)^2, \quad \forall m \in \mathcal{M}, \forall t \in \hat{\mathcal{T}}, \tag{8a}$$

$$\sum_{k=1}^{K} \Gamma_{k,m}^t[k] \leq 1, \quad \forall t \in \mathcal{T}, \forall m \in \mathcal{M}, \tag{8b}$$

$$\sum_{m=1}^{M} \Gamma_{k,m}^t[k] \leq 1, \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \tag{8c}$$

where F_m^{max} is the maximum speed of each UAV-DC$_m$ in m/s and $\hat{\mathcal{T}} = \{1,\ldots,T-1\}$. Let $\Gamma_{k,m}^t[k]$ denotes a binary variable, which indicates that UAV-DC$_m$ is collecting the update of device k at sub-slot k within time-slot t if $\Gamma_{k,m}^t[k] = 1$, and 0 otherwise. The constraint (8a) indicates that the positions UAV-DCs are approximately constant according to the devices at t. The constraints (8b) and (8c) suppose that at each sub-slot k, each dispatched UAV-DC$_m$ only collects the update of at most one device k and each device k is scheduled by at most one UAV-DC$_m$. TDMA is adopted for its simplicity despite its major drawback that forces each covered IoT device to have a fixed allocation of channel time whether or not it has data to transmit, especially when the other covered IoT devices with the same UAV-DC do not plan to transmit data.

Suppose that R_k denotes the transmission power of the device k. Therefore, the instantaneous rate for device k to UAV-DC$_m$ at time-slot t is calculated as follows:

$$O_k^m[t] = \Gamma_{k,m}^t[k] W \log_2\left(1 + \frac{R_k G_k^m[t]}{\sigma^2}\right), \tag{9}$$

where $\sigma^2 = WL_0$ denotes the additive white Gaussian noise at UAV-DC$_m$, where L_0 represents the power spectral density of the Additive White Gaussian Noise (AWGN). W is the channel bandwidth in Hertz (Hz). Consequently, the distance between UAV-DCs and devices defines the amount of data that can be successfully transmitted to UAV-DCs in the uplink. Let $Z_k^m[t]$ denotes the fraction of bits, which is successfully transmitted by device k to UAV-DC$_m$ at time-slot t based on the following equation:

$$Z_k^m[t] = \varrho_k^t O_k^m[t]. \tag{10}$$

where ϱ_k^t is the duration of each k-th sub-slot dedicated to a covered device k.

B. Age of Information (AoI) of moving IoT devices

To calculate the freshness of collected information by UAV-DCs, the AoI metric is calculated for each served device. Indeed, the most popular definition of AoI has been that it is the time elapsed since the latest update generated by device k and received by UAV-DC$_m$. For instance, the AoI of device k that generated an update at $U_k[t]$ and successfully transmitted it at time-slot t is given by:

$$A_k^t = t - U_k[t]. \tag{11}$$

In TEAM system, A_k^t is incremented whenever an update fails to be received by a given UAV-DC$_m$. Otherwise, A_k^t is reset to one. It should be stressed that UAV-DCs calculate the AoI at the end of each time-slot t, and the devices generate the number of bits at the beginning of each time-slot. The AoI evolution estimation of device k can be expressed by:

$$A_k^t = \begin{cases} 1, & \text{if } Z_k^m[t] \geq Z_k^{min} \wedge \Gamma_{k,m}^t[k] = 1, \\ A_k^{t-1} + 1, & \text{Otherwise}, \end{cases} \tag{12}$$

where Z_k^{min} denotes the minimum amount of bits required for ensuring the right decoding of a generated update by a given UAV-DC$_m$. It is worthy to note that the AoI of each device is defined based on the movement of UAV-DCs, the dynamics of devices, the transmission scheduling, and the density of UAVs deployed during the whole flight period. Realistically, the calculation of AoI during the entire data collection mission by considering the movement randomness of IoT devices and the importance of their generated updates is given by $\sum_{k\in\mathcal{K}} \mathbb{E}\left[\sum_{t=1}^{T} \kappa_k A_k^t\right]$, where $\mathbb{E}[.]$ is defined as the expected value of AoI with respect to the randomness of devices' mobility and κ_k denotes the weight associated with each device k depending on the nature and importance of its processing task. After applying equation (12) to a specified device k during a flight period of 12 time-slots, the obtained AoI evolution of device k illustrated in Fig. 2.

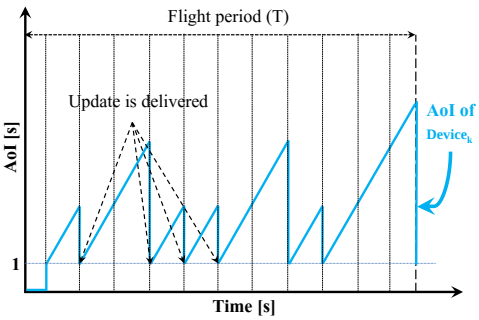


Fig. 2: Associate AoI of device$_k$ in twelve (12) time-slots.

C. Energy consumption of UAVs

UAVs tend to be energy-constrained devices, and their lifetime is highly dependent on energy storage. As mentioned earlier, each UAV i, $i \in \mathcal{M} \cup \mathcal{N}$, is supposed to fly with a maximum speed F_i^{max}. It is worthy of mentioning that the communication energy consumption is relatively small compared to the energy dedicated for propulsion [29], and therefore it is not considered in this work. The energy utilization of all UAVs follows the same model proposed in [30] in which the propulsion energy of all UAVs can be calculated as follows:

$$\begin{aligned} P(F) = & \underbrace{R_b\left(1 + \frac{3F^2}{F_{tip}^2}\right)}_{blade\ profile\ power} \\ & + \underbrace{R_u\left(\sqrt{1 + \frac{F^4}{4z_0^2}} - \frac{F^2}{2z_0^2}\right)}_{induced\ power} + \underbrace{\frac{1}{2} f_0 a z H F^3}_{parasite\ power}, \end{aligned} \tag{13}$$

where F represents the speed of UAVs, F_{tip} denotes the rotor blade's tip speed, z_0 indicates the mean induced velocity, and z is the rotor solidity. R_b and R_u are the blade profile power and the induced power in a static flight (*i.e.,* hovering), respectively. H, a, and f_0 are the rotor disc area, air density, and fuselage drag ratio. However, there is an exception for UAV-ETs, where their energy consumption is also related to energy transmission to IoT devices from their own energy source. In this context, two functionality modes of UAV-ETs are distinguished, where $l_n[t] = 1$ means that UAV-ET$_n$ is executing a WET process, otherwise, $l_n[t] = 0$. The energy consumption of UAV-ET$_n$ and UAV-DC$_m$ until the current time-slot t can be calculated based on the following equations:

$$C_n(\{l_n[t], p_n[t]\}, t) = \underbrace{\int_0^t P(\|f_n[t]\|)dt}_{propulsion\ energy} + \underbrace{\int_0^t l_n[t] R_n dt}_{energy\ transfer}, \tag{14a}$$

$$C_m(\{p_m[t]\}, t) = \underbrace{\int_0^t P(\|f_m[t]\|)dt}_{propulsion\ energy}, \tag{14b}$$

where $f_n[t] = \dot{p_n}[t]$ and $f_m[t] = \dot{p_m}[t]$ represent the velocities of UAV-ET$_n$ and UAV-DC$_m$, respectively. Moreover, $\|f_n[t]\|$ and $\|f_m[t]\|$ denote the speeds of UAV-ET$_n$ and UAV-DC$_m$, respectively. Therefore, the residual energy of UAV-DC$_m$ and UAV-ET$_n$, $\forall m \in \mathcal{M}, \forall n \in \mathcal{N}$ at each time-slot t is expressed as follows:

$$E_n[t+1] = E_n^{max} - C_n(\{l_n[t], p_n[t]\}, t), \quad \forall n \in \mathcal{N}, \tag{15a}$$

$$E_m[t+1] = E_m^{max} - C_m(\{p_m[t]\}, t), \quad \forall m \in \mathcal{M}, \tag{15b}$$

