Reinforcement Learning for Resource Planning in Drone-Based Softwarized Networks

Laura Galluccio*, Christian Grasso*, Guido Maier†, Raoul Raftopoulos*,
Marco Savi‡, Giovanni Schembra*, Sebastian Troia†
*Department of Electrical, Electronic and Computer Engineering, University of Catania, Catania, 95125, Italy
†Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, 20133, Italy
‡Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, 20125, Italy

***Abstract*—In the last few years, some research papers have proposed usage of UAVs organized in Flying Ad-Hoc Network (FANET) in remote areas with a poor or completely non-existent structured network to support novel application scenarios in which data generated by the end users must be processed on site with ultra-low latency. This way, a FANET can be seen as a provider of services and/or slices for extreme-edge 5G networks for delay-sensitive applications. However, keeping a FANET available and active is an ongoing challenge as the autonomy of UAVs is limited and strongly influenced by power consumption of both the engines and the computing element (CE) where the application functions are executed as virtual machines. In this paper, we present an optimization framework capable of increasing the overall duration of the FANET. To this purpose, we apply Reinforcement Learning (RL) based on Double Deep Q-learning (DDQN) to optimize the percentage of available CPU resources for Virtual Function virtualization, and Integer Linear Programming (ILP) to optimize VF placement inside the active UAVs of the FANET.**

***Index Terms*—NFV, Virtual Function Placement, FANET, Resource Allocation, Reinforcement Learning.**

I. Introduction

Use of Unmanned Aerial Vehicles (UAVs) for support and extension of traditional wireless networks has recently gained a great momentum. Indeed, thanks to mobility and low cost of UAVs, a plethora of applications have emerged during the past few decades. Historically, UAVs have been primarily used in the military field, mainly deployed in hostile territories, to reduce human losses [1]. As the cost and size of UAV devices continue to decrease, small UAVs are now more easily accessible to the public; hence, numerous new applications have emerged in the civil and commercial domains, with typical examples including search and rescue operations during environmental disasters, custom border surveillance, support for the management of city and forest fires, road traffic monitoring and others [2] [3] [4] [5] [6] [7].

In particular, UAVs have been proficuously used to support novel application scenarios of emerging 5G networks, concerning areas with a poor or completely non-existent structured network, for which the data generated by end users must be sent to a cloud far from the place where they have been produced. In other words, UAVs can be exploited to extend 5G networks with a fleet of UAVs organized in a Flying Ad hoc Network (FANET) [8] [9] [10]

The design of the communication architecture in a FANET for UAVs is strictly linked to the specific application. This is mainly due to the rapid change in network topology and the corresponding routing protocol convergence problems, i.e. routing table updates must be sophisticated enough to cope with rapid topology changes. Also, the use of a peer-to-peer network topology is needed to support collaborative coordination among peers, so as to provide services and functions to be delivered to on-demand users and devices on the ground. As such, FANETs can be seen as providers of services and/or slices for 5G networks. Network slices are end-to-end logical networks with computing, storage and networking capabilities, deployed over a common physical network infrastructure, and used to run 5G services, such as enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), and massive machine type communications (mMTC).

In this light, FANETs can provide computing capabilities in the form of Virtual Functions (VF) to guarantee diversified service requirements within 5G network slices. An example is represented in [11] where a FANET was exploited to support video monitoring applications in wide rural areas not covered by Internet access.

However, keeping a FANET available and active is an ongoing challenge as the autonomy of its UAVs is limited [12]. Each UAV is equipped with two main elements: a limited battery and a computing element (CE). The latter aims at running one or more VFs for 5G services delivered to end users on the ground.

The total power consumed by the CE is comparable with the one consumed by the engines [13] [14] [15] and consists of two main contributions: 1) a constant power to keep the CE active and 2) a variable power depending on the amount of data processed by each VF. The latter is expressed in terms of CPU usage.

Therefore, the minimization of the power consumption represents the main challenge to be addressed. The more resources the CE consumes, the faster the battery runs out, thus reducing the overall FANET service availability.

Specifically, when the battery charge of a UAV is below a certain threshold, the UAV must temporarily leave the FANET to reach the nearest charging station. During this period, the VFs that were running on that UAV need to be placed within the remaining UAVs, causing an increase of their

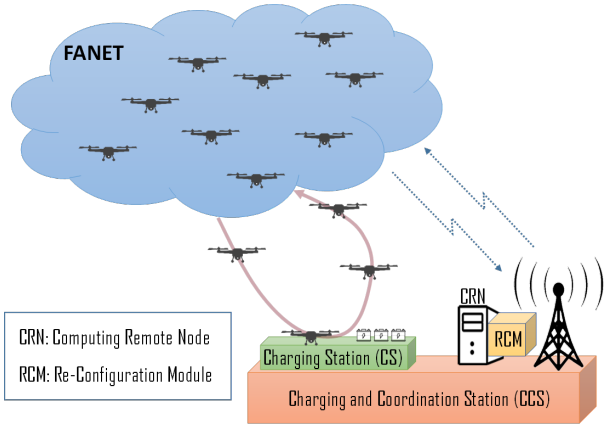

Fig. 1: System Description

energy consumption and the consequent reduction of the flight duration. If the number of flying UAVs is not sufficient, the FANET could not be able to deliver services to the ground devices.

