Toward Proactive Service Relocation for UAVs in MEC

Oussama Bekkouche[1], Somayeh Kianpisheh[1], and Tarik Taleb[1,2]

[1]Communications and Networking Department, Aalto University, Finland. Email: firstname.lastname@aalto.fi
[2]Centre for Wireless Communications, University of Oulu, Finland.

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685066

***Abstract*—Multi-Access Edge Computing (MEC) is considered as one of the key enablers of Unmanned Aerial Vehicles (UAVs) use cases. However, the envisioned MEC deployments introduce new challenges related to the management of the mobility of services across the distributed MEC hosts, following the UAVs movements and possible handovers to ensure sustainable Quality-of-Service (QoS). A major challenge for MEC service mobility is the decision-making on where and when to relocate services. In this paper, we motivate the use of the predefined flight plans of UAVs for devising proactive relocation strategies that can deal efficiently with realistic asynchronous relocation processes. Moreover, we formulate the Proactive Service Relocation for UAV (PSRU) problem using linear programming, and we validate the gains introduced by the proactive relocation strategy and the use of the predefined flight plans of UAVs.**

***Index Terms*—Unmanned Aerial Vehicles (UAVs), 5G, Beyond 5G, MEC, Service Migration, Mobile Networks, and Linear Programming.**

I. Introduction

The ever-increasing usage of Unmanned Aerial Vehicles (UAVs) in a myriad of new applications such as cargo delivery, infrastructure inspection, and public safety is faced with several challenges that hinge on the availability of a reliable, low-latency, and Beyond Visual Line-of-sight (BVLOS) communication between the UAVs and the ground control services and applications. In this context, Multi-Access Edge Computing (MEC) [1] is considered as one of the main enablers of low-latency communications in the 5^{th} generation of mobile networks. By leveraging the cloud-computing capabilities of MEC at the vicinity of the Radio Access Network (RAN), UAVs can directly communicate with the MEC-hosted services and applications without passing through intermediates networks (e.g., transport and core networks), which ensures a reduced latency and an increased bandwidth because of the absence of "bottleneck segments".

In a MEC system, each MEC host is associated with one or more Base Stations (BSs) that play the role of relay nodes between the UAVs and the services deployed on that host. Nevertheless, the coverage area of the set of BSs associated with a specific MEC host is limited, which means that when a handover (HO) event is triggered due to the mobility of the UAVs, and in order to reap the benefits of low latency, there might be a need for changing the hosting MEC host by relocating the running application toward the serving MEC host associated with new BS [2].

The relocation decisions are usually taken in a reactive manner after the HO events that change the serving BSs and MEC hosts and cause the degradation of the Quality-of-Service (QoS). However, a major issue with reactive service relocation is the relatively large time window required for relocating the MEC applications from one MEC host to another. Indeed, the relocation process is known to be an asynchronous process [3] that can take a considerable amount of time before having the MEC application running in the new MEC host. As a result, when a relocation is triggered reactively due to the degradation of the QoS, the users will continue to be served with this degraded QoS during the relocation time window as well, which might not be tolerable for certain applications such as UAVs' applications.

Another alternative that can overcome the challenges of the reactive relocation approach is the proactive relocation approach, in which the HOs events are first predicted, and then, based on these predictions, the relocations are triggered before the occurrence of the actual HOs events and QoS degradation [4]. However, as presented in [5], the proactive relocation approach has the requirement of having prior knowledge about the mobility of the users (e.g., trajectory and speed), which is difficult to fulfill for the majority of MEC use cases.

Fortunately, UAVs will be operating according to predefined flight plans (i.e., a set of waypoints to follow by the UAV) that are first approved and then shared by the UAV Traffic Management (UTM) system. Research work conducted in [6]–[9], discuss the integration of UTM system in 5G for network optimization purposes. Moreover, 3GPP in [10] suggested the inclusion of flight plan information in the radio link signaling of aerial user equipment.

On the other hand, UAVs have a high likelihood to experience Line-of-Sight (LoS) propagation conditions to multiple base stations (BSs) simultaneously, which makes the management of the mobility of UAVs more complex and challenging. Results, presented in [11], show that UAVs experience a higher number of HOs and frequent HO failures compared to terrestrial UEs. Thus, triggering a proactive relocation each time a HO event is predicted can negatively impact the Quality-of-Experience (QoE). Hence, optimization is required when planning proactive relocations, and blind reliance on the predicted HOs should be avoided.

To deal with the above challenges, in this paper, we leverage the predefined flight plans of UAVs for planning proactive relocations that maximize the availability of the UAV applications in the serving MEC host along the UAV's flight while taking into consideration the time required for relocating the applications from one edge host to another. The flight plan is used as an input for a Linear Integer Programming (LIP) model that provides the optimal relocation policy, whereby an optimal policy is defined as the one that ensures that the UAV application is running in the serving MEC host for the maximum amount of time.

The remainder of this paper is organized as follows. Section II presents the related works. Section III discusses a motivating example. The problem space and the formulation of the proposed solutions are detailed in Section IV. Section V presents the performance evaluation and the analysis of the results of the proposed solution. Finally, Section VI concludes the paper.

