A Distributed Coalition Game Approach to Femto-Cloud Formation

S. M. Shahrear Tanzil, Omid Namvar Gharehshiran, and Vikram Krishnamurthy, *Fellow, IEEE*

Abstract—This paper studies distributed formation of femto-clouds in a UMTS LTE network. Femtocell access points (FAPs) are equipped with computational resources. They share their resources with neighboring FAPs and form local clouds with the aim to avoid the remote cloud costs while improving the user quality of experience (QoE) in terms of handling latency. In exchange for sharing their excess resources, FAPs receive monetary incentives proportional to their contribution in performing computational tasks in the femto-cloud. The resource sharing problem is formulated as an optimization problem and a myopic procedure is presented that enables FAPs to collaboratively find its solution in a distributed fashion. In such an optimal femto-cloud structure, the local computational resources of FAPs are maximally exploited, yet the incentive earned by each femto-cloud is divided among the FAPs in a fair fashion. Numerical simulations using NS-3 verify superior QoE of users as well as higher incentives provided to FAP owners as compared with alternative heuristic schemes. Numerical results also show that the grand femto-cloud—the largest collaborative cloud comprising of all FAPs—is not always the optimal structure.

Index Terms—Mobile cloud computing, femto-clouds, distributed coalition formation, cooperative game theory, quality of experience (QoE).

1 Introduction

To increase the semantic richness of sensed data in personal assistant applications such as Apple Siri, Google Now, and Microsoft's Cortana, high data rate sensors such as vision-based sensors are required [1]. Analyzing real-time video and images captured by such sensors, however, requires intensive computational capacity, which makes it costly (in terms of energy consumption) to be processed in mobile devices. Therefore, offloading-based mechanisms have been developed to support vision-based functionalities [1], [2], [3].

One such solution is mobile cloud computing (MCC) [4] that augments the computational capacity of mobile devices by offloading computation and storage to a remote cloud. The interactive response essential for real-time video/image processing is, however, limited by two major bottlenecks in MCC, namely, energy consumption and latency [2], [5], [6], [7]. Therefore, the concept of *cloudlet* has been introduced in [2]: A trusted local cloud comprised of multi-core computers that is connected to the Internet and is available for use within the proximity of mobile users. Mobile devices use Wi-Fi network to offload the computation tasks to the cloudlet, which saves them considerable amount of energy as compared to offloading over the 3G/Long Term Evolution (LTE) cellular network to remote cloud [3], [8]. This prolongs the battery lifetime of mobile devices and, by reducing network latency, improves user's quality of experience (QoE) [9]. In a European project, namely, TROPIC [7], the cloudlet has further been integrated into small-cell access points, such as femtocell access points (FAPs) [6], [7], [9], to perform computations on behalf of mobile devices. The advantage is that femtocells, in contrast to Wi-Fi, work under the same communication standard as the LTE cellular network.

The main idea in this work is to allow FAPs augmented with computational resources to cooperate with each other and form local computational pools, namely, *femto-clouds*. FAPs share the computational resources exceeding their demands in femto-clouds. Therefore, by maximally exploiting FAPs' local resources, such femto-clouds reduce latency[1] and, hence, improve end-user QoE. We assume that FAPs are deployed by different residential users. To motivate FAP owners to share their excess resources, it is natural to assume an incentive mechanism. The maximal use of FAP resources then translates into both lower handling latency and higher incentives to FAP owners. The question that this paper focuses on is then: How should FAPs decide on formation of such femto-clouds in a distributed fashion?

The data transfer delay and limited computational capacity of FAPs impose stringent constraints that naturally prohibit formation of the grand coalition to which all FAPs join, namely, grand femto-cloud. Since offloading tasks to other FAPs within a femto-cloud incurs delay, it is not beneficial to collaborate with FAPs that are far away. On the other hand, the computation tasks exceeding the computational capacity of the femto-clouds have to be transported to the remote cloud. This incurs both data transfer delay and remote cloud costs. If such a cost exceeds the associated incentives, all FAPs within the femto-cloud will be responsible for the loss. Formation of the grand femto-cloud produces a huge pool of tasks, and increases the probability of such losses.

- *S. M. S. Tanzil and V. Krishnamurthy are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, V6T 1Z4, Canada. E-mail: {stanzil, vikramk}@ece.ubc.ca.*
- *O. N. Gharehshiran is with the Department of Statistical Sciences, University of Toronto, Toronto, M5S 3G3, Canada. E-mail: omidn@utstat.toronto.edu.*
- *This research was supported by an NSERC Strategic grant.*

1. Latency can be formulated as the sum of computational delay and data transfer delay.

Therefore, FAPs form femto-clouds in a way to minimize tasks that are needed to be transported to the remote cloud. The proposed femto-cloud formation scheme identifies such optimal localized femto-clouds, to which only a subset of FAPs subscribe, in a distributed fashion.

The main results in this paper are:

1) Formulation of the incentive-based femto-cloud formation problem: The resource sharing problem is formulated as an optimization problem with the objective to maximize the overall utility of all femto-clouds with constraints on the fair division of incentives among individual FAPs within a femto-cloud. The utility function of each femto-cloud takes into account the profile of request arrivals in individual femtocells, previous cooperative behavior of FAPs, data transfer delay, and computational capacity of FAPs to determine the overall incentive available to each femto-clouds. Therefore, solving the formulated problem translates into finding the femto-cloud structure that maximizes utilization of FAPs' local resources (taking into account users' experience), yet provides incentives to FAPs for sharing their resources such that no FAP is willing to give up collaboration within its current femto-cloud to join another femto-cloud.

2) Distributed femto-cloud formation algorithm: The similarities between the formulated femto-cloud formation problem and coalition formation games enable us to employ the dynamic coalition formation algorithm in [10] to devise a procedure that prescribes individual FAPs how to revise their decisions as to which femto-cloud to join so as to reach the solution of formulated problem (i.e., core of the underlying coalition formation game) in a distributed fashion.

3) Numerical results: Finally, numerical simulations implemented on the LTE protocol stack in NS-3 illustrate superior performance of the proposed scheme in terms of both handling latency and incentives provided to FAP owners over alternative heuristic femto-cloud formation schemes. They further confirm that forming a grand femto-cloud, comprising of all FAPs in the network, is not always the optimal choice.

The material presented in the current paper extends the conference version [11] in several respects:

1) Problem formulation: The formulation in Sec. 3 extends that in [11] by considering the task request variability as well as a trust parameter that captures the previous collaboration performance of individual FAPs. This paper further considers task request statistics instead of its distribution which is more realistic to monitor in practice.

2) Implementation considerations: Section 4.2 has been added to shed light on the details crucial to implementing the proposed distributed femto-cloud formation scheme.

3) Simulation results: The numerical results substantially extends those in [11] by considering different task types handled simultaneously and various scenarios to better evaluate its efficacy. Finally, to better illustrate the performance gains achievable from the proposed scheme, the simulations are performed on a larger network.

1.1 Related Work

Here, we provide a brief description of relevant works in the literature.

1.1.1 Collaboration among cloud providers

There is a large body of research devoted to studying cooperation in cloud computing framework; see, e.g., [12], [13], [14]. Cooperation among mobile cloud service providers is studied in [12] for pooling computational resources with the goal to maximize revenue. The authors then use Shapley value to distribute the revenue among the collaborating cloud service providers. In [15], a cooperative outsourcing strategy is proposed which prescribes the providers whether to satisfy users' requests locally or to outsource to a certain provider. Dynamic cloud federation formation is also studied in [16].

1.1.2 Collaboration among femtocells

Coalition formation in femtocell network has been extensively studied in the literature; see, e.g., [17], [18], [19]. For instance, [20] studies coalition formation among femtocells in order to mitigate interference in the network. In [19], an interference management model is developed in a femtocell network wherein the cooperation problem is formulated as a coalition formation game with overlapping conditions. Rami *et al.* [21] also consider resource and power allocation in cooperative femtocell networks. All these works consider cooperation among femtocells with the aim to improve physical-layer throughput.

1.1.3 Incentives for cooperation in femtocell network

Femtocells are typically deployed by mobile network operators in an open/hybrid access mode, in which FAPs are willing to accommodate guest users; see, e.g., [22], [23], [24], [25]. To motivate FAP owners to adopt such an access mode, several incentive schemes have been studied in the literature, e.g., [22], [23], [24], [25], [26], [27]. Incentives can be categorized as *reputation* or *remuneration* [28]. Reputation reflects the willingness of wireless nodes' to cooperate with other nodes. Nodes receive services from other nodes based on their past behavior—misbehaving nodes are deprived from receiving services. In contrast, remuneration-based mechanisms provide monetary incentives for cooperation, e.g., micropayment, virtual currency, E-cash, and credit transfer [29], [30], [31], [32].

