A Bayesian Game Theoretic Approach to Task Offloading in Edge and Cloud Computing

Anna V. Guglielmi*, Marco Levorato+, and Leonardo Badia*
*Dept. of Information Engineering, University of Padova, via Gradenigo 6B, 35131 Padova, Italy
+Donald Bren School of Information and Computer Sciences, University of California, Irvine, USA
email: {guglielm, badia}@dei.unipd.it, levorato@uci.edu

***Abstract*—This paper addresses the management of computational offloading in a three-tier hierarchical architecture comprising mobile devices, edge computing, and cloud computing. Although edge and cloud devices have lower data processing time than mobile devices, the simultaneous transmission of heavy data streams may overload the local wireless network, resulting in an overall larger delay. Herein, a game theoretic framework is proposed for the distributed decision making in a scenario where mobile users share the same network resource and do not have a priori information in the wireless links. We evaluate at first the rational gameplay of the nodes in a scenario with complete knowledge, and we compare it with a scenario with incomplete information modeled as a Bayesian game. In particular, we consider network positions, and therefore channel gain and distance-related parameters, to be uniformly distributed within a given range, and the nodes only have this knowledge available as a prior. The analysis demonstrates that rationality (implying selfish behavior) of the mobile users does not necessarily lead to a more efficient allocation and actually the scenario of incomplete information leads to a socially better outcome, thereby suggesting an interesting guideline for the design of computational offloading strategies in realistic scenarios.**

***Index Terms*—Mobile cloud computing; Mobile edge computing; Game theory; Bayesian games; Mobile devices.**

I. Introduction

THE growing complexity of data analysis and processing algorithms makes their deployment in resource constrained devices (*e.g.*, sensors and mobile devices) increasingly challenging. The recent fog and edge computing paradigms [1] address this issue by placing compute-capable devices within low-latency one-hop wireless topologies.

Offloading local computation tasks to the edge processors can significantly speed-up their completion, but necessitates to transport the data over local wireless networks. Especially in modern architectures, where multiple technologies share the same spectrum resource, interference from exogenous wireless terminals, or from other mobile users offloading tasks, may heavily affect the capacity of the wireless links, so that the delivery of data to edge or cloud processors becomes a relevant component of the overall delay to completion.

Herein, we analyze a three-tiered communication/processing infrastructure, consisting of a local tier of mobile nodes, a middle tier of nearby computing nodes (i.e., edge servers), typically co-located with the wireless Base Stations (BS) and characterized by a limited amount of processing resources, and a remote tier of cloud servers connected to the BS through the core wired network. The mobile users can choose to locally process the data, or offload computation to edge or cloud resources. If the mobile devices decide to offload the processing task either to the edge or cloud server, then the associated data need to be transmitted over the local wireless network. As the wireless links have finite capacity, this first step introduces a delay, but also a coupling between the users, as the decision to offload the processing task of one user increases the network load and, thus, the delay to the completion of the task of all the other users.

The main challenge is that in practical cases, mobile users do not have a priori knowledge of the interference load and computation power of edge and cloud processors. Additionally, the decision of each mobile device influences the overall network load. In fact, unlike in-device processing, if a mobile user offloads a computational task to the edge or cloud resources through the BS, it creates interference that reduces the capacity of the wireless links of the other users.

To model and optimize this distributed decision making process with partial information, we propose a game theoretic framework. Specifically, due to the stochastic characterization of some of the influential variable of the system, such as exogenous interference load and channel gain, we adopt a Bayesian Game (BG) formulation. Each user can decide if locally execute the task on its mobile device, or offloading the task computation towards either the remote cloud or an edge server through the BS. The objective of each user is to minimize its computation cost, evaluated both in terms of computation time and energy needed to accomplish the task execution.

Under this game-theoretic analysis, we characterize the equilibrium for different parameter regions, as well as evaluate the impact of selfish actions by the users. Our result indicate that the uncertainty on network parameters of the other players can actually be beneficial to the overall social welfare as it deters users to aggressively cause interference to others, which would result in a low-efficiency equilibrium. This suggests that preventing the users from gaining information about the other nodes in the network may actually be a good strategy to avoid that the offloading option is abused by the terminals.

The rest of the paper is organized as follows. In Section II prior work related to the present contribution is discussed. Section III and Section IV describe the system model and

This work was partially supported by the NSF under grant IIS-1724331 and CNS-1702950.

978-1-5386-4328-0/18/$31.00 ©2018 IEEE

provide a game theoretic analysis, respectively. Numerical results and computation of the equilibria are presented in Section V. Section VI concludes the paper.

