Profit-Maximizing Stochastic Control for Mobile Crowd Sensing Platforms

Yang Han, Yanmin Zhu
Shanghai Jiao Tong University
hyhx_2008@sjtu.edu.cn, yzhu@cs.sjtu.edu.cn

Abstract—**In this paper, we consider the crucial problem of maximizing the profit of a crowd sensing platform which receives sensing requests from various subscribers and completes the requests by leveraging sensing time of participating smartphones. The profit of the platform equals the total charges of sensing request minus the payments to smartphones. It is of great challenge to the maximal profit for the platform, because of stochastic arrivals of sensing requests, dynamic participation of smartphones, and high complexity in allocating requests to smartphones. In response to the challenges, we propose an optimal online control framework which can efficiently utilize the limited sensing time on each smartphone. Based on the stochastic Lyapunov optimization techniques combined with the idea of weight perturbation, our control framework makes online control decisions including sensing requests admission and dispatching control, sensing time purchasing control and sensing time allocation control, without requiring any knowledge of the future patterns. Rigorous mathematical analysis and comprehensive simulation results show that our control framework can achieve a time averaged profit that is arbitrarily close to the optimum, while still maintaining strong system stability.**

I. Introduction

In recent years, mobile smartphones have become an indispensable part of people's lives. Most of smartphones embed a rich set of built-in sensors, such as accelerometer, gyroscope, microphone, GPS, and camera [1]. As a consequence, it is unprecedentedly easier for one to collect sensing information around surroundings and share such sensing information. As a new compelling paradigm for large-scale sensing information collection, *mobile crowd sensing* [2] collects sensing data from a multitude of individual smartphones, underpinning a variety of valuable knowledge discovery, environment monitoring and decision making applications. So far, there have been a number of crowd sensing deployments, such as noise mapping [3], personal environmental impact analysis [4], and road traffic monitoring [5].

There are two types of mobile crowd sensing, depending on the way of smartphone user participation, i.e., *participatory sensing* and *opportunistic sensing* [1][6]. Participatory sensing requires participants to actively engage in sensing activities by manually determining how, when, what and where to sense. In opportunistic sensing, however, sensing activities are typically automated, without requiring user intervention to actively and consciously perform sensing tasks. In practice, opportunistic sensing applications may run in the background and the phone users may not be aware of active execution of sensing applications. In other words, opportunistic applications are usually transparent to phone users. The benefit of opportunistic sensing is that it significantly lowers the burden of phone users, allowing higher participation, which is crucial for wide adoption of mobile crowd sensing.

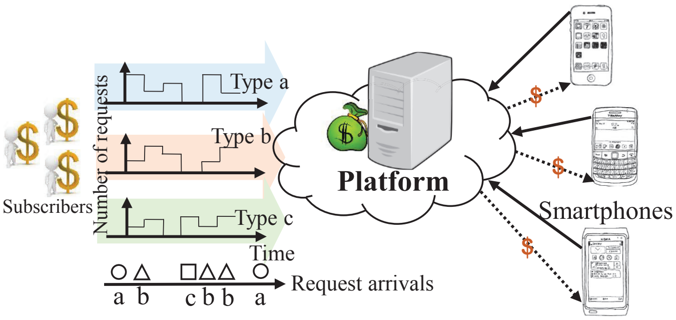

Fig. 1. An illustration of a crowd sensing system. The platform serves multiple types of sensing requests which can arrive dynamically.

This paper concentrates on opportunistic sensing based mobile crowd sensing. In most cases, mobile crowd sensing applications are built for specific purpose and requirements. The sensing data generated by the application are usually only available for single use. To realise the greater vision of a mobile crowdsensing application, common platforms that facilitates easy development and deployment of crowd sensing applications are developed, e.g., MOSDEN [7], and MECA [8]. Fig. 1 describes a typical crowd sensing system consisting of a platform deployed in the cloud. Smartphones with various embedded sensors are connected to the platform via wireless networks.

The platform can provide sensing services to its subscribers by processing their sensing data collection requests. The subscribers can be different sensing applications which use different types of sensing data. Sensing requests from subscribers arrive at the platform dynamically. The platform then dispatches sensing requests to smartphones for processing. At last it aggregates sensing data of smartphones and forwards the results to subscribers. In this way, the platform processes sensing requests from subscribers and charges them for that service. In other words, the platform can generate revenue (utility) by processing sensing requests. In this paper, we consider different types of heterogeneous sensing requests. They may differ from each other in required sensors and can generate different utility.

We can find that smpartphones play a significantly important role in such a crowd sensing system since that the service availability of the platform crucially depends on participation of smartphones. However, the users of smartphones, named participants, will suffer some cost during contribution to the platform, such as battery energy cost, data processing and transmission cost and their normal use of smartphones may be affected when processing sensing requests. So that participants would not allow the platform to utilize their devices without limitation for free.

Therefore we consider the platform pays the participants to motivate their participation in sensing requests processing, according to their *sensing time* which means the time consumption for processing sensing requests. To let the participants decide how much sensing time they are willing to provide, we consider that they can sell sensing time to the platform for requests processing. After purchasing some sensing time, the platform then can utilize participants for processing sensing requests. But it cannot utilize more than the available sensing time that has been purchased. Note that when and how much time to sell completely depends on the participants' willingness. For example, a participant may not be willing to sell much sensing time when he or she is busy on smartphone but may be willing to sell more sensing time when he or she doesn't use smartphone frequently. The amount of sensing time that participant is willing to sell represents how willing the participant wants to participate in the system and is totally dynamic and unpredictable.

The objective of the platform is to maximize the profit, which equals to the revenue minus payment, subject to the constraints that the average system backlog is finite (system stability), and the *sensing time-availability* constraint is met, i.e., the sensing time consumed on each smartphone is no more than the time that is available, at all time. To achieve the goal, four important control decisions need to be made in the crowd sensing system: (1) how many requests of each type to be admitted at any given time; (2) how to dispatch the admitted sensing requests to a number of smartphones; (3) how to allocate limited available sensing time to each type of sensing requests for processing, on each smartphone; and (4) how much sensing time to purchase when participants are willing to sell some.

We see that it is quite challenging to achieve the optimal profit subject to the constraints. The biggest challenge behind the problem is that the information of sensing requests' arrival is not a priori knowledge, which is unpredictable and even bursty. And each participant's willingness to participate is also unknown and unpredictable in advance. In other words, we do not know the amount of sensing time that participants are willing to sell and when they are willing to sell. Thus, conventional deterministic control methods are not applicable here. Second, the *sensing time-availability* constraint greatly complicates the design of an efficient control algorithm, due to the fact that the current sensing time expenditure decision may lead to sensing time outage in the future and thereby affect the future decisions. Third, there exists an intrinsic tradeoff among system stability, sensing time purchasing and the processing throughput which impacts the profit. What's more, all the control decisions are deeply coupled and affect with each other.

