ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838752

An Incentivization Mechanism for Green Computing Continuum of Delay-Tolerant Tasks

Maria Diamanti*, Eirini Eleni Tsiropoulou†, and Symeon Papavassiliou*
{mdiamanti@netmode.ntua.gr, eirini@unm.edu, papavass@mail.ntua.gr}
* School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
† Dept. of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

***Abstract*—Capitalizing on the different available computing options across the network, the concept of computing continuum has recently emerged to efficiently manage the exaggerated computation demands of the numerous Internet-of-Things (IoT) users and applications. Nevertheless, the edge computing's attractiveness to the users, in terms of its reduced incurred time and energy overhead, acts as an impediment in the realization of the envisioned computing continuum. In this paper, recognizing the potential of forwarding delay-tolerant tasks to upper computing layers, we design an incentivization-based mechanism for the offloading users, aiming to shift their preference from the edge to the upper fog computing layer. The corresponding mechanism comprises two stages, in which different models of Contract Theory are adopted. In the first stage, a users-to-edge server contract is formulated to determine the optimal amount of each user's initially offloaded task at the edge that is allowed to be further forwarded to the fog, based on the user's delay tolerance. Subsequently, an edge-to-fog server contract is formulated to account for the edge server's tradeoff between the local execution and transmission overheads, deriving the most beneficial amount of the users' tasks that ultimately reaches the fog. The overall mechanism is evaluated via modeling and simulation regarding its operation and efficiency under different scenarios.**

***Index Terms*—Computing Continuum, Delay-Tolerant Computing, Contract Theory, Adverse Selection, Moral Hazard.**

I. Introduction

The ubiquity of intelligence that is required for the operation of an Internet-of-Things (IoT) environment has provoked the increase of computationally intensive user applications. To facilitate the computationally constrained user devices, while at the same time reducing the computation time and energy overheads, the concept of computation task offloading at the edge has become extremely popular. Nevertheless, the overexploitation of the edge servers will gradually lead to their performance degradation and increased energy consumption. This practically shifts the overall computational burden from the user devices to the finite-capacity edge servers. To alleviate the traffic and ameliorate the overall system's energy efficiency, a heterogeneous multi-layer computing architecture is envisioned, in which servers from different computing layers (e.g., edge, fog and cloud) across the network, cooperate with each other, realizing the concept of computing continuum [1].

Indeed, the diversity that characterizes the offloaded tasks in terms of their intensity and delay (in)sensitivity creates a solid ground for the exploitation of the different computing options within the computing continuum. An appropriate candidate that can leverage this concept and its merits is the delay-tolerant tasks, which can be further offloaded from the edge at the fog - situated anywhere between the network edge and the cloud [2] - or even the cloud, without violating the Quality-of-Service (QoS) requirements of the user application. Nevertheless, the edge computing's appealing attributes related to its proximity to the users (e.g., reduced incurred transmission cost), along with the users' selfish behavior, typically acting as strict utility and personal satisfaction maximizers, come against the cooperative principle that the computing continuum seeks to cultivate among the different computing layers.

In this paper, we aim to exactly fill this gap, by designing and proposing an incentivization-based mechanism, which determines the amount of each user's initially offloaded task at an edge server that can be further forwarded for computation at a fog server, based on the user's delay tolerance. The mechanism includes two stages, in which appropriate contracts following the principles of Contract Theory [3] are designed to capture the users-to-edge server and edge-to-fog server relationships, respectively, realizing a delay-tolerant computing paradigm. The ultimate goal of our work and proposed mechanism is to promote the computing continuum and thus, enhance the resource utilization efficiency across the network.

A. Related Work

There exist some works in the literature that introduce the cooperation between different computing layers, such as edge-fog, edge-cloud or fog-cloud, and deal with different resource allocation problems. For instance, in [4], the authors suggest the migration of frequently invoked tasks from the cloud to the edge in order to minimize the response time. In [5], the cooperation among different fog servers, as well as the cooperation between the fog and the cloud is considered for the execution of computationally- intensive and delay-sensitive services. Following a similar cooperation approach among different distributed edge/fog and cloud servers, the problem of application partitioning and placement is studied in [6], to minimize the execution time and energy consumption of resource-hungry applications. However, the aforementioned

This research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: HFRI-FM17-2436).

works scrutinize the prospect of computation offloading under delay-sensitive tasks, when a single computing layer struggles to meet the specific QoS prerequisites. On the contrary, our objective is to promote the usage of the whole spectrum of computing continuum under delay-tolerant tasks, as a means of better resource and energy utilization across the network.