II. Related work

Game Theory (GT) is a tool from applied mathematics that has been widely used to study how rational players, whose objective is usually seen as the maximization of a *utility function*, interact to determine an equilibrium point. In the context of network communication, this can be seen as the problem to obtain a distributed resource allocation procedure upon which nodes, seen as individual agents, agree. Recently, there has been soaring interest in this kind of applications to wireless communication and networking scenarios [2]. For instance, in [3] a game theoretic model for random access with carrier sensing is investigated. Games for resource management, network selection, and admission control in wireless systems were studied in [4], [5]. In [6], the authors investigate opportunistic communications in hierarchical cognitive networks. Finally, evolutionary coalition games for wireless networking and communications were studied in [7]. A comprehensive literature review of GT formulations for energy efficiency in wireless sensor networks can be found in [8].

Closely related to this contribution, [9], [10] consider offloading problems and propose techniques to determine which tasks should be offloaded to improve the overall system performance. Most of the formulations presented in those papers focus in a single node scenario, where an application is represented as a weighted graph, whose nodes are tasks.

To the best of our knowledge, a relatively small number of contributions addressed a multi-users scenario in the context of edge and cloud computing. In [13], the authors propose a cooperative centralized optimization problems aiming at distributing the processing tasks of multiple users among the computation resources available locally at the mobile devices and globally at the cloud level. The objective is to minimize the average application delay for all users. A Mixed Integer Linear Programming formulation is used. In [14], the problem of distributing computation resources among data streams is studied. Multiple users share the wireless network as well as the computation resources in the cloud. The goal is that of maximizing the throughput of the data streams. Different from the present contribution, a heuristic genetic algorithm is used to solve the optimization problem, and a two-tier architecture composed of mobile users and remote cloud servers is considered.

Analogous to that adopted in this paper, [11], [12] considers a three-tier architecture with multiple mobile users. The authors assume that the cloud resources are constrained. The optimization problem is solved using a greedy centralized heuristic. In [15], a three-tier architecture with no centralized control is studied, and GT is used to study a scenario where multiple selfish users whether and where to offload their computation tasks. Different from our study, a non-cooperative, and non-Bayesian, formulation of the game is used, and there is no consideration of the interactions between the users from a wireless communications standpoint. In other words, all of the previous papers applying game theory to this scenario apparently do so from an idealized perspective where network nodes not only avoid causing congestion to each other but also are fully and instantaneously aware of all the network parameters and can therefore make fully-informed rational (that is, selfish) decision. This aspect is particularly relevant for what concerns the contribution of the present paper, since, as will be shown in the following, the Bayesian component of the game, used to determine incomplete information available to the users, actually lead to an improvement in the overall resulting efficiency of the allocation, an aspect related to the fact that selfish users only act for their own good.

III. System model

We consider a scenario where a set of M mobile devices indicated as n_i, $i \in \{1, 2, .., M\}$, are connected to a BS s providing access to the global network infrastructure. The BS is attached via a high capacity link to an edge processor, which provides low-latency computation services to the users. The BS is also connected to a more powerful cloud computing resource through the Internet.

We assume that each user has a computationally intense task to be completed. A three-tier architecture is considered, where user can decide to compute the task locally in-device or to offload the task to the edge or cloud servers. We define a quasi-static scenario, in which the set of mobile device users remains unchanged for a period comparable to the completion of their tasks. We leave to future studies the analysis of scenario case where mobile users can depart and leave dynamically.

The mobile device users share the wireless channel to the BS s. $\mathcal{C}_{n_i} = (b_{n_i}, d_{n_i})$, $i \in \{1, 2, .., M\}$, describes the computation task of the node n_i, where b_{n_i} and d_{n_i} are the size of the input data and the total number of CPU cycles necessary to complete the task $\mathcal{C}_{n_i}$.

We denote the computation offloading decision of mobile device user n_i as a_{n_i}. Specifically, $a_{n_i} = 0$ if n_i computes the task locally, $a_{n_i} = 1$ if n_i offloads the computation task to the remote cloud server, finally, $a_{n_i} = 2$ if n_i offloads the computation task to the edge server. The vector $\mathbf{a} = (a_{n_1}, a_{n_2}, .., a_{n_M})$ is the decision profile of the mobile device users.