Mobile crowd sensing has received increasingly extensive research study. Unfortunately, little work has studied the problem of maximizing the profit of the crowd sensing platforms subject to the constraints of system stability and sensing time availability. In particular, little work of the crowd sensing platform has noticed the fact of dynamic request arrival and hasn't considered the problem of user motivation. As a consequence, most of the existing studies [7] [8] simply assume the sensing requests are known and within the system's capacity, which is not practical. And they utilize the smartphone for sensing data collection without limitation and payment.

In this paper, we propose a profit maximizing algorithm for crowd sensing platforms. To tackle the aforementioned challenges, we take advantage of Lyapunov optimization technique developed in [9] to design a novel *optimal online control framework* which can independently and simultaneously make all the four control decisions in the crowd sensing system. We utilize the basic Lyapunov optimization technique together with the idea of *weight perturbation*, e.g., [10] and [11]. The idea is to construct the algorithm based on a quadratic Lyapunov function, but carefully perturb the weights used for decision making, so as to *push* the available sensing time volume towards certain nonzero values on each smartphone to avoid underflow. Based on this idea, our control framework can achieve a time-averaged profit which is arbitrarily close to the optimum, while still remain strong stability. Specifically, our control framework is able to achieve a time-averaged profit that is within $O(1/V)$ of the optimum for any $V > 0$, and guarantees that the system backlog is deterministically bounded by $O(V)$.

The remainder of this paper is organized as follows. In Section II, we formulate our system model as well as the objective. In Section III, we present the details of our optimal control framework. We evaluate the performance of our control framework based on simulation results in Section IV. And we discuss related work in Section V. Finally, a brief conclusion of this work is given in Section VI.

II. System Model

In this section, we formulate a typical crowd sensing system model in Fig. 2. It consists of a platform deployed in the cloud and a set of smartphones denoted by $\mathcal{N} = \{1, 2, \cdots, n\}$. The system is to serve m types of heterogenous sensing requests $\mathcal{M} = \{1, 2, \cdots, m\}$ with diverse arrival rates. These sensing requests may differ in required sensors or other smartphone resources (e.g. processor, memory) and can generate different utility. We consider a system that operates in slotted time where the time slot length can range from a few seconds to minutes. In every time slot $t(= 0, 1, 2, \cdots, \tau, \cdots)$, a number of sensing requests of different types may arrive at the platform. And we use $R_i(t)$ to denote the number of the i-th type of sensing requests that arrive at the platform in time slot t.

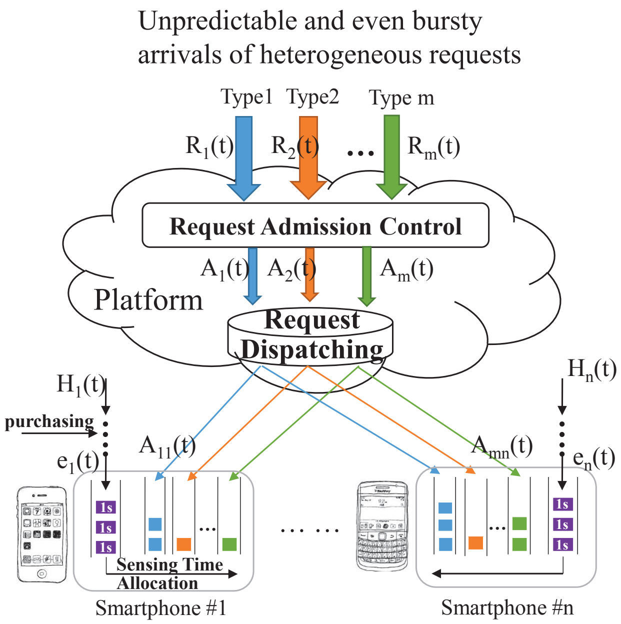

Fig. 2. A crowd sensing system with four control decisions including request admission control, request dispatching control, sensing time purchasing control, and sensing time allocation control.

We assume that the number of arrival requests $R_i(t)$, $\forall i \in \mathcal{M}$, is independent and identically distributed (i.i.d) over time slots, and is independent of the current amount of unfinished workload in the system. And we denote the time averaged arrival rate of requests of the i-th type as $\overline{r_i} = \mathbb{E}\{R_i(t)\}$. We also assume that $0 \leq R_i(t) \leq R_i^{\max}$, $\forall i \in \mathcal{M}$, with some finite $R_i^{\max}$ in all time slots. Since the arrival of sensing requests is highly dynamic and usually unpredictable, and can even spike abruptly, we do not assume any priori knowledge of the statistics of $R_i(t), \forall i \in \mathcal{M}, \forall t$.

A. Control Decisions

In every time slot, the system needs to make four important control decisions, as introduced in Fig. 2.

1) Admission control of sensing requests: The first control decision of the system is to decide how many sensing requests of each type to admit into the system. In each time slot, heterogenous sensing requests arrive at the platform. To prevent system overload, only a subset of each type of requests $A_i(t), \forall i \in \mathcal{M}$, can be admitted into the system to be processed. Therefore, admission control decisions $A_i(t)$ are made in every time slot, subject to the constraint $0 \leq A_i(t) \leq R_i(t)$.

2) Dispatching control of sensing requests: As soon as a subset of sensing requests of each type $A_i(t), \forall i \in \mathcal{M}$, are admitted into the platform, the second control decision is to dispatch (allocate) $A_i(t)$ to each smartphone for processing. We use $A_{ij}(t)$ to denote the number of the requests of the i-th type that are dispatched to smartphone j in time slot t. Obviously, the dispatching control decisions need to satisfy the constraint: $\sum_{j\in\mathcal{N}} A_{ij}(t) = A_i(t), \forall i \in \mathcal{M}, \forall t$.

Since it is impossible for smartphones to finish processing all dispatched (allocated) requests at once, each smartphone maintains a queue for each type of sensing requests where that type of requests will wait to be processed. We define the *request queue backlog* $Q_{ij}(t)$ as the number of requests that are waiting in the queue of type i on smartphone j at the beginning of time slot t, as Fig. 2 shows.

3) Purchasing control of sensing time: To let sensing requests be processed on each participant's smartphone, the platform needs to purchase the sensing time of each participant at first. We suppose that sensing time is measured in second. As explained in Section I, according to the participants' willingness and practical situation, the amount of sensing time that a participant is willing to sell is typically not fixed and varies over time. The participant can sell any amount of sensing time in any time slot. For convenient illustration, we name the amount of sensing time that participant is willing to sell as *purchasable sensing time*. We use $H_j(t)$ to denote the amount of purchasable sensing time on smartphone j in time slot t. In other words, $H_j(t)$ denotes how many seconds the participant is willing to sell to the platform for processing sensing requests. [1] We assume that $H_j(t), \forall j \in \mathcal{N}$ is i.i.d over time slots and there exists a finite H_{max} such that $0 \leq H_j(t) \leq H_{max}, \forall j \in \mathcal{N}, \forall t$. And we denote the time averaged purchasable sensing time on smartphone j as $\overline{h_j} = \mathbb{E}\{H_j(t)\}$.