II. Related work

The problem of service relocation in edge cloud systems has been tackled in the litterateur from different angles that mainly differ in the followed relocation strategy. Work in [12] addresses the mobility-driven service relocation problem using Markov Decision Process (MDP) while considering 1-D stochastic mobility patterns. Wang *et al.* [13] extended the work done in [12] to consider a more realistic 2-D mobility pattern. Authors in [14] used Lyapunov optimization to decompose the mobility-aware dynamic service relocation problem into a series of real-time optimization problems that do not require a priori knowledge about user mobility. Similar in spirit to [15], [16], Tang *et al.* [17] leverage Reinforcement Learning (RL) for optimizing the service relocation decisions based on the communication delay, the power consumption, and the migration cost. These works consider time-slotted models in which the relocation decisions taken at a given time slot t are reflected instantly in the system (i.e., at the same time slot t or at the next time slot $t+1$). While such an assumption alleviates the need for proactive relocations, it does restrict the practicability of these solutions. Indeed, unless the length of the time slots is sufficiently long to reflect the time required for relocating the services from one edge host to another, the practicability of these solutions can be very limited. On the other hand, a largely time-slotted model introduces delays in decision making which implies that the system can operate in undesired states for longer time periods.

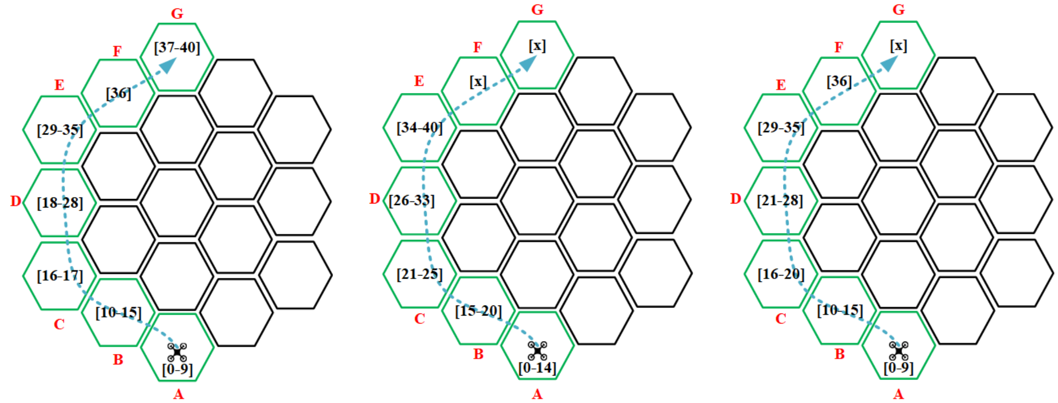
In [18], authors proposed another MDP-based solution for service relocation in edge cloud. However, in this work, the MDP model is only used to decide on the relocation target. The decisions on whether to relocate the service or not are based on the forecasted QoS provided by an Autoregressive Integrated Moving Average predictor (ARIMA). While the QoS predictor can implicitly trigger proactive relocations, it needs to collect real-time data about the state of the network and edge servers. Moreover, the proposed QoS predictor considers a 1-D stochastic mobility pattern. Finally, work in [19] addresses the proactive service relocation by deploying multiple instances of the same MEC application in neighboring MEC hosts, so to rely on already available instances whenever a HO to a different MEC host is necessary. The main drawback of this solution is the irrational usage of the limited MEC resources.

III. Motivating Example

Hereafter, we motivate the need for our solution via the example illustrated in Fig. 1. For the sake of simplicity and without any loss of generality, we consider a cellular network composed of a set of BSs, that are associated with a set of MEC hosts in a one-to-one fashion. Also, we consider that the flight plan of the UAV is slotted to relatively small time slots, where the location of the UAV is updated according to the flight trajectory at the beginning of each time slot. By leveraging the coverage map that correlates the UAV locations to the serving BSs, it is possible to extrapolate the time interval on which the UAVs will be using each BS. As illustrated in Fig. 1(a), given the flight trajectory represented by the dashed arrow and the speed of the UAV, it is possible to deduce the time intervals on which the UAV will use each BS. For instance, the base station **A** will be used from time slot 0 until time slot 9; the UAV will then perform a HO and change the serving BS to **B** and use it for 6 time slots (i.e., from time slot 10 to 15). In addition to the speed and mobility plan of the UAV, the time spent in each BS depends on the size of the coverage area of this BS in the 3D space.

A. Reactive relocation

In order to showcase the limitations of the reactive relocation approach when considering realistic relocation delays, we execute a *"Reactive-always relocate"* algorithm on the example of Fig. 1(a). The execution trace is depicted in Fig. 1(b), where the intervals labeled on each BS indicate the time interval during which the UAV has used the MEC host associated with that BS before having the MEC application completely moved to a new MEC host. The *"Reactive-always relocate"* algorithm always triggers a relocation whenever the MEC host associated with the serving BS differs from the MEC host in which the MEC application is running. Also, it is worth noting that for practical purposes, we consider that when the relocation of a given application is ongoing, no other relocations of that application are possible, and any new relocation decision is ignored. As depicted in Fig. 1(a), the UAV will be using BS **A** during the time interval $[0-9]$; hence, no relocation is triggered during this time interval. However, at time slot 10, the UAV will perform a HO and change the serving BS to **B**. As a result, and in order to sustain the low-latency communication with the UAV application, the *"Reactive-always relocate"* policy will enforce a service relocation from MEC host **A** to MEC host **B**. Nevertheless, the relocation process will take some time before having the application up and running in MEC host **B**. In our example, the relocation time is set to 5 time slots, thereby the UAV application continues to run in MEC host **A** until the end of time slot 14, and it is not available in MEC host **B** until time slot 15. The same process is repeated when the UAV



(a) The serving base stations during the UAV flight.