1.1.4 Femto-clouds

Femto-clouds are relatively recent and only few studies can be found in the literature. For instance, [9] proposes a mechanism for joint optimization of communication and computational resources. In [6], [33], [34], an offloading strategy is proposed for femto-clouds. All these works consider the cloud offloading mechanism while assuming that FAPs are already grouped into coalitions. Femto-clouds differ substantially from cloud radio access networks (CRAN) [35] in that FAPs are endowed with computational resources and the offloaded computations are preferred to be performed locally rather than in a centralized cloud (e.g. remote radio head in CRAN) to reduce handling latency.

Jessica *et al.* propose cluster formation strategies in [36] to handle a single user's requests in femto-clouds. These strategies are devised with different objectives, e.g., to minimize the experienced latency or to reduce power consumption in the cluster. This work is extended to a multi-user

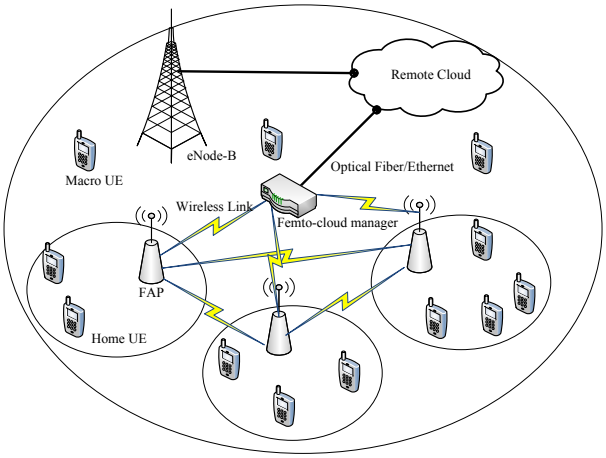

Fig. 1. A typical femto-cloud architecture. The macrocell and femtocell base stations are referred to as eNode-B and femtocell access point (FAP), respectively, and the end users are referred to as user equipment (UE). FAPs are connected to their closest femtocell cloud manager (FCM) via the Z interface while FCM is linked with the remote cloud via optical fiber/ethernet. The FAPs are also connected to the neighbouring FAPs via the Z interface.

scenario in [37] where clusters are formed for each unserved request according to the strategies proposed in [36]. Their model, however, is suitable only for enterprise femtocell environments where all FAPs share their computational resources with each other. Moreover, cluster formation for each unserved request significantly increases the signaling overhead. To the best of our knowledge, the formulation and distributed scheme proposed in this paper for formation of femto-clouds considering a remuneration incentive mechanism and taking into account the delay involved in migrating tasks between FAPs have not been studied before.

The rest of this paper is organized as follows: System architecture is described in Sec. 2. The utility function is defined in Sec. 3. The distributed femto-cloud formation algorithm is presented in Sec. 4. Numerical studies are provided in Sec. 5. Finally, Sec. 6 concludes the paper.

2 System Architecture

We consider a UMTS LTE architecture with K FAPs/Home eNode-Bs (HeNBs) endowed with heterogeneous computational capacity. Each FAP is located in a separate room and possibly different floor of a multi-story building. The FAPs share bandwidth with a macro base station (BS) as shown in Fig. 1, and are deployed by different residential users.

We assume that there exist N_F femtocell cloud managers (FCMs) in the building, where $N_F < K$. The FAPs are connected to their closest FCMs via Z interface according to the proposed standalone FCM architecture in [7]. FCMs are responsible for:

(i) gathering task request information of the connected FAPs, and exchanging this information with neighboring FCMs;
(ii) implementing the incentive mechanism by monitoring the tasks completed by each FAP;
(iii) performing computations for the femto-cloud formation mechanism proposed in this paper.

TABLE 1
Notations and Terminology

System Parameters	Description
K	Number of FAPs
N_F	Number of FCMs
R_k	Trust/reputation value of FAP k
$d_k^{\max}$	Computational capacity of FAP k
$D_{\mathcal{C}}^{\max}$	Overall computational capacity of femto-cloud $\mathcal{C}$
$b_{k,l}$	Uplink data transmission rate from FAP k to FAP l
b_k	Uplink data transmission rate from FAP k to FCM
L	WAN latency for sending tasks to remote cloud

Task Request	Description
N_B	Data size
$\overline{d}_k$	Sample mean of task requests received by FAP k
$\overline{D}_{\mathcal{C}}$	Sample mean of task requests in femto-cloud $\mathcal{C}$
$\overline{H}_{\mathcal{C}}$	Entropy of total task requests in femto-cloud $\mathcal{C}$

Utility Function	Description
m_r	Revenue per unit task
m_p	Proportionality constant for trust
c_r	Remote cloud charges per unit task
c_o	Offloading delay cost
c_u	Penalty for demand uncertainty

FCMs are connected to the remote cloud via optical fiber links, hence, can offload the computational tasks of the connected FAPs to the remote cloud with no intervention of the core network. The FCMs substantially reduce the traffic generated by the MCC in the core network. It is therefore natural to assume that FCMs are installed and maintained by the mobile network operators.

It is assumed that FAPs are connected to the core network via wireless backhaul, and can be deployed by the residential users in a plug-and-play fashion. The FAPs use the 2.6 GHz licensed bandwidth to connect with FCMs, and communicate with other FAPs via the Z interface in a multicast fashion. Since FAPs and FCMs are located in different rooms/floors of the building, the FAP-FAP and FAP-FCM signal propagation undergo several losses. Here, we only consider external wall loss, shadowing loss, and the 2.6 GHz path loss models. As a result, the FAP-FAP communication delay depends mainly on the location of the FAPs.

3 Formulation of the Femto-Cloud Formation Problem

This section formulates the femto-cloud formation problem. We first formulate the utility function that quantifies the performance of individual femto-clouds in Sec. 3.1. The global femto-cloud formation problem with fair allocation of incentives to FAPs is then formalized in Sec. 3.2. We finally discuss the similarities between the formulated problem and coalition formation games. Table 1 summarizes the notations used in this section.

3.1 Local Femto-Clouds and Their Utility

Mobile devices make decisions on offloading their tasks to FAPs based on the handling latency [33]. If offloaded to FAPs, they will then decide whether to perform computations locally or send them to the remote cloud taking into account the users' QoE requirements, their computational capacity and workload. The main goal in this paper is to motivate a cooperation protocol to maximally exploit FAPs' local resources. Neighboring FAPs form collaborative coalitions to increase local computational capacity. Since FAPs are densely deployed, sending the data for the requested tasks to such local *femto-clouds* incurs less latency as compared to the remote cloud. This improves users' QoE while enabling FAP owners to earn incentive by sharing their excess resources.

Resource sharing problems can generally be formulated as constrained optimization problems with a utility function that trades-off the benefits and costs associated with collaboration by sharing resources. Consider a set of FAPs, indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$, and let $\mathcal{C} \subseteq \mathcal{K}$ denote a coalition of FAPs formed for a fixed time interval over which the parameters described below remain constant. The case $|\mathcal{C}| > 1$ is referred to as a *femto-cloud*, whereas $|\mathcal{C}| = 1$ is referred to as an *isolated FAP*. Here, $|\cdot|$ denotes the cardinality operator. The performance of femto-clouds are then quantified by the function $U : 2^{\mathcal{K}} - \emptyset \rightarrow \mathbb{R}$, where $2^{\mathcal{K}}$ denotes the power set of the set of FAPs $\mathcal{K}$. This function quantifies the total *incentive* earned by a femto-cloud as the result of FAPs sharing their resources, which is then divided among the FAPs in the femto-cloud, and is formulated as

$$U(\mathcal{C}) = U^r(\mathcal{C}) - U^c(\mathcal{C}) + U^p(\mathcal{C}), \tag{1}$$

where each term on the right hand side is described below:

The first term $U^r(\mathcal{C})$ models the revenue earned by the femto-cloud and is formulated as

$$U^r(\mathcal{C}) = m_r \cdot \overline{D}_{\mathcal{C}}, \tag{2}$$

where m_r is the revenue per unit task (\$/task). Further, $\overline{D}_{\mathcal{C}}$ denotes the sample mean of the task requests received by femto-cloud $\mathcal{C}$ over the past time slots since the femto-cloud has been modified/formed. If the requested tasks for a particular FAP exceed its computational capacity, the FAP offloads tasks to femto-cloud members and shares the incentive with them. Since femto-clouds are formed for several time slots, rather than dealing with instantaneous offloaded tasks, the incentive function relies on the previously observed statistics of requests.