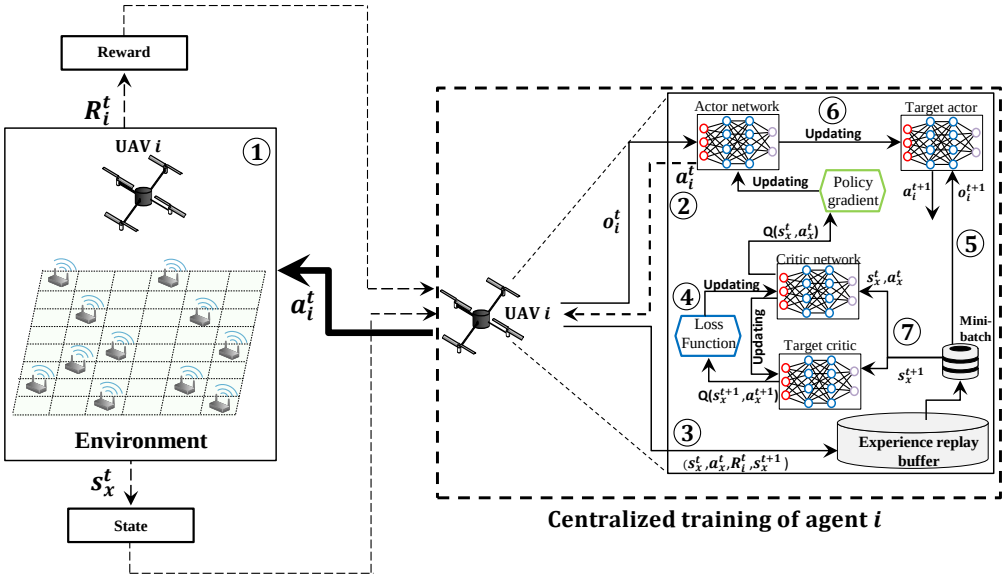


Fig. 3: Structure of UAV i agent.

where E_n^{max} and E_m^{max} are the maximum energy capacity of UAV-ET$_n$ and UAV-DC$_m$, respectively. The average residual energy of all UAVs at each time-slot t can be calculated as follows:

$$\bar{E}[t] = \frac{\left(\sum_{n=1}^{N} E_n[t] + \sum_{m=1}^{M} E_m[t]\right)}{M+N}. \tag{16}$$

D. Problem formulation

The target of this work is to optimize the trajectories of UAV-ETs and UAV-DCs with the aim to enhance several parameters, such as the expected AoI of all devices, the transmission scheduling, the WET process, and the energy consumption of UAVs, all under mobility constraint of IoT devices. For the convenient presentation, let $P_{ET} = \{p_n[t], \forall n \in \mathcal{N}\}$ and $P_{DC} = \{p_m[t], \forall m \in \mathcal{M}\}$ be the set of locations of UAV-ETs and UAV-DCs at each time-slot t, respectively. Thus, an optimization problem can be formulated as follows:

$$\max_{\{P_{DC}\},\{P_{ET}\},\{\Gamma_{k,m}^t[k]\}} \mathbb{E}\left[\left(\frac{\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{k\in\mathcal{K}} O_k^m[t]}{\sum_{k\in\mathcal{K}}\sum_{t=1}^{T}\kappa_k A_k^t}\right)\left(\frac{\sum_{t=1}^{T}\bar{E}[t]}{T}\right)\right] \tag{17}$$

$$\text{s.t.} \quad \textbf{C1:} \quad \|p_i[t] - p_i[t-1]\|^2 \leq (F_i^{max}\Theta)^2,$$
$$\textbf{C2:} \quad \sum_{k=1}^{K} \Gamma_{k,m}^t[k] \leq 1,$$
$$\textbf{C3:} \quad \sum_{m=1}^{M} \Gamma_{k,m}^t[k] \leq 1,$$
$$\textbf{C4:} \quad E_k[t] \geq \varrho_k^t R_k,$$
$$\textbf{C5:} \quad d_i^j[t] \geq L,$$
$$\textbf{C6:} \quad E_i[t] > 0.$$

Note that κ_k is considered as a positive weight of IoT device k, indicating the nature and relative importance of the IoT device's processing task. The definition of κ_k can be carried out manually according to the importance of the AoI for different processes. The higher is the weight, the higher will be the priority of the generated information from the IoT device. **C1** represents the distance traveled by all UAVs at each time-slot t, $\forall t \in \hat{\mathcal{T}}, \forall i \in \mathcal{M} \cup \mathcal{N}$. **C2** and **C3** indicate the scheduling limit where each device k can be served by at most one UAV-DC$_m$ and one UAV-DC$_m$ can schedule at most one device k at each sub-slot k, $\forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \forall k \in \mathcal{K}$. **C4** indicates that the residual energy of each device k should be sufficient for the data transmission towards UAV-DCs, $\forall t \in \mathcal{T}, \forall k \in \mathcal{K}$. The condition **C4** will be satisfied all the time if, and only if, the team of UAV-ETs are held responsible for each unsuccessful data collection (*i.e.,* when condition $E_k[t] < \varrho_k^t R_k$ is verified in equation (21)). The satisfaction of **C4** allows us to guarantee that the IoT devices are always sufficiently charged with energy prior to any data transmission towards UAV-DCs. **C5** denotes that the distance between all UAVs should take into account the safety distance L at each time-slot t, $\forall t \in \mathcal{T}, \forall i, j \in \mathcal{M} \cup \mathcal{N}$, where $i \neq j$. It should be noted that the distance $d_i^j[t] = \sqrt{\|p_i[t] - p_j[t]\|^2 + (h_i[t] - h_j[t])^2}$. **C6** imposes that the energy consumed by all UAVs at each time-slot t should be within their available energy, $\forall i \in \mathcal{M} \cup \mathcal{N}, \forall t \in \mathcal{T}$. Our optimization aims to find a near-optimal control policy $\pi(.)$ to adequately move and synchronize UAVs in each team while simultaneously: (i) minimizing the AoI and maximizing the throughput of all IoT devices, (ii) reducing the energy consumption of UAVs and enhancing the energy transfer towards IoT devices, and (iii) ensuring the proper functionality of WIT and WET while preventing UAVs from colliding with each other. However, achieving all of these goals simultaneously is somewhat tricky due to two factors. On the one hand, to minimize the average AoI, UAVs in both teams should continuously move around to appropriately serve IoT devices. But, on the other hand, to minimize the energy consumption of UAVs, the mobility of UAVs has to be reduced

to save more energy. It is distinguished that (17) is mainly a non-convex mixed-integer optimization problem, which could be relieved based on some heuristic approaches. However, due to the lack of knowledge of the mobility of IoT devices, it would be impossible to probe and be adjusted to the IoT environment's dynamics.

IV. MADRL-BASED FRAMEWORK: TEAM

To solve the problem (17) based on an efficient solution, an MADRL method is involved in learning the environment and optimally planning the trajectories of UAVs. Indeed, multiple AI-based agents are located at a satellite or a terrestrial BS level. As shown in Fig. 3, each agent $i, i \in \mathcal{N} \cup \mathcal{M}$, consistently monitors the IoT environment and routinely explores UAV trajectory and scheduling policy, and synchronizes between dispatched UAVs. Moreover, a sequence of observations, actions, and rewards, results from the interaction between each agent and the IoT environment. Next, the background of MADRL is provided, and then the MADRL-based method will be described and used for solving the multi-UAV cooperative AoI aware WCPN problem.

A. A MADRL Background

A traditional Reinforcement Learning (RL) setup is modeled as a Markov Decision Process (MDP) consisting of a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$ representing the spaces of state, action, reward, and transition probability. At each time-slot t, every agent has to discover the state $s_t \in \mathcal{S}$ and takes the action $a_t \in \mathcal{A}$ based on the policy $\pi(s_t, a_t)$. After that, the agent gets a reward $r_t \in \mathcal{R}$ and updates the current state s_t to $s_{t+1} \in \mathcal{S}$. $\mathcal{P}$ is the transition probability that leads to the new state s_{t+1} after performing an action a_t at the state s_t. This process is repeated by exploiting the tuple (s_t, a_t, r_t, s_{t+1}) until the convergence of π to the optimal policy. However, it is worthy to note that RL is not adequate for complex environments that are characterized by continuous state-action spaces. To address this issue, RL leverages a Deep Neural Network (DNN) to enhance the learning speed and the performance of RL algorithms, and thus creating the concept of DRL. DDPG [31] is a DRL algorithm that can handle continuous control problems. A DDPG algorithm maintains two DNNs, called Actor and Critic neural (AC) networks, where the actor neural networks $\pi(s_t|\eta^\pi)$ generates the optimal action a_t according to the current state s_t. As for the critic neural network $Q(s_t, a_t|\eta^Q)$, it is updated based on the Bellman equation the same applied in Q-learning [32]. The actor neural network $\pi(s_t|\eta^\pi)$ is trained using the chain rule from the start distribution J to the expected reward r_t with respect to the weights (parameters) of the actor as in [33] as follows:

$$\nabla_{\eta^\pi} J(\eta^\pi) \approx \mathbb{E}\left[\nabla_{\eta^\pi} \pi(s|\eta^\pi)|_{s=s_t} \nabla_a Q(s, a|\eta^Q)|_{s=s_t, a=\pi(s_t)}\right]. \tag{18}$$

The problem (17) can be solved based on a multi-agent Markov Decision Process (MA-MDP), which is called an observable Markov game [34]. In this work, it is supposed that there are $M + N$ agents interacting with a dynamic IoT environment, which are characterized by a series of actions $\mathcal{A} \triangleq \{a^t_{ET}, a^t_{DC}, t \in \mathcal{T}\}$ and a series of states $\mathcal{S} \triangleq \{s^t_{ET}, s^t_{DC}, t \in \mathcal{T}\}$, respectively. The states are defined as $s^t_{ET} = \{o^t_n, \forall n \in \mathcal{N}\}$ and $s^t_{DC} = \{o^t_m, \forall m \in \mathcal{M}\}$, where o^t_n, o^t_m, are the private observation of UAV-ET$_n$ and UAV-DC$_m$, respectively. The actions are denoted as $a^t_{ET} = \{a^t_n, \forall n \in \mathcal{N}\}$ and $a^t_{DC} = \{a^t_m, \forall m \in \mathcal{M}\}$, where a^t_n, a^t_m are the respective actions of UAV-ET$_n$ and UAV-DC$_m$. After executing their respective actions, UAV-ET$_n$ and UAV-DC$_m$ receive their respective rewards R^t_n and R^t_m and the environment is updated to new states s^{t+1}_{ET} and s^{t+1}_{DC}, respectively. It is important to note that each agent $i, \forall i \in \mathcal{M} \cup \mathcal{N}$ maintains an actor neural network, where $a^t_i = \pi^i(o^t_i|\eta^{\pi^i})$ and a critic neural network $Q^i(s^t_x, a^t_x|\eta^{Q^i})$, where $x \in \{ET, DC\}$.