Since the users share the wireless resource, we adopt an interference model to capture the degradation of individual user links when other transmissions are active. In particular, we define the uplink data rate of n_i, i.e., r_{n_i}, as

$$r_{n_i} = W \log_2 \left(1 + \frac{q_{n_i} g_{n_i}}{\omega_0 + \mathcal{I}}\right) \tag{1}$$

where $\mathcal{I} = \sum_{n_j \in \mathcal{M} - \{n_i\}} q_{n_j} g_{n_j} I\{a_{n_i}, a_{n_j} \in \{1, 2\}\}$, W is the channel bandwidth, q_{n_i} is n_i's transmission power, g_{n_i} is the channel gain between n_i and s, ω_0 is the background noise power, and $I\{a_{n_i}, a_{n_j} \in \{1, 2\}\}$, with $i, j \in \{1, 2, .., M\}$, is the indicator function representing the interference from the other user conditioned on its transmission decision.

We define $f_{n_i}^{LC}$, $f_{n_i}^{EC}$, and, $f_{n_i}^{CC}$, as the computation capability measured in terms of CPU cycles per second of the mobile devices, the computation capability assigned by the edge server to n_i, and the one assigned by the cloud, respectively. We assume that $f_{n_i}^{EC} < f_{n_i}^{CC}$. If n_i decides to locally compute C_{n_i}, the computation execution time is $t_{n_i}^{LC} = \frac{d_{n_i}}{f_{n_i}^{LC}}$. On the other hand, if n_i offloads the computation task it would incur the extra overhead for transmitting the input data either to the edge server or remote cloud through the wireless access behind the time needed to execute C_{n_i}.

To assess the offloading decision, we define two cost metrics corresponding to the time and energy consumption to complete the task execution. In particular, the total time to complete the task in the edge and cloud computing are

$$t_{n_i,off}^{EC} + t_{n_i,exe}^{EC} = \frac{b_{n_i}}{r_{n_i}} + \frac{d_{n_i}}{f_{n_i}^{EC}} \tag{2}$$

and

$$t_{n_i,off}^{CC} + t_{n_i,exe}^{CC} = \frac{b_{n_i}}{r_{n_i}} + D_i + \frac{d_{n_i}}{f_{n_i}^{CC}}, \tag{3}$$

where D_i is the delay caused by propagating the data through the wired data transmission to the cloud server.

If the task is completed locally, the energy expense is equal to $e_{n_i}^{LC} = \alpha_{n_i} d_{n_i}$, where α_{n_i} the consumed energy per CPU cycle. If the task is offloaded to the edge server, then the energy consumption is $e_{n_i,off}^{EC} = q_{n_i} \frac{b_{n_i}}{r_{n_i}} + L_{n_i}$, where L_{n_i} denotes the tail of the transmission energy due to the fact that the mobile device user will continue to occupy the channel for a while even after the data transmission. Finally, for the offloading towards the remote cloud we assume that $e_{n_i,off}^{CC} = q_{n_i} \left(\frac{b_{n_i}}{r_{n_i}} + D_i \right) + L_{n_i}$.

Consequently, we define the computation cost for each of the three possible computing decisions as

$$\begin{aligned} K_{n_i}^{LC} &= \lambda_{n_i}^t t_{n_i}^{LC} + \lambda_{n_i}^e e_{n_i}^{LC}, \\ K_{n_i}^{EC} &= \lambda_{n_i}^t \left(t_{n_i,off}^{EC} + t_{n_i,exe}^{EC} \right) + \lambda_{n_i}^e e_{n_i,off}^{EC}, \\ K_{n_i}^{CC} &= \lambda_{n_i}^t \left(t_{n_i,off}^{CC} + t_{n_i,exe}^{CC} \right) + \lambda_{n_i}^e e_{n_i,off}^{CC}. \end{aligned} \tag{4}$$

$\lambda_{n_i}^t, \lambda_{n_i}^e$ are positive weights in the range $[0, 1]$ associated with the computational time and energy for n_i. The weights can be adapted to the state of the devices. For instance, if a mobile user's battery is close to depletion, $\lambda_{n_i}^e$ can be set to 1 and $\lambda_{n_i}^t$ to 0. Instead, when the tasks generated by a mobile device are delay-sensitive, the delay can be set to $\lambda_{n_i}^t = 1$ and $\lambda_{n_i}^e = 0$. We neglect the time needed to receive the outcome of processing. This component is typically smaller compared to the cost of transporting the data due to the smaller size of feedback, but can be easily added to our metrics.