In this paper we present an optimization framework capable of increasing the overall duration of the FANET. To do this, we act on two fronts: the first, in the long-term, is to optimize the percentage of available CPU resources for VF computing; the second, in the short-term, consists in optimizing the VF placement inside the active UAVs of the FANET.

The proposed optimization framework uses a Reinforcement Learning (RL) approach, based on Double Deep Q-learning (DDQN) [16], with goal of setting the amount of available CPU, and an Integer Linear Programming (ILP) algorithm to optimize the VF placement.

The rest of the paper is organized as follows. Section II gives an overall description of the system, specifically of the considered scenario and the functional architecture of the main elements. Then, in Section III, we introduce the proposed orchestration framework, describing the long-term FANET behavior optimization, which is based on RL, and the short-term VF placement optimization. Section IV proposes a use case for performance evaluation. Finally, Section V draws conclusions.

II. System Description

A. Scenario and high-level system description

The system we consider in this paper is depicted in Fig. 1. We consider a fleet of 5G-enabled Unmanned Aerial Vehicles (UAVs) that is used to create a *Flying Ad-hoc Network* (FANET). The FANET is deployed in support to Application Services (ASs) that are dynamically requested by users in a given ground area, and typically spans up to few squared kilometers. More specifically, each UAV acts as an NFV Infrastructure Point-of-Presence (NFV-PoP), that is, as a *micro-datacenter* where both virtual network functions and virtual application functions (related to given ASs) are executed as Virtualized Functions (VF). Each UAV can be interconnected to other UAVs and to users by means of standalone 5G technology [17]. This way, each UAV provides application flows generated by ground devices with the required VFs. To this purpose, each VF has to be run on a UAV to manage the aggregated flow coming from ground (specifically, from the devices that require it) taking into account the performance requirements specified for that VF.

Being the battery charge of UAVs limited, a *Charging and Coordination Station* (CCS) is assumed to be deployed on the ground, within a range of few kilometers from the FANET. When the battery charge of a UAV is below a given threshold, the UAV has to temporarily leave the FANET and fly to the CCS for recharging. Every time a UAV leaves the FANET, the FANET has to be re-configured, meaning that the VFs currently executed by that UAV must be migrated to other UAVs, with the goal of guaranteeing service continuity to the users. However, if not done appropriately, such a re-configuration may impact on battery consumption and, consequently, on flight duration of the UAVs involved in VF migration, with a potential impossibility for the FANET to guarantee service continuity if too few UAVs are present in the FANET at the same time. A *energy-aware FANET re-configuration* needs thus to be performed. A *Re-Configuration Module* (RCM) that runs on a dedicated computing node placed in the CCS and named *Computing Remote Node* (CRN) is in charge of this task.

CCS-UAVs communication is guaranteed by the most appropriate wireless network technology, such as a *Private 5G Network* [18] or a *Low-Power Wide-Area Network* (LPWAN) (e.g. based on LoRaWAN) [19]. The latter case is preferable from a UAV energy consumption minimization perspective, but *(i)* it requires that each UAV is equipped with a LoRaWAN TX/RX module and *(ii)* the available bandwidth is limited. The former case is more energy-hungry but does not need any additional TX/RX hardware, as the already-available 5G antenna is enough to enable the UAV-CCS communication. A thorough analysis aimed at identifying the most appropriate solution is out the scope of this paper, and strongly depends on the considered context.

B. Functional architecture

The main architectural elements of the system, as shown in Fig. 1, are the UAVs and the CCS. Their functional architecture and components are described in the following.

a) Unmanned Aerial Vehicle: The four main components of any UAV are (see Fig. 2) the *Engines*, the *Computing Element*, the *TX/RX Module* and the *Battery*. A *UAV Local Manager* is in charge of coordinating the behavior of the above modules, as specified below.

The *Battery* is used to supply all the three former modules. Its current charge level is communicated to the UAV Local Manager such that it can take its decisions regarding the amount of CE computing power using and when landing for battery recharging.

The Computing Element (CE) is the core component of the UAV, since it ensures that the UAV can act as a NFVI-PoP, being able to execute VFs inside Virtual Machines (VM). For

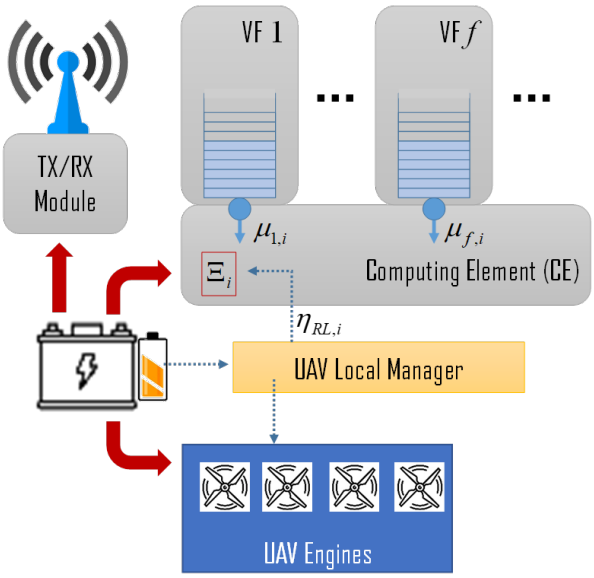

Fig. 2: Functional architecture of a UAV

this reason, a hypervisor (e.g. CentOS) is installed on each CE to provide VMs hosting VFs with a virtualization of the underlying hardware resources.