B. TEAM Description

As depicted in Fig. 3, the TEAM algorithm is based on a centralized training framework combined with a distributed execution. During the learning phase, each agent $i, \forall i \in \mathcal{M} \cup \mathcal{N}$, sends its own action a^t_i to the environment, and then the reward R^t_i and the state $s^t_x, x \in \{ET, DC\}$ which consists of the observations of all the agents are sent back to each agent. Moreover, each agent does not have only the knowledge of its private information, but also other extra information related to other agents (*e.g.,* their coordinates). Therefore, the critic neural network is trained based on all agents' different observations and actions belonging to a given team. In what follows, the different components of TEAM framework are described.

1) State Space: The environment state s^t_x of a given team $x \in \{ET, DC\}$ is composed of the private observations of all its agents, which is expressed as $s^t_x = \{o^t_i, \forall i \in \mathcal{M} \oplus \mathcal{N}\}$. Each observation o^t_i includes the following information:

- $p_i[t] = [x_i(t), y_i(t), h_i(t)], \forall i \in \mathcal{M} \cup \mathcal{N}$: the current locations of all UAVs.
- $p_k[t] = [x_k(t), y_k(t)], \forall k \in \mathcal{K}$: the current locations of all IoT devices.
- $A^t_k, \forall k \in \mathcal{K}$: the current AoI of all IoT devices.
- $E_k[t], \forall k \in \mathcal{K}$: the current residual energy of all IoT devices.
- $E_i[t], i \in \mathcal{M} \cup \mathcal{N}$: the current residual energy of UAV i.
- $\Gamma^t_{k,m}[k] \in \{0, 1\}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$: denotes the uploading status of all devices, which is represented as a variable that could take the value of 1 if device k has currently started an upload of its data (*i.e.,* device k is served), and 0 otherwise.

At each time-slot t, the format of the observation o^t_i is represented as $o^t_i = [p_1[t], \ldots, p_{M+N}[t], p_1[s], \ldots, p_K[t], A^t_1, \ldots, A^t_K, E_1[t], \ldots, E_K[t], \Gamma^t_{1,m}[1], \ldots, \Gamma^t_{K,m}[K], E_i[t]]$ with a cardinality of $(M + N + 4K + 1)$. To accelerate the learning process, all the elements of o^t_i are normalized. In detail, all elements that can take values greater than 1 are divided into their maximum corresponding values.

2) Action Space: The action a^t_i of each UAV i consists of three parts:

- $\omega^t_i \in [0, 2\pi[$: the horizontal flying direction of UAV i at time-slot t.

- $d_i^t \in [0, d_{max}]$: the flying distance of UAV i under the constraint of the flying speed $F \in [0, F_i^{max}]$.
- $h_i[t] \in [h_{min}, h_{max}]$: the interval altitudes of UAV i.

At each time-slot t, the DDPG algorithm defines the action a_i^t based on the trained control policy. Moreover, multi-rotor UAVs are considered adequate devices to perform all possible actions from the action space due to their small size, maneuverability, and low inertia. These parameters are the reasons behind adopting such a kind of UAVs in TEAM evaluation. However, deploying other types of UAVs, such as fixed-wing UAVs, may lead to neglect specific actions, e.g., hovering in place, making sharp turns (*e.g.,* $\frac{\pi}{2}$, or $\frac{\pi}{4}$), and moving backward. Thus, the deployment of such UAVs will be inefficient as they cannot adapt to the dynamics of the environment, and it will take a long period of time for such UAVs to learn it.

3) Reward Functions: The main goals of this work are to minimize the energy consumption of the deployed UAVs, maximize the throughput, and reduce the expected AoI of the whole system. Therefore, a reward is given to each UAV-ET$_n$ according to both the energy harvested by IoT devices and the data collection successfully made by UAV-DCs from fully charged IoT devices. Moreover, each UAV-DC$_m$ is rewarded when it collects information from IoT devices with low AoI. R_n^t and R_m^t are the rewards for UAV-ET$_n$ and UAV-DC$_m$, respectively, as:

$$R_n^t = E_n[t] \left(\frac{\sum_{k=1}^{K} E_k[t]}{K} \right) - \sum_{k \in \mathcal{K}} \Omega_k^t - P_n^t, \tag{19}$$

$$R_m^t = E_m[t] \left(\frac{\sum_{k=1}^{K} O_k^m[t]}{\sum_{k=1}^{K} \kappa_k A_k^t} \right) - \sum_{k \in \mathcal{K}} \varpi_k^t - P_m^t, \tag{20}$$

where P_n^t and P_m^t are the penalties incurred at each time-slot t by UAV-ET$_n$ and UAV-DC$_m$, respectively. Ω_k^t represents the number of times where a device k tries to upload its data to UAV-DC$_m$, but it fails due to its insufficient energy, whereas ϖ_k^t denotes the number of IoT devices that are not served. Ω_k^t and ϖ_k^t are expressed as follows:

$$\Omega_k^t = \begin{cases} \Gamma_{k,m}^t[k], & \text{if } E_k[t] < \varrho_k^t R_k, \\ 0, & \text{Otherwise,} \end{cases} \tag{21}$$

$$\varpi_k^t = \begin{cases} 1, & \text{if } \Gamma_{k,m}^t[k] = 0, \\ 0, & \text{Otherwise.} \end{cases} \tag{22}$$

Generally speaking, the reward function of UAV-ETs' agents focuses on increasing the average residual energy levels of IoT devices while maintaining the energy consumption of UAV-ETs at its lowest level. Moreover, UAV-ETs are held responsible for each unsuccessful data collection due to the insufficient energy of IoT devices. The reward function of UAV-DCs focuses on maximizing the throughput of all devices while minimizing their AoI. Also, UAV-DCs are rewarded when they do not consume much energy while holding them responsible for each IoT device not served. When a given UAV i runs out of energy and fails, it will undoubtedly penalize and disturb not only their corresponding team. To address these problems and avoid incorporating a novel mechanism of charging UAVs, the respective agents of the failed UAVs will receive a zero or negative reward during the exploitation phase. This considerably helps the UAVs decrease their movements in the following training episodes and charge IoT devices only when required.

4) Expected Penalties: Two crucial constraints should be considered when all UAVs select actions, maintaining a safe distance between UAVs and flying UAVs inside the zone of interest. Consequently, two penalties are incurred by UAVs:

$$PEN_i^1[t] = \begin{cases} 0, & \text{if } x_i(t), y_i(t) \in [0, l_{max}], \\ \Upsilon_1, & \text{Otherwise,} \end{cases} \tag{23}$$

$$PEN_i^2[t] = \begin{cases} \Upsilon_2, & \text{if } \ d_i^j[t] < L, \\ 0, & \text{Otherwise.} \end{cases} \tag{24}$$

The penalties Υ_1 and Υ_2 are incurred by each UAV i whenever an action a_i^t would result in crossing the target zone boundaries or violating the safety distance L, respectively. Then, at each time-slot t, all penalties are summed for each UAV i, $P_i^t = PEN_i^1[t] + PEN_i^2[t]$ and incurred from its respective reward R_i^t, $\forall i \in \mathcal{M} \cup \mathcal{N}$. At each time-slot t, each IoT device k selects the adequate UAV-DC$_m$ for data transmission based on the following expression:

$$\Gamma_{k,m}^t[k] = \begin{cases} 1, & m = \text{argmax}_{m \in \mathcal{M}} \{O_k^m[t]\}, \\ 0, & \text{Otherwise.} \end{cases} \tag{25}$$

Specifically, after each movement of UAV-DCs, each device k selects the most suitable UAV-DC$_m$ for data collection, which has the maximum data rate. Otherwise, device k does not transmit its data. Generally, to address the problem of non-stationary environment, we consider a non-stationary multi-agent MDP, where the environment state $s_t \in \mathcal{S}$, the actions of UAVs $a_t \in \mathcal{A}$, the possible state-action combinations $\mathcal{P}_t \subset \mathcal{S} \times \mathcal{A}$, and the rewards $\mathcal{R} : \mathcal{P}_t \to \mathbb{R}$. For a clear understanding, each agent in TEAM continuously interacts with the dynamic IoT environment and observes the positions and energy consumption of UAVs and the AoI, throughput, and energy usage of each IoT device. The values of these parameters enable it to select the adequate actions for its corresponding UAV. This strategy allows UAVs to adapt to the realistic deployment of IoT devices and their dynamics, and maximizes the cumulative rewards across the flight period T.