The second term $U^c(\mathcal{C})$ in (1) represents the costs incurred by forming a femto-cloud, and is comprised of four terms

$$U^c(\mathcal{C}) = U^c_r(\mathcal{C}) + U^c_{o,r}(\mathcal{C}) + U^c_{o,m}(\mathcal{C}) + U^c_u(\mathcal{C}) \tag{3}$$

where each term is described below:

1) Remote cloud cost: When the accumulated task requests within a femto-cloud exceeds its computational capacity, the excess tasks have to be offloaded to the remote cloud to avoid processing delays. This incurs two types of costs:

a) Remote cloud processing cost: The term $U^c_r(\mathcal{C})$ in (3) models the remote cloud processing cost

$$U^c_r(\mathcal{C}) = c_r \cdot \left|\overline{D}_{\mathcal{C}} - D^{\max}_{\mathcal{C}}\right|^+, \tag{4}$$

where c_r is the remote cloud charges in \$/task. Further, $|x|^+ = \max\{0, x\}$, and $D^{\max}_{\mathcal{C}} = \sum_{k\in\mathcal{C}} d^{\max}_k$ is the overall computational capacity of femto-cloud $\mathcal{C}$, where $d^{\max}_k$ represents the computational capacity[2] of the k-th FAP. This term motivates FAPs to form coalitions with FAPs with low workload to computational capacity ratio.

b) Remote cloud offloading delay cost: The second term $U^c_{o,r}(\mathcal{C})$ in (3) is the penalty associated with the data transfer delay in offloading excess femto-cloud workload to the remote cloud, and is formulated as

$$U^c_{o,r}(\mathcal{C}) = c_o \cdot \left(\left|\overline{D}_{\mathcal{C}} - D^{\max}_{\mathcal{C}}\right|^+ \cdot \left(\frac{N_B}{\min_{k\in\mathcal{C}} b_k} + L\right)\right). \tag{5}$$

Here, N_B denotes the data size, in bytes, of a task, b_k is the uplink data transmission rate, in bytes/sec, from k-th FAP to FCM, L represents the wide area network (WAN) latency introduced by transporting the task to the remote cloud via the FCM, and c_o (\$/sec) is the dimension for proportionality constant.

2) Multicast offloading delay to FAPs: The term $U^c_{o,m}(\mathcal{C})$ in (3) represents the penalty for the delay in transmitting data, associated with the tasks exceeding FAPs' computational, to the femto-cloud into a monetary penalty. It provides incentive for FAPs to collaborate with neighboring FAPs to decrease the handling delay and improve the QoE of users, and is formally given by

$$U^c_{o,m}(\mathcal{C}) = c_o \cdot \left(\sum_{k\in\mathcal{C}} \left|\overline{d}_k - d^{\max}_k\right|^+ \cdot \frac{N_B}{\min_{l\in\mathcal{C}-\{k\}} b_{k,l}}\right). \tag{6}$$

Here, $b_{k,l}$ denotes the uplink data transmission rate from the k-th FAP to the l-th FAP, $\overline{d}_k$ is the sample mean of the task request in the k-th FAP over the past time slots since the femto-cloud has been modified/formed. Finally, $\overline{d}_k - d^{\max}_k$ is the number of tasks that exceeds the computational capacity of the k-th FAP, and have to be sent to the cloud.

3) Demand uncertainty cost: Since femto-clouds are formed for multiple time slots and we use sample statistics rather than instantaneous task requests, it is important to account for deviation from the mean demand so as to avoid remote cloud costs. The last component of the cost function captures such uncertainty in the overall femto-cloud demand, and is formulated as

$$U^c_u(\mathcal{C}) = c_u \cdot \overline{H}_{\mathcal{C}}, \tag{7}$$

where $\overline{H}_{\mathcal{C}}$ denotes the sample entropy of the accumulated task request time series. This term simply motivates FAPs to form femto-clouds with FAPs with less variability around their mean computational demand.

Finally, the last term $U^p(\mathcal{C})$ in (1) models the priority value of the coalition $\mathcal{C}$. With each FAP, there corresponds a *trust value*, denoted by R_k, that captures the quality of its previous cooperative behavior [38]. By joining femto-clouds and successfully performing computations offloaded

2. One unit of computational capacity is equal to one unit of workload.

by other cloud members, FAPs earn trust. Femto-cloud comprising of FAPs with higher trust values are expected to perform tasks in a timely manner; therefore, the service provider is willing to provide them with higher monetary incentives as they improve the users QoE. This further eliminates free-rider FAPs that join coalitions to obtain incentives without performing tasks.

We formulate $U^p(\mathcal{C})$ as follows:

$$U^p(\mathcal{C}) = m_p \cdot \left(\sum_{k \in \mathcal{C}} R_k \cdot \frac{\min\{d_k^{\max}, f\}}{\overline{d}_k} \right), \tag{8}$$

where m_p (\$) is the proportionality constant that determines the relative weight of trust in formation of femto-clouds, and f is a system parameter[3] that depends on the overall task requests in the system. Note in the above formulation that higher priority is placed on FAPs with lower mean demand to computational capacity ratios and higher trust values. It is assumed that FCMs are responsible for updating the trust values for their neighboring FAPs. The mechanism for updating these trust values is however out of the scope of this paper, and merits a separate publication. We further assume that R_k remains constant for several time slots while the FCM monitors the k-th FAP cooperative behavior, and is only updated when the femto-clouds structure is being modified.

3.2 Optimization of the Femto-clouds with FAP Incentives

As mentioned in Sec. 3.1, FAPs expect incentives for sharing their excess resources. Let $\mathbf{r} = (r_1, \ldots, r_K)$ denote the incentive allocation vector. Each element r_k represents the share of each FAP k from the total incentive obtained by the femto-cloud $\mathcal{C}$ that FAP k have joined. To make the problem mathematically tractable, the set of incentive values is confined to a finite set. Suppose Δ (\$) is the smallest incentive unit. Each FAP's demand is then restricted to the set

$$\mathcal{P} = \left\{ \widehat{n}\Delta; \widehat{n} \in \mathbb{N}, 0 \leq \widehat{n}\Delta \leq \max_{\mathcal{C} \in 2^{\mathcal{K}} - \emptyset} U(\mathcal{C}) \right\}, \tag{9}$$

where $\mathbb{N}$ represents the set of all natural numbers, and the function $U(\cdot)$ is defined in Sec. 3.1. Let further $\mathcal{B}$ denote the set of all possible femto-cloud structures. Each *femto-cloud structure* $\mathcal{S}$ is a partition on the set $\mathcal{K}$, i.e., $\cup_{\mathcal{C} \in \mathcal{S}} \mathcal{C} = \mathcal{K}$. The femto-cloud formation problem is then formulated as

$$\begin{aligned} \max_{\mathcal{S} \in \mathcal{B}} \quad & \sum_{\mathcal{C} \in \mathcal{S}} \lfloor U(\mathcal{C}) \rfloor_\Delta, \\ \text{s.t.} \quad & r_k \in \mathcal{P}, \\ & \sum_{k \in \mathcal{C}} r_k = \lfloor U(\mathcal{C}) \rfloor_\Delta, \qquad \forall \mathcal{C} \in \mathcal{B}, \\ & \sum_{k \in \mathcal{C}'} r_k \geq \lfloor U(\mathcal{C}') \rfloor_\Delta, \quad \forall \mathcal{C}' \subseteq \mathcal{K}, \mathcal{C}' \neq \emptyset. \end{aligned} \tag{10}$$

where $\lfloor x \rfloor_\Delta = \lfloor \frac{x}{\Delta} \rfloor \cdot \Delta$ denotes the greatest integer multiple of the smallest divisible incentive unit Δ, and $\mathcal{P}$ is defined in (9).

3. Taking the minimum in (8) is a technicality to avoid obtaining excess priority for computational capacity that exceeds the femto-cloud demands.