(b) The serving MEC hosts when considering "Reactive-always relocate" strategy

(c) The serving MEC host when considering "Proactive-always relocate" strategy

Fig. 1. Example showing the serving cells vs the serving MEC hosts when considering different relocation strategies.

changes the serving BS from **B** to **C** and from **C** to **D**. By comparing the execution trace of the *"Reactive-always relocate"* algorithm with the connectivity trace in Fig. 1(a), we notice that the UAV was not served from the optimal MEC host for a significant amount of time (e.g., the UAV application was never placed in the MEC hosts **F** and **G**).

B. Proactive relocation

A trivial enhancement to the previous algorithm is to use a look-ahead window to verify whether the UAV will change the serving BS in the future, with the size of the look-ahead window equal to the relocation time. The execution trace of such an algorithm is presented in Fig. 1(c). For instance, at time slot 5, the UAV is served by MEC host **A**, and by using a look-ahead window of 5 time slots, it is possible to realize that the UAV will change the serving MEC host to **B** starting from time slot 10, and hence, the relocation should be triggered before that. Even that this strategy can considerably enhance the availability of MEC applications at the optimal MEC hosts, it is still not optimal. Indeed, by having a closer look at the execution trace of Fig. 1(c), we notice that the relocation from **B** to **C** was completed at the right time (i.e., the application was available in **C** at the beginning of time slot 16, which is the time slot at which the UAV change the serving BS from **B** to **C**). However, the UAV spent only two time slots: 16 and 17 in BS **C** and then moved to BS **D** starting from time slot 18. Nonetheless, in order for the application to be available in **D** starting from time slot 18, a relocation from **B** to **D** at time slot 13 is required but this was not possible because an already triggered relocation (at time slot 11) was ongoing from **B** to **C**. By relocating the application from **B** to **C**, the application was hosted in the optimal MEC host **C** for two time slots. However, if the application was relocated to **D** instead of **C**, the availability of the UAV application at MEC host **D** will be increased from two to five time slots. The same issue arise when relocating the application from **E** to **F** instead from **E** to **G**. This issue occurred because of very fast HOs from **C** to **D** and from **F** to **G**. Since UAVs are characterized by frequent and Pin-Pong HOs [11], [20], addressing this issue is mandatory to sustain an acceptable QoS and unlock the full potential of MEC for UAVs. Therefore, an optimized solution is required, which is the focus of this paper.

IV. System Model and Problem Formulation

A. System Model

As shown in Fig. 1, we consider a scenario where a UAV flies according to a predefined flight plan, on top of a geographical area covered by a set of base stations $\mathcal{B}$ and MEC hosts $\mathcal{H}$. Each base station $b \in B$ is associated with one MEC host $h_b \in \mathcal{H}$ and each MEC host $h \in \mathcal{H}$ is associated with one or more base stations. Moreover, we consider a time-slotted model, in which the UAV's location and the serving base station may only change at beginning of a time slot. For this purpose, we sample the finite time horizon T equal to the duration of the predefined flight plan to discrete time slots of equal length Δt. We define $\mathcal{S}_t$ as the serving MEC host that runs the UAV's application at time slot $t \in T$, and $\mathcal{L}_t$ as the local MEC host directly connected to the serving base station at time slot $t \in T$. The identity of the serving base station at time slot $t \in T$ is provided by a function $\delta(t)$ that can reflect the RAN dynamics, and the mapping between a base station $b \in \mathcal{B}$ and its MEC host $h_b \in \mathcal{H}$ is provided by a function $\phi(b) = h_b$. As a result, $\mathcal{L}_t = \phi(\delta(t))$.

At the beginning of each time slot t, the relocation controller i.e., the MEC Orchestrator (MEO) [1] takes a relocation decision $\mathcal{A}_t \in \{1 \,..\, |\mathcal{H}|\}$ that reflects one of the following control options:

1) Triggering a relocation: in this case, $\mathcal{A}_t$ is equal to the relocation target, and it should differ from the serving MEC host $\mathcal{S}_t$. Without any loss of generality, we assume that the time required for relocating the UAV's application from $\mathcal{S}_t$ to $\mathcal{A}_t$ depends only on the

application size and type (i.e., stateful or stateless), it is thus known a priori and we denote it by $\mathcal{T}$ (expressed in terms of time slots). Note that during the relocation process, the UAV will continue to be served from the original MEC host until the relocation is done (i.e., the MEC system is assumed to support the Make-before-Break mechanism).

2) No relocation is triggered: in this case, the target of relocation $\mathcal{A}_t$ is equal to the current serving MEC host $\mathcal{S}_t$.