The *TX/RX Module* includes a 5G antenna, and may also include a LoRaWAN module as specified in the previous subsection. Since power consumption of this module is few hundred mW, while power consumption of engines and CE is dozens of Watt, we can neglect the former, considering in the sequel only the latter in the overall power load of the UAV Battery.

As already pointed out in previous work [13] [14] [15], the amount of energy consumed by the CE is comparable with that consumed by the Engines. For this reason, a careful optimization of the *CE power management* and *VF placement* (i.e., FANET (re)-configuration) is needed to avoid an unsustainable reduction of flight duration. This task is one of the main duties of the CCS. In addition, it is needed that FANET management guarantees that the number of UAVs does not decrease too much in order to limit the possibility of not being able to guarantee the service to the requesting flows.

The UAV Local Manager of each UAV interacts with the CCS at each re-configuration instance by sending the current level of battery of all the UAVs that are active in the FANET, and receiving updates on both the computation power to be set in the CE and the VFs to be run until the next re-configuration instance.

b) Charging and Coordination Station: The CCS is the place where UAV batteries are re-charged and where FANET re-configuration decisions are taken. It includes two main components: the *Charging Station* and a *Computing Remote Node*. With respect to the Charging Station, the CCS is designed in such a way that the outage period of UAVs with residual low battery power is minimized. There, for this reason, backup batteries are over-provisioned and a sufficient number of electrical plugs is available to charge all the spare batteries. When a UAV lands, it thus always finds an available battery that can replace the discharged one. The replacement is done fast by an automated system without human intervention [20], and the UAV can fly to its spot in the FANET back right after. In this way, the outage period for a UAV is only related to the time needed for replacing the battery at the CCS and the time to fly to the CCS and back.

The *Computing Remote Node* (CRN) is a powered computing device, with TX/RX capabilites, that runs the Re-Configuration Module. RCM is able to periodically retrieve all the relevant information from the UAVs (i.e. the battery status cited so far, together with the currently-executed VFs and the amount of traffic towards VFs) needed for taking energy-aware decisions on the *best* FANET re-configuration, and for communicating its decisions to the UAVs for re-configuration execution. Typically, the *best* re-configuration is that minimizing the FANET energy consumption as a whole while ensuring service continuity to all Application Services over time. RCM is re-ran each time the FANET status changes. This may happen for some different reasons: *(i)* The FANET topology changes after that a UAV leaves the FANET to fly to CCS or a UAV comes back after battery replacement at the CCS; *(ii)* The input traffic towards one or more VFs has considerably changed, and current configuration is not energy efficient anymore; *(iii)* New VFs have to be executed by the FANET upon ground devices' requests. In all these cases, the RCM uses its TX/RX module (a 5G antenna + virtualized 5G core capabilities, or a LoRaWAN Gateway) to send to UAVs the new configuration to be implemented.

III. FANET Resource Orchestration

Orchestration of FANET resources aims at minimizing the service downtime. To this purpose, it is realized according two successive steps that are executed at the occurrence of each event that modifies the FANET composition. This happens when one UAV leaves the FANET for battery recharging or comes back to the FANET after battery recharging, when a flow requests a new VF service, or when a flow already using a VF service changes its behavior (e.g. its mean bitrate). Artificial intelligence for decision-making is applied by the RCM running in the CRN to take the following decisions, which are communicated to the active UAVs in the FANET:

1) the fraction $\eta_{RL,i}$ of the computing power $\mu_i^{(P)}$, to be made available by each UAV i on its CE;
2) the set F_i of VFs that each UAV i has to execute.

The target is twofold. On one side, in the short term, the overall service request, that is, all aggregated flows, each requesting a given VF with a given Quality of Service (QoS) requirement in terms of maximum tolerated latency, should be satisfied. On the other side, service downtimes due to a too low number of active UAVs in the FANET should be minimized in the long term.

These objectives are strictly correlated to each other. In fact, a myopic placement of the VFs on any UAV should minimize energy consumption and service downtime at the short time,

but could cause that some UAVs consume their battery charge very soon, in the same time interval. In this case, they are forced to land almost simultaneously, so leaving the FANET with too few UAVs to provide all the flows with the required services.

For this reason, the RCM makes the following two decision steps at each event: first, the fraction $\eta_{RL,i}$ of the computing power $\mu_i^{(P)}$ is calculated for each UAV i by means of Reinforcement Learning, in order to minimize service downtime at the long-term. These values are used as input of an (instantaneous) optimization problem to decide VF placement with the constraint that latency for packet processing for the aggregated flow requesting the VF f be less a given delay threshold D_f.