Before proceeding to provide an intuitive interpretation of (10), a few definitions are in order. Let $\mathbf{r}$ and $\mathbf{r}'$ denote two $K \times 1$ incentive vectors. The product ordering $\mathbf{r} \leq \mathbf{r}'$ holds if and only if $r_k \leq r'_k$ for all $k \in \mathcal{K}$. An incentive allocation $\mathbf{r}$ is then called *efficient* if the sum of incentives of all FAPs is equal to the maximum total incentive, achievable under the most desirable femto-cloud structure. In addition, if a group of FAPs can form a femto-cloud $\mathcal{C}'$ where the division of coalition's incentive guarantees $\mathbf{r}' \geq \mathbf{r}$, then $\mathcal{C}'$ will *block* the currently formed femto-cloud $\mathcal{C}$ and the associated incentive vector $\mathbf{r}$. An incentive vector $\mathbf{r}$ is called *non-blocking* if for all possible femto-clouds $\mathcal{C}'$, the associated incentive $\mathbf{r}'$ satisfies $\mathbf{r} \geq \mathbf{r}'$. The second constraint in (10) ensures that the incentives allocated to FAPs are efficient. The third constraint in (10) is the non-blocking condition, and can be interpreted as a *fairness* criterion on the division of incentives among FAPs in each femto-cloud. An incentive allocation vector is called fair if no FAP can gain higher incentive by sharing its resources with a different group of FAPs. The solution to (10) can thus be considered as the optimal femto-cloud structure in that: i) the computational capacity of all FAPs is maximally exploited, and ii) the FAP incentives are distributed in a fair fashion within each femto-cloud.

Coalition Formation Game Interpretation: The femto-cloud formation problem with FAP incentives outlined above fits well within the context of *coalition formation games*. The coalition formation games encompass cooperative games where the coalition structure plays a major role, and are defined by the pair $(\mathcal{G}, V)$, where $\mathcal{G}$ denotes the set of players and $V : 2^{\mathcal{G}} - \emptyset \rightarrow \mathbb{R}$ is the *characteristic function*[4]. This function associates with any non-empty coalition a number that quantifies the total payoff that can be gained by the coalition. A cooperative game is called *superadditive* if for any two disjoint coalitions $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{G}$:

$$V(\mathcal{C}_1 \cup \mathcal{C}_2) \geq V(\mathcal{C}_1) + V(\mathcal{C}_2).$$

In superadditive games, the *grand* coalition—the coalition consisting all players—forms the stable coalition structure. The *coalition formation* games encompass cooperative games where the coalition structure plays a major role. These games are generally non-superadditive; therefore, the optimal coalition structure may be comprised of several disjoint coalitions. Due to the data transfer delay and limited computational capacity of FAPs, it is intuitive that the optimal structure of femto-clouds has to incorporate several disjoint coalitions of FAPs. It is thus natural to formulate the femto-cloud formation problem as a coalition formation game with $\mathcal{G} = \mathcal{K}$ and $V(\cdot) = U(\cdot)$. In particular, the solution of the femto-cloud formation problem (10) is identical to a solution notion in coalition formation games, namely, *modified core* [10]. Therefore, solving (10) is equivalent to finding the modified core of the underlying coalition formation game. The interested reader is referred to [39], [40], [41] for further details.

4. The term characteristic function is as used in cooperative games and is unrelated to characteristic functions in probability theory.

4 Distributed Femto-Cloud Formation and Convergence to the Core

This section presents a distributed femto-cloud formation algorithm that guarantees convergence to the solution of (10) almost surely, and elaborates on its implementation considerations.

4.1 Distributed Femto-Cloud Formation Algorithm

Define network state pair by $\omega = (\mathcal{S}, \mathbf{r})$, which contains the femto-clouds structure $\mathcal{S}$ and the incentive vector of FAPs $\mathbf{r}$. The distributed femto-cloud formation procedure relies on the dynamic coalition formation algorithm proposed in [10] and is summarized below in Algorithm 1. The advantage of using the decentralized procedure in Algorithm 1 over centralized solutions is that it retains autonomy of FAP owners as whether to collaborate and better captures the dynamics of the negotiation process among them [10]. In a centralized solution, FAP owners have to be forced to follow the calculated optimal femto-cloud structure. In fact, if an FAP owner decides to not follow the prescription, the implemented femto-cloud structure is no longer the optimal solution. In contrast, the decentralized solution implemented in Algorithm 1 mimics the natural procedure that FAP owners will follow to form collaborative groups—they explore their options and settle in the femto-cloud that provides the highest feasible incentive. The implementation considerations will be addressed in the next subsection.

The myopic best-reply strategy implemented in Step 2.1-2.3 of Algorithm 1 defines a finite-state Markov chain, namely, *best-reply process* [10]. Standard results on finite state Markov chains show that, no matter where the process starts, the probability that the best-reply process reaches a recurrent set of states after n iterations tends to one as n tends to infinity. The outcome that which of these ergodic states will eventually be reached is determined by the initial state. Under the best-reply process, absorbing states do not necessarily guarantee reaching the solution of (10). To address this issue, *perturbation* has to be introduced. That is, to allow FAPs deviate from optimal strategies and choose sub-optimal strategies with a small probability with the hope of achieving higher incentives. The interested reader is referred to [10] for details and further discussion.

Deviation from the best-reply process, namely, *experimentation*, is formally defined as follows: In any state, when there exists a potential femto-cloud $\mathcal{C}' \in 2^{\mathcal{K}}$ such that

$$\textstyle\sum_{k\in\mathcal{C}'} r_k < \lfloor U(\mathcal{C}') \rfloor_\Delta, \tag{13}$$

each FAP $k \in \mathcal{C}'$ follows the best-reply process of Step 2.1-2.3 with probability $1-\varepsilon$. With the remaining probability ε, it randomly joins an existing femto-cloud, and demands the surplus incentive that the femto-cloud expects to achieve as the result of FAP k joining it. The blocking condition (13) is checked in Step 1 of Algorithm 1. This modified best-reply process defines a finite-state Markov chain, namely, *best-reply process with experimentation* [10], with the same state space as the best-reply process (without experimentation) and slightly modified transition probabilities.

The limiting distribution of the best-reply process with experimentation summarized in Algorithm 1 assigns probability one to the states $(\mathcal{S}^n, \mathbf{r}^n)$ that solve the femto-cloud formation problem (10). This result is summarized in the following theorem.

Algorithm 1 Distributed Femto-Cloud Formation

Initialization. Set $0 < \varepsilon, \rho < 1$, where ρ is the probability of revising strategy and ε is the experimentation probability. Initialize $\omega^0 = (\mathcal{S}^0, \mathbf{r}^0)$, where

$$\mathcal{S}^0 = \big\{\{1\}, \ldots, \{K\}\big\}, \mathbf{r}^0 = \big(\widehat{r}_1, \ldots, \widehat{r}_K\big), \text{and } \widehat{r}_k = U(\{k\}).$$

Step 1. Find blocking coalitions by FCM:
Let $\mathcal{A}^n = \emptyset$. For all $\mathcal{C} \in 2^{\mathcal{K}} - \emptyset$,

$$\text{if } \textstyle\sum_{k\in\mathcal{C}} r_k^n < \lfloor U(\mathcal{C}) \rfloor_\Delta, \text{ then } \mathcal{A}^n \leftarrow \mathcal{A}^n \cup \mathcal{C}.$$

Step 2. Each FAP $k \in \{1, \ldots, K\}$ independently performs:

Step 2.1. With probability ρ, continue with Step 2.2. With the remaining probability $1-\rho$, stay in the same coalition, set $r_k^{n+1} = r_k^n$, and go to Step 2.5.

Step 2.2. Compute

$$\widetilde{\mathcal{C}}_k^{n+1} = \underset{\mathcal{C}\in\mathcal{S}^n\cup\emptyset}{\operatorname{argmax}} \left(\lfloor U(\mathcal{C}\cup\{k\}) \rfloor_\Delta - \sum_{l\in\mathcal{C}, l\neq k} r_l^n \right) \tag{11}$$

$$\widetilde{r}_k^{n+1} = \big\lfloor U\big(\widetilde{\mathcal{C}}_k^{n+1}\cup\{k\}\big) \big\rfloor_\Delta - \sum_{l\in\widetilde{\mathcal{C}}_k^{n+1}, l\neq k} r_l^n \tag{12}$$

Step 2.3. If $k \in \mathcal{A}^n$, with probability ε, go to Step 2.4. With the remaining probability $1-\varepsilon$, sample uniformly from the set $\mathcal{S}^n \cup \emptyset$, denote it by $\widetilde{\mathcal{C}}_k^{n+1}$, and set $r_k^{n+1} = \widetilde{r}_k^{n+1}$, where $\widetilde{r}_k^{n+1}$ is computed according to (12). Go to Step 2.5.

Step 2.4. Set $r_k^{n+1} = \widetilde{r}_k^{n+1}$ and, if non-singleton, randomize among $\widetilde{\mathcal{C}}_k^{n+1}$ uniformly.

Step 2.5. If $k \neq K$, continue with the next FAP.