B. Proactive Service Relocation for UAV (PSRU) – Problem Formulation

1) Decision variables: Hereafter, we introduce the decision variables used for the linear formulation of the PSRU problem.

a) ***Elapsed relocation time****:* We define the integer variable $\mathcal{R}_t$ that captures the number of time slots elapsed by the end of time slot t since the MEC controller triggered the relocation.

$$\forall t \in T : \mathcal{R}_t = \begin{cases} \mathrm{R}_{t-1}+1 & \text{If a relocation is still ongoing at the beginning of time slot } t \text{ (if } \mathcal{T} > \mathcal{R}_{t-1} > 0).\\ 0 & \text{If no relocation is ongoing at the beginning of time slot } t.\\ 1 & \text{If a relocation is triggered at the beginning of time slot } t.\end{cases}$$

b) ***Relocation status****:* We define the Boolean variable $\mathcal{O}_t$ that captures whether an already triggered relocation is still ongoing at the beginning of time slot t. This variable is used for capturing the variable $\mathcal{R}_t$. $\mathcal{O}_t$ is defined as follows:

$$\forall t \in T : \mathcal{O}_t = \begin{cases} 1 & \text{If an old relocation is ongoing at the beginning of time slot } t \text{ (i.e., } \mathcal{T} > \mathcal{R}_{t-1} > 0).\\ 0 & \text{Otherwise (i.e., } \mathcal{R}_{t-1} = \mathcal{T} \text{ or } \mathcal{R}_{t-1} = 0 \text{).}\end{cases}$$

The definition of the variable $\mathcal{O}_t$ implies that the relocation is considered to be ongoing at the beginning of a time slot t only and only if the elapsed relocation time at the end of the previous time slot is greater than 0 and smaller than $\mathcal{T}$. If this condition is not satisfied, no relocation is considered to be ongoing at the beginning of a time slot t, either because no relocation has been previously triggered (i.e., $\mathcal{R}_{t-1} = 0$), or a previously triggered relocation has ended (i.e., $\mathcal{R}_{t-1} = \mathcal{T}$).

c) ***Triggered relocation****:* We define the binary variable $\mathcal{K}_t$ that captures whether the decision $\mathcal{A}_t$ taken at the beginning of time slot t can trigger new relocation of the UAV's application from the MEC host $\mathcal{S}_t$ to the MEC host $\mathcal{A}_t$. The variable $\mathcal{K}_t$ is also used for capturing $\mathcal{R}_t$.

$$\forall t \in T : \mathcal{K}_t = \begin{cases} 1 & \text{If the action } \mathcal{A}_t \text{ can trigger a relocation at the beginning of time slot } t.\\ 0 & \text{Otherwise.}\end{cases}$$

d) ***Relocation decisions****:* As discussed in section IV-A, we define the integer variable $\mathcal{A}_t$ that captures the relocation decision taken by the MEO at the beginning of time slot t.

$$\forall t \in T : \mathcal{A}_t = a; a \in [1 \,..\, |\mathcal{H}|]$$

e) ***Serving MEC host****:* As aforementioned, we introduce the integer variable $\mathcal{S}_t$ that captures the identity of the serving MEC host during time slot t. As explained in section IV-B2d, $\mathcal{S}_t$ is captured by leveraging the variables $\mathcal{R}_t$ and $\mathcal{A}_t$.

$$\forall t \in T : \mathcal{S}_t = h; h \in [1 \,..\, |\mathcal{H}|]$$

f) ***Optimal MEC host****:* We define the Boolean variable $\mathcal{U}_t$ that shows whether the UAV's application is hosted in the optimal MEC host. The value of the variable $\mathcal{U}_t$ depends on the value of $\mathcal{S}_t$, and it is defined as follows:

$$\forall t \in T : \mathcal{U}_t = \begin{cases} 1 & \text{If } \mathcal{L}_t = \mathcal{S}_t.\\ 0 & \text{Otherwise.}\end{cases}$$

2) Constraints: The values of the previously defined variables are enforced using the following set of constraints. Hereafter, we denote by $\mathcal{M}$ a big positive number approaching infinity, and by $\overline{\mathcal{X}}$ the complement of the binary variable $\mathcal{X}$.

a) ***Elapsed relocation time constraints****:* The semantic of the variable $\mathcal{R}_t$ as defined in IV-B1 can be enforced using the following constraints 1, 2, 3, 4, 5, and 6.

$$\mathcal{R}_0 = \mathcal{K}_0 \tag{1}$$

Constraint 1 ensures that the elapsed relocation time by the end of the first time slot $t = 0$ is set to either 0 or 1, depending on whether new relocation was triggered at the beginning of this time slot.

$$\forall t \in T \setminus \{0\} : \mathcal{R}_{t-1} + 1 - \overline{\mathcal{O}_t} \times \mathcal{M} \leq \mathcal{R}_t \tag{2}$$

$$\forall t \in T \setminus \{0\} : \mathcal{R}_t \leq \mathcal{R}_{t-1} + 1 + \overline{\mathcal{O}_t} \times \mathcal{M} \tag{3}$$

In case a previously triggered relocation is still ongoing by time step t, constraints 2 and 3 ensure that the elapsed relocation time at the end of time slot t is equal to the previously elapsed relocation time plus one (i.e., $\mathcal{R}_t = \mathcal{R}_{t-1} + 1$).