The long-term optimization based on RL will be described in Section III-A, while the short-term optimization, which is approached by means of Integer Linear Programming (ILP), will be introduced in Section III-B.

A. Long-term FANET behavior optimization

In a typical Reinforcement Learning (RL) problem, an *agent* interacts with its *environment*. The environment, in return, provides *rewards* and a *new state* based on the *actions* of the agent. Therefore, in reinforcement learning, we do not teach an agent how it should do something, but presents it with rewards whether positive or negative based on its actions. Mathematically, we need to formulate this problem by means of a Markov Decision Process (MDP).

Therefore, in this section we describe the MDP used to support the RCM in taking the best energy-aware decisions that minimize the FANET service downtime. Specifically, a RL agent is deployed inside the RCM to choose the best reduction factor $\eta_{RL,i}$ of each UAV computing hardware mounted on board.

Let us define the MDP used by the RL agent to achieve this goal. A MDP can be defined by its environment, space state, action space and reward function.

The *environment* in which the agent takes action is the system described so far, as seen by the RCM. It is constituted by the state of charge of the battery of each UAV in the FANET, which is used to supply the engines, the computing element and the TX/RX Module. Its current charge level is communicated to the UAV Local Manager such that it can take its decisions regarding the amount of CE computing power using and when landing for battery recharging.

The *state observations* are the system state observed by the agent at the beginning of each decision epoch. Specifically, each time a UAV leaves the FANET to fly to the CCS or a UAV comes back after battery replacement, a decision epoch is triggered, and each UAV computing hardware reduction factor $\eta_{RL,i}$ is chosen.

The *action space* is the set of reduction factors to be applied to each UAV in the FANET, H_{RL}. Specifically, at each decision epoch the agent chooses one reduction factor $\eta_{RL,i}$ for each UAV in the FANET, from a discrete set of values ranging from 0 to 1 with a constant step of 0.25:

$$H_{RL} = \{\eta_{RL,0}, \ldots, \eta_{RL,N}\}, \text{ for } \eta_{RL,i} \in [0,1], \forall i. \quad (1)$$

Finally, we recall that the objective of the long-term reconfiguration is to ensure service continuity to all Application Services over time. For this reason, we design the *reward* so that its maximization would minimize the total service downtime. Therefore, at the end of each decision epoch n, the RCM will retrieve the amount of time in which some VF could not be deployed, and therefore users experienced service downtime, $S_{down}(n)$. Let us set the reward r_n as the opposite of the service downtime:

$$r_n = -S_{down}(n) \quad (2)$$

Recall that the goal of the agent is to maximize the total reward G_n it receives with a discounting factor γ, calculated as:

$$G_n = \sum_{k=0}^{+\infty} \gamma^k \cdot r_{n+k+1} \quad (3)$$

Therefore, the long-term goal of the agent is to minimize the amount of service downtime experienced over a large time horizon.

B. VF Placement short-term Optimization

Let V be the set of UAVs, including both the ones that are flying in FANET, so providing service, and the ones that are temporarily left the FANET for battery recharging. Let $i \in V$ be the generic UAV, and F the set of available VF instances to be deployed on the FANET as a whole. We refer to an aggregated flow as the superposition of all the flows that require the same VF with the same QoS requirement. Let λ_f be the packet rate of the aggregated flow requiring the VF f.

The short-term optimization problem consists in deciding the VF placement while minimizing the total power consumption and the percentage of flows not served because the required VF has not been placed. To this purpose, we define an optimization problem as follows.

Let u_i be the variable representing whether the UAV i is used to host at least one VF or not:

$$u_i = \begin{cases} 1 & \text{if the UAV } i \text{ hosts at least one VF} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Let $v_{f,i}$ be the variable representing whether the VF f is assigned to the UAV i or not:

$$v_{f,i} = \begin{cases} 1 & \text{if the VF } f \text{ is assigned to the UAV } i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Let p_f be the variable representing whether the VF f is placed on a UAV or not. Since each VF is placed at most on one UAV only, this variable can be defined as follows:

$$p_f = \sum_{i=1}^{N} v_{f,i}. \quad (6)$$

The objective function weighing the contribution of the overall power consumption, P_T, and the lack of service, L_T, is defined as follows:

$$\Psi = \gamma_F \cdot P_T + \gamma_L \cdot L_T; \tag{7}$$

where γ_F and γ_L are two constants that are used to give different importance to the two components of the objective function.