Step 3. Form $\omega^{n+1} = (\mathcal{S}^{n+1}, \mathbf{r}^{n+1})$.
Set $n \leftarrow n+1$ and go to Step 1.

Theorem 4.1. Let $\omega^c = (\mathcal{S}^c, \mathbf{r}^c)$ denote the states that solve the femto-cloud formation problem (10). Then, the sample path of $\omega^n = (\mathcal{S}^n, \mathbf{r}^n)$ generated by Algorithm 1 converges almost surely to the core, i.e.,

$$P\big(\lim_{n\to\infty} \omega^n = \omega^c\big) = 1, \tag{14}$$

for all initializations ω^0 if the solution set is non-empty.

Proof: The proof relies on the results of [10] and the analogy between the femto-cloud formation problem (10) and the modified core of the underlying coalition formation game; see Sec. 3.2 for details. It is shown in [10] that the best-reply process with experimentation implemented by Algorithm 1 converges almost surely to the modified core of the coalition formation game; see [10] for the detailed proof. Comparing the definition of modified core in [10] with (10) then completes the proof. □

4.2 Implementation Considerations

a) Decentralized Implementation: The proposed algorithm, independently followed by each FAP, provides a

decentralized solution to (10). This decentralized implementation relies on collaboration among the FCMs. It is assumed that FAPs monitor their users task request statistics over an interval comprising several time slots, and periodically transmit this information to their neighboring FCM. The FCMs then exchange this information with each other so as to be able to evaluate the femto-cloud characteristic function (1) and detect for blocked FAPs (Step 1 in Algorithm 1) in their neighborhood. Note that data size of user request information is negligible compared to the task data size. The FCMs are further responsible for providing FAPs that decided to revise their cooperation strategies with the feasible incentive (the term inside parentheses in (11)) in the associated femto-cloud, and to inform the blocked FAPs of their potential for obtaining higher incentives in other femto-clouds. Finally, having been enabled to communicate with each other, it is the task of FCMs to collaboratively update the network state parameter ω^n in Algorithm 1.

b) Time-scales: We assume that the femto-cloud structure remains constant for several time slots, and FAPs update their user request statistics with the same frequency. During this period, FAPs run Algorithm 1 based on the most recent sample statistics of the user requests. Once convergence to the solution takes place, the femto-cloud structure and associated incentives will be followed in the next decision epoch that the femto-cloud structure is being revised. Note that, since FAPs and FCMs are both static, the FAP-FAP and FAP-FCM channel responses vary slowly. In the utility function, average data transmission rate is considered over which femto-cloud structures are assumed to remain constant.

c) Characteristic Function Parameters: The parameters m_r, c_r, c_o, c_u, and m_p in (1) could be mathematically be interpreted as weight factors that determine the relative importance of the different factors considered in formulation of the characteristic function such as the delay cost, the demand uncertainty cost, and remote cloud processing cost. Clearly, the values of these parameters affect the optimal femto-cloud structure and incentive allocations. The particular choice of these values will depend on the specific application. For instance, in some applications users may be willing to incur longer delays to pay less for using the femto-cloud, in which case c_o should be smaller relative to m_r. In others, users may not tolerate delay, where c_o should be set very large. We further emphasize that the utility function formulated in Sec. 3.1 is only an example that exhibits how to incorporate different factors into the implementation of femto-clouds. Depending on the application specifics, certain terms could be added or omitted.

d) Empty Core: Finally, imposing conditions on the utility function to ensure existence of a solution (modified core of the underlying game) could be inherently complex in some applications. To address this issue, the experimentation factor ε in Algorithm 1 can be made to diminish to zero with time, e.g., one can replace ε with $\varepsilon_n = 1/n^\alpha$ for $0 < \alpha < 1$. This ensures that Algorithm 1 converges to the absorbing states of the best-reply process (Steps 2.1-2.3 in Algorithm 1) if the core is empty. Extensive simulations in Sec. 5 numerically verify that the results still outperform alternative schemes.

5 Numerical Results

This section provides numerical examples to evaluate the performance of the proposed incentive-based femto-cloud formation scheme.

5.1 Object Recognition Tasks

We focus on the processing associated with the object recognition task from images and videos captured via vision-based sensors in mobile devices, which is required to support mobile augmented reality applications. The formulation, however, is general enough to be adapted to various computationally intensive applications such as face recognition, pattern recognition, and optical character recognition from images/videos[5]. In particular, applications with different computational requirements can be split into several equal-sized computational sub-tasks. The utility function only requires how many sub-tasks can be executed in the femto-cloud and the predicted demand of sub-tasks in the coalition.

Feature extraction is typically the most computationally intensive task in object recognition at the deployment stage [42]. We assume that FAPs are equipped with graphics processing units (GPUs), and are capable of performing parallel computations in their GPUs. Therefore, the feature extraction procedure can be performed either on the UE's local processor, or on the FAPs. When both UE and FAPs are busy or lack sufficient computation capacity, the task is outsourced to the remote cloud. Once extracted, the feature vectors are sent to the application server, which compares them with the training models, and sends the best matched result(s) to the UE. In the examples to follow, we consider feature extraction tasks on both images and videos. At each time, each UE can either offload an image or a video to the FAP for the feature extraction task. In the numerical examples, gPb[6] is used for feature extraction. We assume that the duration of a video is uniformly distributed between 1 to 10 seconds.

Here, one unit of workload/demand associated with feature extraction is considered to be 144 Giga floating point operations per second (GFLOPs), which is also used to define one unit of computational capacity[7]. We assume that a 3264×2448 pixels image is divided into 9 sub-images [44], [45] with each sub-image containing 1088×816 pixels and occupying 2.1 mega bit (Mb) memory. Similarly, videos are divided up into 1 second segments. Each 1 second video of 640×480 pixels and 30 frame rate occupies 2.2 Mb memory. In both cases, the feature extraction task requires approximately 144 GFLOPs which is equivalent to one unit of workload or computational capacity.

5. Different object recognition applications may require different feature descriptors. The choice of the descriptor, however, is not crucial to the problem formulation and the proposed femto-cloud formation mechanism; it only affects the parameters of the utility function defined in Sec. 3.1.

6. The global probability algorithm (gPb) is a contour detection algorithm that achieves the best performance among all such schemes [43]. The computational requirement of gPb is 158,600 FLOPS per pixel [42].

7. 72 cores, each with 1000 MHz clock speed, are grouped together and considered as one unit of computational capacity.

5.2 Simulation Setup

Throughout this section, the NS-3 simulator is used as a realistic simulation of the entire LTE system architecture. We consider a city environment and use the LTE module developed by the LENA project [46], [47] as follows: We use LENA's *RandomRoomPositionAllocator* function to randomly locate 15 FAPs inside a 10-story building made of concrete and comprising 20 apartments, as depicted in Fig. 2. There exist 2 FCMs in the building located close to FAP-2 and FAP-15, respectively. The FCMs are connected to the remote cloud via $1Gbps$ optical fiber link. LENA's *HybridBuildingsPropagationLossModel* and *3kmphTraceFadingLossModel* functions (i.e., slowly varying Nakagami-m fading model) are used for propagation loss and channel fading between UEs and FAPs, respectively. We further use the *Kun2600MhzPropagationLossModel* and the *NakagamiPropagationLossModel* functions as the propagation loss model and channel fading for FAP-FAP and FAP-FCM communication. The handover is handled via the LENA's *AddX2Interface* function. UEs are further randomly located inside the building and connected to FAP using the *AttachToClosestEnb* function. At each time slot, sub-channels are allocated to users in each FAP according to the proportional fair (PF) scheduling policy with hybrid automatic repeat request (HARQ) re-transmission mechanism. Further, the UEs and FAPs are equipped with multiple input multiple output (MIMO) antennas, and support adaptive modulation and coding. UEs transmit UDP packets to the FAP. FAPs also transmit UDP packets for multi-cast communication. The data transfer rates are calculated from the *RLCTrace* files generated by the NS-3 simulator. Other NS-3 simulation parameters are listed in Table 2. Finally, the UE is considered to be an iPhone 5S and can perform 76.6 Giga floating point operations per second.