IV. Bayesian game theoretic approach

In GT, individual agents, called *players*, perform actions purely based on their individual interests, which may differ from those of the other players [16]. Each player usually acts towards the maximization of its own utility; however, this is jointly determined by the actions played by all players. Throughout this paper, for readability reasons, we will actually consider an individual *cost* function that the players try to minimize; therefore, the same reasoning of classical setups apply but to consider a "utility function" or a "payoff" one should take the opposite value to what we consider here. Thus, we consider a *static game of complete information* as defined by a triple $\mathcal{G}$ = ($\mathcal{A}$, $\mathcal{S}$, $\mathcal{K}$), where $\mathcal{A}$ is the set of players, $\mathcal{S}$ is the set of all strategies allowed to the players, and $\mathcal{K}$ is a set of cost functions, one per each user, depending on the strategies chosen by the players. Note that this reflects that a player's chosen strategy influences the cost paid by the other players; at the same time, rational players are in turn able to anticipate the effect of the strategies chosen by the other players. We search for a Nash Equilibrium (NE) seen as a joint strategy profile where all players locally minimize their paid cost.

However, we will also focus on a Bayesian game, i.e., a game of incomplete information in which rational anticipation of the game outcome made by the players is hindered by the lack of precise knowledge of the cost function of the other players. This is usually modeled by introducing a *type* for the players, which in turn translates in a different cost function; in our case the type will be given by a specific parameter that affects the transmission costs, and specifically we will consider the node distance from the BS as this parameter. Players are clearly aware of their own type, but as for the others they can only treat them as random variables; still, they can exploit a prior distribution on the other players' types, that is common knowledge. Note that this description is more accurate to describe mobile computing systems where many parameters such as the channel gain, can only be estimated but never known with certainty. Even though a mobile user can reasonably well estimate its own channel gain, it is simplistic to assume that it knows with perfect precision that parameter for the other players; thus, a Bayesian setup is definitely more realistic.

In our scenario, we consider two mobile devices corresponding to the set of players, i.e., $\{n_1, n_2\}$. The set of actions for each player is {Local, Cloud, Edge}, where Local, Cloud, and Edge are associated with executing the task locally on the device, offloading the computation task to the remote cloud, and offloading the task execution to the edge server, respectively. Each player acts towards the minimization of its computation cost.

For the sake of simplicity, we consider a symmetric system, where all the following parameters are the same for both users (and thus we drop subscripts "n_1" and "n_2" for better readability: q, b, α, f^{LC}, f^{EC}, f^{CC}, λ^t, λ^e. Moreover, we assume L_{n_1} and L_{n_2} to be negligible.

In the BG context, each player has a type defined by its corresponding channel gain, whose value is not known to its opponents. However, we assume that the prior distribution of the types is known, for instance based on historical data accumulated at the edge processor. Types of player 1 and 2 are channel gains g_{n_1} and g_{n_2}, respectively. In the literature, multi-path propagation is usually considered and modeled, e.g., as Rayleigh fading. For the sake of simplicity, we evaluate the channel gain as determined by path loss only, i.e., $g_{n_i} = \delta(n_i, s)^{-\gamma}$, for $i \in \{1, 2\}$, where $\delta(n_i, s)$ is the distance between n_i and s, and we choose exponent $\gamma = 4$ to describe a sub-urban scenario. We assume that distance $\delta(n_i, s)$ is uniformly distributed in $[\delta_{min}\ \delta_{max}]$. This assumption is made only to simplify the computations in what follows (especially to determine the Bayesian NE) but the conclusion we draw are actually valid as long as there is any kind of uncertainty on the channel gain. In particular, note that none of our conclusions are limited to the choice of uniform distribution for the positions and/or only considering the path loss in the evaluation; those are choices that can easily be relaxed and would just lead to more complex computations with analogous conclusions. Also for notational brevity, put

$$A = \frac{b}{W \log_2 \left(1 + \frac{q g_{n_i}}{\omega_0 + q g_{n_j}}\right)} \tag{5}$$

$$B = \frac{b}{W \log_2 \left(1 + \frac{q g_{n_i}}{\omega_0}\right)}. \tag{6}$$

Now, the cost paid by the two players depending on their moves can be directly written. Specifically:

– if both players choose Local, they both incur a cost equal to $\lambda^t \frac{d}{f^{LC}} + \lambda^e \alpha d$;

– if both players choose Cloud, their cost is also the same: $\lambda^t \left[A + D + \frac{d}{f^{CC}}\right] + \lambda^e \left[q\left(A + D\right)\right]$;

– if both players choose Edge, their identical cost is: $\lambda^t \left[A + \frac{d}{f^{EC}}\right] + \lambda^e q A$;

– if one player chooses Local and the other Cloud, then the cost paid by the former is $\lambda^t \frac{d}{f^{LC}} + \lambda^e \alpha d$, instead for the latter $\lambda^t \left[B + D + \frac{d}{f^{CC}}\right] + \lambda^e \left[q\left(B + D\right)\right]$;

– if one player chooses Local and the other Edge, then the cost paid by the former is $\lambda^t \frac{d}{f^{LC}} + \lambda^e \alpha d$, instead for the latter $\lambda^t \left[B + \frac{d}{f^{EC}}\right] + \lambda^e q B$;

– if one player chooses Cloud and the other Edge, then the cost paid by the former is $\lambda^t \left[A + D + \frac{d}{f^{CC}}\right] + \lambda^e \left[q\left(A + D\right)\right]$, instead for the latter $\lambda^t \left[A + \frac{d}{f^{EC}}\right] + \lambda^e q A$.