$$\forall t \in T \setminus \{0\} : -(1 - \overline{\mathcal{K}_t} \times \overline{\mathcal{O}_t}) \times \mathcal{M} \leq \mathcal{R}_t \tag{4}$$

$$\forall t \in T \setminus \{0\} : \mathcal{R}_t \leq (1 - \overline{\mathcal{K}_t} \times \overline{\mathcal{O}_t}) \times \mathcal{M} \tag{5}$$

In case there is no previously triggered relocation that is still ongoing by time step t and no new relocation is triggered at the beginning of time slot t, constraints 4 and 5 ensure that the elapsed relocation time at the end of time slot t is set to 0.

$$\forall t \in T \setminus \{0\} : 1 - (1 - \mathcal{K}_t \times \overline{\mathcal{O}_t}) \times \mathcal{M} \leq \mathcal{R}_t \tag{6}$$

$$\forall t \in T \setminus \{0\} : \mathcal{R}_t \leq 1 + (1 - \mathcal{K}_t \times \overline{\mathcal{O}_t}) \times \mathcal{M} \tag{7}$$

In case there is no previously triggered relocation that is still ongoing by time step t, and a new relocation is triggered at the beginning of time slot t, constraints 6 and 7 ensure that the elapsed relocation time at the end of time slot t is set to 1.

However, constrains 4 and 5 are not linear due to the product of the binary variables $\overline{\mathcal{K}_t} \times \overline{\mathcal{O}_t}$, and constraints 6 and 7 are not linear due to the product of the binary variables $\mathcal{K}_t \times \overline{\mathcal{O}_t}$. Details on how these constraints can be linearized are given later in Section IV-C.

b) ***Relocation status constraints****:* The values of the variables $\mathcal{O}_t$ as defined in Section IV-B1 can be enforced using the constraints and temporary variables defined in this subsection. We first define the temporary binary variables $\mathcal{X}_{1,t}$ and $\mathcal{X}_{2,t}$ as follows:

$$\forall t \in T \setminus \{0\}: \ \mathcal{X}_{1,t} = \begin{cases} 1 & \text{If } \mathcal{T} > \mathcal{R}_{t-1}. \\ 0 & \text{Otherwise.} \end{cases}; \ \mathcal{X}_{2,t} = \begin{cases} 1 & \text{If } \mathcal{R}_{t-1} > 0. \\ 0 & \text{Otherwise.} \end{cases}$$

$\mathcal{X}_{1,t}$ is subject to the following constraints:

$$\forall t \in T \setminus \{0\}: 0 \leq \mathcal{R}_{t-1} - \mathcal{T} + \mathcal{X}_{1,t} \times \mathcal{M} \tag{8}$$

$$\forall t \in T \setminus \{0\}: \mathcal{R}_{t-1} - \mathcal{T} + \mathcal{X}_{1,t} \times \mathcal{M} \leq \mathcal{M} - 1 \tag{9}$$

$\mathcal{X}_{2,t}$ is subject to the following constraints:

$$\forall t \in T \setminus \{0\}: 0 \leq -\mathcal{R}_{t-1} + \mathcal{X}_{2,t} \times \mathcal{M} \tag{10}$$

$$\forall t \in T \setminus \{0\}: -\mathcal{R}_{t-1} + \mathcal{X}_{2,t} \times \mathcal{M} \leq \mathcal{M} - 1 \tag{11}$$

The variable $\mathcal{O}_t$ is set to 1 only and only if $\mathcal{T} > \mathcal{R}_{t-1}$ and $\mathcal{R}_{t-1} > 0$. Thus, $\mathcal{O}_t$ can be expressed as the product of the two Boolean variables $\mathcal{X}_{1,t}$ and $\mathcal{X}_{2,t}$.

$$\mathcal{O}_0 = 0 \tag{12}$$

$$\forall t \in T \setminus \{0\}: \mathcal{O}_t = \mathcal{X}_{1,t} \times \mathcal{X}_{2,t} \tag{13}$$

The non-linear constraint 13 can be linearized by following the same linearization procedure as in section IV-C.

c) ***Triggered relocations constraints****:* A new relocation can be triggered only if the action $\mathcal{A}_t$ is different from the serving MEC host $\mathcal{S}_t$. Thus, the variable $\mathcal{K}_t$ is equal to 1 only and only if $\mathcal{A}_t \neq \mathcal{S}_t$. We first present the logical expression that captures the semantic of the variable $\mathcal{K}_t$, then we define the set of constraints that enforce it.

$$\mathcal{K}_t = (\mathcal{S}_t > \mathcal{A}_t) \vee (\mathcal{A}_t > \mathcal{S}_t) \tag{14}$$

We define the temporary variables $\mathcal{X}_{3,t}$ and $\mathcal{X}_{4,t}$ as follows:

$$\forall t \in T: \quad \mathcal{X}_{3,t} = \begin{cases} 1 & \text{If } \mathcal{S}_t > \mathcal{A}_t. \\ 0 & \text{Otherwise.} \end{cases}; \ \mathcal{X}_{4,t} = \begin{cases} 1 & \text{If } \mathcal{A}_t > \mathcal{S}_t. \\ 0 & \text{Otherwise.} \end{cases}$$