5) Algorithm: The pseudo-code of TEAM is executed by each agent, which controls the movements of its respective UAV i (*see* Algorithm 1 and Fig. 3).

At the beginning of the algorithm, each UAV i initialize its replay buffer $\mathcal{B}_i$ of size B (Line 2). At Line 3, the critic $Q^i(.)$ and actor $\pi^i(.)$ networks are initialized with their respective weights η^{Q^i} and η^{π^i}. The target critic $Q^{i'}(.)$ and actor $\pi^{i'}(.)$

Algorithm 1: TEAM pseudo-code.

1 **begin**
2 Initialize replay buffer $\mathcal{B}_i$ to capacity B, where ($\mathcal{B}_i = \emptyset$);
3 Randomly initialize actor network $\pi^i(.)$ and critic network $Q^i(.)$ with their respective parameters η^{π^i} and η^{Q^i};
4 Initialize target networks $\pi^{i'}(.)$ and $Q^{i'}(.)$ with weights $\eta^{\pi^{i'}} \leftarrow \eta^{\pi^i}$ and $\eta^{Q^{i'}} \leftarrow \eta^{Q^i}$;
5 $EPS \leftarrow$ Number of episodes;
6 $T \leftarrow$ Number of time-slots;
7 Initialize the action noise ε;
9 **for** $Episode \leftarrow 0, \ldots, EPS$ **do**
10 Initialize the location of UAV i;
11 Initialize observation $o_i^0, \forall i \in \mathcal{M} \cup \mathcal{N}$;
// o_i^0 takes the initial values of all the components in the environment
13 **for** $t \leftarrow 0, \ldots, T$ **do**
14 $a_i^t = \pi^i(o_i^t|\eta^{\pi^i}) + \varepsilon$;
15 **Execute**: action $a_i^t = [\omega_i^t, d_i^t, h_i[t]], \forall i \in \mathcal{M} \cup \mathcal{N}$;
17 **if** $P_i^t > 0$ **then**
18 Cancel action a_i^t of UAV i and update $s_x^{t+1}, x \in \{ET, DC\}$;
19 **foreach** *device* $k \in \mathcal{K}$ **do**
20 **Calculate**: obtain $o_k^m[t]$ based on (9), $\forall m \in \mathcal{M}$;
21 **Define**: get data collection decision $\Gamma_{k,m}^t[k]$ based on (25);
22 **Evaluate**: get reward R_i^t based on (19) or (20) according to the team to which UAV i belongs;
23 **Observe**: obtain a new state $s_x^{t+1}, x \in \{ET, DC\}$;
24 Store transition sample $(s_x^t, a_x^t, R_i^t, s_x^{t+1})$ into experience buffer replay $\mathcal{B}_i$, $x \in \{ET, DC\}$
// Store tuples directly in the experience replay buffer
25 Sample random minibatch of size Δ samples of transitions $(s_x^\delta, a_x^\delta, R_i^\delta, s_x^{\delta+1})$ from $\mathcal{B}_i$;
26 Set target value TGT_δ^i:
27 $TGR_\delta^i = R_i^\delta + \zeta Q^{i'}(s_x^{\delta+1}, \pi^{i'}(s_x^{\delta+1}|\eta^{\pi^{i'}})|\eta^{Q^{i'}})$;
28 Update weight η^{Q^i} of $Q^i(.)$ by minimizing the loss ($L(\eta^{Q^i})$):
29 $Loss(\eta^{Q^i}) = \frac{1}{\Delta}\sum_{\delta=1}^{\Delta}(TGR_\delta^i - Q^i(s_x^\delta, a_x^\delta|\eta^{Q^i}))^2$;
30 Update weight η^{π^i} of $\pi^i(.)$ by:
31 $\nabla_{\eta^{\pi^i}} J(\eta^{\pi^i}) \approx \frac{1}{\Delta}\sum_{\delta=1}^{\Delta} \nabla_{\eta^{\pi^i}} \pi^i(o_i^t|\eta^{\pi^i}) \nabla_{a_i^t} Q^i(s_x^t, a_x^t|\eta^{Q^i})|, \forall i \in \mathcal{M} \cup \mathcal{N}, \forall t \in \mathcal{T}, x \in \{ET, DC\}$
32 Update the corresponding target network weights $\eta^{Q^{i'}}$ of $\eta^{\pi^{i'}}$ by:
33 $\eta^{Q^{i'}} = \chi\eta^{Q^i} + (1-\chi)\eta^{Q^{i'}}$;
34 $\eta^{\pi^{i'}} = \chi\eta^{\pi^i} + (1-\chi)\eta^{\pi^{i'}}$;

networks are initialized while updating their respective weights $\eta^{Q^{i'}}$ and $\eta^{\pi^{i'}}$ (Line 4). Then, the number of episodes and epochs are initialized. The action noise ε is defined, which follows a normal distribution with a variance of 1 and zero mean (Lines 5, 6, and 7).

The second part of the pseudo-code TEAM (Lines 9-23) denotes the training process of TEAM over EPS episodes, each of which consists of T time-slots. In Lines 10 and 11, the environment is initialized where the location and settings of each UAV i and each IoT device k are defined to their initial values. Moreover, each UAV i receives its initial observation o_i^0 ①. At each time-slot t, each UAV i selects a trajectory action a_i^t based on its actor network $\pi^i(o_i^t|\eta^{\pi^i})$. For a better exploration, a random noise parameter ε is added, which decays over time-slots with the rate of 0.9995 ②. Once UAV i executes the action a_i^t, the restrictions (23) and (24) are checked to see if they are satisfied or not. If it not the case, UAV i cancels the action a_i^t and obtains a penalty P_i^t and the state s_x^{t+1} is updated accordingly. Then, for each device k, the most suitable UAV m is selected for data harvesting based on (25). After that, each UAV i receives a reward R_i^t and transit to the next state $s_x^{t+1}, x \in \{ET, DC\}$.

In the third part of TEAM (Lines 24-34), each UAV i collects a transition $(s_x^t, a_x^t, R_i^t, s_x^{t+1})$ of each training epoch, which is stored in its replay buffer $\mathcal{B}_i$ ③. Then, a random mini-batch samples Δ transitions from $\mathcal{B}_i$ to update the actor and critic networks based on four steps ⑤. First, the target value TGT_δ^i is calculated based on the target critic network $Q^{i'}(.)$, where ζ is a discount factor ⑦. Second, the loss function $L(\eta^{Q^i})$ updates the critic network ④. Third, the policy gradient $\nabla_{\eta^{\pi^i}} J(\eta^{\pi^i})$ updates the actor network. Finally, for the sake of stability, the weights $\eta^{Q^{i'}}$ and $\eta^{\pi^{i'}}$ are slowly updated based on the parameter $\chi = 0.001$ (Lines 33-34).

C. Complexity and Convergence Analysis

For the sake of simplicity, each agent's computational complexity in TEAM is mainly related to its neural networks' configuration, where the density of agents is supposed to be linear to the dimension of the input layers (*see* Table IV in Section V-A). To estimate the computational complexity, let $\imath_{Ac,i}$ be the number of the neurons in the ith layer of the actor network. Also, we define the number of neurons in the jth layer of the critic network as $\imath_{Cr,i}$. Therefore, the computational complexity of both networks is expressed as follows:

$$\begin{aligned}\Pi_{cplx} &= \left(2 \times \sum_{i=1}^{\Re-1} \imath_{Ac,i}\imath_{Ac,i+1} + 2 \times \sum_{j=1}^{\Im-1} \imath_{Cr,i}\imath_{Cr,i+1}\right), \\ &= \mathcal{O}\left(\sum_{i=1}^{\Re-1} \imath_{Ac,i}\imath_{Ac,i+1} + \sum_{j=1}^{\Im-1} \imath_{Cr,i}\imath_{Cr,i+1}\right)\end{aligned} \tag{26}$$

where $\Re$ and $\Im$ are the number of fully-connected layers in the actor and critic networks, respectively. Since the networks in Algorithm 1 are trained at the same time and extracting Δ experiences from the replay buffer $\mathcal{B}_i$, the computational complexity of Algorithm 1 is expressed as:

$$\begin{aligned}\Pi_{alg1} = \mathcal{O}\Bigg((N+M) \times T \times EPS \times \Delta \\ \times \left(\sum_{i=1}^{\Re-1} \imath_{Ac,i}\imath_{Ac,i+1} + \sum_{j=1}^{\Im-1} \imath_{Cr,i}\imath_{Cr,i+1}\right)\Bigg),\end{aligned} \tag{27}$$

In the testing phase, each agent uses only its actor network online and therefore the complexity will be $\mathcal{O}\left((N+M) \times T \times \left(\sum_{i=1}^{\Re-1} \imath_{Ac,i}\imath_{Ac,i+1} + \sum_{j=1}^{\Im-1} \imath_{Cr,i}\imath_{Cr,i+1}\right)\right)$. As for the convergence of TEAM, a gradient descent method

is adopted to train critic $Q^i(.)$ and actor $\pi^i(.)$ networks for each agent to update their respective weights η^{Q^n} and η^{π^n} while decaying the learning rates with iterations. After a certain number of iterations, the weights will converge to given values that allow the convergence of TEAM. According to [35] and [36], the theoretical convergence analysis is very complicated to be performed before the training phase. Instead, the readers can observe the convergence of TEAM during the simulation part in Section V-B.