The platform needs to decide whether or not to purchase sensing time on each smartphone, when there are purchasable sensing time in time slot t, i.e., $H_j(t) > 0$. We model the purchasing decision as $e_j(t) \in [0, H_j(t)]$, which denotes the amount of sensing time that are purchased on smartphone j in time slot t. We assume that the sensing time purchased on each smartphone can be accumulated for further use, and the sensing time purchased in time slot t is assumed to be available for use from the next time slot $t+1$. We model such an accumulation as a *sensing time queue*. And we use the sensing time queue size on smartphone j in time slot t, denoted by $S_j(t)$, to measure the amount of available sensing time that has been purchased on smartphone j in time slot t.

4) Sensing time allocation: In order to process each type of sensing requests waiting in the corresopnding queue maintained by each smartphone, the smartphone needs to allocate sensing time to each type of request in every time slot. Let $p_{ij}(t)$ denote the amount of sensing time allocated to the i-th type of sensing request on smartphone j in time slot t. It is clear that, in any time slot t, the sensing time allocation decisions must satisfy the following *sensing time-availability* constraint:

$$\sum_{i\in\mathcal{M}} p_{ij}(t) \leq S_j(t), \forall j \in \mathcal{N}. \tag{1}$$

That is, the consumed sensing time must be no more than what is available. Since processing too many sensing requests of the same type in one time slot may cause a high occupancy rate of certain smartphone resources, such as a certain sensor

[1] The value of $H_j(t)$ can be obtained by participant's declaration to the system. In each time slot t, if the participant declares that he or she is willing to sell some sensing time, $H_j(t) > 0$; If the participant doesn't declare, $H_j(t) = 0$.

and processor, so that result in participants' discomfort when using their smartphones, we limit the amount of sensing time allocated to each type of sensing requests as: $0 \leq p_{ij}(t) \leq p_{max}$, for some finite p_{max}.[2]

Given the sensing time allocation decision, the processing rate of the i-th type of sensing requests on smartphone j in time slot t is given by the *processing rate-sensing time* function:

$$u_{ij}(t) = u_{ij}(p_{ij}(t)) = \alpha_i \cdot p_{ij}(t), \tag{2}$$

where α_i means the number of processable sensing requests of i-th type by consuming one second sensing time. And we denote $\alpha_{max} = \max_i \alpha_i$.

B. Queueing Dynamics

Recall that $Q_{ij}(t)$ denotes the request queue backlog of type i on smartphone j. In each time slot t, the number of sensing requests dispatched to each queue is $A_{ij}(t)$. And the number of requests that can be processed is $u_{ij}(t)$. Therefore, we can obtain the following queueing dynamics over time for each queue on each smartphone as:

$$Q_{ij}(t+1) = max[Q_{ij}(t) - u_{ij}(t)), 0] + A_{ij}(t), \tag{3}$$

with $Q_{ij}(0) = 0, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}$. In this paper, we say that the system is *stable* if all queues are stable, that is the following is met:

$$\lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(Q_{ij}(\tau)) < \infty, \forall i, \forall j. \tag{4}$$

Similarly, $S_j(t)$ is the sensing time queue size on smartphone j in time slot t, which denotes the amount of available (purchased) sensing time for processing sensing requests on smartphone j. Under the sensing time-availability constraint (1), we find that on each smartphone j, the sensing time queue $S_j(t)$ evolves according to the following:

$$S_j(t+1) = S_j(t) - \sum_{i\in\mathcal{M}} p_{ij}(t) + e_j(t), \tag{5}$$

with $S_j(0) = 0, \forall j \in \mathcal{N}$.

C. Profit Model and Objective Formulation

1) Profit Model: The objective of our control framework is to maximize the profit of the platform in crowd sensing systems, which equals to the revenue minus cost. The revenue is the utility produced by processing sensing requests while the cost is the payment for participants in purchasing their sensing time.

The utility usually depends on the system throughput. Here, the sensing requests processing throughput of the crowd sensing system is the number of requests that can be admitted and processed. Specifically, for each type of sensing request $\forall i \in \mathcal{M}$, we define the *time averaged throughput* of type i as $\overline{a_i} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(A_i(\tau))$. Obviously, there is $\overline{a_i} \leq \overline{r_i}$, since the time averaged throughput $\overline{a_i}$ cannot exceed the time averaged arrival rate $\overline{r_i}$ for any type of sensing requests.

Intuitively, the larger throughput is achieved, the more utility can be obtained. So the utility brought by the i-th type of requests processing throughput can be given by a linear utility-throughput function:

$$U_i(A_i(t)) = \beta_i \cdot A_i(t), \tag{6}$$

where β_i is a type-specific positive coefficient for characterizing different utility produced by processing each type of sensing request. And we denote $\beta_{max} = \max_i \beta_i$.

The cost of the platform is the payment for each participant. We define the payment for purchasing sensing time of smartphone j as:

$$C_j(e_j(t)) = k \cdot e_j(t), \tag{7}$$

where k is the unit price for one second sensing time. And we denote the time averaged sensing time purchased on smartphone j as $\overline{e_j} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}(e_j(\tau))$.

2) Objective Formulation: The objective of our control framework is to make a series of control decisions in every time slot, including admitting the right amount of sensing requests $A_i(t)$, dispatching requests to each smartphone, purchasing appropriate sensing time on each smartphone, and allocating available sensing time to each type of sensing requests subject to the sensing time-availability constraint (1), so as to maximize the time averaged profit of the platform as follows:

$$\max \quad \sum_{i\in\mathcal{M}} U_i(\overline{a_i}) - \sum_{j\in\mathcal{N}} C_j(\overline{e_j}) \tag{8}$$

subject to the system stability constraint (4).

It is of great challenges to make optimal control decisions for maximizing the profit of the platform: *First*, the information of sensing requests' arrival and the purchasable sensing time on each smartphone is time-varying and unpredictable, which makes it infeasible to precisely calculate optimal solution in an offline manner. *Second*, the sensing time-availability constraint (1) couples the current sensing time allocation decision and the future decisions, for that a current decision may incur the sensing time queue to be empty and thereby block some sensing time allocation actions in the future. In response, our control framework offers an online and distributed control algorithm in Section III based on a modified Lyapunov function, which is able to efficiently make all four important decisions.

III. Constructing Online Control Framework

In this section, we present our profit optimal online control framework which is able to concurrently make all control decisions shown in Fig.2. The design of our control framework is based on Lyapunov optimization technique developed in [9]. The idea is to construct a Lyapunov control algorithm combined with *perturbed weights* for making control decisions.

[2] Note that, the maximum possible allocated sensing time in one time slot mp_{max}(seconds) can not exceeds the length of time slot.

By carefully perturbing the weights, it can be guaranteed that whenever a smartphone allocates sensing time to the sensing requests, there is always enough available time in the sensing time queue. We will show that our control algorithm can be proved to achieve a time averaged profit that is arbitrarily close to optimum, while still maintaining system stability.