The term P_T is the overall power consumption of the FANET, defined as follows:

$$P_T = \sum_{i=1}^{N} u_i \Big[P_i^{(E)} + \eta_{RL,i} \cdot P_i^{(CE)} + \\ + \sum_{f \in F} v_{f,i} \left(P_f^{(VM)} + P_f^{(VF)} \right) \Big]; \tag{8}$$

where:

- $P_i^{(E)}$ is the power consumption due to the engines of the UAV i; it is aways present if the UAV i is involved in the FANET mission;
- $P_i^{(CE)}$ is the power absorbed by the computing hardware mounted on board of UAV i, independently of the number of running VFs; it is reduced by a factor $\eta_{RL,i}$ decided by the RCM, which calculates it by RL to minimize service downtime, as described in Section III-A.
- $P_i^{(VM)}$ is the additional power consumption due to the execution of the VM where the VF f is executed;
- $P_f^{(VF)}$ is the power consumption absorbed to process the flow requiring the VF f. It depends on the packet flow rate, λ_f, and the energy $e_f^{(P)}$ needed by the VF f to process one packet as follows:

$$P_f^{(VF)} = \lambda_f \cdot e_f^{(P)} \tag{9}$$

The term L_T in (7) represents the deployment capability of all the VFs, defined as the fraction of flows that the FANET is able to serve. It can be calculated as follows:

$$L_T = \sum_{\forall f \in F} p_f \cdot \lambda_f \tag{10}$$

Each queue associated with a VF instance (see Fig. 1) is modeled as a M/M/1 system, which is a single-server queueing system with Poisson-distributed arrivals and exponentially-distributed packet service times.

Let Ξ_i be the processing rate of the CE installed onboard UAV i, expressed in FLoating point Operations Per Second (FLOPS), and $\eta_{RL,i}$ the reduction factor decided by the RCM installed on the CCS on ground. Assuming that all VFs are provided with no priority, i.e. the overall processing power is equally shared among the VFs running on the UAV i, the packet processing rate for the generic VF f, coinciding with the service rate of the M/M/1 queue assigned to it, is:

$$\mu_{f,i} = \frac{\Xi_i \cdot \eta_{RL,i}}{\Phi_i} \cdot \frac{1}{\Omega_f} \tag{11}$$

where Ω_f is the number of floating-point operations required by the VF f to process a packet, while Φ_i is the number of VFs placed on the UAV i, that is:

$$\Phi_i = \sum_{\forall f \in F} v_{f,i} \tag{12}$$

According to the M/M/1 queueing theory, the average response time or sojourn time (i.e. total time a packet spends in the M/M/1 queueing system) suffered by packets of the flow requiring the VF f, if deployed on the UAV i, is:

$$d_f = \frac{1}{\mu_{f,i} - \lambda_f}. \tag{13}$$

Now, we can formulate the optimization problem that maximizes the objective function defined in (7). We find the optimum set of UAVs to be included in the placement, $U = [u_i]_{1 \times N}$, and the optimum placement, i.e. $V = [v_{f,i}]_{|F| \times N}$, $|F|$ being the cardinality of F, that is, the number of all the VFs requested to the FANET.

The optimization problem is formulated as follows:

$$\min_{U,V} \Psi \tag{14}$$

$$\text{s.t. C1: } \widetilde{i} = 0, \ \forall \widetilde{i} \in \widetilde{S} \tag{15}$$

$$\text{C2: } 0 \leq u_i \leq min\{1, \Phi_i\}, \ \forall i \in V \tag{16}$$

$$\text{C3: } \sum_{\forall f \in F} \lambda_f \cdot v_{f,i} < \mu_{f,i}, \ \forall i \in V \tag{17}$$

$$\text{C4: } \delta_f < D_f^{(MAX)} \ \forall f \in F \tag{18}$$

where $\widetilde{S}$ is a subset of S containing non-available UAVs. Constraint C1 imposes that no VF can run in non-available UAVs. Constraint C2 shows that the Boolean variable u_i, for each UAV i, is upper-bounded by the number Φ_i of VFs running on the UAV i. In other words, u_i cannot assume the value 1 if $\Phi_i = 0$. Constraint C3 imposes that the overall packet rate arriving to any VFs deployed on each UAV i is less than the fraction $\mu_{f,i}$ assigned to the VF f on the UAV i. Finally, constraint C4 imposes that the processing delay suffered by packets of the flow using the VF f is less than the maximum tolerable delay for that flow, $D_f^{(MAX)}$.

IV. Numerical Results

In this section, we will present a use case to evaluate performance of the proposed framework.

The experiments were performed in a simulator based on OpenAI Gym [21] and Stable-Baselines3 [22]. We used a Proximal Policy Optimization (PPO) [23] agent, which is based on the Actor-Critic architecture, and is one of the most powerful RL algorithms in the current literature, by providing consistency and stability with little parameter tuning. In our agent both actor and critic networks have two fully connected layers with 64 neurons each. We also used the Adam Optimizer with a learning rate of $3 \cdot 10^{-4}$ and set the discount factor $\gamma = 0.95$.

Let us stress that, for network sizes like the ones considering in this paper and characterizing FANET scenarios, derivation

TABLE I: Packet flow rate

Function	λ_f (packet/s)	$P_f^{(VM)}$ (W)	$e_f^{(P)}$ (mJ)
f_1	4688	8.63	6.04
f_2	4276	10.78	4.24
f_3	3896	1.22	4.63
f_4	3164	15.41	5.55
f_5	5100	20.01	4.9
f_6	4521	3.69	5.31
f_7	3786	1.22	8.74
f_8	3520	2.78	7.44
f_9	5021	4.13	6.18
f_{10}	3762	10.16	4.1

of the solution of the optimization problem defined in (14) by means of an entry-level computing hardware requires few minutes. Also, consider that it does not need to be executed at runtime, but it can be run on the Computing Remote Node offline to save the results on a table that should be modified each time the FANET behavior changes, on a larger timescale.