5.3 Numerical Examples

With the above simulation set-up, in the following examples, the effect of a single parameter is studied on the formation of femto-clouds while other parameters are kept constant. We set $\varepsilon = 0.3$, $\rho = 0.2$, $\Delta = 1$, and $\alpha = 0.5$ in Algorithm 1. Table 4 summarizes the parameters of all FAPs. These parameters are chosen so as to enable illustrating different scenarios. Each point on the graphs of Figs. 3-6 are averaged over 1000 i.i.d. realizations. The results are compared with two alternative heuristic schemes for femto-cloud formation. Scheme-1 is based on the relative distance of the FAPs. That is, κ FAPs with the least relative distances form a local femto-cloud. Scheme-2 relies on the computational capacity, the sample mean and sample entropy of demand at the FAPs. That is, FAPs are ranked based on the value of $d_k^{\max} - \overline{d}_k - \overline{H}_k$. Then, $\overline{\kappa}$ FAPs with the highest ranks are collected to form a local femto-cloud with $\underline{\kappa}$ lowest ranked FAPs. The procedure continues until all FAPs form/join a coalition. Coalition structures in heuristic schemes are listed in Table 3.

5.3.1 Example 1

The first example studies the effect of data transfer delay in the formation of femto-clouds. This scenario represents an enterprise environment where all FAPs are owned by the same authority. Therefore, we set $m_r = c_r = c_u = m_p = 0$, and $c_o = 1$ \$/sec. The goal will thus be to reduce the overall handling delay by forming local femto-clouds. Fig. 3 shows the average data transfer delay in the femto-clouds versus the computational capacity of FAP-1. The 'Isolated FAPs' case refers to the scenario where no FAP is willing to cooper-

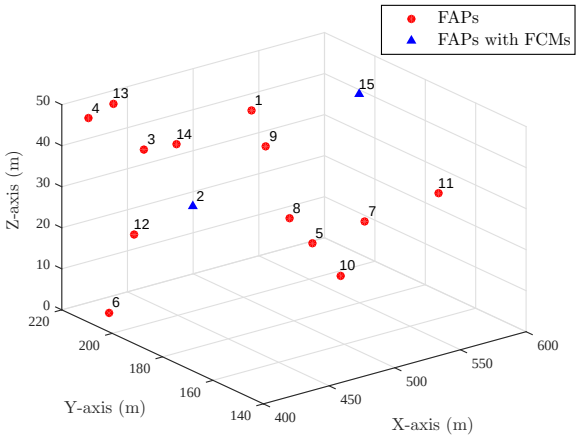

Fig. 2. FAPs and FCMs locations inside the building. UE arrival at each FAP follows a Poisson distribution. The number of UEs in the simulation depends on the user arrival rate at each FAP (see Table 4).

TABLE 2
Simulation setup: LTE system parameters in NS-3

Parameters	Value/Type
Adaptive Modulation & Coding	*PiroEW2010*
Bit Error Rate	0.0005
MIMO	2×2
FAP Antenna	*IsotropicAntennaModel*
External Wall Loss	10 dB
Shadowing Loss	5 dB
EPS Bearer	*GBR_CONV_VIDEO*
FAP Transmission Power	20 dbm
FAP Noise Figure	5 dbm
UE Transmission Power	10 dbm
UE Noise Figure	5 dbm
Macrocell Bandwidth	20 MHz
Mobility Model	*ConstantPosition*
Scheduler	*PfFfMacScheduler*

TABLE 3
Femto-cloud coalition structures in heuristic schemes

	Femto-Clouds Coalition Structure
Scheme-1	{1,2,5,8,9},{3,4,12,13,14},{6,7,10,11,15}
Scheme-2 (FAP-1 comp. capacity 0–4)	{1,2,6,10,15},{3,7,9,11,12},{4,5,8,13,14}
Scheme-2 (FAP-1 comp. capacity 6–8)	{2,6,7,10,15},{1,3,9,11,12},{4,5,8,13,14}
Scheme-2 (FAP-1 comp. capacity 10–14)	{1,2,7,10,15},{3,9,11,12,14},{4,5,6,8,13}
Scheme-2 (FAP-1 comp. capacity 16–20)	{2,7,10,12,15},{3,4,9,11,14},{1,5,6,8,13}
Scheme-2 (FAP-1 arrival rate 1)	{1,7,10,12,15},{2,3,4,9,14},{5,6,8,11,13}
Scheme-2 (FAP-1 arrival rate 2)	{1,2,7,10,15},{3,9,11,12,14},{4,5,6,8,13}
Scheme-2 (FAP-1 arrival rate 3–5)	{1,2,6,10,15},{3,7,9,11,12},{4,5,8,13,14}

TABLE 4
Simulation setup: FAP parameters in the numerical example

FAP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Trust Value	0.1	0.5	0.5	0.4	0.1	0	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0	0.5
Comp. Capacity	10	10	30	10	10	5	5	20	20	15	15	5	10	10	30
User Arrival Rate	2	1	2	2	1	2	3	2	3	1	2	2	1	3	1
Mean Process. Requests	20.47	13.13	17.63	15.7	13.51	16.46	20	11.4	17.77	8.13	17.53	11.67	10.64	21.63	13.16
Entropy	3.55	2.96	3.38	3.23	3.02	3.23	3.29	2.97	3.28	2.75	3.29	2.97	2.8	3.38	2.99

ate and operates individually—that is, there exist no femto-cloud. In contrast, the 'Grand femto-cloud' refers to the case where all FAPs form one large collaborative femto-cloud. In the 'Isolated FAPs' case, as the computational capacity increases, FAP-1 can perform more tasks locally and offloads fewer tasks to the remote cloud. This leads to the reduction of WAN latency. Therefore, the data transfer delay of FAP-1 decreases and, hence, the overall data transfer delay in the femto-clouds decreases.

As can be seen in Fig. 3, the data transfer delay in the femto-cloud structures prescribed by Algorithm 1 is the lowest. This is in contrast to the grand femto-cloud which provides the highest delay. This is mainly because some FAPs are located far away in the building; hence, the multicast delay in the grand femto-cloud is high. The data transfer delay in alternative scheme-1 is higher than alternative scheme-2. This is due to the fact that some FAPs have more requests than their computational capacity, in which case tasks are transported to the remote cloud and, hence, the WAN latency increases. The 'Isolated FAPs' case ignores cooperation among FAPs, which naturally results in higher delay.

The femto-cloud structures are listed in Table 5 for various values of computational capacity for FAP-1. FAP-1 forms a femto-cloud with FAP-8 and FAP-15 when its individual computational capacity is low. In this case, FAP-1 offloads a portion of the requested tasks to the femto-cloud and reduces WAN latency as compared to transporting tasks to the remote cloud. However, as the computational capacity of FAP-1 goes beyond its demand, it joins in a different femto-cloud so as to be able to process tasks exceeding the capacity of the femto-cloud members. This reduces the overall handling delay in the femto-cloud and improves users' QoE.

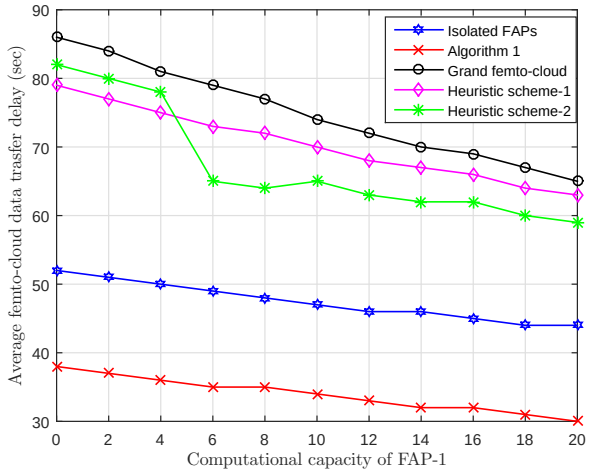

Fig. 3. Computational capacity of FAP-1 vs. average data transfer delay in the femto-clouds ($c_0 = 1$).

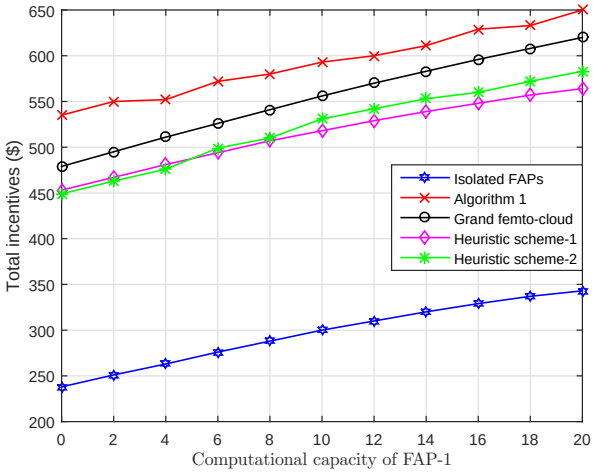

Fig. 4. Computational capacity of FAP-1 vs. femto-cloud incentive.