A. Problem Transformation Based on Lyapunov Optimization

To start, we first choose a perturbation parameter $\theta_j, j \in \mathcal{N}$ (to be specified later) and then define a *perturbed Lyapunov function* as follows:

$$L(t) \triangleq \frac{1}{2} \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{M}} Q_{ij}^2(t) + \frac{1}{2} \sum_{j \in \mathcal{N}} [S_j(t) - \theta_j]^2. \tag{9}$$

This function represents not only a scalar measure of request queue congestion, but also the sensing time queue level on each smartphone. By keeping the Lyapunov function value small, we in fact *push* the request queue backlogs towards small values which suggests the system stability, and *push* the sensing time queue size $S_j(t)$ towards θ_j. Thus, by carefully choosing the value of θ_j, we can guarantee that the sensing time queue always has enough available sensing time whenever the smartphone decides to allocate sensing time to process sensing requests.

We use $\mathbf{Q}(t) = (Q_{ij}(t))$ and $\mathbf{S}(t) = (S_j(t))$ to denote the matrix of the request queues and sensing time queues on smartphones and denote $\mathbf{Z(t)} = (\mathbf{Q(t)}, \mathbf{S(t)})$. To keep system stable and the volume of available sensing time that has been purchased staying around a certain level by consistently pushing the Lyapunov function towards a small value, we introduce a *one-slot conditional Lyapunov drift*, which represents the change in Lyapunov function from one slot to the next, as follows:

$$\Delta(t) \triangleq \mathbb{E}\{L(t+1) - L(t) | \mathbf{Z}(t)\}. \tag{10}$$

The expectation here is taken over the randomness of sensing requests' arrival and participants' willingness to sell sensing time and the (possibly random) control decisions made in response to those information.

According to Lyapunov optimization, the underlying objective of our optimal control algorithm is to make control decisions to minimize an upper bound of the following *drift-minus-profit* expression in each time slot:

$$\Delta_V(t) \triangleq \Delta(t) - V\mathbb{E}\{\sum_{i \in \mathcal{M}} U_i(A_i(t)) - \sum_{j \in \mathcal{N}} C_j(e_j(t)) | \mathbf{Z}(t)\}. \tag{11}$$

Insight: The control parameter $V \geq 0$ represents an *importance weight* on how much we emphasize the profit maximization compared to system stability. We want to make $\Delta(t)$ small to push request queue backlogs towards lower congestion state, but we also want to make $\mathbb{E}\{\sum_{i \in \mathcal{M}} U_i(A_i(t)) - \sum_{j \in \mathcal{N}} C_j(e_j(t)) | \mathbf{Z}(t)\}$ big so that we can obtain a large profit. So this parameter V enables various tradeoffs between system stability and profit.

We have the following lemma regarding the *drift-minus-profit* expression:

***Lemma** 1*: In each time slot t, under any feasible control decisions, including requests admission, dispatching, sensing time purchasing and allocation subject to the sensing time-availability constraint (1), we have:

$$\begin{aligned} \Delta(t) - V\mathbb{E}\{\sum_{i \in \mathcal{M}} U_i(A_i(t)) - \sum_{j \in \mathcal{N}} C_j(e_j(t)) | \mathbf{Z}(t)\} \leq B \\ - \sum_{i \in \mathcal{M}} \mathbb{E}\{V\beta_i A_i(t) - \sum_{j \in \mathcal{N}} Q_{ij}(t) A_{ij}(t) | \mathbf{Z}(t)\} \quad (12) \\ + \sum_{j \in \mathcal{N}} \mathbb{E}\{Vke_j(t) + [S_j(t) - \theta_j] e_j(t) | \mathbf{Z}(t)\} \quad (13) \\ - \sum_{j \in \mathcal{N}} \mathbb{E}\{\sum_{i \in \mathcal{M}} Q_{ij}(t) \alpha_i p_{ij}(t) \\ + [S_j(t) - \theta_j] \sum_{i \in \mathcal{M}} p_{ij}(t) | \mathbf{Z}(t)\} \quad (14) \end{aligned}$$

where $B = \frac{n \sum_{i \in \mathcal{M}} [\alpha_i^2 p_{\max}^2 + (R_i^{\max})^2] + n(H_{\max} + mp_{\max})^2}{2}$ is a finite constant parameter.

Proof: First, squaring both sides of (3), and using the fact that for any $Q \geq 0, u \geq 0, A \geq 0, (max[Q - u, 0] + A)^2 \leq Q^2 + u^2 + A^2 - 2Q(u - A)$, and combining with Eq. (2), we have:

$$\begin{aligned} Q_{ij}^2(t+1) - Q_{ij}^2(t) &\leq \alpha_i^2 p_{ij}^2(t) + A_{ij}^2(t) - \\ &\quad 2Q_{ij}(t)[\alpha_i p_{ij}(t) - A_{ij}(t)] \\ &\leq \alpha_i^2 p_{max}^2 + (R_i^{max})^2 - \\ &\quad 2Q_{ij}(t)[\alpha_i p_{ij}(t) - A_{ij}(t)]. \end{aligned}$$

Then, according to Eq. (5), we have:

$$\begin{aligned} & [S_j(t+1) - \theta_j]^2 - [S_j(t) - \theta_j]^2 \\ = \; & [S_j(t+1) - S_j(t)][S_j(t+1) + S_j(t) - 2\theta_j] \\ = \; & [e_j(t) - \sum_{i \in \mathcal{M}} p_{ij}(t)]^2 - 2[S_j(t) - \theta_j][\sum_{i \in \mathcal{M}} p_{ij}(t) - e_j(t)] \\ \leq \; & (H_{\max} + mp_{\max})^2 - 2[S_j(t) - \theta_j][\sum_{i \in \mathcal{M}} p_{ij}(t) - e_j(t)]. \end{aligned}$$

Based on the above two inequations, we further have:

$$\begin{aligned} L(t+1) - L(t) \leq B - Q_{ij}(t)[\alpha_i p_{ij}(t) - A_{ij}(t)] \\ -[S_j(t) - \theta_j][\sum_{i \in \mathcal{M}} p_{ij}(t) - e_j(t)], \end{aligned} \tag{15}$$

by defining $B = \frac{n \sum_{i \in \mathcal{M}} [\alpha_i^2 p_{\max}^2 + (R_i^{\max})^2] + n(H_{\max} + mp_{\max})^2}{2}$.

Taking expectations on both sides of (15) over the randomness of control decisions conditioning on $\mathbf{Z}(t)$, subtracting the term $V\mathbb{E}\{\sum_{i \in \mathcal{M}} U_i(A_i(t)) - \sum_{j \in \mathcal{N}} C_j(e_j(t)) | \mathbf{Z}(t)\}$ from both sides, combining with Eq. (6) and Eq. (7), and rearranging the terms, we can see that lemma 1 holds.