V. Performance Evaluation

In this section, the numerical results are presented and the performance of TEAM algorithm is analyzed. The simulation experiments of TEAM are performed in two phases: (i) Learning phase and (ii) Testing phase. The learning phase is offline centralized training, which can carry out all communications and calculation overheads. This phase studies the convergence of the TEAM algorithm over 2000 episodes compared to other DRL algorithms, such as MADQN [37] and DDQN [38] under the same parameters. After the learning phase, the network parameters are saved for the testing phase that is carried out online. This phase evaluates and compares the performance of TEAM with those of two baseline methods, namely random and greedy. In the random method, each UAV i randomly selects an action, while in the greedy method, each UAV i selects an action that can maximize its own reward R_i^t, all under the constraint of the area boundary. In the next subsections, the simulation settings and the TEAM training configurations are comprehensively presented, followed by the interpretation and discussion of the obtained results in each phase.

A. Simulation setup

To evaluate the performance of TEAM, a set of experiments is conducted using Python 3.6.9 and Tensorflow 1.14.0. A square area of width $l_{max} = 5$ km and a surface of 25 km^2 is considered, which comprises 100 randomly moving IoT devices. It should be stressed that IoT devices are considered as a part of the environment, and their configurations cannot be modified during the learning and testing phases. Table III clearly outlines the different considered simulation parameters.

TABLE III: Simulation Setup.

Parameter	Description	Value
Surface	Area size	25 km^2
l_{max}	Area width	5 km
h_i	Altitude of UAV i	50m–150m
UAV density	Nb. of UAV-ETs and UAV-DCs	2–20
R_n	Transmission power of UAV-ET$_n$	40 dBm
R_k	Transmission power of device k	-20 dBm
F_{max}	Maximum speed of UAVs	20m/s
α	Path loss factor	2
f_c	Carrier frequency	700 MHz
W	Bandwidth	1 MHz
ξ_k	Energy conversion	0.1
η_0	Reference channel gain	-30 dB
σ^2	Noise power	-100 dBm
Z_k^{min}	Status-update size	10 bits

TEAM algorithm is trained over 2000 episodes with 100 steps each. Four fully-connected hidden layers are defined in both actor and critic networks, which comprises in the order 400, 400, 300, and 300 neurons. Each neuron uses Rectified Linear Unit (ReLU) as an activation function. In addition, Hyperbolic tangent (tanh) is used as an activation function in the output layer of the actor-network to limit the movement of each UAV according to its maximum travel distance. The input of each critic network is represented as a concatenation of actions and observations, and its output is a scalar that assesses the states according to the global policy. The parameters of the learning phase can be found in Table IV.

TABLE IV: Parameters of TEAM.

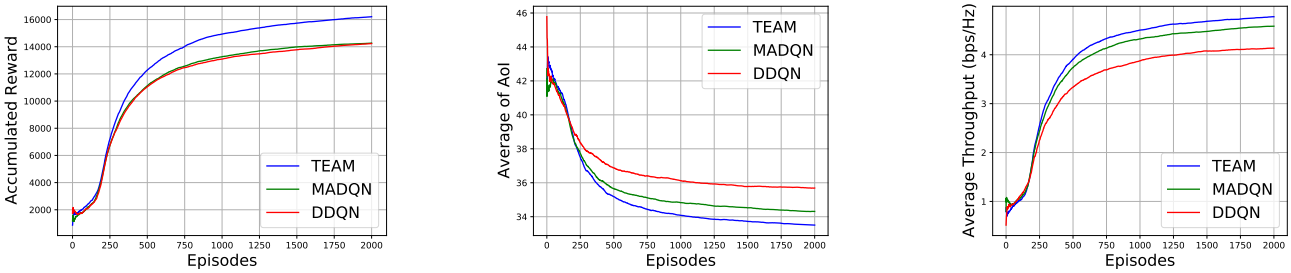
Parameters of actor neural network			
Layers	**Number**	**Size**	**Act. Function**
Input	1	$M + N + 4K + 1$	–
Hidden	4	400, 400, 300, 300	ReLU
Output	1	3	Tanh
Parameters of critic neural network			
Layers	**Number**	**Size**	**Act. Functions**
Input	1	$(M + N) \times (M + N + 4K + 4)$	–
Hidden	4	400, 400, 300, 300	ReLU
Output	1	1	–
Key parameters of the training stage			
Parameter		**Value**	
Memory size $\mathcal{B}$		10^5	
Mini-batch size U		256	
Actor learning rate		0.0002	
Critic learning rate		0.0001	
Optimizer method		Adam	
Updating steps		1000	
Reward discount, ζ		0.99	
Υ_1, Υ_2		10.0	
RL Comparisons		MADQN, Double DQN	

The simulation setup, as well as the parameters of TEAM, are selected from the evaluations of related and relevant UAV-enabled WPCN and MADRL-based solutions [5], [8]. Furthermore, all these parameters are validated through repetitive experiments, which help us select the adequate ones.

B. Learning phase

The training curve of TEAM in Fig. 5 is obtained by deploying 8 UAVs (*i.e.,* 4 UAV-ETs and 4 UAV-DCs) to serve 100 devices. It is clearly shown that the obtained reward for each episode remains under 2.5×10^6 at the beginning and starts increasing from the $100th$ episode until convergence. Indeed, in the beginning, each agent calculates random actions to explore the IoT environment and its dynamics. Then, our TEAM model is trained using all the experiences learned from this step to optimally serve dynamic IoT devices with continuous action space. These two steps allow TEAM agents to avoid the different penalties and slowly optimize the placements of their corresponding UAVs. This can significantly increase the rewards obtained by each agent until convergence. It should be stressed that due to the non-stationary environment, the rewards vary around their average while overall increasing with more learning.

Under the same density of UAVs and devices, the convergence of TEAM is evaluated in terms of accumulated reward, average AoI of devices, and average throughput. Overall, it has been distinguished that the learning phase



(a) Cumulative reward. (b) Average AoI of devices. (c) Average throughput.

Fig. 4: Performance comparisons over episodes (Nb. of UAV-DCs=4, Nb. of UAV-ETs=4, Nb. of IoT devices=100).

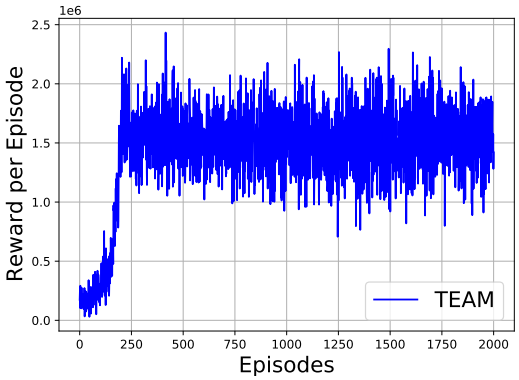
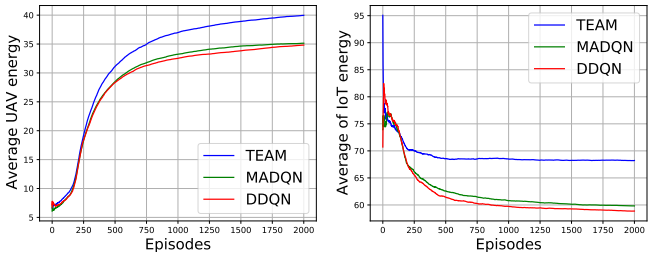


Fig. 5: Reward per episode in TEAM (Nb. of UAV-DCs=4, Nb. of UAV-ETs=4, and Nb.of IoT devices=100).

has converged at about 1000-1500 episodes, and it remains approximately stable afterward (*c.f.,* Fig. 4). For instance, it is observed from Fig. 4(a) that in the first 750 episodes, the obtained rewards are not stable and remain at their lowest levels. Then, the rewards increase with the number of episodes and never decrease after. This is because, during the learning phase, UAVs are moving randomly over the dynamic devices without considering neither their energy capacity nor their mobility, and thus penalties are severely incurred by UAVs. After a certain number of learning episodes, UAVs learn and try to synchronize with each other (UAV-ETs and UAV-DCs) so that they can play their role at the appropriate time. Next, in Fig. 4(b), the average of AoI is analyzed. It is clearly seen that TEAM achieves the best performance, followed by MADQN and DDQN. This is due to the optimal policy built by TEAM to control the trajectory of UAVs and, more particularly, ensures that the devices are fully charged with energy and ready for uploading their updates to UAV-DCs. In Fig. 4(c), it is distinguished that the average throughput increases very rapidly at the outset of the training step (until 750 episodes), and then the rise would be comparatively sluggish. This is explained by the fact that more and more devices will be served at each time by UAV-DCs. It should be stressed that in Fig. 4, UAVs in MADQN cannot perform continuous actions, and thus slow convergence and not good performances as in TEAM. Moreover, the actions made by UAVs in DDQN are performed through a single agent, and therefore it delays in providing appropriate UAV actions.