Importantly, the value of channel gain g_{n_i} in the numerator of (1) is known to player i. Conversely, the term g_{n_j} at the denominator is not known to player i as it represents the type of player j. In this Bayesian setup, we aim at the evaluation of the expected costs computed weighing the costs defined above with respect to the distribution of types. Before proceeding with the Bayesian evaluations, some considerations arisen from the complete knowledge case follow. In this case, we assume that players have complete knowledge of the system, meaning that each of them knows its opponent's type.

For symmetry reasons, a NE in pure strategies must be to one of the three outcomes (Local, Local), (Cloud, Cloud), and (Edge, Edge). Moreover, a relationship between the value of D and the two outcomes (Cloud, Cloud) and (Edge, Edge) has been observed. If and only if $D < D_{thr}$, where

$$D_{thr} = \frac{d(f^{CC} - f^{EC})}{f^{CC} f^{EC} (1 + \Lambda q)}, \tag{7}$$

then (Cloud, Cloud) has lower computation cost than (Edge, Edge). Thus, we can compare possible outcome (Local, Local) with the "non-local" outcome (N,N) where N can stand for either "Cloud" or "Edge" depending on (7).

V. Equilibrium and Numerical results

We now consider a system with specific parameter choices reported in Table I. According to the table, $\delta(n_i, s)$ is uniformly distributed between $\delta_{\min} = 10$ m and $\delta_{\max} = 100$ m. Moreover, we considered D in [4 13]ms.

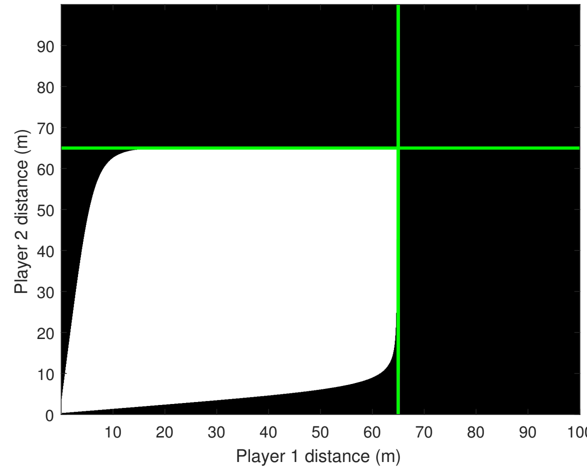

Fig. 1. Validity regions for NEs.

TABLE I
Values of relevant parameters

Parameters	Values
b	40 kBytes
d	10^9 CPU cycles
f^{LC}	0.5 GHz
f^{EC}	10^2 GHz
f^{CC}	10^4 GHz
q	0.01 W
ω_0	5 μW
W	1 GHz
λ^t	0.5
λ^e	0.5
α	4
g	30^{-4}
$\delta(n_i$, s)	U([10 100] m)

First of all, we consider a baseline solution of the game with *complete* information, against which we assess the performance of our Bayesian setup. According to what discussed previously, rational players in this setup end up in playing a symmetric outcome. To determine whether in this scenario the players will eventually both play the Local strategy or strategy "N," we remark that the game is isomorphic to a Prisoner's dilemma [16]; however, the choice of which one of the symmetric allocations is the Nash equilibrium (and whether this is also Pareto efficient) depends on the channel gain, and in turn on the distances. Thus, we numerically computed the resulting equilibrium for all the possible values of $\delta(n_1, s)$ and $\delta(n_2, s)$. We recall that we are under the assumption of *full knowledge* on both users of their mutual positions, so both values are known to both players. The result is shown in Fig. 1: the figure reports, for all pairs ($\delta(n_1, s)$, $\delta(n_2, s)$), what would be the NE of rational (i.e., selfish) players, with the black region corresponding to the choice of locally executing the computation task, and the white region representing task offloading towards either the cloud or edge server, according to D.

Aside from some border effects, we can infer that a threshold behavior is present, meaning that the players will offload their computation only if they are close enough to the base station. Such a threshold δ^{thr}, which can be numerically computed (see later Fig. 2) to be around 65 meters, is also plotted in the figure. The presence of this threshold is not coincidental, as we will show next.