$\mathcal{X}_{3,t}$ is subject to the constraints 15 and 16.

$$\forall t \in T: 0 \leq \mathcal{A}_t - \mathcal{S}_t + \mathcal{M} \times \mathcal{X}_{3,t} \tag{15}$$

$$\forall t \in T: \mathcal{A}_t - \mathcal{S}_t + \mathcal{M} \times \mathcal{X}_{3,t} \leq \mathcal{M} - 1 \tag{16}$$

$\mathcal{X}_{4,t}$ is subject to the constraints 17 and 18.

$$\forall t \in T: 0 \leq \mathcal{S}_t - \mathcal{A}_t + \mathcal{M} \times \mathcal{X}_{4,t} \tag{17}$$

$$\forall t \in T: \mathcal{S}_t - \mathcal{A}_t + \mathcal{M} \times \mathcal{X}_{4,t} \leq \mathcal{M} - 1 \tag{18}$$

Finally, $\mathcal{K}_t$ can be expressed as a logical "or" of the variables $\mathcal{X}_{3,t}$ and $\mathcal{X}_{4,t}$, and it is subject to constraints 19, 20, and 21.

$$\forall t \in T: \mathcal{K}_t \geq \mathcal{X}_{3,t} \tag{19}$$

$$\forall t \in T: \mathcal{K}_t \geq \mathcal{X}_{4,t} \tag{20}$$

$$\forall t \in T: \mathcal{K}_t \leq \mathcal{X}_{3,t} + \mathcal{X}_{4,t} \tag{21}$$

d) ***Serving MEC host constraints****:* The variable $\mathcal{S}_t$ is subject to constraint 22 to 26.
$\forall t \in T \setminus \{0 \,..\, \mathcal{T}-1\}$:

$$\mathcal{A}_{t-\mathcal{T}} - \mathcal{X}_{1,t} \times \mathcal{M} \leq \mathcal{S}_t \tag{22}$$

$$\mathcal{S}_t \leq \mathcal{A}_{t-\mathcal{T}} + \mathcal{X}_{1,t} \times \mathcal{M} \tag{23}$$

$$\mathcal{S}_{t-1} - \overline{\mathcal{X}_{1,t}} \times \mathcal{M} \leq \mathcal{S}_t \tag{24}$$

$$\mathcal{S}_t \leq \mathcal{S}_{t-1} + \overline{\mathcal{X}_{1,t}} \times \mathcal{M} \tag{25}$$

$\forall t \in \{0 \,..\, \mathcal{T}-1\}$:

$$\mathcal{S}_t = H_0 \tag{26}$$

In case a relocation is completed by the end of time slot $t-1$ (i.e., $\mathcal{X}_{1,t}$ is equal to 0), constraints 22 and 23 ensures that $\mathcal{S}_t$ is set to the action $\mathcal{A}_{t-\mathcal{T}}$ taken $\mathcal{T}$ time slots earlier (i.e., $\mathcal{S}_t$ is updated to the relocation target). Moreover, constraints 24 and 25 ensure that $\mathcal{S}_t$ is set to $\mathcal{S}_{t-1}$ when no relocation is ongoing or a relocation still in progress (i.e., when $\mathcal{X}_{1,t}$ is equal to 1).

e) ***Relocation decisions constraints****:* The relocation decisions $\mathcal{A}_t$ taken by the MEO are subject to constraints 27 and 28.

$$\forall t \in T: \mathcal{A}_t \geq 1 \tag{27}$$

$$\forall t \in T: \mathcal{A}_t \leq |\mathcal{H}| \tag{28}$$

*f) **Optimal MEC host constraints**:* The variable $\mathcal{U}_t$ can be expressed logically as follows:

$$\mathcal{U}_t = \overline{(\mathcal{S}_t > \mathcal{L}_t)} \wedge \overline{(\mathcal{L}_t > \mathcal{S}_t)} \tag{29}$$

We define the temporary variables $\mathcal{X}_{5,t}$ and $\mathcal{X}_{6,t}$ as follows:

$$\forall t \in T: \quad \mathcal{X}_{5,t} = \begin{cases} 1 & \text{If } \mathcal{S}_t > \mathcal{L}_t. \\ 0 & \text{Otherwise.} \end{cases} ; \; \mathcal{X}_{6,t} = \begin{cases} 1 & \text{If } \mathcal{L}_t > \mathcal{S}_t. \\ 0 & \text{Otherwise.} \end{cases}$$

$\mathcal{X}_{5,t}$ is subject to the constraints 30 and 31.

$$\forall t \in T: 0 \leq \mathcal{L}_t - \mathcal{S}_t + \mathcal{M} \times \mathcal{X}_{5,t} \tag{30}$$

$$\forall t \in T: \mathcal{L}_t - \mathcal{S}_t + \mathcal{M} \times \mathcal{X}_{5,t} \leq \mathcal{M} - 1 \tag{31}$$

$\mathcal{X}_{6,t}$ is subject to the constraints 32 and 33.

$$\forall t \in T: 0 \leq \mathcal{S}_t - \mathcal{L}_t + \mathcal{M} \times \mathcal{X}_{6,t} \tag{32}$$

$$\forall t \in T: \mathcal{S}_t - \mathcal{L}_t + \mathcal{M} \times \mathcal{X}_{6,t} \leq \mathcal{M} - 1 \tag{33}$$

Finally, $\mathcal{U}_t$ can be expressed as the product of the compliments of the two variables $\mathcal{X}_{5,t}$ and $\mathcal{X}_{6,t}$.

$$\forall t \in T: \mathcal{U}_t = \overline{\mathcal{X}_{5,t}} \times \overline{\mathcal{X}_{6,t}} \tag{34}$$

Constraint 34 can be linearized by following the procedure described in section IV-C.

