Adaptive Resource Initialization for IoMT Task Offloading in NTNs

Alejandro Flores C., Konstantinos Ntontin, and Symeon Chatzinotas
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg.
e-mails:{alejandro.flores, kostantinos.ntontin, symeon.chatzinotas} @uni.lu

***Abstract*—In this work we study the offloading decision and computing resource allocation of tasks generated by internet-of-medical-things (IoMT) devices into a non-terrestrial network, comprised of local coordinating low-altitude platforms (LAPs) working also as multi-access edge computing (MEC) servers, and a common low Earth-orbit (LEO) satellite, which acts as a common MEC server across the LAPs. We solve the problem of total delay minimization across the tasks in the system. Given the NP-hard nature of the offloading decision problem, we solve it with a sequential greedy heuristic. To avoid biasing the offloading decision due to the computing resource initialization, we formulate a mechanism for dynamically initializing the resources at each step. We propose several methods for the computing resource initialization, and show in simulations, regions where each method is the most effective depending on the parameters of the system.**

***Index Terms*—multi-access edge computing (MEC), non-terrestrial networks, resource allocation, task offloading, internet of medical things (IoMT).**

I. Introduction

The internet-of-medical-things (IoMT) aims to interconnect internet-of-things (IoT) devices designed for healthcare purposes, to healthcare providers, and is envisioned as an enabler of the Healthcare 4.0 paradigm. In the context of remote patient monitoring (RPM), some examples of IoMT devices include continuous glucose monitoring sensors [1], photoplethysmography sensors [2] and portable electrocardiograms [3]. The continuous monitoring of biosignals of the patient allows for a more personalized healthcare. Particularly, it can help with the timely detection of adverse conditions within the patient, such as fever from temperature sensors [4], cough from acoustic sensors [5] and fatigue from heart rate [6]. IoMT devices can be resource constrained, for which the remote computing of their generated tasks at remote multiaccess edge computing (MEC) servers, allows for a better use of resources, with faster execution times.

Nevertheless, an IoMT device may find itself without access to the network in remote areas, or in situations of network overload. Indeed, only 15% of the surface of the Earth has mobile coverage [7]. Such scenarios benefit from the inclusion and use of non-terrestrial networks (NTNs). These networks comprise air nodes such as low-altitude platforms (LAPs), high altitude platfoms (HAPs), as well as space nodes, such as low Earth-orbit (LEO), medium Earth-orbit (MEO) or geostationary Earth-orbit (GEO) satellites. In particular, LEO satellites are closer to the surface of the Earth, for which the propagation delays and attenuation of wireless signals are less significant than for higher-altitude satellites.

The task offloading decision is an NP-hard problem due to its binary nature. Common approaches involve alternating optimization techniques in which the resources not optimized at a given stage are kept fixed. On the other hand, LEO satellites can have very large coverage regions, potentially providing service to a large amount of devices. However, LEO satellites do not have as many computing resources as terrestrial infrastructure, for operational reasons. Initializing the computing resources of a low-resource LEO MEC in a fixed manner across a large number of nodes may bias the nodes towards not offloading their tasks to the LEO satellite.

With this in mind, in this work we explore how to properly initialize and coordinate the computing resources of a LEO MEC so that they are effectively utilized across an increased number of ground IoMT devices for task offloading, with the objective of minimizing the total delay of the tasks across the system. In order to facilitate coordination, the ground IoMT devices are sectioned into clusters, of which, dedicated LAPs gather the tasks, which they can compute locally or further offload them to the common LEO satellite for remote computing. We develop a low-complexity greedy heuristic for the offloading decision problem, and propose several methods for dynamic resource initialization of the LEO satellite computing resources.

The remainder of this paper is structured as follows: Section II presents the related works, Section III introduces the system model together with the parameters of interest, whereas Section IV presents the optimization problem to be solved. Section V develops the resource allocation closed-form solution and offloading decision algorithm, while Section VI develops several mechanisms for the computing resource initialization of the resources at the LEO satellite. In Section VII simulation results are presented, and finally, in Section VIII the conclusions are drawn.