Let us consider a FANET composed by $|V| = 5$ UAVs, and $|F| = 10$ VFs that have to be allocated in the FANET to provide functionalities to end users or ground devices.

The packet flow rate λ_f, as well as the additional power consumption $P_f^{(VM)}$ to maintain the VMs hosting the VFs switched on, and the energy $e_f^{(P)}$ needed by the VF f to process one packet, are listed in Table I. These parameters were estimated on a real deployment of a softwarized network at the *UniCT 5G&B Lab* of the University of Catania, where a set of VNFs implemented by students for teaching purposes are running to execute different functions. Flow statistics have been achieved during some measurement experiment on flows generated by groups of users during their normal academic activities.

Regarding the UAVs, we consider five small equal quad-copters, consuming a total power for the engine $P_i^{(E)}$ equal to 66.56W, and each UAV is equipped with an INTEL NUC miniPC working as CE with a maximum execution capacity Ξ_i equal to 8.32 gigaFLOP/s. Finally, all the considered VFs have to guarantee the same requirement in terms of maximum tolerable processing delay suffered by each packet $D_f^{(MAX)}$. In our experiments, we set this delay equal to 1ms [24]. Finally, the two weights of the objective function Ψ are set as $\gamma_F = 0.5$ for the overall power consumption of the FANET, and $\gamma_L = 0.1$ for the deployment capabilities of all VFs.

Performance of the framework is evaluated by varying the mean value t_{CS} of the round-trip time needed by a UAV to reach the Charging Station, replace the battery and come back to the place of the mission, and the maximum battery capacity B. The actual value of round-trip time is randomly generated as a Gaussian random variable with average t_{CS} and standard deviation equal to 10% of t_{CS}. Results are provided with a 95% confidence interval on the third decimal place. In Table II, there is a summary of all the variables and the corresponding value considered during the experiments.

In Fig. 3, the average number of flying UAVs is represented. It is easy to understand that, as the round-trip time increases,

TABLE II: Simulation Parameters

Parameter	Value
$\|V\|$	5
$\|F\|$	10
$P_i^{(E)}$	66.5 W, $\forall i \in V$
Ξ_i	8,32 gigaFlop/s, $\forall i \in V$
t_{CS}	[40,90,140,190,240,290,340] s
B	[40,60,80]Wh
η_{RL}	$0, 0.25, 0.5, 0.75, 1$
γ_F	0.5
γ_L	0.1
$D_f^{(MAX)}$	1ms
Layer of the networks	2
Number of neurons for each layer	64
Learning Rate	$3 \cdot 10^{-4}$
γ	0.95

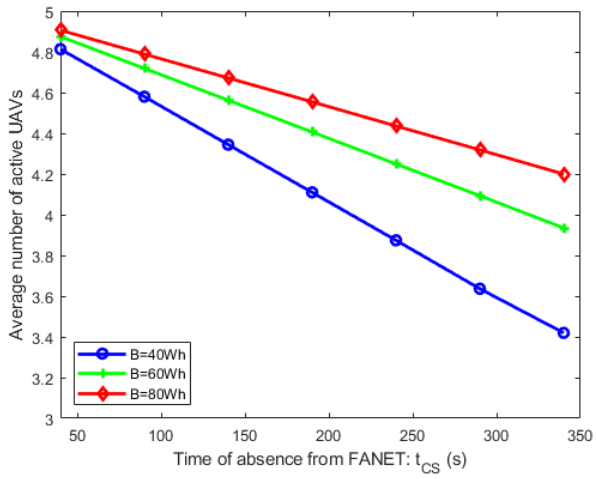

Fig. 3: Average number of flying UAVs

each UAV remains more time away from the FANET, and therefore there are fewer flying UAVs. Obviously, the maximum battery capacity has an influence on the average number of flying UAVs, since the greater the battery capacity, the less time each UAV is forced to return to the Charging Station for battery replacement.

Another consequence of the longer round-trip time is the need to place the VFs hosted by the UAV that has temporarily left the FANET on those UAVs remained active in flight. This causes an increase in the average number of VFs that each flying UAV will have to host to ensure that the FANET is able to satisfy the requests coming from users as much as possible (see Fig. 4). Also in this case, the maximum battery capacity has its impact: as the battery capacity increases, more UAVs are active in the FANET simultaneously, each one having to host a smaller number of VFs.