C. Constrains linearization

In this section, we briefly explain how constraints 4, 5, 6, 7, 13, and 34 can be linearized. Let $\mathcal{Y}_1$ and $\mathcal{Y}_2$ be two binary variables. The non-linear product $\mathcal{Y}_1 \times \mathcal{Y}_2$ can be captured using an additional Boolean variable $\mathcal{P}$ that is subject to the following constrains:

$$\mathcal{P} \leq \mathcal{Y}_1 \tag{35}$$

$$\mathcal{P} \leq \mathcal{Y}_2 \tag{36}$$

$$\mathcal{P} \geq \mathcal{Y}_1 + \mathcal{Y}_2 - 1 \tag{37}$$

Constrains 35 and 36 ensure that the value of $\mathcal{P}$ is set to 0 when either or both $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are equal to 0. Constraints 37 ensure that $\mathcal{P}$ is set to 1 only when both $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are equal to 1.

D. Final optimization model

The final model to optimize is given below, where the objective is to maximize the availability defined as the ratio of the number of time slots on which the UAV application is available in the optimal MEC host, on the total number of time slots.

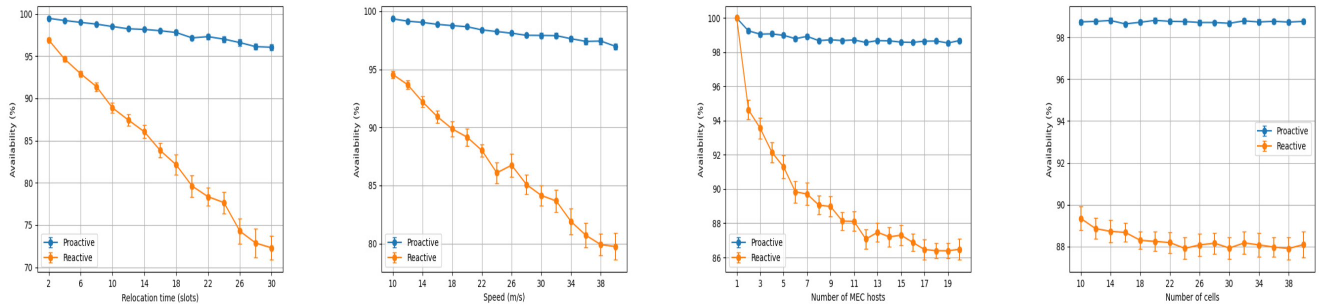
$$\mathbf{max} \frac{\sum_{t=0}^{T} \mathcal{U}_t}{T}$$

Subject to constraints:
(1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (15), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25), (26), (27), (28), (30), (31), (32), (33), and (34).

V. Evaluation

In this section, we evaluate our solution in terms of its ability to achieve its design goals under different configurations. It is to be noted that the plotted results present the mean and 95% confidence interval of the 100 repetitions, where in each repetition we consider a new random network topology and a new random flight plan. Unless otherwise specified, the network topology is composed of 25 BSs and 10 MEC hosts, randomly dispersed across a square geographical area of $10Km \times 10Km$. Each BS is associated with the nearest MEC host. The size of the time slots Δt is set to $1s$. Hereafter, we consider the "Reactive-Always relocate" algorithm discussed in III-A as a comparison baseline.

Fig. 2(a) shows the impact of different values of relocation time $\mathcal{T}$ on the availability of the UAV's application at the optimal MEC host. We notice that the performance of the reactive strategy drops off rapidly when increasing the relocation time from $2s$ to $30s$. This is mainly due to the increased number of "too late relocations". Whereas the proposed solution adapts itself to different values of relocation time, and its performance drops off slightly from 99% to 96%. The results depicted in 2(b) show the impact of different values of the speed on the performance of the reactive and proactive strategies. In the same way as for the relocation time, increasing the speed of the UAV from $10m/s$ to $40m/s$ causes a drop in the performance of the reactive strategy from 94.5% to 79%, whereas the performance of the proactive strategy is slightly impacted. This can be explained by the fact that the reactive strategy cannot keep up with the high speed of the UAV and fast successive HOs, whilst the proactive approach leverages the mobility plan to trigger the relocations before the UAV performs the HOs. Fig. 2(c) illustrates the impact of the number of available MEC hosts $|\mathcal{H}|$ on the performance of the two relocation strategies. In this experiment, we fixed the number of BSs to 20, and for each value of $|\mathcal{H}|$, we deploy the MEC hosts at random locations, then each BS is associated with the nearest MEC host. Increasing the number of MEC hosts implies increasing the need for relocating the applications. For instance, if only one MEC host is available in the network, there will be no need for relocation. However, increasing the number of MEC hosts will increase the probability of changing the serving MEC host and thus increasing the number of relocations. This explains the drop-off in the performance of the reactive strategy, as more late relocations will result in less availability. On the other hand, the performance of the proposed proactive solution is somewhat stable at 98.5%. The last experiment shows the impact of the number of the BSs $|\mathcal{B}|$ on the performance of the two relocation strategies. In this experiment, we fixed the number of MEC hosts to 9 and increased the density of BSs. For each value $|\mathcal{B}|$, we deploy the BSs at random locations, and then each BS is associated with the nearest MEC host. Increasing the number of BSs implies increasing the number of HOs. However, this does not imply increasing the number of relocations. For example, a set of neighboring



(a) Impact of relocation time on service availability.