II. Related Works

There is extensive literature on the allocation of MEC resources, such as segment selection and caché [8] service placement [9] and demand-aware caching [10]. Particularly, the resource allocation in NTNs for task offloading is a research area that has raised interest in the last years. Considering single-layer solutions, [11] jointly optimizes the deployment and task offloading for MEC-enabled LAPs, [12] optimizes LAP deployment, IoT device association, task offloading and resource allocation in an industrial IoT setting, whereas in [13] the optimization of task offloading, computing resources, caching decisions and bandwidth in a HAPS-assisted and MEC-enabled intelligent transportation system, is considered. For LEO-enabled task offloading, while [14] considers the joint optimization of bandwidth and transmit power for LEO-assisted task offloading of data generated by IoT devices, [15] performs the joint optimization of association, transmit power, task scheduling and bandwidth in a three-tier satellite MEC, ground cloud infrastructure. Moreover, multi-layered NTNs have also been studied, such as in [16] where the computation offloading and resource allocation are jointly optimized in a multi-layer NTN with MEC-enabled HAPS connected through a LEO-enabled backhaul, and [17], which considers the joint optimization of access strategy, transmit power, computing resources and offloading in a space-air-ground integrated network.

III. System Model

Assume U clusters of IoMT devices distributed over a large area without access to the terrestrial infrastructure, where each cluster u is comprised of N_u IoMT nodes, and the set of IoMT devices is $\mathcal{I}$. To provide them with cellular access, a LAP from a set of LAPs $\mathcal{U}$, with $|\mathcal{U}| = U$, is deployed to each cluster to gather the tasks offloaded from their respective IoMT nodes. Each LAP gathers IoMT tasks and can either process them, or offload them to a LEO satellite acting as a MEC server through orthogonal channels, with the set of tasks gathered by LAP u being $\mathcal{I}_u$. Each task is characterized as $\boldsymbol{\psi}_i = [d_i, c_i]$, where d_i is the number of bits in the task and c_i is its computational density (in CPU cycles per bit). Each node is located in Cartesian coordinates $\mathbf{r}_k = [x_k, y_k, z_k]^T$, with $k \in \mathcal{I} \cup \mathcal{U} \cup \{s\}$, as seen in Fig. 1.

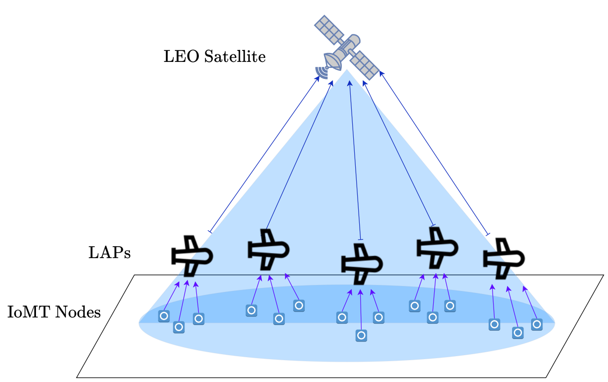

Fig. 1. System Model

Assume a fixed communication rate between LAP $u \in \mathcal{U}$ and the LEO satellite as $R_{u,s}$ which is broadly based on their distances to the LEO satellite and the current atmospheric conditions. Let β_i be a binary variable that indicates if task $\boldsymbol{\psi}_i$ is offloaded to the LEO satellite ($\beta_i = 1$) or not ($\beta_i = 0$). Then, the transmission delay of offloading all of the intended tasks from LAP u to the common LEO is given as

$$\tau_{u,s} = \frac{\sum\limits_{i \in \mathcal{I}_u} \beta_i d_i}{R_{u,s}} + 2\underbrace{\frac{||\mathbf{r}_k - \mathbf{r}_u||_2}{c}}_{\tau_{\mathrm{RTT}}}, \tag{1}$$

which considers the propagation delay of offloading the task and receiving its response, where c is the speed of light.

Each LAP u has F_u computing resources available to them, while the LEO satellite has F_s resources available for task computation. Assuming that task $\boldsymbol{\psi}_i$ is computed at node $k \in \mathcal{U} \cup \{s\}$, and is allocated a share of resources f_k^i, then the computing delay due to the processing of the task is given as

$$\tau_i^k = \frac{d_i c_i}{f_i^k} \tag{2}$$

Then, the total delay experienced by all the tasks in the system due to offloading and computing is given as

$$\tau = \sum_{i \in \mathcal{I}} \left(\beta_i \left(\tau_{u,s} + \tau_i^s\right) + (1 - \beta_i)\tau_i^u\right) \tag{3}$$