For the Bayesian case, it is possible to prove that a threshold behavior is indeed present, according to the following reasoning. Player i does not know whether player j is actually playing an offloading move; nevertheless, i may get a Bayesian belief that j will be playing according to a threshold strategy (meaning that j will offload only if close enough to the base station). Actually, this is the kind of self-enforcing assumption that automatically gets confirmed due to the symmetry of the players, and it is also easy to see that this threshold strategy would be the best response to itself. Hence, in Bayesian game terms, this is a Bayesian NE. Note that a similar reasoning has been formally proven in [17].

This remark can serve to compute the expected cost obtained by the players in the Bayesian case. In particular, for each player i, if $\delta(n_i, s) > \delta^{thr}$, player i will decide for the local computation of the task execution, choosing offloading otherwise. The actual value of the threshold δ^{thr} can be found by imposing symmetry between the two outcomes as NEs, and therefore considering: $\delta(n_1, s) = \delta(n_2, s)$ and also that the cost of local computation is exactly equal to offloading. The value of δ^{thr} can be precisely found in this way.

Consequently, for the computation of the Bayesian expected cost we need to sum 4 different contributions denoted as K_1 – K_4. Specifically, looking at Fig. 1 and considering for example player 1, they are:

$$K_1 = \int_{\delta_{min}}^{\delta^{thr}} \int_{\delta_{min}}^{\delta^{thr}} \left(\lambda^t \left[A + D + \frac{d}{f^{CC}} \right] + \lambda^e \left[q(A+D) \right] \right) \mathrm{d}x \mathrm{d}y$$

if $D < D_{thr}$,

$$K_1 = \int_{\delta_{min}}^{\delta^{thr}} \int_{\delta_{min}}^{\delta^{thr}} \left(\lambda^t \left[A + \frac{d}{f^{EC}} \right] + \lambda^e q A \right) \mathrm{d}x \mathrm{d}y$$

otherwise,

$$K_2 = \int_{\delta_{min}}^{\delta^{thr}} \int_{\delta^{thr}}^{\delta_{max}} \left(\lambda^t \frac{d}{f^{LC}} + \lambda^e \alpha d \right) \mathrm{d}x \mathrm{d}y$$

$$K_3 = \int_{\delta^{thr}}^{\delta_{max}} \int_{\delta^{thr}}^{\delta_{max}} \left(\lambda^t \frac{d}{f^{LC}} + \lambda^e \alpha d \right) \mathrm{d}x \mathrm{d}y$$

and

$$K_4 = \int_{\delta^{thr}}^{\delta_{max}} \int_{\delta_{min}}^{\delta^{thr}} \left(\lambda^t \left[B + D + \frac{d}{f^{CC}} \right] + \lambda^e \left[q(B+D) \right] \right) \mathrm{d}x \mathrm{d}y$$

if $D < D_{thr}$,

$$K_4 = \int_{\delta^{thr}}^{\delta_{max}} \int_{\delta_{min}}^{\delta^{thr}} \left(\lambda^t \left[B + \frac{d}{f^{EC}} \right] + \lambda^e q B \right) \mathrm{d}x \mathrm{d}y$$

otherwise. Player 1 expected cost is given by $K_1 + K_2 + K_3 + K_4$. Player 2 expected cost can be evaluated in a similar way.

Fig. 2 shows how the distance threshold δ^{thr} varies considering several D's values. Figs. 3 and 4 depict player 1's cost considering the complete knowledge case and player 1 expected cost, respectively, as a function of D. In general, we can observe that as D increases, the computation cost increases until it reaches a saturation value. When the value of D is small enough (i.e., $D < D_{thr}$), the best action for the mobile device is to offload the task execution to the remote cloud instead of the edge server as the additional delay is compensated for by the larger computation speed. When the threshold D_{thr} is reached, the best action is to offload to the edge server. However, it is relevant to notice that the Bayesian case implies that the nodes incur a *lower* cost when they have imperfect knowledge on the distance of the other player. The reason for this apparently counterintuitive behavior is a consequence of multi-agent multi-objective optimization performed via game theory. Recalling that different users have a different objective in the game (i.e., minimization of their individual cost in a selfish way), the "ignorance is bliss" principle applies [16]. Differently from single-person optimization, where more knowledge always corresponds to a better outcome, in a game theoretic setup having less information about the other players may turn out, as in this case, to be advantageous, as players will be less aggressive and do not tend to abuse offloading if they are not sure that the other player is doing it too. Given the symmetry of the setup, it turns out that the best course of action for the players may be to abstain from unnecessary (or