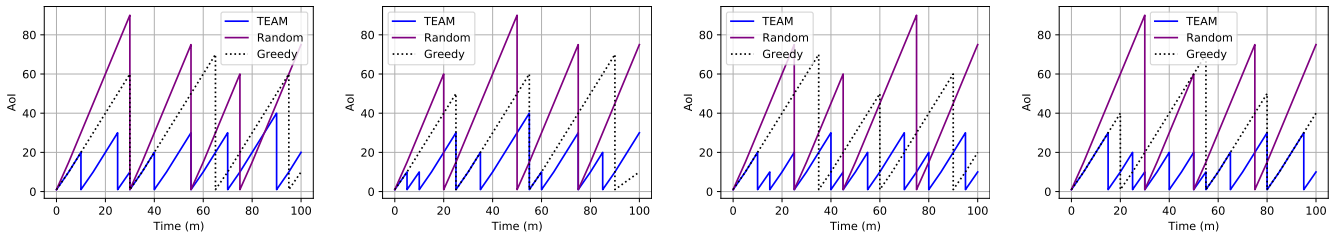
(a) Residual energy of UAVs. (b) Residual energy of devices.

Fig. 6: Residual energy of IoT devices and UAVs (Nb. of UAV-DCs=4, Nb. of UAV-ETs=4, Nb. of IoT devices=100).

In Fig. 6, the energy consumption of IoT devices and UAVs is calculated under the same number of episodes. Indeed, as expected, we distinguish that TEAM in both figures outperforms MADQN and DDQN. This is because TEAM has quickly built an optimal policy compared to MADQN and DDQN, which allows controlling the movements of UAVs according to both the mobility of IoT devices and the energy level (*i.e.,* residual energy) of each UAV (*see* Fig. 6(a)). As for Fig. 6(b), it is clearly observed that TEAM preserves the residual energy of devices up to 9% better than MADQN and DDQN. This is because UAV-ETs in TEAM learn faster to place themselves in the right places where IoT devices need energy.

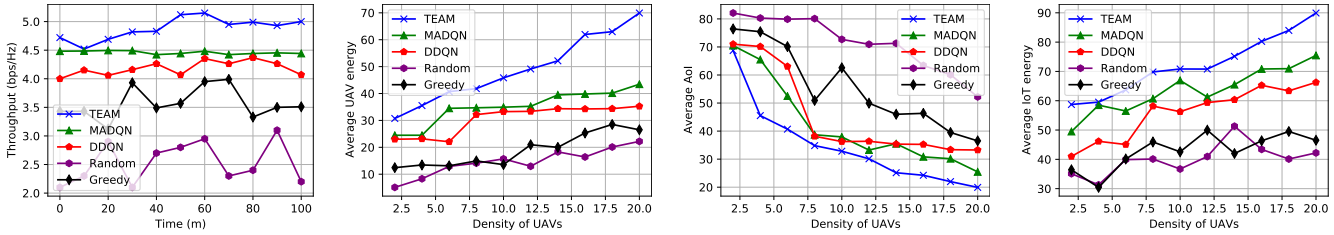
C. Testing phase

At a first step, four IoT devices are randomly selected, and the AoI evolution is measured for each of them according to the different considered policies (*c.f.,* Fig. 7). Overall, it is noticed that by adopting TEAM, the real-time AoI of the four devices is noticeably smaller than that of the baseline techniques. As explained above, TEAM quickly learns the dynamics of the environment and builds an optimal policy to allow UAVs to fly closer to IoT devices, harvest updates from them, and supply them with energy when needed. Regarding the baseline methods, particularly the random method, it is noticed that the AoI of the four devices is much larger than the AoI obtained by TEAM, which is due to the fact that



(a) Device 1. (b) Device 2. (c) Device 3. (d) Device 4.

Fig. 7: The evolution of AoI of four IoT devices based on different policies.

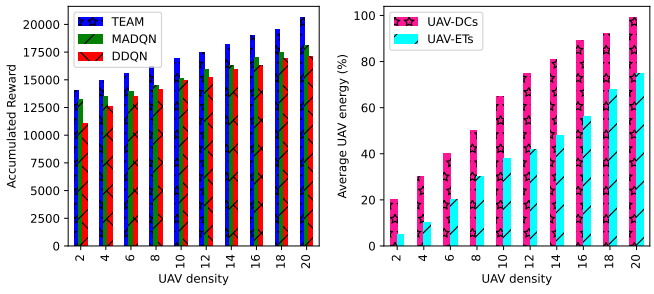


(a) Throughput of devices. (b) Energy of UAVs. (c) AoI of devices. (d) Energy of IoT devices.

Fig. 8: The performance of TEAM in terms of different metrics.

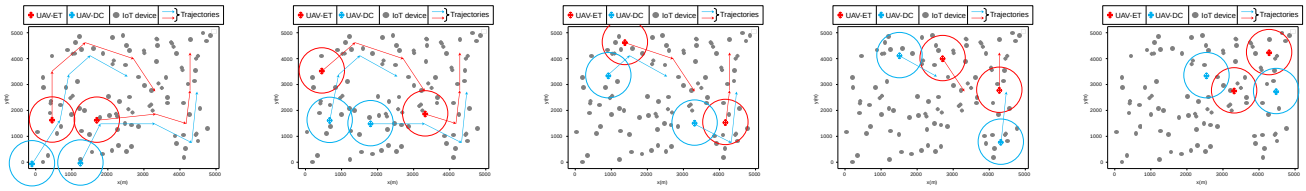
UAVs find it challenging to adapt to the dynamics of devices. As for the greedy method, it is observed that the obtained AoI is reduced than that of the random method, which is explained by the selection of the device with higher AoI at each time-slot t to increase the reward. However, the greedy method is not effective as TEAM because of the long flight period carried out by UAVs to find the appropriate UAVs.

At a second step, the average throughput is studied during the whole flight period (*see* Fig. 8(a)). It is clearly observed that TEAM has significantly optimized the average throughput compared to other policies. To explain this observation, two reasons can be provided. First, based on the built optimal policy, TEAM allows UAV-DCs to serve more IoT devices, *i.e.,* more time-slots will be available in which devices can take advantage of the channel gain. Second, UAV-ETs inspect as many devices as possible and prepare them for an eventual service of UAV-DCs. In Fig. 8(b), the effect of UAV density on UAV energy consumption (Residual energy on UAVs) is carefully studied. Overall, TEAM outperforms the other algorithms in terms of energy utilization (*i.e.,* average of energy levels) for each density of UAVs, representing an enhancement of 25% over DRL algorithms and more than 40% over baseline methods. It is because TEAM has quickly learned the dynamics of the environment, and therefore UAVs can quickly find the appropriate places to maximize their rewards and minimize their mobility. Moreover, it is also distinguished that the energy consumption is high at low densities of UAVs, which is due to the constant movements of UAVs looking to serve the maximum number of IoT devices. However, at high densities of UAVs, the number of IoT devices to serve is getting weaker and weaker, which considerably minimizes the movements of UAVs, and thus their energy consumption. As for the baseline methods, at any density, UAVs are permanently moving either randomly or looking for a more accumulating reward, which significantly increases the energy consumption. In general, Fig. 8(c) shows that with the increased density of UAVs, the average AoI becomes lower and lower. This is because the density of UAVs defines how devices have been served, *i.e.,* by increasing the number of UAVs, more devices will be served either by UAV-ETs (energy) or by UAV-DCs (data harvesting). Furthermore, the performance gap between the different policies demonstrates the efficiency of the speed of the UAV trajectory optimization deployed by each policy with the increasing density of UAVs. The average residual energy of devices is another performance metric, which has been evaluated and depicted in Fig. 8(d). The average residual energy of devices is calculated at each flight period by updating the number of UAVs at each time. Clearly, TEAM increases the embedded energy in IoT devices compared to other policies, which is explained by the provided optimal trajectories of UAV-ETs to charge IoT devices directly after uploading their updates. Consequently, TEAM outperforms the other DRL approaches and baseline methods due to its high adaptability to the environment and optimal trajectories provided for UAVs, respectively.