IV. Optimization Problem

The objective of the LAPs across the system is to minimize the total delay of all tasks handled by them. The computational resources at the LAPs are limited, so they make use of the resources of the LEO satellite to aid in the processing of the tasks within the system. To this end, the computing resources across the system, as well as the decision to offload each task to the LEO satellite or to compute it locally at its corresponding LAP are leveraged to minimize the total task delay. Let $\boldsymbol{\beta} = [\beta_1, ...\beta_I]$, $\mathbf{F} = [\mathbf{f}^1, ..., \mathbf{f}^U, \mathbf{f}^s]$, with $\mathbf{f}^k = [f_1^k, ..., f_I^k]^T$ for $k \in \mathcal{U} \cup \{s\}$. Then, the optimization problem to be solved is formulated as follows:

$$\mathcal{P}: \quad \min_{\boldsymbol{\beta}, \mathbf{F}} \quad \hat{\tau} \tag{4a}$$

$$\text{s.t.} \quad \mathrm{C}_1: \quad \sum_{i \in \mathcal{I}} f_i^k \leq F_k^{\max}, \quad \forall k \in \mathcal{U} \cup \{s\} \tag{4b}$$

$$\mathrm{C}_2: \quad f_i^k \geq 0, \quad \forall i \in \mathcal{I}, k \in \mathcal{U} \cup \{s\} \tag{4c}$$

$$\mathrm{C}_3: \quad \beta_i \in \{0, 1\}, \quad \forall i \in \mathcal{I} \tag{4d}$$

where C_1 constraints the maximum computing resources available at the LAPs and the LEO satellite, C_2 indicates that the computing resources are non-negative, and C_3 indicates that the decision variables are binary.

V. Offloading Decision and Resource Allocation Algorithm

Problem $\mathcal{P}$ is NP-hard, due to the binary offloading decision variables, thus, it is not convex and does not have a simple and efficient optimal solution. For this reason, we solve it as a two-stage optimization procedure: first, we solve the offloading decision problem through an iterative, dynamic initialization heuristic, and then, we solve the optimal computing resource allocation at each LAP and the LEO satellite. For simplicity, the offloading decision algorithm is solved centrally at a single designated LAP, which acts as a coordinator and sends the offloading results to every other LAP and the LEO, while the resource allocation algorithm is ran at every computing node parallelly.

A. Optimal Computing Resource Allocation

Consider node $k \in \mathcal{U} \cup \{s\}$, which allocates its available resources F_k to its offloaded tasks $\mathcal{I}^k$. Then, to minimize its total task delay, we consider problem $\mathcal{P}$ with only the terms related to the computing resources, such that node k solves the following convex optimization problem

$$\min_{\{f_k^i\}_{i\in\mathcal{I}^k}} \quad \sum_{i\in\mathcal{I}^k} \frac{d_i c_i}{f_k^i} \tag{5a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}^k} f_k^i \leq F_k \tag{5b}$$

$$f_k^i \geq 0 \qquad \forall i \in \mathcal{I}^k \tag{5c}$$

We write the partial Lagrangian considering the first constraint as

$$\mathcal{L}(\mathbf{f}, \lambda) = \sum_{i\in\mathcal{I}^k} \frac{d_i c_i}{f_k^i} + \lambda \left(\sum_{i\in\mathcal{I}_n} f_k^i - F_k^n \right) \tag{6}$$

Taking the derivative of the partial Lagrangian with respect to f_k^i and equaling to zero we obtain

$$f_k^i = \frac{1}{\sqrt{\lambda}} \sqrt{d_i c_i} \tag{7}$$

Replacing this expression in constraint (5b) and working out the algebra, we obtain

$$\frac{1}{\sqrt{\lambda}} = \frac{F_k}{\sum\limits_{i\in\mathcal{I}^k} \sqrt{d_i c_i}} \tag{8}$$

Let us write for simplicity $\xi_i = \sqrt{d_i c_i}$. Then, replacing in (7), the optimal point is given as

$$f_k^{i,*} = \frac{\xi_i}{\sum\limits_{j\in\mathcal{I}^k} \xi_j} F_k \qquad \forall i \in \mathcal{I}^k \tag{9}$$

B. Optimal Offloading Decision

We can write problem $\mathcal{P}$ considering only the offloading decision variables, and with some algebraic manipulation, as

$$\mathcal{P}: \quad \min_{\mathbf{b},\mathbf{F}} \sum_{i\in\mathcal{I}} \beta_i \left(d_i c_i \left(\frac{1}{f_i^s} - \frac{1}{f_i^u} \right) + \frac{d_i}{R_{u,s}} + \frac{\sum\limits_{\substack{j\in\mathcal{I}_u \\ j\neq i}} \beta_j d_j}{R_{u,s}} + \tau_{\mathrm{RTT}} \right) \tag{10a}$$

$$\text{s.t.} \quad \beta_i \in \{0, 1\}, \quad \forall i \in \mathcal{I} \tag{10b}$$