Further improving the overall available computing resource utilization efficiency, by shifting the users' preference to upper computing layers - when the offloaded tasks' delay tolerance allows for - calls for the creation and provisioning of appropriate incentives. In this context, Contract Theory [3] has been widely adopted and employed under different settings and applications. Recent efforts pertaining to the incentives behind computation offloading can be found in [7], [8]. In the former, the cooperation between potential offloading request nodes and offloading nodes under a Heterogeneous Cloud Radio Access Network (H-CRAN) setting is studied, whereas in the latter, a hierarchical computation offloading framework is developed. Based on this hierarchical framework, the edge computing operator seeks to incentivize potential temporary edge nodes to take over the computation offloading of the users. Another interesting approach, different from the computation offloading domain, but tailored to the delayed traffic offloading in cellular networks, is presented in [9]. This work suggests that the users capitalize on their delay and price sensitivity and forward their traffic through the available Delay Tolerant Networks (DTNs) or WiFi networks, in exchange for reduced service cost.

B. Contributions & Outline

Despite the popularity that computation offloading has gained over the past years, the problem of incentives towards a green computing continuum under delay-tolerant task execution remains notably unexplored. In this paper, our objective is to design and propose an incentivization mechanism that promotes the users-to-edge server and edge-to-fog server cooperation, towards better utilizing the computational resources across the network and hence, increasing the overall system's efficiency. The ultimate purpose of the mechanism is to determine the amount of offloaded tasks by the users at the edge server that can be further forwarded to the fog for processing, based on the users'/tasks' delay tolerance and computational intensity.

In particular, a two-stage incentivization mechanism is proposed based on Contract Theory. In the first stage, a contract between the users and the edge server is designed to solve the Adverse Selection problem, according to which each user, having private information about its offloaded task's delay tolerance and intensity, autonomously selects the most appropriate amount of task that can be forwarded to the fog in exchange for some service discount. In the second stage, a contract between the edge and fog servers is formulated to solve the Moral Hazard problem, according to which the edge server scrutinizes the tradeoff between the local execution and the transmission energy overhead of the tasks that are allowed to be offloaded at the fog. The outcome of the second stage is the most beneficial amount of tasks for both the users and the edge server that reaches the fog. Indicative simulation results of the two contract-theoretic stages are provided, demonstrating the main characteristics of the designed contracts.

The remainder of this paper is organized as follows. Section II presents the system model and the contract bundles and utilities related to the contracts of the two stages. Section III introduces the formulation of the Adverse selection problem and the users-to-edge server contract, whereas in Section IV the formulation of the Moral Hazard and the edge-to-fog server contract is provided. Finally, Section V presents the performance evaluation and Section VI concludes the paper.

II. Framework & System Model

A computing continuum environment is considered, consisting of a set of users $N = \{1, \ldots, |N|\}$, a single edge server and a single fog server. Each user n has a computing application $A_n = (B_n, C_n, \phi_n, T_n)$, where B_n [Bytes] denotes the total input bytes, C_n [CPU cycles] is the number of CPU cycles required for the application's execution referred to as "task" in the following, ϕ_n [CPU cycles/Byte] indicates the application's intensity such that $C_n = \phi_n B_n$ and T_n [msec] is the application's/task's completion time requirement that characterizes its level of delay tolerance. We assume that each user n communicates with the edge server, which is in close proximity to the users, and offloads the task C_n for remote computation. Then, the fog server that is considered to lie between the edge and cloud/core network [2], serves - among others - the purpose of computation alleviation of the edge. As a result, assuming that a task C_n can be arbitrarily partitioned into subsets of any size, part of the tasks that have been initially offloaded at the edge, can be further forwarded and processed at the fog.

A. Users-to-Edge Server Contract Bundle & Utilities

Each user communicating with the edge server is characterized by a type, which captures the tradeoff between its application's delay-tolerance and intensity levels and is defined as $\theta_n = w_1 \frac{T_n}{\max\{T_n, \forall n \in N\}} + w_2 \frac{\phi_n}{\max\{\phi_n, \forall n \in N\}}, \theta_n \in (0, 1]$, where $w_1, w_2 \in \mathbb{R}^+$ are appropriate weight factors, such that $w_1 + w_2 = 1$. According to its type, each user provides its effort to the edge server, i.e., the percentage of the task C_n that can be forwarded and processed at the fog. By denoting as c_n [CPU cycles] the amount of the task that can be offloaded at the fog, the user's n effort to the edge server is defined as $p_n = \frac{c_n}{C_n}, p_n \in [0, 1]$. Subsequently, the edge server offers to the user n a reward $r_n, r_n \in [0, 1]$ (e.g., in the form of a monetary discount), to account for the user's intention to participate in the computing continuum. Considering that the reward can be drawn from the edge server's energy savings, which are proportional to the user's effort p_n, we model the user's n reward as $r_n = \sqrt{\theta_n p_n}$. Hence, a higher user type indicates a user that has offloaded a more delay-tolerant task compared to the other users, and thus, can provide a higher effort to the edge server, by allowing a higher percentage of its task to be offloaded at the fog. Obviously, a higher user effort yields a higher reward offered by the edge server back to the user. Accordingly, a user's contract comprise the amount of offloaded task that will be further processed at the fog and the corresponding reward.