B. Profit Optimal Online Control Algorithm

In this subsection, we will present our profit optimal online control algorithm ***(POC)***. The idea of the algorithm is to approximately minimize the upper bound of the *drift-minus-profit* given above, subject to the sensing time-availability constraint (1). In each time slot t, according to online observation of $\mathbf{Q}(t)$ and $\mathbf{S}(t)$, *POC* takes the following four phases of control actions, including requests admission and dispatching control, sensing time purchasing, sensing time allocation and queue update.

1) Request Admission and Dispatching: For requests of each type $i \in \mathcal{M}$, the requests admission decisions $A_i(t)$ and dispatching decisions $(A_{i1}(t), A_{i2}(t), \cdots, A_{in}(t))$ as shown in Fig. 2 can be made by maximizing term (12). Since the admission decisions and dispatching decisions of different types are independent from each other, we can concurrently choose $A_i(t)$ and $A_{ij}(t)$ to be the optimal solutions of the following optimization problem:

$$\max_{A_i(t), A_{ij}(t)} \quad V\beta_i A_i(t) - \sum_{j\in\mathcal{N}} Q_{ij}(t) A_{ij}(t) \tag{16}$$

$$\text{s.t.} \quad 0 \leq A_i(t) \leq R_i(t), \forall i \in \mathcal{M}, \quad \sum_{j\in\mathcal{N}} A_{ij}(t) = A_i(t).$$

This problem seems a little complicated due to the coupling of $A_i(t)$ and $A_{ij}(t)$. We can consider the problem in a simple way: if the value of $A_i(t)$ is given in advance, the problem can be rewritten as follows:

$$\min_{A_{ij}(t)} \quad \sum_{j\in\mathcal{N}} Q_{ij}(t) A_{ij}(t) \tag{17}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}} A_{ij}(t) = A_i(t), \forall i \in \mathcal{M}.$$

Note that in (17), the amount of the i-th type requests dispatched to smartphone j is weighted by the current queue backlog. Therefore, the optimal dispatching decision for i-th type request, $\forall i \in \mathcal{M}$, is to dispatch as many admitted requests as possible to the smartphone with the shortest queue:

$$A_{ij}(t) = \begin{cases} A_i(t), & \text{if } j = j_i^*, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

where $j_i^* = \arg\min_{j\in\mathcal{N}} Q_{ij}(t)$, that is, the queue of the i-th type request maintained by smartphone j_i^* is the shortest queue among all n queues for the i-th type request. Such a control strategy is like an intuitive *"Dispatch to the Shortest Queue"* strategy for load balance. We also find that this strategy can reduce processing delay of newly admitted sensing requests since they are dispatched to the shortest queues.

Based on the dispatching strategy in (18), the request admission control decision $A_i(t)$ in (16) can be solved as:

$$\max_{A_i(t)} \quad V\beta_i A_i(t) - Q_{ij_i^*}(t) A_i(t) \tag{19}$$

$$\text{s.t.} \quad 0 \leq A_i(t) \leq R_i(t), \forall i \in \mathcal{M}.$$

Problem (19) is a simple linear programming problem with respect to $A_i(t)$ and the optimal solution is:

$$A_i(t) = \begin{cases} R_i(t), & V\beta_i > Q_{ij_i^*}(t), \\ 0, & \text{otherwise.} \end{cases} \quad \forall i \in \mathcal{M}. \tag{20}$$

Insight: We can see that the strategy in (20) is a *threshold-based* admission control strategy. When the backlog of the shortest queue for type i, $Q_{ij_i^*}(t)$, is smaller than a threshold $V\beta_i$, all the arrived sensing requests in current time slot are admitted into the system. This increases the system throughput $A_i(t)$ when the system is of a low congestion state. But when the backlog of shortest queue exceeds the threshold, all the arrived requests will be denied to guarantee the system stability.

2) Sensing Time Purchasing: On each smartphone $j \in \mathcal{N}$, the sensing time purchasing control decisions can be made by minimizing term (13). Since the purchasing decisions on different smartphones are independent from each other, each $e_j(t)$ can be concurrently decided by solving the following optimization problem:

$$\min_{e_j(t)} \quad Vke_j(t) + [S_j(t) - \theta_j] e_j(t) \tag{21}$$

$$\text{s.t.} \quad 0 \leq e_j(t) \leq H_j(t), \forall j \in \mathcal{N}.$$

It is easy to get the optimal solution:

$$e_j(t) = \begin{cases} H_j(t), & S_j(t) < \theta_j - Vk, \\ 0, & \text{otherwise.} \end{cases} \quad \forall j \in \mathcal{N}. \tag{22}$$

Insight: Actually, there is no need to make any purchasing decisions if no sensing time is purchasable on smartphones, i.e., $H_j(t) = 0$. If there is sensing time purchasable, this strategy will perform sensing time purchasing on smartphone j only when the available sensing time volume is less than $\theta_j - Vk$, and hence $S_j(t) \leq \theta_j - Vk + H_{max}$ for all t. In this way, we can avoid purchasing too much sensing time when there is enough available sensing time for processing sensing requests. As we will show later, by this strategy, our algorithm can achieve a profit that is within $O(1/V)$ of the optimal, by limiting the volume of available sensing time that has been purchased within $O(V)$.

3) Sensing Time Allocation: First, we define the *perturbed* weight of the i-th type sensing request on smartphone j in time slot t as:

$$W_{ij}(t) \triangleq Q_{ij}(t) - R_i^{max}. \tag{23}$$

Observing that the sensing time allocation decisions $p_{ij}(t)$ are independent among different smartphones. Therefore, in each time slot t, the sensing time allocation decisions on each smartphone $j \in \mathcal{N}$, $(p_{1j}(t), p_{2j}(t), \cdots, p_{mj}(t))$, can be determined in a fully distributed manner, by solving the following maximization problem:

$$\max_{p_{ij}(t)} \quad \sum_{i\in\mathcal{M}} W_{ij}(t)\alpha_i p_{ij}(t) + [S_j(t) - \theta_j] \sum_{i\in\mathcal{M}} p_{ij}(t) \tag{24}$$

$$\text{s.t.} \quad p_{ij}(t) \in \{0, 1, \cdots, p_{max}\}, \quad \text{sensing time-availability constraint (1)}, \quad \forall i \in \mathcal{M}, \forall j \in \mathcal{N}.$$

The term (24) can be rewritten as $\sum_{i\in\mathcal{M}}[W_{ij}(t)\alpha_i + S_j(t) - \theta_j] p_{ij}(t)$, where $W_{ij}(t)\alpha_i + S_j(t) - \theta_j$ can be viewed as the weight of the decision variable $p_{ij}(t)$. The optimal solution for the maximization problem would prefer to make $p_{ij}(t)$ with larger positive weight as big as possible. Following this intuition, each smartphone j adopts a simple but effective *greedy strategy* that ranks each allocation decision variable $p_{ij}(t)$ according to their weight. Then it searches from the one with largest positive wight to the one with the smallest positive weight and allocates the sensing time value to each $p_{ij}(t)$ as big as possible subject to the constraints. The remaining variables with negative weight are set to zero.