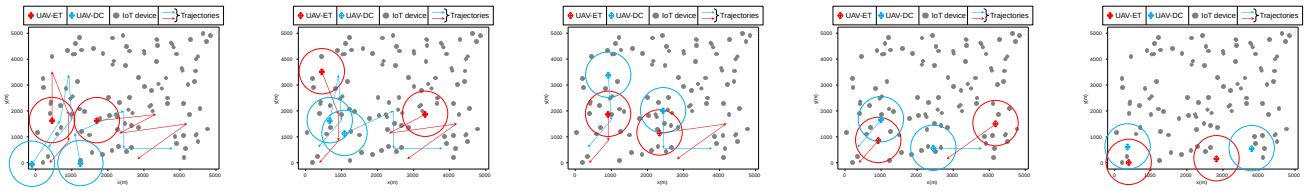
(a) System performance. (b) Residual energy of UAVs.

Fig. 9: System performance and energy efficiency of TEAM (Nb. of UAV-DCs=Nb. of UAV-ETs=$\frac{UAV\ \ Density}{2}$).



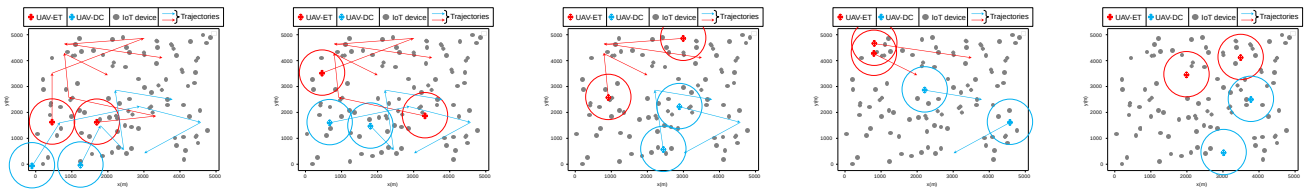
(a) Step=1. (b) Step=2. (c) Step=3. (d) Step=4. (e) Step=5.

Fig. 10: Paths obtained by the TEAM method for a scenario of 2 UAV-ETs and 2 UAV-DCs serving 100 IoT devices.



(a) Step=1. (b) Step=2. (c) Step=3. (d) Step=4. (e) Step=5.

Fig. 11: Paths obtained by the Greedy method for a scenario of 2 UAV-ETs and 2 UAV-DCs serving 100 IoT devices.



(a) Step=1. (b) Step=2. (c) Step=3. (d) Step=4. (e) Step=5.

Fig. 12: Paths obtained by the Random method for a scenario of 2 UAV-ETs and 2 UAV-DCs serving 100 IoT devices.

From Figs. 9, the system performance in terms of accumulated rewards and UAV energy levels are increasing as the density of UAVs increases because the number of unserved IoT devices is getting weaker and weaker. Fig. 9(a) shows the system performance in terms of accumulated rewards. Generally, TEAM has optimally increased the accumulated rewards compared to other methods, whatever the density of UAV-DCs and UAV-ETs. This is because TEAM is mainly based on a continuous control mechanism to adapt to the dynamics of the IoT environment. In Fig. 9(b), we clearly distinguish that the residual energy levels of UAV-ETs are less than those of UAV-DCs. It is due to the WET process that consumes more energy from their embedded batteries.

D. Trajectory analysis

Figs. 10, 11, and 12 show the trajectories of two UAVs in both teams, which are provided by TEAM framework, Random method, and Greedy method. The evaluation is carried out during five time-steps where UAVs are serving 100 IoT devices. In Fig. 10, it is clearly demonstrated that TEAM framework prompts UAVs in both teams to cooperate to optimally serve the IoT devices. Indeed, UAV-DCs are always following UAV-ETs that ensure the powering of IoT devices and make them fully charged with energy and ready for uploading their updates to UAV-DCs. Moreover, we distinguished that the trajectories provided by TEAM framework have no collisions and very low interference to IoT devices. However, it is not the case of Random and Greedy methods (*see* Figs. 12 and 11), where UAVs in both teams are not synchronized and cannot optimally serve IoT devices. This can significantly increase the AoI of IoT devices and cause an excessive energy consumption of UAVs.

VI. Conclusion

In this work, a multi-UAV system is deployed in which two teams of UAVs are cooperatively dispatched to behave as data collectors and energy transmitters to supply a large scale of dynamic IoT devices. This work aims to jointly minimize the AoI of IoT devices and maximize their throughput while reducing the energy utilization of UAVs. To do so, the trajectories and resource allocation of UAVs are optimized by considering the dynamics of the IoT environment. All the more explicitly, the optimization problem was formulated as a non-convex mixed-integer program that turned out to be challenging to solve straightforward. Therefore, a MADRL-based method, called TEAM, is proposed to address the trajectory design issue of UAVs and learn the dynamicity of the IoT environment. Numerical results show that TEAM has a significant performance gain over the compared benchmark algorithms and baseline mechanisms.

In future work, several approaches could be generated from this work. For example, TEAM framework could be split into two parts, each dedicated to a specific task (*i.e.,* energy charging or data collection) using the appropriate

technologies. Moreover, the TEAM strategy could be exploited in mobile edge computing scenarios assisted by multiple UAVs to support the computing task offloading of IoT devices while charging those running out of energy. Also, UAV-ETs could be exploited to serve not only IoT devices, but also UAV-DCs. These are just a few proposals that require further efforts and adaptation.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C1008589)

REFERENCES

[1] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *Journal of Network and Computer Applications*, vol. 166, p. 102706, 2020.

[2] O. S. Oubbati, M. Atiquzzaman, A. Baz, H. Alhakami, and J. Ben-Othman, "Dispatch of uavs for urban vehicular networks: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 174–13 189, 2021.

[3] O. S. Oubbati, M. Atiquzzaman, A. Lakas, A. Baz, H. Alhakami, and W. Alhakami, "Multi-UAV-enabled AoI-aware WPCN: A Multi-agent Reinforcement Learning Strategy," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2021, pp. 1–6.

[4] C. Zhou, H. He, P. Yang, F. Lyu, W. Wu, N. Cheng, and X. Shen, "Deep RL-based trajectory planning for AoI minimization in UAV-assisted IoT," in *Proceedings of the 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2019, pp. 1–6.

[5] M. Shokry, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of Information Aware Trajectory Planning of UAVs in Intelligent Transportation Systems: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 1939–9359, 2020.

[6] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy efficient UAV communication with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1913–1927, 2019.

[7] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered Public Safety IoT: A UAV-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.

[8] Z. Chen, K. Chi, K. Zheng, G. Dai, and Q. Shao, "Minimization of Transmission Completion Time in UAV-Enabled Wireless Powered Communication Networks," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1245–1259, 2019.

[9] B. Zhao, J. Liu, Z. Wei, and I. You, "A deep reinforcement learning based approach for energy-efficient channel allocation in satellite Internet of things," *IEEE Access*, vol. 8, pp. 62 197–62 206, 2020.

[10] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-Efficient Data Collection and Device Positioning in UAV-Assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, 2019.

[11] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[12] M. Yi, X. Wang, J. Liu, Y. Zhang, and B. Bai, "Deep Reinforcement Learning for Fresh Data Collection in UAV-assisted IoT Networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 1–6.

[13] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "Age of Information Aware Trajectory Planning of UAVs in Intelligent Transportation Systems: A Deep Learning Approach," *IEEE Transactions on Vehicular Technology*, 2020.

[14] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K.-K. Wong, "Minimum Throughput Maximization for Multi-UAV Enabled WPCN: A Deep Reinforcement Learning Method," *IEEE Access*, vol. 8, pp. 9124–9132, 2020.

[15] B. Liu, H. Xu, and X. Zhou, "Resource allocation in unmanned aerial vehicle (UAV)-assisted wireless-powered Internet of Things," *Sensors*, vol. 19, no. 8, p. 1908, 2019.

[16] Q. Wu, J. Xu, Y. Zeng, D. W. K. Ng, N. Al-Dhahir, R. Schober, and A. L. Swindlehurst, "A Comprehensive Overview on 5G-and-Beyond Networks with UAVs: From Communications to Sensing and Intelligence," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 2912–2945, 2021.

[17] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2019.

[18] D. Orfanus, E. P. De Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military UAV relay networks," *IEEE Communications letters*, vol. 20, no. 4, pp. 804–807, 2016.

[19] A. Bensalem and D. E. Boubiche, "EBEESU: ElectriBio-inspired energy-efficient self-organization model for unmanned aerial ad-hoc network," *Ad Hoc Networks*, vol. 107, p. 102236, 2020.

[20] Z. Guan, N. Cen, T. Melodia, and S. M. Pudlewski, "Distributed joint power, association and flight control for massive-MIMO self-organizing flying drones," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1491–1505, 2020.