Since the total energy consumption also depends on the consumption due to the instantiation of the VMs that host the VFs ($P_f^{(VM)}$) and the processing load due to the input flow of each VFs, it is evident that the higher the number of VFs allocated on the same UAV, the greater the average consumption of it. The UAV will be forced to use a higher

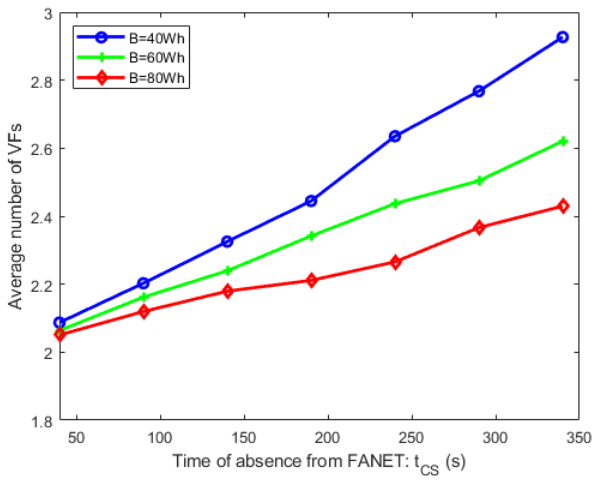

Fig. 4: Average number of VFs running on each UAV

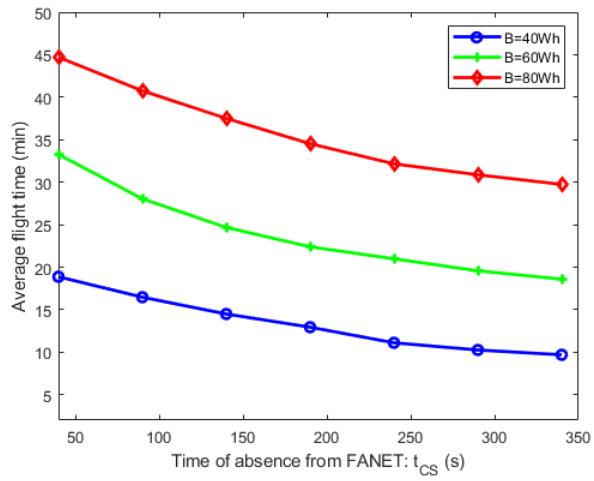

Fig. 6: Average flight time of each UAV

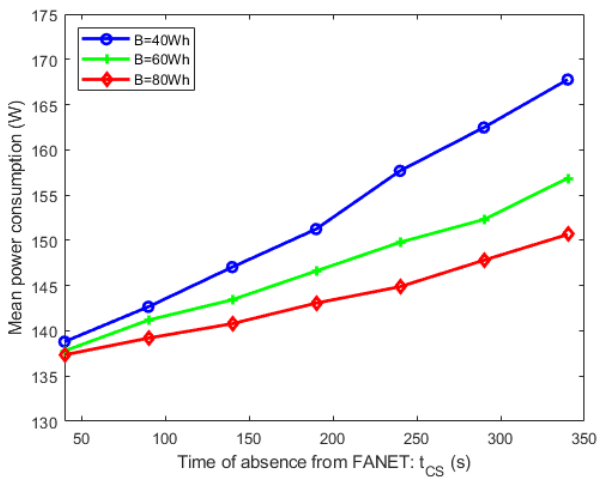

Fig. 5: Mean power consumption of active UAVs

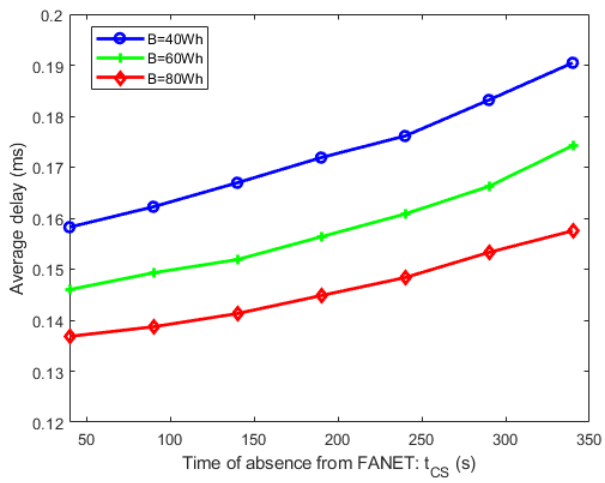

Fig. 7: Average FANET processing delay

percentage of CPU in order to process a higher input load if compared to the case in which the UAVs are all present within the FANET and the VFs are distributed more uniformly. In Fig. 5, the average power consumption suffered by each UAV remaining active in the FANET is shown. The curves clearly justify what has just been said.

Coming back to what is shown in the previous figures, the possibility of having a higher battery capacity means that the UAVs are forced to return to the Charging Station less frequently. Accordingly, remaining more UAVs in flight, the VFs do not weigh down individual UAVs, which are not forced to increase the percentage of CPU to use, and therefore consume less.

As already said in the Introduction, the more resources the CE consumes, the faster the battery of the UAV runs out, thus reducing the overall FANET service availability. In Fig. 6, the average flight time of each UAV is represented. This trend is obviously correlated with the power consumption shown in Fig. 5, since the more a UAV consumes, the less its battery charge will be, and therefore the less its flying time inside the FANET.