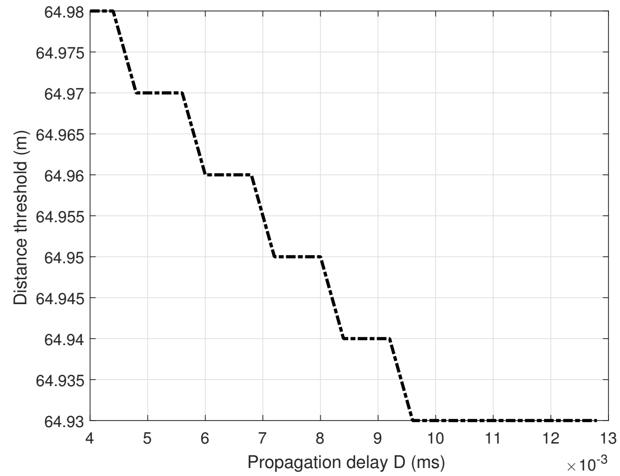

Fig. 2. δ^{thr} development varying D.

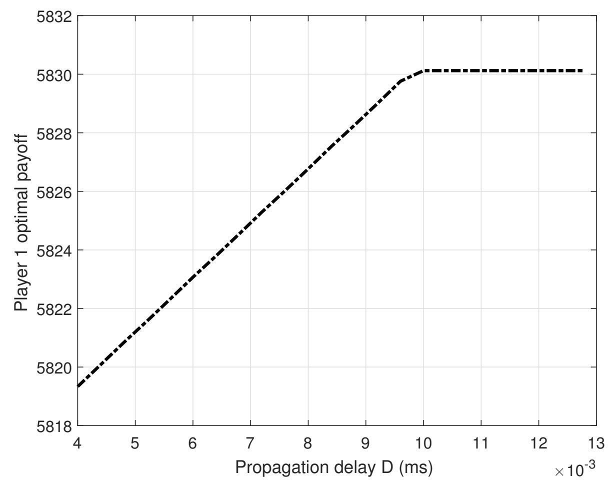

Fig. 3. Player 1 cost in a complete knowledge scenario varying D.

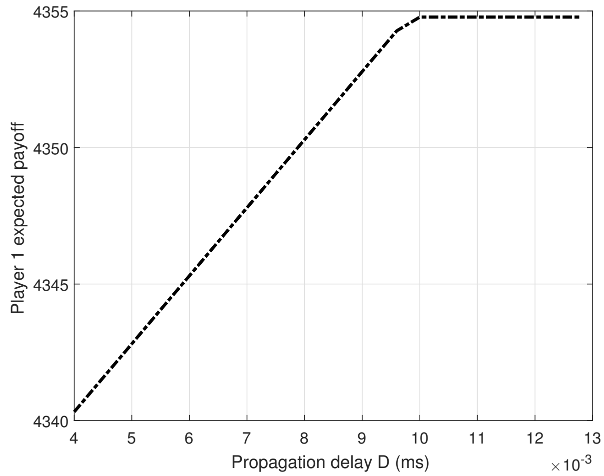

Fig. 4. Player 1 expected cost varying D.

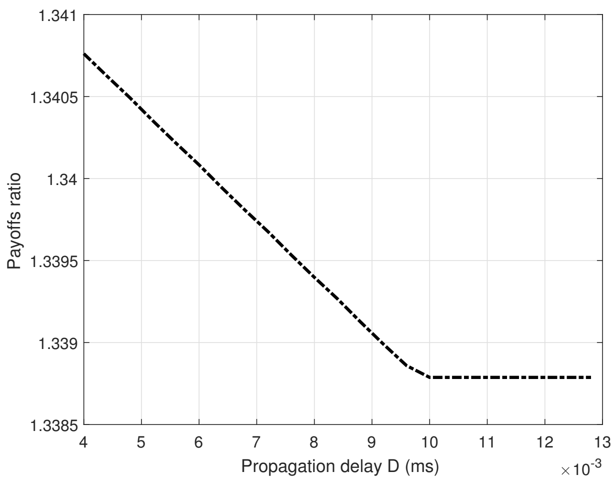

Fig. 5. Ratio between player 1 optimal cost and player 1 expected cost varying D.

harmful for the other player) offloading operations.