Following this discussion, each user's n utility function, capturing its satisfaction from participating in the contract and the green computing continuum, is defined as follows:

$$U_{n,e}(p_n) = \theta_n e(r_n) - \kappa p_n, \tag{1}$$

where $e(r_n)$ is the user's evaluation function of reward, which is strictly increasing and concave with respect to r_n, and $\kappa \in \mathbb{R}^+$ represents each user's unit cost of provided effort to the edge server. In this paper, we assume $e(r_n) = \sqrt{r_n}, \forall n \in N$.

On the other hand, the edge server's utility from each user's n participation in the contract is $U_{e,n} = p_n - \xi r_n$. Thus, the function of its overall expected utility is written as follows:

$$U_e(\mathbf{p}_n) = \sum_{\forall n \in N} [\lambda_n(p_n - \xi r_n)], \tag{2}$$

where $\lambda_n, \lambda_n \in [0,1]$ is the probability of user n of being of type θ_n as estimated by the edge server that is unaware of the actual users' types, for which it holds that $\sum_{\forall n \in N} \lambda_n = 1$, and $\xi \in \mathbb{R}^+$ is the edge server's unit cost of offered rewards to the users. Also, $\mathbf{p}_n$ is the vector of users' efforts. Without loss of generality and for simplicity in the representation, we assume that the user types can take values from a discrete set of types, while the equivalent contract's formulation under a continuous user-type space follows our work in [10].

B. Edge-to-Fog Server Contract Bundle & Utilities

After the completion of the first-stage contract, the edge server derives the total bytes that can be potentially transmitted to the fog as $D_e = \sum_{\forall n \in N} \frac{p_n C_n}{\phi_n}$ [Bytes], by knowing each user's effort p_n. The purpose of the second contract for the edge server is to determine the most beneficial amount of bytes d_e from the total D_e that strikes a good balance between local execution, backhaul transmission overhead and available incentives from the fog. As a result, we model the edge server's effort a_e at the fog as the percentage of D_e bytes that are ultimately transmitted to the fog, i.e., $a_e = \frac{d_e}{D_e}, a_e \in [0,1]$. Nevertheless, due to the constrained backhaul, there may occur incorrect transmissions and hence, the actual performance that is perceived by the fog server is a noisy signal of the edge server's effort. We denote as $q_e = a_e + \epsilon$ the edge server's performance to the fog, where $\epsilon \sim N(0, \sigma^2)$ is the normally distributed error between the effort and the performance with zero mean and variance σ^2. By not knowing the edge server's actual effort a_e, the fog server offers a compensation, which includes both a fixed t_e and a performance-related s_e reward so as to be as fair as possible. The overall compensation offered to the edge server is $w_e = t_e + s_e q_e, w_e \in [0,1]$ and can be either a monetary or a communications'-related reward.

Due to the constrained backhaul, we consider that the edge server exhibits a Constant Absolute Risk Averse (CARA) behavior as its compensation from the fog increases. Thus, the edge server's utility function is given by:

$$U_{e,f}(w_e, a_e) = -e^{-\eta[w_e - \psi(a_e)]}, \tag{3}$$

where $\eta \in \mathbb{R}^+$ is the edge server's coefficient of risk aversion, a high value of which dictates the more conservative behavior of the edge server towards providing its effort. Also, $\psi(a_e)$ is the edge server's cost of effort defined as $\psi(a_e) = \frac{1}{2}\mathcal{C}a_e^2$, with $\mathcal{C} \in \mathbb{R}^+$ denoting its unit cost of effort.

In contrast to the edge server, the fog server is considered as risk neutral and the function of its expected utility is:

$$U_f(w_e, a_e) = E[q_e - w_e] = (1 - s_e)a_e - t_e, \tag{4}$$

where $E[\cdot]$ is the expectation operator.

III. Users-to-Edge Server Contract based on Adverse Selection Problem

A. Contract Design under Incomplete Information

Although that the edge server is unaware of the users' types in the considered realistic case, it should at least guarantee that the designed contracts bear specific properties in order for the users to participate depending on their type. A contract agreement between the users and the edge server is termed as feasible if the following two conditions hold true.
(i) *Individual Rationality (IR)*: Each user's utility yields a non-negative value, i.e., $U_{n,e}(p_n) = \theta_n e(r_n) - \kappa p_n \geq 0, \forall n \in N$.
(ii) *Incentive Compatibility (IC)*: Each user selects the contract bundle that best fits its type, i.e., $\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_{n'}) - \kappa p_{n'}, \forall n, n' \in N, n \neq n'$.