(b) Impact of the speed of the UAV on service availability.

(c) Impact of the number of MEC hosts on service availability.

(d) Impact of the number of BSs on service availability.

Fig. 2. Performances evaluation.

BSs might be associated with the same MEC host, and thus no relocation is required when a HO occurs. This explains the results shown in Fig. 2(d) where both strategies present a somewhat stable performance when increasing the number of BSs. Nonetheless, the proactive strategy always outperforms the reactive strategy.

VI. Conclusion

This paper discussed the practical need for a proactive relocation strategy in MEC, especially for the use cases in which the mobility plan of the user can be made available. We studied the use case of UAVs, for which the mobility plans are obtained from the UTM system. We have first motivated the need for an optimized proactive relocation strategy by showcasing the limitations of the reactive and the non-optimized proactive relocation strategies. This has been addressed by formulating the PSRU problem as a LIP that provides the optimal relocation strategy. The performance evaluation of the proposed solution validated our claims for the possible introduced gains and the limitations of the reactive relocation strategies. Further investigation of the impact of errors in predicting of the residual time of the UAV in each cell, and probabilistic mobility plans, is left to future work.

Acknowledgment

This work was partially supported by the European Unions Horizon 2020 Research and Innovation Program through the 5G!Drones Project under Grant No. 857031. It was also supported in part by the Academy of Finland 6Genesis project under Grant No. 318927.

References

[1] ETSI GS MEC 003, "Multi-access Edge Computing (MEC); Framework and Reference Architecture," *ETSI GROUP SPECIFICATION*, 2020.

[2] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Network*, vol. 34, no. 1, pp. 84–93, 2020.

[3] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2018.

[4] T. Taleb and K. B. Letaief, "A cooperative diversity based handoff management scheme," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1462–1471, 2010.

[5] ETSI GR MEC 018, "Mobile Edge Computing (MEC); End to End Mobility Aspects ," *ETSI GROUP REPORT*, 2017.

[6] O. Bekkouche, M. Bagaa, and T. Taleb, "Toward a utm-based service orchestration for uavs in mec-nfv environment," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[7] S. Si-Mohammed, M. Bouaziz, H. Hellaoui, O. Bekkouche, A. Ksentini, T. Taleb, L. Tomaszewski, T. Lutz, G. Srinivasan, T. Jarvet, and P. Montowtt, "Supporting unmanned aerial vehicle services in 5g networks: New high-level architecture integrating 5g with u-space," *IEEE Vehicular Technology Magazine*, vol. 16, no. 1, pp. 57–65, 2021.

[8] O. Bekkouche, K. Samdanis, M. Bagaa, and T. Taleb, "A service-based architecture for enabling uav enhanced network services," *IEEE Network*, vol. 34, no. 4, pp. 328–335, 2020.

[9] T. Taleb, K. Adlen, H. Hellaoui, and O. Bekkouche, "On supporting uav based services in 5g and beyond mobile systems,," *IEEE Network*, (to appear).

[10] 3GPP TS 36.331, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," *3GPP Technical Specification*, 2020.

[11] 3GPP TR 36.777, "Enhanced LTE support for aerial vehicles," *3GPP Technical Report*, 2018.

[12] A. Ksentini, T. Taleb, and M. Chen, "A markov decision process-based service migration procedure for follow me cloud," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1350–1354.

[13] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge computing based on markov decision process," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019.

[14] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.

[15] R. A. Addad, D. L. C. Dutra, T. Taleb, and H. Flinck, "Ai-based network-aware service function chain migration in 5g and beyond networks," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021.

[16] ——, "Toward using reinforcement learning for trigger selection in network slice mobility," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2241–2253, 2021.

[17] Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, "Migration modeling and learning algorithms for containers in fog computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 712–725, 2019.

[18] W. Zhang, Y. Hu, Y. Zhang, and D. Raychaudhuri, "Segue: Quality of service aware edge cloud service migration," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 344–351.

[19] I. Farris, T. Taleb, H. Flinck, and A. Iera, "Providing ultra-short latency to user-centric 5g applications at the mobile network edge," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, p. e3169, 2018, e3169 ett.3169. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3169

[20] Y. Chen, X. Lin, T. Khan, and M. Mozaffari, "A deep learning approach to efficient drone mobility support," in *MobiCom '20: The 26th Annual International Conference on Mobile Computing and Networking*, ser. DroneCom '20. Association for Computing Machinery, 2020, p. 67–72.