4) Queue Update: Finally, the request queue backlogs of each type i maintained by each smartphone j, $Q_{ij}(t)$, can be updated according to the dynamic (3), with the optimal values of $A_{ij}(t)$ and $p_{ij}(t)$ determined by the above strategies. Similarly, the sensing time queues $S_j(t)$ on smartphones can be updated according to dynamic (5), based on the optimal values of $e_j(t)$ and $p_{ij}(t)$.

C. Implementation of POC

We note that *POC* only requires the knowledge of the queue sizes $\mathbf{Q}(t)$ and $\mathbf{H}(t)$. It can make online control decisions without any priori knowledge of $R_i(t)$ and $H_j(t)$. This is very useful in practice since the future information of sensing requests arrival and participants' sensing time sale is difficult to obtain or predict.

As mentioned above, the sensing time purchasing and allocation decisions on different smartphones are independent from each other. Therefore, these two decisions can be made on each smartphone in a *fully distributed manner*. We can divide *POC* into two parts. The first part is the request admission and dispatching control performed on the platform, with the knowledge of queue sizes $\mathbf{Q}(t)$ of all smartphones. And the second part is the sensing time purchasing and allocation decisions performed on each smartphone locally, with its own queue sizes information. Then, the platform and smartphones communicate with each other to exchange the control decisions and update the queue sizes for the next time slot. This distributed control decision making method can significantly lighten the computation load of the central platform when the number of smartphones becomes larger and larger, which is desirable in practical implementation.

D. Performance Analysis

We now analyze the performance of *POC*, in terms of a close to optimum time-averaged profit and strong stability of the system. First, we define the parameter θ_j as:

$$\theta_j \triangleq \alpha_{max}\beta_{max}V + mp_{max}. \tag{25}$$

We see that the value of θ_j can be easily determined since it only requires the knowledge of the maximum coefficients of the utility function (6) and the processing rate-sensing time function (2), and the maximum sensing time expenditure in one time slot, and requires no knowledge of $R_i(t)$ or $H_j(t)$. This feature is very desirable for practical implementations.

***Theorem** 1:* For arbitrary sensing request arrivals of each type $(R_1(t), R_2(t), \cdots, R_m(t))$ and arbitrary purchasable sensing time of each participant $(H_1(t), H_2(t), \cdots, H_n(t))$, under the *POC* algorithm with θ_j defined in (25), we have the following:

a) The request queues and the sensing time queues satisfy the following over all time slots:

$$0 \le Q_{ij}(t) \le \beta_i V + R_i^{max}, \tag{26}$$

$$0 \le S_j(t) \le \theta_j - Vk + H_{max}. \tag{27}$$

What's more, when a smartphone allocates nonzero sensing time to any type of request, $S_j(t) \ge mp_{max}$.

b) The gab between the time averaged profit achieved by *POC* and the optimal profit ρ^* is within $\widetilde{B}/V$:

$$\sum_{i\in\mathcal{M}} U_i(\overline{a_i}) - \sum_{j\in\mathcal{N}} C_j(\overline{e_j}) \ge \rho^* - \frac{\widetilde{B}}{V}, \tag{28}$$

where $\rho^* = \sum_{i\in\mathcal{M}} U_i(a_i^*) - \sum_{j\in\mathcal{N}} C_j(e_j^*)$, a_i^* and e_j^* are the optimal solution to problem (8), and $\widetilde{B} = B + mnp_{max}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i$, which is a finite constant independent of V.

Proof: (Part a)) We prove (26) at first. Obviously, it holds for $t = 0$ since $Q_{ij}(0) = 0, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}$. Now assume that $Q_{ij}(t) \le \beta_i V + R_i^{max}, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}$ in time slot t, we would like to show that it also holds for time slot $t+1$. First, if there is no sensing requests of type i dispatched to smartphone j, then $Q_{ij}(t+1) \le Q_{ij}(t) \le \beta_i V + R_i^{max}$; Second, if there are sensing requests of type i dispatched to smartphone j, according to *request admission and dispatching* strategy of *POC*, j must be the smartphone with the shortest queue for type i among all n queues for type i. Hence, according to (18) and (20), we must have $Q_{ij}(t) < \beta_i V$. Since any queue for type i can receive at most R_i^{max} sensing requests of type i in any time slot, $Q_{ij}(t+1) \le Q_{ij}(t) + R_i^{max} < \beta_i V + R_i^{max}$. Then (26) is proved.

As we have explained when introducing *sensing time purchasing* strategy of *POC* that $S_j(t) \le \theta_j - Vk + H_{max}$, then (27) is proved. We now show that, when $S_j(t) < mp_{max}$, (24) will be maximized by choosing $p_{ij}(t) = 0$ for all i on smartphone j: Since $Q_{ij}(t)$ is upper bounded by $\beta_i V + R_i^{max}$, then $W_{ij}(t) \le \beta_i V$. Therefore, the weight of $p_{ij}(t)$, $W_{ij}(t)\alpha_i + S_j(t) - \theta_j \le \alpha_i\beta_i V + mp_{max} - (\alpha_{max}\beta_{max}V + mp_{max}) \le 0$. We can easily see that (24) is maximized by $p_{ij}(t) = 0, \forall i \in \mathcal{M}$. This shows that $S_j(t) \ge mp_{max}$ whenever smartphone j allocates sensing time to any type of request. Hence all sensing time allocation decisions are feasible. It shows that the constarint (1) is indeed redundant in *POC* and completes the proof of part a).

(Part b)) Next, we prove part b) which requires the following lemma 2.

***Lemma** 2*: (Existence of Optimal Randomized Stationary Policy): For arbitrary sensing request arrival rate of each type $(\overline{r_1}, \overline{r_2}, \cdots, \overline{r_m})$ and arbitrary purchasable sensing time of each participant $(\overline{h_1}, \overline{h_2}, \cdots, \overline{h_n})$, there exists a randomized stationary control policy σ that chooses feasible control decisions $A_i^\sigma(t), A_{ij}^\sigma(t), e_j^\sigma(t)$ and $p_{ij}^\sigma(t), \forall i \in \mathcal{M}, \forall j \in \mathcal{N}, \forall t$, independent of the current queue sizes, and yields the following steady state values:

$$\begin{aligned} \mathbb{E}\{A_i^{\sigma}(t)\} &= a_i^* \\ \mathbb{E}\{\alpha_i p_{ij}^{\sigma}(t)\} &\geq \mathbb{E}\{A_{ij}^{\sigma}(t)\} && (29) \\ \mathbb{E}\{e_j^{\sigma}(t)\} &= e_j^* \\ \mathbb{E}\{\sum_{i\in\mathcal{M}} p_{ij}^{\sigma}(t)\} &= \mathbb{E}\{e_j^{\sigma}(t)\}. && (30) \end{aligned}$$

Since Lemma 2 can be proved using similar techniques in [9], we omit the details for brevity. Note that in Lemma 2, (29) shows the rate of dispatched requests to smartphone j of type i is no more than the processing rate; (30) says that the rate of sensing time purchasing is equal to the sensing time consumption rate on smartphone j.