We can write the objective function as $\Phi = \sum_{i\in\mathcal{I}} \beta_i \Phi_i$. The value of Φ_i is negative when the delay from offloading and computing in the LEO satellite is smaller than locally computing at the LAP. Then, as a low-complexity solution to the NP-hard offloading decision problem, we adopt a simple greedy heuristic in which we choose iteratively the tasks with the lowest, most negative, value of Φ_i with fixed β_j values for the rest of the tasks. With this heuristic, at each step, the task that presents the best improvement from being offloaded to the LEO, is the one set for offloading, until there is no task in the system that would decrease its delay by offloading to the LEO.

Furthermore, the initialization of the computing resources at the LEO can bias the offloading strategy obtained. For that reason, we adopt a dynamic initialization approach in which the computing resources at the LEO are initialized before every task allocation. The details on this initialization are presented in the next section. An outline of the greedy offloading heuristic can be seen in Algorithm 1.

Algorithm 1: Greedy Computing-Adaptive Offfloading Strategy

foreach $u \in \mathcal{U}$ **do**
- Initialize $\boldsymbol{\beta}_u = \mathbf{0}_{I_u \times 1}$;
- Initialize $\hat{\mathcal{I}}_u = \mathcal{I}_u$;

repeat
- **foreach** $u \in \mathcal{U}$ **do**
  - Initialize f_i^u as in (9) $\forall i \in \hat{\mathcal{I}}_u$;
  - Initialize f_i^s as in Section VI $\forall i \in \hat{\mathcal{I}}_u$;
  - Compute Φ_i as in (10a) $\forall i \in \hat{\mathcal{I}}_u$;
- **if** $\min\{\Phi_i\}_{i\in\mathcal{I}} \geq 0$, **then return**;
- Choose $i^* = \operatorname{argmin}\{\Phi_i\}_{i\in\mathcal{I}}$;
- Set $\hat{\mathcal{I}}_{u^*} := \hat{\mathcal{I}}_{u^*} \setminus \{i^*\}$;
- Set $\beta_{i^*} = 1$;
- Update $F_k := F_k - f_{i^*}^s$;

VI. Adaptive Non-Orthogonal Computing Resource Initialization

The offloading decision problem highly depends on the initialization of the computing resources from available computing nodes (LAPs and LEO satellite), and can highly bias the offloading decision. Assume that the computing resources in s are initialized as f_i^s for all the tasks in the system $i \in \mathcal{I}$. If the initialized resources at s for task i are less than the computing resources allocated by its LAP, that is $f_i^s < f_i^u$, then the task will never be offloaded. This is clear by the fact that smaller computation resources imply larger computation delay, and that accessing the LEO satellite incurs in higher delay due to transmission and propagation. If the number of nodes $|\mathcal{I}|$ is large, and the resources available F_s are not too large, then it is clear that having $f_i^s < f_i^u$ for all $i \in \mathcal{I}$ is likely, even when $F_s >> f_i^u$ for any given i. In this situation, no task will be offloaded to s, and its resources are not utilized, even when there is a clear advantage to using them.

To solve this problem, we propose an overlapping resource advertisement of the computing resources of the LEO satellite, to all of the LAPs. Let F_s^u be the resources of the LEO satellite that are advertised to LAP u, so that it initializes the resources for its corresponding tasks f_i^s for $i \in \mathcal{I}_u$. We set the values of F_s^u such that $\sum_{u\in\mathcal{U}} F_s^u > F_s$, which is to say, that the total resources advertised by the LEO satellite, are larger than the ones available. By allocating one task at a time, the situation in which more resources than available at the LEO satellite are allocated, is prevented.

A. Overlapping Resource Advertisement Model

The share of resources of s allocated to u can be expressed as

$$F_s^u = \frac{\nu_u}{\sum\limits_{m \in \mathcal{U}} \nu_m} \theta F_s \qquad \forall u \in \mathcal{U} \tag{11}$$

where ν_u is the corresponding weight of LAP u and θ is a term that amplifies or reduces the perceived resources of s across the system. If $\theta \leq 1$, then the resources of s across the system are advertised without overlap, whereas if $\theta > 1$ there is resource overlap across LAPs.