Additionally, the three conditions listed in Propositions 1-3 must hold true to conclude to a feasible contract.

Proposition 1. *For any feasible contract, the following must hold: $r_n > r_{n'} \iff \theta_n > \theta_{n'}$ and $r_n = r_{n'} \iff \theta_n = \theta_{n'}$.*

Proof. To prove the first part of Proposition 1, i.e., $r_n > r_{n'} \iff \theta_n > \theta_{n'}$, we add the following two IC conditions by parts: $\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_{n'}) - \kappa p_{n'}$ and $\theta_{n'} e(r_{n'}) - \kappa p_{n'} \geq \theta_{n'} e(r_n) - \kappa p_n$, and by factorization we get $(\theta_n - \theta_{n'})[e(r_n) - e(r_{n'})] \geq 0$. Given that $\theta_n > \theta_{n'}$ and $e(r_n)$ is a strictly increasing function of r_n, it is concluded that $r_n > r_{n'}$. On the other hand, given that $r_n > r_{n'}$ and hence, $e(r_n) > e(r_{n'})$, we conclude that $\theta_n > \theta_{n'}$. By using a similar procedure and argumentation, the second part of Proposition 1, i.e., $r_n = r_{n'} \iff \theta_n = \theta_{n'}$, can also be proven. □

Proposition 2. *A higher-type user, i.e., $\theta_1 < \cdots < \theta_n < \cdots < \theta_{|N|}$, will receive a greater reward from the edge server, i.e., $r_1 < \cdots < r_n < \cdots < r_{|N|}$, and will provide a higher effort, i.e., $p_1 < \cdots < p_n < \cdots < p_{|N|}$.*

Proof. The proof stems intuitively from Proposition 1. □

Proposition 3. *A higher-type user, i.e., $\theta_1 < \cdots < \theta_n < \cdots < \theta_{|N|}$, will receive a higher utility, i.e., $U_{1,e} < \cdots < U_{n,e} < \cdots < U_{|N|,e}$ to be properly incentivized by the edge server.*

Proof. Given two users $n, n' \in N$, $n \neq n'$ of types $\theta_n > \theta_{n'}$ and the IC condition, we get $\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_{n'}) - \kappa p_{n'} > \theta_{n'} e(r_{n'}) - \kappa p_{n'}$, which results in $U_{n,e} > U_{n',e}$. Hence, following the monotonicity of user types, the proof can be inductively concluded. □

Based on the above analysis, the edge server formulates and solves the optimization problem that maximizes its personal utility, while satisfying the aforementioned conditions that guarantee the users' participation in the contract. The outcome of this optimization problem is the set of optimal contract bundles $(r_n^*, p_n^*), \forall n \in N$, each of them tailored to each user type θ_n. The edge server announces the available contract bundles to each user and each user autonomously selects the bundle that best fits its private information. The corresponding optimization problem solved by the edge server is written as:

$$\max_{(r_n,p_n)_{\forall n\in N}} U_e = \sum_{\forall n\in N} [\lambda_n(p_n - \xi r_n)] \tag{5a}$$

$$\textbf{s.t. } \theta_n e(r_n) - \kappa p_n \geq 0, \forall n \in N \tag{5b}$$

$$\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_{n'}) - \kappa p_{n'}, \forall n, n' \in N, n \neq n' \tag{5c}$$

$$0 \leq r_1 < \cdots < r_n < \cdots < r_{|N|}. \tag{5d}$$

The optimization problem in Eq. (5a)-(5d) is non-convex and to obtain a tractable solution we reduce its IR and IC constraints, described earlier at the beginning of section III-A. First, given that $\theta_n > \theta_1$ and the IC condition, we get $\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_1) - \kappa p_1 > \theta_1 e(r_1) - \kappa p_1 \geq 0$, which means that if the IR constraint of the lowest user type θ_1 is met, then the IR constraints of all other user types will be automatically satisfied. This reduces Eq. (5b) to Eq. (6b), below. Next, to reduce the IC constraints, we consider the Downward (DIC) and Upward (UIC) IC conditions between the users n and n', with $n' \in \{1, \ldots, n-1\}$ and $n' \in \{n+1, \ldots, |N|\}$, respectively. Then, Propositions 4-5 hold true.