Fig. 5 evaluates this further by considering the ratio between the ideal scenario with full knowledge and the expected cost in the Bayesian setup. Again there is a dependance on D that keeps increasing until a ceiling level is hit. In general, in the complete knowledge case, each player i knowing the opponent j distance $\delta(n_j, s)$ decides to offload the computation task only when this choice represents a NE, i.e., (Cloud, Cloud) or (Edge, Edge). In the Bayesian case, player i does not know $\delta(n_j, s)$, as a consequence, it might happen that it decides to offload the computation task even if the other player j choice is to locally execute its own task. In this way, player i has a lower computation cost. Indeed, the pairs of actions (Cloud, Local), (Edge, Local), (Local, Cloud), and (Local, Edge) give a lower computation cost with respect to (Local, Local) for the player that is choosing the task offloading. These asymmetrical pairs are not considered in the optimal case since they do not represent NEs, however they might arise in the Bayesian case due to the uncertainty on the players' distances from the BS.

VI. Conclusions

We applied Bayesian game theory to a network scenario where mobile users can offload computations task to edge or cloud resources. Our objective is to make efficient offloading in a scenario where the mobile devices are unaware of the network load and mutual interference reduces the capacity of the wireless links connecting to the local base station. Numerical results illustrate regions where the strategy of the devices converges to different points of equilibrium.

The remarkable finding of our analysis is that a Bayesian scenario, i.e., with imperfect information, actually achieves better welfare (or lower social cost) than one with full knowledge by the users. This is due to the competitive nature of the game, where selfish rational users would try offloading computational task even when it is not efficient for the entire network that they do so. Our results suggest an interesting guideline in the design for offloading strategies, i.e., to make network parameters less known to the users in order to improve their cooperative participation to resource sharing.

References

[1] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog computing and its role in the Internet of Things," *Proc. Mobile Cloud Comput.*, pp. 13–16, 2012.

[2] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, "A survey on applications of model-free strategy learning in cognitive wireless networks, *IEEE Commun. Surv.& Tut.*, vol. 18, no. 3, pp. 1717-1757, Jan. 2016.

[3] M. Ghazvini, N. Movahedinia, K. Jamshidi, and N. Moghim, "Game theory applications in CSMA methods," *IEEE Commun. Surv.& Tut.*, vol. 15, no. 3, pp. 1062-1087, 2013.

[4] D. Niyato and E. Hossain, "Radio resource management games in wireless networks: an approach to bandwidth allocation and admission control for polling service in IEEE 802.16," *IEEE Wirel. Commun.*, vol. 14, no. 1, pp. 27-35, 2007.

[5] R. Trestian, O. Ormond, and G.-M. Muntean, "Reputation-based network selection mechanism using game theory," *Phys. Commun.*, vol. 4, no. 3, pp. 156-171, 2011.

[6] G. Scutari and D. P. Palomar, "MIMO cognitive radio: a game theoretical approach," *IEEE Trans. Sig. Proc.*, vol. 58, no. 2, pp. 761-780, Feb. 2010.

[7] M. A. Khan, H. Tembine, and A. V Vasilakos, "Evolutionary coalitional games: design and challenges in wireless networks," *IEEE Wirel. Commun.*, vol. 19, no. 2, pp. 50-56, 2012.

[8] T. AlSkaif, M. G. Zapata, and B. Bellalta, "Game theory for energy efficiency in wireless sensor networks: latest trends," *J. Netw. Comput. Appl.*, vol. 54, pp. 33-61, 2015.

[9] E. Abebe and C. Ryan, "Adaptive application offloading using distributed abstract class graphs in mobile environments," *J. Syst. Softw.*, vol. 85, no. 12, pp. 2755–2769, 2012.

[10] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," *Proc. EuroSys*, pp. 301–314, 2011.

[11] R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. Vasilakos, "MAPCloud: mobile applications on an elastic and scalable 2-tier cloud architecture," *Proc. IEEE UCC*, pp. 83–90, 2012.

[12] R. Rahimi, N. Venkatasubramanian, and A. Vasilakos, "MuSIC: on mobility-aware optimal service allocation in mobile cloud computing," *Proc. IEEE Cloud*, 2013.

[13] L. Yang, J. Cao, and H. Cheng, "Resource constrained multi-user computation partitioning for interactive mobile cloud applications," *tech. rep., Dept. of Computing*, Hong Kong Polytechnic Univ., 2012.

[14] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *Sigmetrics Perf. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, 2013.

[15] V. Cardellini, V. De Nitto Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. Lo Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Progr.*, vol. 157, no. 2, pp. 421–449, 2016.

[16] S. Tadelis, *Game theory: an introduction,* Princeton Univ. Press, 2013.

[17] A. V. Guglielmi and L. Badia, "Bayesian game analysis of a queueing system with multiple candidate servers," *Proc. IEEE CAMAD*, 2015.