B. Total Overlapping Resources

If we set $\theta = U$ and $\nu_u = 1$ for all $u \in \mathcal{U}$, we obtain a total overlap of the LEO resources, in which every LAP can freely allocate all of the resources available at the LEO satellite having $F_s^u = F_s$ for all $u \in \mathcal{U}$. However, there are some issues with this method:

- **Early LAP biasing:** when a given LAP u allocates a task to s, the total resources F_s are reduced. This implies that, in the next allocation step $f_j^s(t+1) < f_j^s(t)$ for all $j \in \mathcal{I} \setminus \mathcal{I}_u$, while $f_i^s(t+1) = f_i^s(t)$ for $i \in \mathcal{I}_u$. This biases the resources of s in favor of the tasks in $\mathcal{I}_u$, so it is more likely that the next allocation of resources from s is also given to a node in $\mathcal{I}_u$. This leads to a higher likelihood of all of the resources from s being allocated solely to $\mathcal{I}_u$.
- **Early resource depletion:** if one of the LAPs u has only one task to be offloaded $\boldsymbol{\psi}_i$, then all of the remaining resources of s will be initialized completely to task $\boldsymbol{\psi}_i$ as $f_i^s = F_s$. This will make it more likely that task $\boldsymbol{\psi}_i$ is offloaded to the remote node that has the larger resource pool available. This is an issue, because it may waste a very large pool of resources in a single task, which otherwise could be used to serve a larger number of tasks.

These problems cause an uneven and suboptimal use of the resources of the remote nodes.

C. Optimality-based Overlapping Resources

Another natural choice for ν_u, which would follow an optimal allocation across the whole system is

$$\nu_u = \sum_{i \in \mathcal{I}_u} \xi_i \tag{12}$$

so that the resources of remote node k allocated for device i would be given as

$$f_s^i = \frac{\xi_i}{\sum\limits_{j \in \mathcal{I}_u} \xi_j} \left(\frac{\nu_u}{\sum\limits_{m \in \mathcal{U}} \nu_m} \theta F_s \right) \tag{13}$$

$$= \frac{\xi_i}{\sum\limits_{j \in \mathcal{I}_u} \xi_j} \left(\frac{\sum\limits_{i \in \mathcal{I}_u} \xi_i}{\sum\limits_{m \in \mathcal{U}} \sum\limits_{j \in \mathcal{I}_m} \xi_j} \theta F_s \right) \tag{14}$$

$$= \frac{\xi_i}{\sum\limits_{j \in \mathcal{I}} \xi_j} \theta F_s \tag{15}$$

Nevertheless, this type of allocation does not guarantee that a node is allocated more resources than there are available at s for an arbitrary θ. To ensure this, we derive the following condition

$$f_s^i \leq F_s \qquad \forall i \in \mathcal{I} \tag{16}$$

$$\frac{\xi_i}{\sum\limits_{j \in \mathcal{I}} \xi_j} \theta F_s \leq F_s \qquad \forall i \in \mathcal{I} \tag{17}$$

$$\theta \leq \frac{\sum\limits_{j \in \mathcal{I}} \xi_j}{\xi_i} \qquad \forall i \in \mathcal{I} \tag{18}$$

which can be accomplished by setting

$$\theta \leq \frac{\sum\limits_{j \in \mathcal{I}} \xi_j}{\max\limits_{i \in \mathcal{I}} \xi_i} \tag{19}$$

However, this value of θ guarantees that at each iteration at least one of the tasks in the system is allocated all of the resources of s. This is desirable when all of the LAPs have a single task left to allocate, but not in any other situation, since it would potentially cause immediate resource depletion. Thus, let us introduce an adaptive parameter $0 \leq \eta \leq 1$ such that

$$\theta = \eta \frac{\sum\limits_{j \in \mathcal{I}} \xi_j}{\max\limits_{i \in \mathcal{I}} \xi_i} \tag{20}$$

Thus, by adjusting the η parameter we can control the maximum amount of the total resources of the LEO satellite that are initialized to any given node.

1) Fixed $\eta = 1$: By setting $\eta = 1$, the resources of s at a given task i are initialized as

$$f_s^i = \frac{\xi_i}{\sum\limits_{j \in \mathcal{I}} \xi_j} \left(\frac{\sum\limits_{j \in \mathcal{I}} \xi_j}{\max\limits_{j \in \mathcal{I}} \xi_j} \right) F_s \tag{21}$$

$$= \frac{\xi_i}{\max\limits_{j \in \mathcal{I}} \xi_j} F_s. \tag{22}$$

This causes the most amount of resources to be allocated to the tasks that have a larger amount of CPU cycles, allocating all of the resources to the task with the largest amount of cycles. This manner of choosing η risks an immediate depletion of resources after a single task allocation.