Proposition 4. *All the DIC conditions can be reduced to the local DIC conditions between the users $n, n-1, \forall n \in N$.*

Proof. We consider three adjacent user types, i.e., $\theta_{n-1} < \theta_n < \theta_{n+1}$. By combining $\theta_{n+1} > \theta_n$ and the IC condition $\theta_n e(r_n) - \kappa p_n \geq \theta_n e(r_{n-1}) - \kappa p_{n-1}$ we get $\theta_{n+1}[e(r_n) - e(r_{n-1})] > \theta_n[e(r_n) - e(r_{n-1})] \geq \kappa p_n - \kappa p_{n-1}$. By utilizing this property, we get $\theta_{n+1}e(r_{n+1}) - \kappa p_{n+1} \geq \theta_{n+1}e(r_n) - \kappa p_n \geq \theta_{n+1}e(r_{n-1}) - \kappa p_{n-1} \geq \cdots \geq \theta_{n+1}e(r_1) - \kappa p_1$. The latter can be extended and hold for user types θ_{n-1} and θ_n. Hence, if the local DIC conditions between users $n, n-1$ hold, then all DIC conditions are satisfied. □

Proposition 5. *All the UIC conditions can be reduced to the local DIC conditions between the users $n, n-1, \forall n \in N$.*

Proof. Following a similar procedure with the proof of Proposition 4, we can conclude that all UIC conditions are reduced to the local UIC conditions between the users $n, n+1, \forall n \in N$ or $n-1, n, \forall n \in N$, equivalently, while the latter can be easily implied by the local DIC condition. □

Based on the above analysis and constraints reduction, the optimization problem in Eq. (5a)-(5d) can be rewritten as in Eq. (6a)-(6d). It should be, also, noted that in the reformulated problem, the IR and IC conditions are considered as equalities by the edge server so as to obtain the maximum benefit from the users. Finally, it is noted that the problem in Eq. (6a)-(6d) is convex and can be solved by applying the Karush-Kuhn Tucker (KKT) conditions, resulting in the optimal contract bundle vectors $(\mathbf{r}_n^*, \mathbf{p}_n^*)$.

$$\max_{(r_n,p_n)_{\forall n\in N}} U_e = \sum_{\forall n\in N} [\lambda_n(p_n - \xi r_n)] \tag{6a}$$

$$\textbf{s.t. } \theta_1 e(r_1) - \kappa p_1 = 0 \tag{6b}$$

$$\theta_n e(r_n) - \kappa p_n = \theta_n e(r_{n-1}) - \kappa p_{n-1}, \forall n \in N \tag{6c}$$

$$0 \leq r_1 < \cdots < r_n < \cdots < r_{|N|}. \tag{6d}$$

B. Contract Design under Complete Information

In the ideal case that the edge server is aware about the users' types, it seeks to fully exploit their effort and maximize its personal utility, by marginally ensuring their participation in contract. Hence, the edge server solves the following problem:

$$\max_{(r_n,p_n)} p_n - \xi r_n, \quad \forall n \in N \tag{7a}$$

$$\textbf{s.t. } \theta_n e(r_n) - \kappa p_n = 0. \tag{7b}$$

The solution to this linear programming problem can be easily found to be equal to $(r_n^*, p_n^*) = ((\frac{\theta_n}{2\xi\kappa})^2, \frac{\theta_n^2}{2\xi\kappa^2}), \forall n \in N$.

IV. Edge-to-Fog Server Contract based on Moral Hazard Problem

A. Contract Design under Incomplete Information

In the practical case that the fog server has no guarantees about the edge server's effort, its purpose is to offer to the edge server the appropriate compensation that maximizes its personal utility, while at the same time satisfying the edge server's IR and IC conditions, as defined earlier in Section III-A. Thus, the fog server's problem is formulated as follows:

$$\max_{(t_e,s_e,a_e)} U_f = (1 - s_e)a_e - t_e \tag{8a}$$

$$\textbf{s.t. } E[-e^{-\eta[w_e - \psi(a_e)])}] \geq U_{e,f}(\bar{w}_e, a_e = 0) \tag{8b}$$

$$a_e \in \arg\max_{a_e} E[-e^{-\eta[w_e - \psi(a_e)])}], \tag{8c}$$

where $U_{e,f}(\bar{w}_e, a_e = 0)$ is the edge server's minimum acceptable utility when exerting zero effort, while Eq. (8b) and Eq. (8c) represent the IR and IC conditions, respectively.