5.3.2 *Example 2*

This example considers a scenario where FAPs are deployed by residential users. To motivate owners for sharing excess resources, monetary incentives are considered as described in Sec. 3.2. Therefore, FAPs are motivated to cooperate by forming femto-clouds not only to reduce the handling delay, but also to earn incentive. We assume $m_r = 4$ \$/task, $c_r = 5$ \$/task, $c_o = 3$ \$/sec, $c_u = 2$ \$/task, $m_p = 1$, and $f = 200$ in the characteristic function (1).

Figure 4 plots the total incentive earned by all FAPs versus computational capacity of FAP-1. As the capacity of FAP-1 increases, it can serve more tasks exceeding other FAPs' capacities within the femto-cloud; hence, it receives higher incentives, which in turn increases the total incentive. Note that, for lower computational capacity, the incentive obtained by FAP-1 is still higher than the 'isolated FAPs' case. This is because incentives depend not only on the revenue but also on costs associated with delay costs. By forming a femto-cloud, FAP-1 can save on its delay costs as explained in Example 1 and, thus, obtains higher incentives.

Figure 5 also displays the total incentive obtained by all FAPs versus the user arrival rate at FAP-1. As expected, as the user arrival rate at FAP-1 increases, the tasks requested at FAP-1 will increase and the incentives it receives will decrease. This is mainly because FAP-1 (in the isolated case) as well as other FAPs in the femto-cloud need to transport

TABLE 5
Femto-clouds coalition structures in Example 1

FAP-1 Computational Capacity	Femto-Clouds Coalition Structure
0	{1,8,15}, {2}, {3,7}, {4}, {5,10}, {6}, {9}, {11}, {12}, {13}, {14}
2–14	{1,6,8,15}, {2}, {3,4}, {5,10}, {7}, {9}, {11}, {12}, {13}, {14}
16–20	{1,3,4,6,8,9}, {5,10}, {11,12,15}, {2}, {7}, {13}, {14}

TABLE 6
Femto-clouds coalition structures in Example 2

FAP-1 Computational Capacity	Femto-Clouds Coalition Structure
0-10	{1,2,3,4,6,7,8,9}, {11,12,13,14,15}, {5,10}
12-20	{1,6,8,11,12,13,14,15}, {2,3,4,5,7,9,10}
FAP-1 User Arrival Rate	Femto-Clouds Coalition Structure
1-5	{1,2,3,4,6,7,8,9}, {11,12,13,14,15}, {5,10}

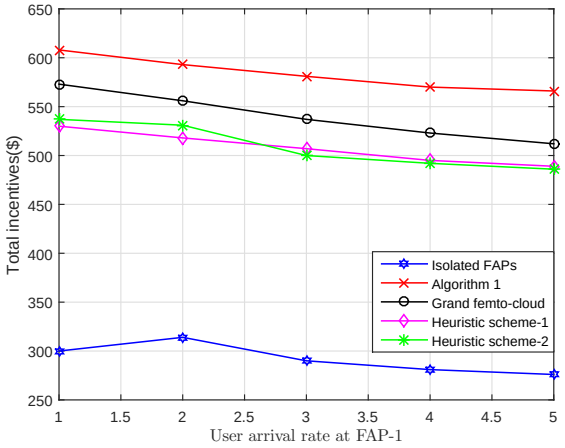

Fig. 5. User arrival rate at FAP-1 vs. femto-cloud incentive.

more tasks to the remote cloud, which increases the delay costs and remote cloud charges and, hence, reduces the incentives offered to FAPs. Note that this example considers the case where the charge per computation in the remote cloud is higher than the revenue obtained per computation in femto-cloud, i.e., $m_r \leq c_r$ in (1). Therefore, for fixed computational capacity, FAP-1's incentives decreases as the user arrival rate increases. The femto-cloud structures are listed in Table 6.

Fig. 6 shows the delay-incentive trade-off for a range of computational capacity of FAP-1. As expected, the femto-cloud data transfer delay for the femto-cloud structures in Example 2 is higher than those obtained in Example 1. This is due to the fact that the main goal of femto-cloud formation in Example 2 is to maximize the incentives where delay cost c_0 is lower than the computational revenue m_r and remote cloud processing cost c_r, whereas the aim of femto-cloud formation in Example 1 was to reduce the data transfer delay.

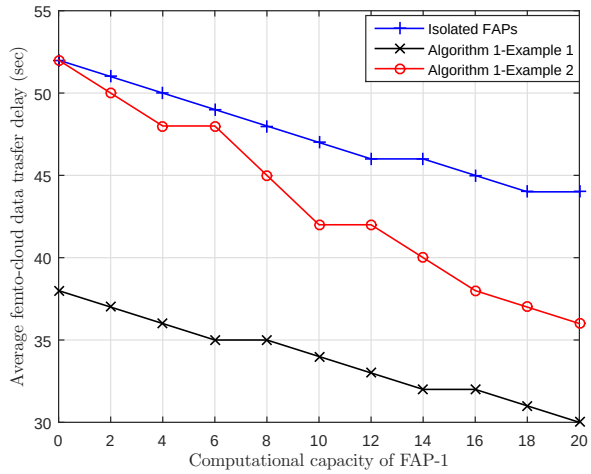

Fig. 6. Computational capacity of FAP-1 vs. average data transfer delay in the femto-clouds.

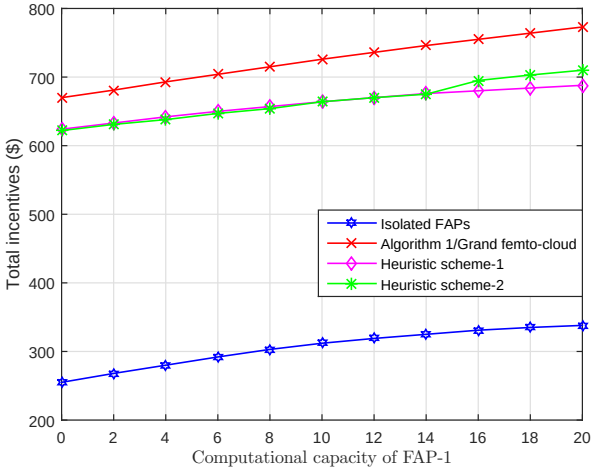

Fig. 7. Computational capacity of FAP-1 vs. femto-cloud incentive.

5.3.3 *Example 3*

In this example, we consider a hotspot scenario where all FAPs are located closely such that the multicast offloading delay among FAPs is negligible. More precisely, in such a case, the uplink data transmission rate from the k-th FAP to the l-th FAP, denoted by $b_{k,l}$ in (6), is much greater than N_B. This results in the $U^c_{o,m}(\mathcal{C})$ term in (3) being negligible compared to other terms.

Figure 7 shows the total incentives earned by all FAPs versus computational capacity of FAP-1. Here, the grand femto-cloud is the optimal coalition structure and provides the highest incentives to the FAP owners compared to other heuristic schemes.

6 Conclusion

To reduce the handling latency and costs associated with offloading computationally intensive tasks to remote clouds, the local computational capacity of femtocell access points (FAPs) should be maximally exploited. To this end, this paper proposed formation of femto-clouds comprising of

several FAPs wherein their excess computational resources are shared. In exchange for sharing their excess resources, FAP owners receive monetary incentives. We formulated the resource sharing problem as an optimization problem with the objective to maximize the overall utilities of all femto-clouds subject to the fair division of incentives among individual FAPs within a femto-cloud. We then presented a distributed femto-cloud formation algorithm that enabled FAPs to reach the optimal solution in a distributed fashion. We further commented on the similarities between the solution of the formulated problem and the modified core of a coalition formation game. Finally, simulation experiments using the LTE protocol stack in NS-3 showed superior performance of the proposed scheme in terms of both handling latency and incentives provided to FAP owners. They confirmed the interesting observation that a femto-cloud comprised of all FAPs is not always optimal—in many cases, multiple disjoint femto-clouds resulted in reduced latency and higher incentives to the FAP owners. The numerical examples further verified the applicability of Algorithm 1 in a wide range of scenarios, e.g., hotspot area, residential, and enterprise femtocell environments.

REFERENCES

[1] S. Agarwal, M. Philipose, and P. Bahl, "Vision: The case for cellular small cells for cloudlets," in *Proc. of the 5th Intl. Workshop on Mobile Cloud Computing & Services*, Bretton Woods, NH, 2014, pp. 1–5.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, 2009.

[3] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Proc. of the 8th Intl. Conf. on Mobile Systems, Applications, and Services*, San Francisco, CA, 2010, pp. 49–62.