2) Fixed $\eta < 1$: The resources of s at a given task i are initialized as

$$f_s^i = \eta \frac{\xi_i}{\max\limits_{j \in \mathcal{I}} \xi_j} F_s. \tag{23}$$

Here, the same behavior is followed, in which the most amount of resources to be allocated to the tasks that have a larger amount of CPU cycles. However, the task with the largest amount of cycles has only a proportion of η of the total resources of F_s initialized. This avoids an early depletion of resources at s, however it also avoids a future full usage of its resources.

3) Adaptive η : To avoid an early depletion of resources, but also allowing more resources to be advertised when there are fewer tasks to be allocated, we design a heuristic for choosing an appropriate value of η after each allocation. We choose the value of η such that the resources initialized across the tasks of any LAP do not exceed the total amount of resources at the LEO satellite.

$$\sum_{i\in\mathcal{I}_u} f_s^i = \sum_{i\in\mathcal{I}_u} \eta \frac{\xi_i}{\max\limits_{j\in\mathcal{I}} \xi_j} F_s \leq F_s \qquad \forall u \in \mathcal{U} \tag{24}$$

$$\Rightarrow \eta \sum_{i\in\mathcal{I}_u} \frac{\xi_i}{\max\limits_{j\in\mathcal{I}} \xi_j} \leq 1 \qquad \forall u \in \mathcal{U} \tag{25}$$

$$\Rightarrow \eta \leq \frac{\max\limits_{j\in\mathcal{I}} \xi_j}{\sum\limits_{i\in\mathcal{I}_u} \xi_i} \qquad \forall u \in \mathcal{U} \tag{26}$$

$$\Rightarrow \eta = \frac{\max\limits_{j\in\mathcal{I}} \xi_j}{\max\limits_{u\in\mathcal{U}} \sum\limits_{i\in\mathcal{I}_u} \xi_i} \tag{27}$$

With this heuristic, the initializations can be written as

$$f_s^i = \frac{\xi_i}{\max\limits_{u\in\mathcal{U}} \sum\limits_{j\in\mathcal{I}_u} \xi_j} F_s \tag{28}$$

VII. Results

We present results gathered from computing simulations performed in Matlab, showing the total task delay obtained by different initialization methods. The methods used in simulations are

- Non-Overlapping Optimal: as in (9) across all of the tasks.
- Total Overlapping: as in (11) with $\theta = U$ and $\nu_u = 1$ for all $u \in \mathcal{U}$.
- Adaptive η: as in (28).
- Custom η value: as in (23).

As per 3GPP recommendations [18], the carrier frequency for the LAP-LEO link is in the Ka band, chosen as 28GHz with bandwidth $B_s = 200$MHz. We assume 16 clusters with 14 IoMT devices per cluster generating tasks of parameters $c_i = 100$ and d_i following an uniform distribution $U(100\text{Kb}, 10\text{Mb})$, where larger task sizes correspond to data acquired from IoMT devices with high computational demands, such as wearable ultrasound systems used for image processing or continuous high-frequency time-series data from wearable ECG monitors. In contrast, smaller task sizes are associated with less intensive workloads, including time-series processing from IoMT devices such as wireless stethoscopes or wearable pulse oximeters. We consider the altitude of LAPs as $120m$, which is the maximum allowed altitude by the European Union Aviation Safety Agency (EASA) [19]. We consider a LEO satellite at an altitude of 500 km at horizontal coordinates $[0, 0]$.

In Fig. 2 the task average delay is plotted for varying LAP computing resources F_u, under the different resource initialization modes. The available resources at the LEO satellite are of $F_s = 1$GHz. It can be clearly seen that different initialization modes perform best under different conditions of the resources at the LAPs. When the resources at the LAPs are small, the resources at the LEO satellite are much larger, thus the non-overlapping initialization has good performance, since this is enough to incentivize tasks being offloaded to the LEO satellite. This is also the case for small η values, which allow the resources of the LEO satellite to be distributed across a larger number of tasks without depleting the LEO resources. When the computing resources at the LAPs increase, the better initialization methods are the adaptive and total overlapping, in which the resources of the LEO satellite are adaptively given to tasks that benefit the most from a bigger share of resources from the LEO satellite. Also, note that the very large average delays on the left region are due to the large-sized tasks, by the computing of tasks at the LAP with very low computing resources, and by the overall low amount of computing resources in the system, for a large number of tasks.