We now show that ***POC*** approximately minimizes the upper bound of the *drift-minus-profit* expression (11). Note that ***POC*** indeed minimizes the following function in each time slot t:

$$\begin{aligned} D(t) \;=\; &-[\sum_{i\in\mathcal{M}} V\beta_i A_i(t) - \sum_{j\in\mathcal{N}} Q_{ij}(t)A_{ij}(t)] \\ &+ \{\sum_{j\in\mathcal{N}} Vke_j(t) + [S_j(t)-\theta_j]e_j(t)\} \\ &- \{\sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} [Q_{ij}(t) - R_i^{max}]\alpha_i p_{ij}(t) \\ &\qquad + [S_j(t)-\theta_j]\sum_{i\in\mathcal{M}} p_{ij}(t)\}, \end{aligned}$$

subject to only the constraints: $A_i(t) \in [0, R_i(t)], e_j(t) \in [0, H_j(t)]$ and $0 \leq p_{ij}(t) \leq p_{max}$, i.e., without the sensing time-availability constraint (1). We define $\widetilde{D}(t)$ as:

$$\begin{aligned} \widetilde{D}(t) \;=\; &-[\sum_{i\in\mathcal{M}} V\beta_i A_i(t) - \sum_{j\in\mathcal{N}} Q_{ij}(t)A_{ij}(t)] \\ &+ \{\sum_{j\in\mathcal{N}} Vke_j(t) + [S_j(t)-\theta_j]e_j(t)\} \\ &- \{\sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} Q_{ij}(t)\alpha_i p_{ij}(t) + [S_j(t)-\theta_j]\sum_{i\in\mathcal{M}} p_{ij}(t)\}, \end{aligned}$$

which is in fact the function inside the expectation of the upper bound of the drift-minus-profit. It is clear that,

$$D(t) = \widetilde{D}(t) + \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i p_{ij}(t).$$

Since ***POC*** minimize $D(t)$, we have:

$$\begin{aligned} &\widetilde{D}^P(t) + \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i p_{ij}^P(t) \\ &\qquad \leq \widetilde{D}^{ALT}(t) + \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i p_{ij}^{ALT}(t), \end{aligned}$$

where the superscript P denotes the ***POC*** algorithm, and ALT denotes any other alternate policy. Combined with the fact that,

$$0 \leq \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i p_{ij}(t) \leq mnp_{max}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i,$$

we further have:

$$\widetilde{D}^P(t) \leq \widetilde{D}^{ALT}(t) + mnp_{max}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i. \qquad (31)$$

That is, the value of $\widetilde{D}(t)$ under ***POC*** is no greater than the value under any other policy plus a constant. Moreover, it has been explained that the sensing time-availability is naturally satisfied under ***POC*** algorithm without explicitly being enforced in part a). According to the definition of $\widetilde{D}(t)$, we have:

$$\begin{aligned} &\Delta(t) - V\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(t)) - \sum_{j\in\mathcal{N}} C_j(e_j^P(t))|\mathbf{Z}(t)\} \\ &\qquad\qquad \leq B + \mathbb{E}\{\widetilde{D}^P(t)|\mathbf{Z}(t)\}. \end{aligned}$$

Based on (31), we have:

$$\begin{aligned} &\Delta(t) - V\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(t)) - \sum_{j\in\mathcal{N}} C_j(e_j^P(t))|\mathbf{Z}(t)\} && (32) \\ &\qquad\qquad \leq \widetilde{B} + \mathbb{E}\{\widetilde{D}^{ALT}(t)|\mathbf{Z}(t)\}, \end{aligned}$$

where $\widetilde{B} = B + mnp_{max}\sum_{i\in\mathcal{M}} R_i^{max}\alpha_i$. Plugging the optimal randomized stationary policy σ in Lemma 2 into (32) yields:

$$\Delta(t) - V\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(t)) - \sum_{j\in\mathcal{N}} C_j(e_j^P(t))|\mathbf{Z}(t)\} \leq \widetilde{B} - V\rho^*,$$

where $\rho^* = \sum_{i\in\mathcal{M}} U_i(a_i^*) - \sum_{j\in\mathcal{N}} C_j(e_j^*)$. Then taking expectations over $\mathbf{Z}(t)$ and summing the above over time slots $\tau \in \{0, 1, \cdots, t-1\}$, we have:

$$\begin{aligned} &\mathbb{E}\{L(t) - L(0)\} - V\sum_{\tau=0}^{t-1}\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(\tau)) \\ &\qquad\qquad - \sum_{j\in\mathcal{N}} C_j(e_j^P(\tau))\} \leq t\widetilde{B} - tV\rho^*. \end{aligned}$$

Rearranging terms and considering the fact that $L(t) \geq 0$ and $L(0) = 0$, dividing both sides by Vt yields:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(\tau)) - \sum_{j\in\mathcal{N}} C_j(e_j^P(\tau))\} \geq \rho^* - \frac{\widetilde{B}}{V}. \quad (33)$$

Since the utility function $U_i(*)$ and the cost function $C_j(*)$ are linear, we have:

$$\begin{aligned} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\sum_{i\in\mathcal{M}} U_i(A_i^P(\tau))\} &= \sum_{i\in\mathcal{M}} U_i(\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}(A_i^P(\tau))), \\ \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\sum_{i\in\mathcal{M}} C_j(e_j^P(\tau))\} &= \sum_{i\in\mathcal{M}} C_j(\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}(e_j^P(\tau))). \end{aligned}$$

By plugging the above two equations into (33) and taking a lim as $t \to \infty$, we finally have:

$$\sum_{i\in\mathcal{M}} U_i(\overline{a_i}) - \sum_{j\in\mathcal{N}} C_j(\overline{e_j}) \geq \rho^* - \frac{\widetilde{B}}{V}, \qquad (34)$$

This completes the proof of part b).

Insight: Part a) of Theorem 1 provides a powerful deterministic guarantee of upper bounds on all request queue backlogs and the volume of available sensing time that has been purchased within $O(V)$ in any time slot. This feature shows that *POC* can ensure the system stability and avoid purchasing too much sensing time which may not be required in the future. Moreover, part a) shows that the sensing time-availability constraint (1) is naturally satisfied under *POC* without explicitly being enforced. Part b) suggests that the difference between the time averaged profit achieved by *POC* and the optimal profit is within $O(1/V)$. If the value of parameter V is large enough, the time averaged profit obtained by *POC* can become arbitrarily close to the optimum. But a large V will also result in large request queue backlogs according to (26), which implies great processing delays for sensing requests.