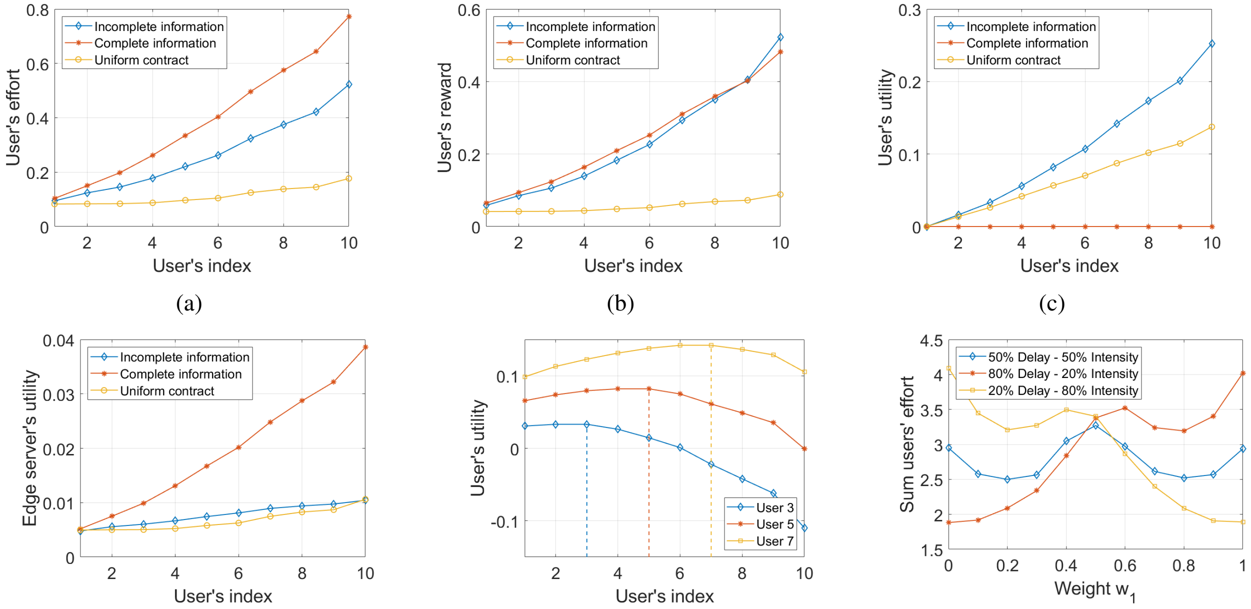
Elaborating more on the edge server's utility function we get $E[-e^{-\eta[w_e - \psi(a_e)])}] = -e^{-\eta(t_e + s_e a_e - \frac{1}{2}\mathcal{C}a_e^2)}E[e^{-\eta s_e \epsilon}]$.

Proposition 6. *Given a normal random variable $\epsilon \sim N(0, \sigma^2)$ and a constant $\gamma \in \mathbb{R}^+$, then it holds that $E[e^{\gamma\epsilon}] = e^{\frac{\gamma^2\sigma^2}{2}}$.*

Proof. Given that $f(\epsilon) = \frac{e^{-\epsilon^2/2\sigma^2}}{\sigma\sqrt{2\pi}}$ is the probability density function of ϵ, we have $E[e^{\gamma\epsilon}] = \int_{-\infty}^{+\infty} e^{\gamma\epsilon} f(\epsilon) d\epsilon = e^{\frac{\gamma^2\sigma^2}{2}}$, since $\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{\frac{-(\epsilon - \gamma\sigma^2)^2}{2\sigma^2}} d\epsilon = 1$, as the area under a normal distribution with mean $\gamma\sigma^2$ and variance σ^2. □

Following Proposition 6 and letting that $\gamma = -\eta s_e$, the edge server's utility is equal to $E[-e^{-\eta[w_e - \psi(a_e)])}] = -e^{-\eta(t_e + s_e a_e - \frac{1}{2}\mathcal{C}a_e^2)} e^{\frac{\eta^2 s_e^2 \sigma^2}{2}} = -e^{-\eta\hat{w}(a_e)}$, where $\hat{w}(a_e)$ is the edge server's certainty equivalent compensation [3] defined as:

$$\hat{w}(a_e) = t_e + s_e a_e - \frac{1}{2}\mathcal{C}a_e^2 - \frac{\eta}{2}s_e^2\sigma^2. \tag{9}$$



(a) (b) (c)

(d) (e) (f)

Fig. 1: Evaluation of users-to-edge server contract under different comparative scenarios.

The exponential form of the edge server's expected utility function, enables the reformulation of the optimization problem in Eq. (8a)-(8c) with respect to the edge server's certainty equivalent compensation $\hat{w}(a_e)$ as follows:

$$\max_{(t_e,s_e,a_e)} U_f = (1-s_e)a_e - t_e \tag{10a}$$

$$\textbf{s.t. } \hat{w}(a_e) \geq \bar{w}_e \tag{10b}$$

$$a_e \in \arg\max_{a_e} \hat{w}(a_e), \tag{10c}$$

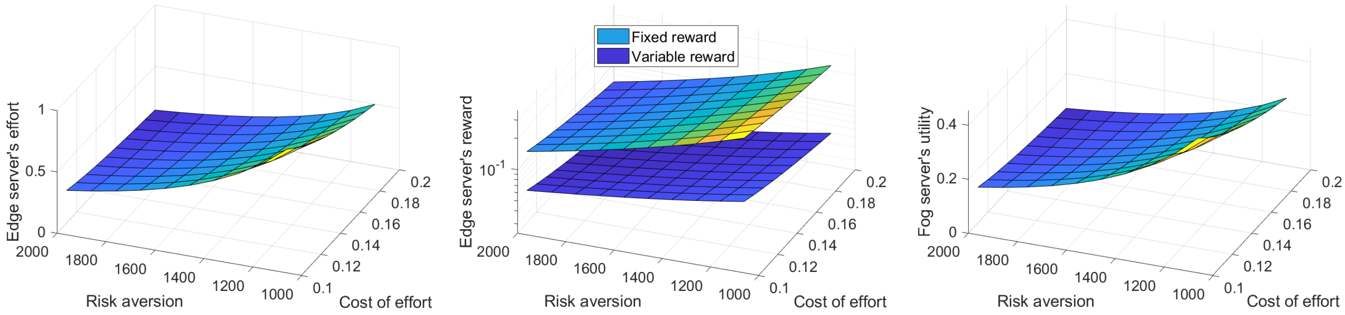
where $\bar{w}_e$ is the edge server's minimum acceptable compensation when exerting zero effort.

By calculating the first order derivative of Eq. (10c) and setting it equal to zero, we obtain $a_e^* = \frac{s_e}{\mathcal{C}}$. Then, substituting to the certainty equivalent compensation $\hat{w}(a_e)$ in Eq. (9) and considering the constraint in Eq. (10b) as equality, since the fog server seeks to make the most of the edge server's effort, the optimization problem in Eq. (10a)-(10c) reduces to a two-variable linear program. The solution of this linear program can be easily found to be $t_e^* = \bar{w}_e - \frac{1-\eta\mathcal{C}\sigma^2}{2\mathcal{C}(1+\eta\mathcal{C}\sigma^2)^2}$ and $s_e^* = \frac{1}{1+\eta\mathcal{C}\sigma^2}$. Hence, the solution of the edge-to-fog server contract is $(t_e^*, s_e^*, a_e^*) = (\bar{w}_e - \frac{1-\eta\mathcal{C}\sigma^2}{2\mathcal{C}(1+\eta\mathcal{C}\sigma^2)^2}, \frac{1}{1+\eta\mathcal{C}\sigma^2}, \frac{s_e}{\mathcal{C}})$.

V. Evaluation & Results

In this section, we enclose a numerical evaluation of both the users-to-edge and the edge-to-fog server contracts, which is obtained via modeling and simulation. In our evaluation, we consider $|N| = 10$ users, communicating with the edge server, with application characteristics randomly generated such that $B_n \in [1, 5]$ MBytes, $\phi_n \in [500, 2000]$ CPU cycles/Byte and $T_n \in [500, 2000]$ msec, $\forall n \in N$ [8]. Considering the users-to-edge server contract, the users' and edge server's cost of effort and reward, respectively, are set as $\kappa = 0.9$ and $\xi = 0.8$, while we assume that the edge server estimates the occurrence of different user types based on the uniform distribution, such that $\lambda_n = \frac{1}{|N|}, \forall n \in N$. Regarding the edge-to-fog server contract, we set $\sigma^2 = 0.1$ and $\bar{w}_e = 0$, while different values of the edge server's cost of effort $\mathcal{C} \in [1000, 2000]$ and coefficient of risk aversion $\eta \in [0.1, 0.2]$ are scrutinized in the sequel.

In Fig. 1, we perform a comprehensive evaluation of the users-to-edge contract, accounting for different comparative scenarios. In particular, Fig. 1a-1d presents the performance of the proposed users-to-edge contract under the realistic case of incomplete information, compared against the ideal case of complete information availability, as described in Section III-B, and a "uniform contract" approach, in which the edge server is unaware of the user types and offers them a uniform and user type-agnostic reward defined as $r_n = 0.5p_n, \forall n \in N$. In Fig. 1a-1d, the horizontal axis represents the users sorted in ascending order with respect to their types, while the vertical axes depict the values of the users' efforts, rewards and utilities, as well as the edge server's utility $U_{e,n}$ attained by each user's participation in the contract. All graphs in Fig. 1a-1d verify the monotonic behavior of the contracts, according to which a user of higher type provides a higher effort to the edge server, and hence, is rewarded more, yielding higher utilities for both the users and the edge server. Evidently, in the complete information case, the users agree to exert a higher effort in exchange for a quite equal reward to the incomplete information case. As a result, each user's utility is equal to zero, while the edge server achieves a significantly higher utility per user in this ideal complete information availability case. Considering the uniform contract case, the results show that the users are not adequately distinguished with respect to their types, resulting in quite similar amounts of efforts and rewards for all users, limiting in this way the potential of higher types to promote the green computing continuum.