[21] J. Park, H. Lee, S. Eom, and I. Lee, "UAV-aided wireless powered communication networks: Trajectory optimization and resource allocation for minimum throughput maximization," *IEEE Access*, vol. 7, pp. 134 978–134 991, 2019.

[22] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2233–2246, 2018.

[23] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, and X.-G. Xia, "3-D beamforming for flexible coverage in millimeter-wave UAV communications," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 837–840, 2019.

[24] S. Lee and R. Zhang, "Distributed wireless power transfer with energy feedback," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1685–1699, 2016.

[25] Y. Liu, H.-N. Dai, H. Wang, M. Imran, X. Wang, and M. Shoaib, "UAV-enabled data acquisition scheme with directional wireless energy transfer for Internet of Things," *Computer Communications*, vol. 155, pp. 184–196, 2020.

[26] M. Badi, S. Gupta, D. Rajan, and J. Camp, "Characterization of the Human Body Impact on UAV-to-Ground Channels at Ultra-low Altitudes," *IEEE Transactions on Vehicular Technology*, 2021.

[27] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[28] K. Xiong, Y. Zhang, P. Fan, H.-C. Yang, and X. Zhou, "Mobile service amount based link scheduling for high-mobility cooperative vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9521–9533, 2017.

[29] R. Zhang, X. Pang, W. Lu, N. Zhao, Y. Chen, and D. Niyato, "Dual-UAV Enabled Secure Data Collection With Propulsion Limitation," *IEEE Transactions on Wireless Communications*, 2021.

[30] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[31] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[32] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *arXiv preprint arXiv:1509.06461*, 2015.

[33] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of ICML*, 2014, pp. 1–9.

[34] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.

[35] F. Wu, H. Zhang, J. Wu, and L. Song, "Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4175–4189, 2020.

[36] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019.

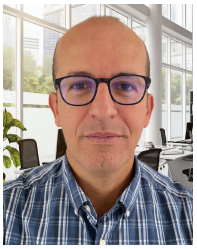
[37] M. Egorov, "Multi-agent deep reinforcement learning," *CS231n: Convolutional Neural Networks for Visual Recognition*, 2016.

[38] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2094–2100.

Omar Sami Oubbati is an Associate Professor at the University Gustave Eiffel in the region of Paris, France. He is a member of the Gaspard Monge Computer Science laboratory (LIGM CNRS UMR 8049). He received his degree of Engineer (2010), M.Sc. in Computer Engineering (2011), M.Sc. degree (2014), and a PhD in Computer Science (2018), all from University of Laghouat, Algeria. From Oct. 2016 to Oct. 2017, he was a Visiting PhD Student with the Laboratory of Computer Science, University of Avignon, France. He spent 6 years as an Assistant Professor at the Electronics department, University of Laghouat, Algeria and a Research Assistant in the Computer Science and Mathematics Lab (LIM) at the same university. His main research interests are in Flying and Vehicular ad hoc networks, Energy harvesting and Mobile Edge Computing, Energy efficiency and Internet of Things (IoT). He is the recipient of the 2019 Best Survey Paper for Vehicular Communications (Elsevier). He has actively served as a reviewer for flagship IEEE Transactions journals and conferences, and participated as a Technical Program Committee Member for a variety of international conferences, such as IEEE ICC, IEEE CCNC, IEEE ICCCN, IEEE WCNC, IEEE ICAEE, and IEEE ICAIT. He serves on the editorial board of Vehicular Communications Journal of Elsevier and Communications Networks Journal of Frontiersin. He has also served as guest editor for a number of international journals. He is a member of the IEEE and IEEE Communications Society.

Mohammed Atiquzzaman received the M.S. and Ph.D. degrees in electrical engineering and electronics from the University of Manchester, U.K., in 1984 and 1987, respectively. He currently holds the Edith J. Kinney Gaylord Presidential Professorship with the School of Computer Science, University of Oklahoma, USA. His research has been funded by the National Science Foundation, National Aeronautics and Space Administration, U.S. Air Force, Cisco, and Honeywell. He coauthored Performance of TCP/IP Over ATM Networks and has authored more than 300 refereed publications. His current research interests include areas of transport protocols, wireless and mobile networks, ad hoc networks, satellite networks, power-aware networking, and optical communications. He Co-Chaired the IEEE High Performance Switching and Routing Symposium (2003, 2011), IEEE GLOBECOM and ICC (2014, 2012, 2010, 2009, 2007, and 2006), IEEE VTC (2013), and SPIE Quality of Service Over Next Generation Data Networks conferences (2001, 2002, and 2003). He was the Panels Co-Chair of INFOCOM'05, and has been on the program committee of many conferences, such as INFOCOM, GLOBECOM, ICCCN, ICCIT, Local Computer Networks, and serves on the review panels at the National Science Foundation. He was the Chair of the IEEE Communication Society Technical Committee on Communications Switching and Routing. He received the IEEE Communication Society's Fred W. Ellersick Prize and the NASA Group Achievement Award for outstanding work to further NASA Glenn Research Center's efforts in the area of the Advanced Communications/Air Traffic Management's Fiber Optic Signal Distribution for Aeronautical Communications project. He received from IEEE the 2018 Satellite and Space Communications Technical Recognition Award for valuable contributions to the Satellite and Space Communications scientific community. He also received the 2017 Distinguished Technical Achievement Award from IEEE Communications Society in recognition of outstanding technical contributions and services in the area of communications switching and routing. He is the Editor in Chief of Journal of Networks and Computer Applications, the founding Editor in Chief of Vehicular Communications, and serves served on the editorial boards of many journals, including IEEE Communications Magazine, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, Real Time Imaging Journal, Journal of Sensor Networks, and International Journal of Communication Systems.

Hyotaek Lim received his B.S. degree in computer science from Hongik University in 1988, the M.S. degree in computer science from POSTECH and the Ph.D. degree in computer science from Yonsei University in 1992 and 1997, respectively. From 1988 to 1994, he had worked for Electronics and Telecommunications Research Institute as a research staff. Since 1994, he has been with Dongseo University, Korea, where he is currently a professor in the Department of Computer Engineering. His research interests include ubiquitous and mobile networking, cloud computing, and storage area networks.



Abderrezak Rachedi is currently working as full professor (Professeur des Universités) at University Paris-Est Marne-la-Vallée (UPEM) since Sep. 2018. He was associate professor (maître de confèrences) between 2009 and 2018 at the same university. He received his Habilitation to Direct Researsh (HDR: habilitation à diriger des recherches) from Paris-Est University in Dec. 2015, and his PhD degree in computer science in Nov. 2008 from the University of Avignon in France. He received his research MS degree in computer science from the University of Savoie in France in 2003, and his engineer degree in computer science from the University of technology and science Houari Boumedienne -USTHB-(Algiers, Algeria) in 2002. His research interests lie in the field of wireless networking, wireless multi-hop networks, Internet of Things (IoT), wireless sensor networks (WSNs), Internet of Vehicles (IoV), Vehicular communication (V2X), quality of services (QoS) with security, Trust models design, Blockchain, Network performance analysis and evaluation. Prof. Rachedi advised multiple Ph.D. and Master's students at Paris-Est University. So far, his research efforts have culminated in more than hundred refereed journal, conference and book publications in a wide variety of prestigious international conferences and journals including the IEEE Transactions on Vehicular Technology (IEEE TVT), Elsevier Ad hoc networks, IEEE ICC, and IEEE GLOBECOM. He participated or still participates to several national and international research and industrial projects. He is co-founder of IT startup "uGetWin" in Feb. 2021 which relies on the results of research work around the Internet of Things, artificial intelligence and digital twins (3 international patents and 3 software). He has served on the editorial board for IEEE ACCESS journal, IEEE Internet of Things, IEEE Journal on Selected Areas in Communications: Series on Network Softwarization & Enablers, John Wiley's International of Communication Systems (IJCS), John Wiley's Security & Privacy, Wireless Communications and Mobile Computing Journal. He is a senior member of the IEEE and has served as Technical Program Committee member and reviewer of many international research projects, journals and conferences.

Abderrahmane Lakas received his MS (1990) and PhD (1996) in Computer Systems from the University of Paris VI, Paris, France. He joined the College of Information Technology, UAE University in 2003. He is teaching various courses on computer networks and network security. Dr. Lakas had many years of industrial experience holding various technical positions in telecommunication companies such as Netrake (Plano, Texas, 2002), Nortel (Ottawa, 2000) and Newbridge (Ottawa, 1998). He spent two years (94-96) as a Research Associate at the University of Lancaster, UK. Dr. Lakas has been conducting research in the areas of network design and performance, voice over IP, quality of service and wireless networks. He is a senior member of the IEEE and IEEE Communications Society. Dr. Lakas is in the editorial board of Journal of Communications (Actapress), and Journal of Computer Systems, Networks, and Communications (Hindawi). He is serving on the technical program committees of many international conferences GLOBCOM, ICC, VTC, etc.