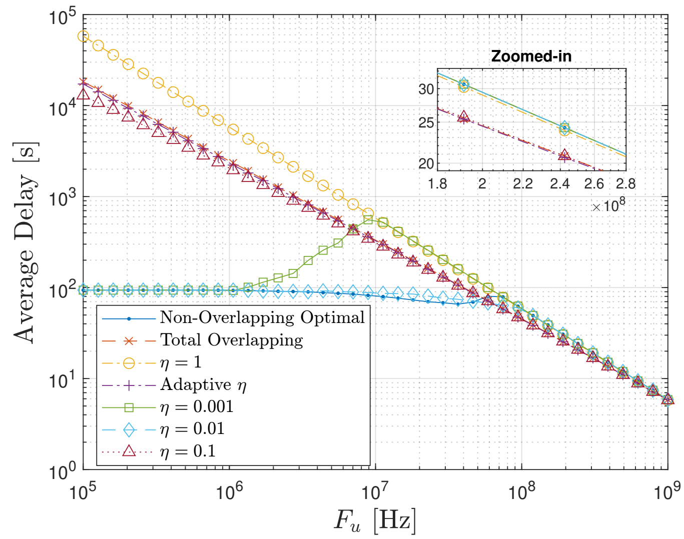

Fig. 2. Average delay of tasks for varying LA available resources F_u.

To show how the choice of value of η impacts on the performance of the algorithm, we show Fig. 3. In this figure, we show the delay obtained by using custom η values, for different amount of resources at the LEO satellite, assuming $F_u = 100$MHz. The performance of the adaptive η method is also shown. In all of the custom η curves, it can be seen that there is a global minimum point. To the left, the value of η is too small, so that it is not beneficial for the tasks to offload, whereas to the right, the value of η is too large, so that there is an earlier depletion of resources unto fewer tasks. Moreover, as the amount of resources available at the LEO satellite are larger, the region of minimum delay is wider, and is obtained further to the left, given that the resources are large enough so that, even when advertised only a small fraction, it is enough for the tasks in the system to prefer offloading to the LEO satellite. Moreover, the adaptive η curves show regions where they outperform the custom η curves, but as F_s increases, the

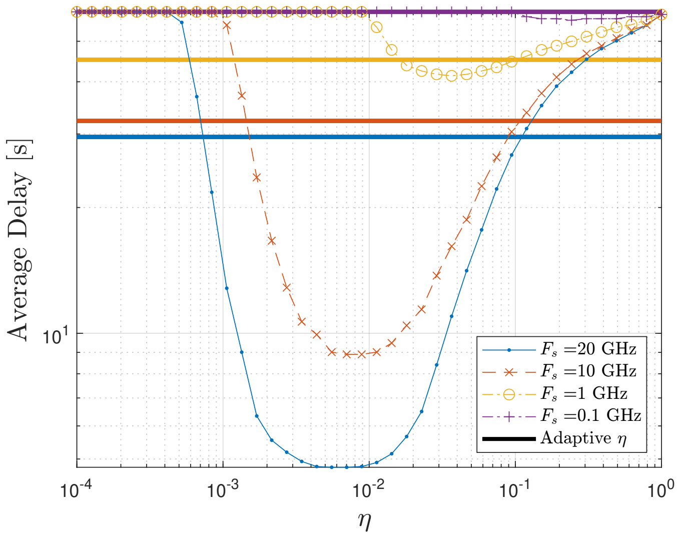

Fig. 3. Average delay of tasks for varying overlapping parameter η and several LEO satellite resources F_s.

regions where custom η outperforms the adaptive case also grow wider. However, for optimal performance, the value of a fix η is chosen empirically.

VIII. Conclusions

In this work we studied the problem of task offloading decision and computing resource allocation of tasks generated by IoMT devices, unto an NTN comprised of LAPs deployed as local coordinators for clusters of ground IoMT devices, which also work as MEC servers, and a common LEO satellite working as a common MEC server. To minimize the total task delay across the tasks in the system, we designed a greedy heuristic for the offloading decision problem, while solving the per-MEC resource allocation problem in closed form. The offloading decision heuristic implements a dynamic resource initialization mechanism in which the computing resources of the LEO satellite are initialized after each task allocation. To allow for better advertisement of the computing resources at the LEO, several resource initialization methods were developed, and simulated to show under which conditions they perform best regarding the resources available at the LAPs and the LEO satellite.

Acknowledgement

This work received funding from Horizon Europe under the Marie Sklodowska Curie actions: ELIXIRION (GA 101120135).