(a) (b) (c)

Fig. 2: Evaluation of edge-to-fog server contract.

Continuing the evaluation of the incomplete information contract, in Fig. 1e, we confirm the incentive compatibility of the obtained contract bundles. Indicatively, the users of index 3, 5 and 7 are selected and their utility values are scrutinized over all contract bundles offered by the edge server, as indicated in the horizontal axis by the user index. Indeed, the utility of the users 3, 5 and 7 is maximized when selecting the contract bundle designed for their specific type. Finally, Fig. 1f illustrates the sum users' effort contracted under the incomplete information case, considering different values of the weight w_1 (and hence, w_2) in the formula of the user type. Specifically, in this evaluation scenario we assume three cases, in which different percentages of the existing users are characterized by either remarkably high delay tolerance or task intensity. Apparently, for low values of the weight w_1, where the users are mainly distinguished based on their task's intensity, the sum users' effort is higher in the ”20% Delay - 80% Intensity” case, where 80% of the user applications have high task intensity. The opposite holds true for high values of w_1, leading to higher sum efforts for the ”80% Delay - 20% Intensity” case that 80% of the user applications are characterized by increased delay tolerance. Intuitively, the graph of the ”50% Delay - 50% Intensity” case lies in between the other two cases, while in all three cases we observe a slight increase in the sum users' effort around $w_1 \in [0.4, 0.6]$, where the impact of both the delay tolerance and intensity is taken almost equally into account in the formation of the user types.

In Fig. 2a-2c, we study the performance of the edge-to-fog server contract under different values of the edge server's cost of effort $\mathcal{C}$ and coefficient of risk aversion η. Specifically, in Fig. 2a-2c, the x and y axes contain the different values of the edge server's cost of effort and risk aversion, respectively, while the vertical z axes depict the values of the edge server's effort and fixed and variable reward, as well as the fog server's utility. Obviously, for low values of both $\mathcal{C}$ and η, the edge server provides a higher effort (indicated by the light yellow color) and thus, is rewarded more by the fog server, increasing in this way the fog server's utility. On the other hand, high values of both $\mathcal{C}$ and η yield lower values of all metrics (indicated by the deep blue color). At this point, it should be reminded that in the edge-to-fog server contract, the edge server's utility is always set equal to zero due to $\bar{w}_e = 0$.

VI. Conclusion and Future Work

In this paper, the paradigm of green computing continuum was exploited, by designing an incentivization-based mechanism that shifts the preference of the users of delay-tolerant tasks from the prevailing edge to the fog computing layer. The devised incentivization mechanism included two sequential stages, in which respective users-to-edge and edge-to-fog server contracts were formulated based on Contract Theory to capture their in-between relationships. Indicative numerical results were presented to verify the effective operation of the proposed mechanism. Our future work focuses on extending this model, by accounting for both the underlying communications and computing resource allocation needs.

References

[1] D. Rosendo, P. Silva, M. Simonin, A. Costan, and G. Antoniu, “E2clab: Exploring the computing continuum through repeatable, replicable and reproducible edge-to-cloud experiments,” in *2020 IEEE International Conference on Cluster Computing (CLUSTER)*, 2020, pp. 176–186.

[2] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing: A taxonomy, survey and future directions,” in *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer Singapore, 2018, pp. 103–130.

[3] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

[4] B. Nour, S. Mastorakis, and A. Mtibaa, “Whispering: Joint service offloading and computation reuse in cloud-edge networks,” in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[5] M. Mukherjee, S. Kumar, Q. Zhang, R. Matam, C. X. Mavromoustakis, Y. Lv, and G. Mastorakis, “Task data offloading and resource allocation in fog computing with multi-task delay guarantee,” *IEEE Access*, vol. 7, pp. 152 911–152 918, 2019.

[6] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, “An application placement technique for concurrent iot applications in edge and fog computing environments,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1298–1311, 2021.

[7] B. Zhang, L. Wang, and Z. Han, “Contracts for joint downlink and uplink traffic offloading with asymmetric information,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 4, pp. 723–735, 2020.

[8] C. Su, F. Ye, T. Liu, Y. Tian, and Z. Han, “Computation offloading in hierarchical multi-access edge computing based on contract theory and bayesian matching game,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 686–13 701, 2020.

[9] Y. Li, J. Zhang, X. Gan, L. Fu, H. Yu, and X. Wang, “A contract-based incentive mechanism for delayed traffic offloading in cellular networks,” *IEEE Trans. Wireless Comm.*, vol. 15, no. 8, pp. 5314–5327, 2016.

[10] M. Diamanti, E. E. Tsiropoulou, and S. Papavassiliou, “Resource orchestration in uav-assisted noma wireless networks: A labor economics perspective,” in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.