Fig. 7 shows the average processing delay offered by the FANET to each packet. As we can see, in all the performed simulations, the FANET is able to satisfy the maximum acceptable delay requirement $D_f^{(MAX)}$. However, we can see that as the time of absence of the UAV from the FANET increases, the delay undergoes a gradual increase. This is due to the fact that, as there are fewer UAVs available, the VFs will be concentrated on those that remain active. This implies that even if the UAV increases the percentage of used CPU, it will have to partition this CPU to a higher number of VFs. Consequently, as the computational resources available to the single VF are smaller, the delay introduced during packet processing will increase. The difference in behavior depending on the maximum battery capacity is justified by what has already been said so far.

V. Conclusions and Future Work

The focus of this paper regards the application of a FANET to provide ground devices deployed in remote areas with edge-computing facilities to support delay-constrained vertical applications. Since flight duration of the component UAVs is strongly limited by battery limitations, exacerbated by power consumption of the Computing Element mounted onboard to run VFs, which is added to the power consumption of UAV engines, management of computation load distribution is challenging in these scenarios. To this purpose, an optimization framework is proposed to decide, at runtime, the number of UAVs that have to participate to the service provisioning, the VF placement on these UAVs, and the computation power of each UAV to be dedicated to edge computing. A constrained optimization problem based on Integer Linear Programming (ILP) is formulated to make the first two decisions, while Reinforcement Learning (RL) based on Double Deep Q-learning (DDQN) is used to optimize the percentage of available CPU resources for VF computing. A use case is used to evaluate performance of the proposed framework.

As a future work, we plan to extend the proposed long-term FANET behavior optimization by including a Long Short-Term Memory (LSTM) recurrent neural network, which could further improve the overall system performance thanks to its inherent ability to keep an internal state that is updated as the episode plays out.

Acknowledgement

The research was partially supported by the research project "Programma ricerca di Ateneo UNICT 2020-22 linea 2-OMNIA" of the University of Catania, the PRIN Project "Liquid Edge" and the POR S6 project.

References

[1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[2] C. Barrado, R. Messeguer, J. Lopez, E. Pastor, E. Santamaria, and P. Royo, "Wildfire Monitoring using a Mixed Air-ground Mobile Network," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 24–32, 2010.

[3] J. George, S. PB, and J. B. Sousa, "Search Strategies for Multiple UAV Search and Destroy Missions," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 355–367, 2011.

[4] Z. Sun, P. Wang, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan, and I. F. Akyildiz, "BorderSense: Border patrol through advanced wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 468–477, 2011.

[5] A. Cho, J. Kim, S. Lee, and C. Kee, "Wind Estimation and Airspeed Calibration using a UAV with a Single-Antenna GPS Receiver and Pitot Tube," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 109–117, 2011.

[6] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the Sky: Leveraging UAVs for Disaster Management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[7] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[8] C. Grasso, R. Raftopoulos, and G. Schembra, "Deep Q-Learning for Job Offloading Orchestration in a Fleet of MEC UAVs in 5G Environments," in *IEEE International Conference on Network Softwarization (NetSoft)*, 2021, pp. 186–190.

[9] B. Li, Z. Fei, and Y. Zhang, "Uav communications for 5g and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2019.

[10] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5g and resilient public safety networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 36–42, 2018.

[11] C. Rametta and G. Schembra, "Designing a Softwarized Network Deployed on a Fleet of Drones for Rural Zone Monitoring," *Future Internet*, vol. 9, no. 1, 2017.

[12] G. Faraci, A. Raciti, S. A. Rizzo, and G. Schembra, "Green wireless power transfer system for a drone fleet managed by reinforcement learning in smart industry," *Applied Energy*, vol. 259, February 2020.

[13] G. Faraci, C. Grasso, and G. Schembra, "Reinforcement-Learning for Management of a 5G Network Slice Extension with UAVs," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 732–737.

[14] G. Faraci, C. Grasso, and G. Schembra, "Fog in the Clouds: UAVs to Provide Edge Computing to IoT Devices," *ACM Trans. Internet Technol.*, vol. 20, no. 3, 2020.

[15] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5G Network Slice Extension With MEC UAVs Managed With Reinforcement Learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2356–2371, 2020.

[16] H. Van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double q-Learning," in *AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[17] G. Liu, Y. Huang, Z. Chen, L. Liu, Q. Wang, and N. Li, "5G Deployment: Standalone vs. Non-Standalone from the Operator Perspective," *IEEE Communications Magazine*, vol. 58, no. 11, pp. 83–89, 2020.

[18] A. Aijaz, "Private 5G: The Future of Industrial Wireless," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 136–145, 2020.

[19] "A Comparative Study of LPWAN Technologies for Large-scale IoT Deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.

[20] D. Lee, J. Zhou, and W. T. Lin, "Autonomous Battery Swapping System for Quadcopter," in *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 118–124.

[21] "Gym: a toolkit for developing and comparing reinforcement learning algorithms," https://gym.openai.com/. Accessed Jan 2022.

[22] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

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

[24] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.