[4] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[5] P. Bahl, R. Y. Han, E. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proc. of the 3rd ACM Workshop on Mobile Cloud Computing & Services*, Ambleside, UK, 2012, pp. 21–28.

[6] S. Barbarossa, P. Di Lorenzo, and S. Sardellitti, "Computation offloading strategies based on energy minimization under computational rate constraints," in *Proc. of the 23rd Europ. Conf. on Networks and Communications*, Bologna, Italy, 2014, pp. 1–5.

[7] F. L. Vilela, A. J. Ferrer, M. A. Puente, Z. Becvar, M. Rohlik, T. Vanek, P. Mach, O. M. Medina, J. V. Manzano, H. Hariyanto *et al.*, "TROPIC-D22 design of network architecture for femto-cloud computing," 2013.

[8] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing," in *Proc. of IEEE INFOCOM*, Turin, Italy, 2013, pp. 1285–1293.

[9] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. of the IEEE 14th Workshop on Signal Processing Advances in Wireless Communications*, Darmstadt, Germany, 2013, pp. 26–30.

[10] T. Arnold and U. Schwalbe, "Dynamic coalition formation and the core," *J. Econ. Behav. Organ.*, vol. 49, no. 3, pp. 363–380, 2002.

[11] S. M. S. Tanzil, O. N. Gharehshiran, and V. Krishnamurthy, "Femto-cloud formation: A coalitional game-theoretic approach," in *Proc. of IEEE GLOBECOM*, San Diego, CA, 2015, pp. 1–6.

[12] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, 2013.

[13] M. Guazzone, C. Anglano, and M. Sereno, "A game-theoretic approach to coalition formation in green cloud federations," in *Proc. of the 14th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing*, Chicago, IL, 2014, pp. 618–625.

[14] C. A. Lee, "Cloud federation management and beyond: Requirements, relevant standards, and gaps," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 42–49, 2016.

[15] T. Truong-Huu and C.-K. Tham, "A novel model for competition and cooperation among cloud providers," *IEEE Trans. on Cloud Comput.*, vol. 2, no. 3, pp. 251–265, 2014.

[16] L. Mashayekhy, M. Movahed Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Trans. on Cloud Comput.*, vol. 3, no. 1, pp. 14–27, 2015.

[17] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Spectrum leasing as an incentive towards uplink macrocell and femtocell cooperation," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 617–630, 2012.

[18] O. N. Gharehshiran, A. Attar, and V. Krishnamurthy, "Collaborative sub-channel allocation in cognitive LTE femto-cells: a cooperative game-theoretic approach," *IEEE Trans. Commun.*, vol. 61, no. 1, pp. 325–334, 2013.

[19] Z. Zhang, L. Song, Z. Han, and W. Saad, "Coalitional games with overlapping coalitions for interference management in small cell networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2659–2669, 2014.

[20] F. Pantisano, M. Bennis, W. Saad, M. Debbah, and M. Latva-aho, "Interference alignment for cooperative femtocell networks: a game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2233–2246, 2013.

[21] R. Langar, S. Secci, R. Boutaba, and G. Pujolle, "An operations research game approach for resource and power allocation in cooperative femtocell networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 675–687, 2015.

[22] S.-Y. Yun, Y. Yi, D.-H. Cho, and J. Mo, "The economic effects of sharing femtocells," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 595–606, 2012.

[23] L. Gao, G. Iosifidis, J. Huang, and L. Tassiulas, "Economics of mobile data offloading," in *Proc. of IEEE INFOCOM Workshops*, Turin, Italy, 2013, pp. 351–356.

[24] Y. Chen, J. Zhang, and Q. Zhang, "Utility-aware refunding framework for hybrid access femtocell network," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1688–1697, 2012.

[25] S. Hua, X. Zhuo, and S. S. Panwar, "A truthful auction based incentive framework for femtocell access," in *Proc. of IEEE WCNC*, Shanghai, China, 2013, pp. 2271–2276.

[26] L. Duan, J. Huang, and B. Shou, "Economics of femtocell service provision," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2261–2273, 2013.

[27] N. Shetty, S. Parekh, and J. Walrand, "Economics of femtocells," in *Proc. of IEEE GLOBECOM*, Honolulu, HI, 2009, pp. 1–6.

[28] Y. Zhang and M. van der Schaar, "Peer-to-peer multimedia sharing based on social norms," *Signal Processing: Image Communication*, vol. 27, no. 5, pp. 383–400, 2012.

[29] M. Jakobsson, J.-P. Hubaux, and L. Buttyán, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, R. N. Wright, Ed., 2003, vol. 2742, pp. 15–33.

[30] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, "Selfish caching in distributed systems: a game-theoretic analysis," in *Proc. of the 23rd Annual ACM Symp. on Principles of Distributed Computing*, Elche, Spain, 2004, pp. 21–30.

[31] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact e-cash," in *Advances in Cryptology*, ser. Lecture Notes in Computer Science, R. Cramer, Ed., 2005, vol. 3494, pp. 302–321.

[32] L. Buttyán and J.-P. Hubaux, "Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks," Tech. Rep., 2001.

[33] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti, "Joint optimization of radio resources and code partitioning in mobile cloud computing," *arXiv preprint arXiv:1307.3835*, 2013.

[34] O. Munoz-Medina, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2014.

[35] C. Chen, "C-ran: the road towards green radio access network," *White paper*, 2011.

[36] J. Oueis, E. C. Strinati, and S. Barbarossa, "Small cell clustering for efficient distributed cloud computing," in *Proc. of IEEE PIMRC*, Washington, DC, 2014, pp. 1474–1479.
[37] ——, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. of IEEE VTC*, vol. Spring, Glasgow, Scotland, 2015, pp. 1–6.
[38] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood, "The value of reputation on ebay: A controlled experiment," *Experimental Economics*, vol. 9, no. 2, pp. 79–101, 2006.
[39] J. P. Kahan and A. Rapoport, *Theories of Coalition Formation*. New York, NY: Psychology Press, 2014.
[40] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Başar, "Coalitional game theory for communication networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 77–97, 2009.
[41] G. Owen, *Game Theory*. New York, NY: Academic Press, 1995.
[42] B.-Y. Su, "Parallel application library for object recognition," Ph.D. dissertation, EECS Department, University of California, Berkeley, 2012.
[43] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proc. of IEEE CVPR*, Anchorage, AK, 2008, pp. 1–8.
[44] J. Li and J. Z. Wang, "Studying digital imagery of ancient paintings by mixtures of stochastic models," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 340–353, 2004.
[45] K. Tan and S. Chen, "Adaptively weighted sub-pattern PCA for face recognition," *Neurocomputing*, vol. 64, pp. 505–511, 2005.
[46] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented LTE network simulator based on ns-3," in *Proc. of ACM MSWiM*, Miami Beach, FL, 2011, pp. 293–298.
[47] N. Baldo, M. Requena-Esteso, M. Miozzo, and R. Kwan, "An open source model for the simulation of LTE handover scenarios and algorithms in ns-3," in *Proc. of ACM MSWiM*, Barcelona, Spain, 2013, pp. 289–298.

S. M. Shahrear Tanzil received the B.Sc. degree in electrical and electronics engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh, in 2011 and the M.A.Sc. degree from the University of British Columbia (UBC), Canada, in 2013. He is currently working towards the Ph.D. degree at UBC and is a member of the Statistical Signal Processing Laboratory. His research interests include resource allocation in wireless networks, mobile cloud computing, and game theory.

Omid Namvar Gharehshiran received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 2015, where he was a Member of the Statistical Signal Processing Group. He currently holds the NSERC Postdoctoral Fellowship at the Actuarial Science and Mathematical Finance Group, Department of Statistical Sciences, University of Toronto, Toronto, ON, Canada. His research interests include stochastic optimization and control, games, and learning theory.

Vikram Krishnamurthy (S'90–M'91–SM'99–F'05) received the bachelors degree from the University of Auckland, Auckland, New Zealand, and the Ph.D. degree from the Australian National University, Canberra, A.C.T., Australia, in 1988 and 1992, respectively. He is currently a Professor and holds the Canada Research Chair with the Department of Electrical Engineering, University of British Columbia, Vancouver, BC, Canada. His research interests include statistical signal processing, computational game theory, and stochastic control in social networks. He authored the book *Partially Observed Markov Decision Processes – Filtering to Controlled Sensing* published by Cambridge University Press in 2016. He served as Distinguished Lecturer for the IEEE Signal Processing Society and Editor-in-Chief of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He was the recipient of an Honorary Doctorate from KTH (Royal Institute of Technology), Sweden, in 2013.