References

[1] B. M. Ahmed, M. E. Ali, M. M. Masud, and M. Naznin, "Recent trends and techniques of blood glucose level prediction for diabetes control," *Smart Health*, vol. 32, p. 100457, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352648324000126

[2] S. A. Lubitz, A. Z. Faranesh, C. Selvaggi, S. J. Atlas, D. D. McManus, D. E. Singer, S. Pagoto, M. V. McConnell, A. Pantelopoulos, and A. S. Foulkes, "Detection of atrial fibrillation in a large population using wearable devices: The fitbit heart study," *Circulation*, vol. 146, no. 19, pp. 1415–1424, 2022.

[3] W. Qi and H. Su, "A cybertwin based multimodal network for ecg patterns monitoring using deep learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6663–6670, 2022.

[4] J. Medina, M. Espinilla, A. García-Fernández, and L. Martínez, "Intelligent multi-dose medication controller for fever: From wearable devices to remote dispensers," *Computers & Electrical Engineering*, vol. 65, pp. 400–412, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579061730561X

[5] Y. Umayahara, Z. Soh, K. Sekikawa, T. Kawae, A. Otsuka, and T. Tsuji, "A mobile cough strength evaluation device using cough sounds," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3810

[6] D. Ichwana, R. Z. Ikhlas, and S. Ekariani, "Heart rate monitoring system during physical exercise for fatigue warning using non-invasive wearable sensor," in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2018, pp. 497–502.

[7] World Economic Forum, "What is a satellite-connected smartphone – and how can it help bridge the digital divide?" https://www.weforum.org/agenda/2022/10/satellite-connected-smartphones-digital-divide/, October 2022, accessed: 2024-10-16. [Online]. Available: https://www.weforum.org/agenda/2022/10/satellite-connected-smartphones-digital-divide/

[8] D. Xenakis, "To dash, or not to dash? optimal video bitrate selection and edge network caching in mec-empowered slice-enabled networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5556–5571, 2024.

[9] T. Mihretu Ayenew, D. Xenakis, N. Passas, and L. Merakos, "A novel content placement strategy for heterogeneous cellular networks with small cells," *IEEE Networking Letters*, vol. 2, no. 1, pp. 10–13, 2020.

[10] T. Mihretu Ayenew, D. Xenakis, L. Alonso, N. Passas, and L. Merakos, "Demand-aware cooperative content caching in 5g/6g networks with mec-enabled edges," *IEEE Networking Letters*, vol. 4, no. 3, pp. 118–122, 2022.

[11] J. Zhang, H. Luo, X. Chen, H. Shen, and L. Guo, "Minimizing response delay in uav-assisted mobile edge computing by joint uav deployment and computation offloading," *IEEE Transactions on Cloud Computing*, vol. 12, no. 4, pp. 1372–1386, 2024.

[12] M. Hevesli, A. M. Seid, A. Erbad, and M. Abdallah, "Task offloading optimization in digital twin assisted mec-enabled air–ground iiot 6g networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17 527–17 542, 2024.

[13] Q. Ren, O. Abbasi, G. K. Kurt, H. Yanikomeroglu, and J. Chen, "Caching and computation offloading in high altitude platform station (haps) assisted intelligent transportation systems," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9010–9024, 2022.

[14] Y. Lyu, Z. Liu, R. Fan, C. Zhan, H. Hu, and J. An, "Optimal computation offloading in collaborative leo-iot enabled mec: A multiagent deep reinforcement learning approach," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 2, pp. 996–1011, 2023.

[15] Y. Zhang, H. Zhang, K. Sun, J. Huo, N. Wang, and V. C. M. Leung, "Partial computation offloading in satellite-based three-tier cloud-edge integration networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 2, pp. 836–847, 2024.

[16] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, "Computation offloading and resource allocation in mec-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 478–21 491, 2022.

[17] Q. Chen, W. Meng, T. Q. S. Quek, and S. Chen, "Multi-tier hybrid offloading for computation-aware iot applications in civil aircraft-augmented sagin," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 399–417, 2023.

[18] 3GPP, "3rd generation partnership project; technical specification group radio access network; nr; user equipment (ue) radio transmission and reception; part 5: Satellite access radio frequency (rf) and performance requirements (release 18)," 3GPP, Tech. Rep. TS 38.101-5 V18.7.0 (2024-09), 2024.

[19] European Union, "Commission implementing regulation (eu) 2019/947 of 24 may 2019 on the rules and procedures for the operation of unmanned aircraft," 2019, accessed: 2024-11-08. [Online]. Available: https://eur-lex.europa.eu/eli/reg_impl/2019/947/oj