IV. Performance Evaluation

We carry out simulations to evaluate our online control algorithm *POC* for crowd sensing platforms. In our simulations, we consider that the platform is to serve m (varies form 3 to 5) heterogenous types of sensing requests. The utility brought by processing different types of request are different according to the type-specific utility coefficient β_i, which are [1,2,3,4,5] for the 5 types of requests in our simulations. The sensing requests of type i arrive at the platform according to a random process of average rate $\overline{r_i}$, which are [30,50,30,30,40]. For each type of sensing request, we set its maximum arrival rate as $R_i^{max} = 2\overline{r_i}$. And we assume that, in each time slot t, the number of newly arrived sensing requests, $R_i(t)$, is uniformly and randomly distributed within $[0, R_i^{max}]$. The coefficient α_i is assumed to be one for all types of sensing requests. Besides, we choose $p_{max} = 1$ and the cost coefficient $k = 1$.

Suppose that there are n (varies from 50 to 300) smartphones participating in the system. On each smartphone, we simulate the purchasable sensing time as: If there is no sensing time being purchasable in time slot t, then $H_j(t) = 0$; Otherwise, $H_j(t)$ takes a random value within [0,1000] with equal probability. And the time (slot) interval between each time sensing time being purchasable is assumed to be within [100,1000] randomly, with equal probability. All the following simulations are run for $10,000$ time slots.

First, we verify the optimality of our algorithm. Fig. 3 shows the simulation results with the growth of parameter V. In this case, we use $m = 4$ types of sensing requests and $n = 100$ smartphones and simulate $V \in \{20, 30, 40, 50, 60, 80, 100, 150, 200, 300, 500\}$. We see that the time averaged profit of the platform achieved by *POC* improves significantly and converges quickly to very close to the optimal value as the value of V increases. Meanwhile, the average request queue backlog and sensing time queue size which means the available (purchased) sensing time volume both grow linearly as the value of V increases. It shows the tradeoffs between the profit and system stability which can be adjusted by parameter V, and verifies Theorem 1 that the time averaged profit achieved by *POC* can approach the optimal profit within a gap of $O(1/V)$ while ensuring deterministic upper bounds of $O(V)$ on request queue backlogs and the volume of sensing time purchased.

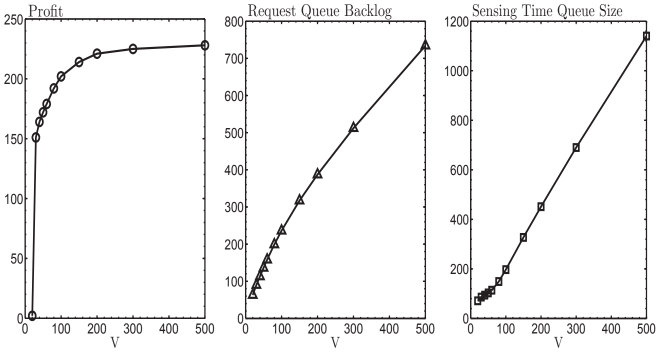

Fig. 3. Simulation results of *POC* with the growth of V.

Second, we compare *POC* with two baseline online control algorithms described as follows.

1) **Random Online Control Algorithm (RANDOM)**: In this algorithm, each smartphone maintains a queue with fixed size, which equals to the upper bound of request queue backlog in *POC*, for each type of sensing request. Control decisions are made randomly except for request dispatching control which consider the load balance by using the *Dispatch to the Shortest Queue* strategy as *POC*. Requests will be denied only if all the queues have already been full.
2) **Greedy Online Control Algorithm (GREEDY)**: This algorithm makes greedy control decisions in each time slot. It admits sensing requests as many as possible, dispatches requests using the *Dispatch to the Shortest Queue* strategy, purchases and allocates sensing time as much as possible.

We use *POC* with $V = 100$ in the comparison. Results are shown in Fig. 4, Fig. 5 and Fig. 6:

In the case of Fig.4, we fix the number of sensing request types $m = 4$ and compare the time averaged profit achieved by three algorithms as the number of smartphones varies from 50 to 300. We find that the profit achieved by *POC* increases as the number of smartphones becomes larger since that more smartphones can result in more available sensing time as well as processing capacity. And the profit achieved by *POC* is always higher than that achieved by the RANDOM and GREEDY algorithms. Note that when the number of smartphones exceeds a certain value, the increasing rate of profit achieved by *POC* becomes slower and slower while the profits achieved by the other two algorithms even begin to decrease. This is because when the number of smartphones is small, the available sensing time and processing capacity are relatively not enough. Therefore *POC* is better than other algorithms since it can utilize the limited available sensing time more efficiently. But when the number of smartphone is large, the sensing requests become relatively not enough with respect to the big processing capacity provided by a large number of smartphones. Hence the other two algorithms may

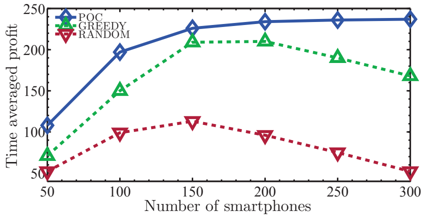

Fig. 4. Time averaged profit vs. number of smartphones

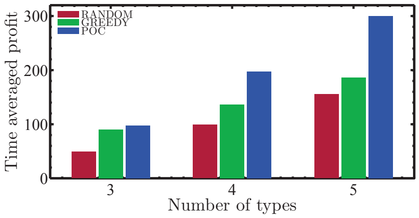

Fig. 5. Time averaged profit vs. number of types

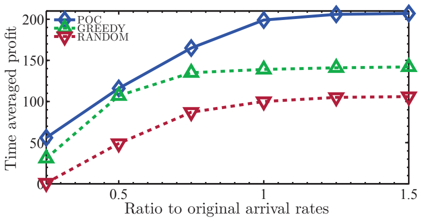

Fig. 6. Time averaged profit vs. ratio to original arrival rates

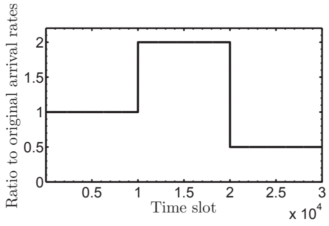

Fig. 7. Varying arrival rates over time

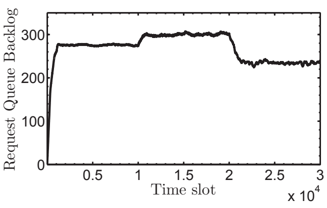

Fig. 8. Average request queue backlog over time

purchase too much sensing time when there are not many requests for processing and result in the decreasing in profit. By contrast, *POC* can avoid purchasing too much sensing time and keep the available (purchased) sensing time volume around a certain level.

In the case of Fig. 5, we use $n = 100$ smartphones and compare the time averaged profit obtained by the three algorithms as the number of sensing request types varies from 3 to 5. We see that *POC* is always better than the RANDOM and GREEDY algorithms and the gap between *POC* and the other algorithms becomes larger when m increases, since *POC* can efficiently utilize the available sensing time.

In Fig. 6, we study the profit achieved by three algorithms under different requests' arrival rates. We change the average arrival rates of each type request from 0.25 of the original value ($0.25\overline{r_i}$) to 1.5 times of the original value ($1.5\overline{r_i}$) and find that *POC* can always achieve better time averaged profit than the other two algorithms regardless of the changing in arrival rates of sensing requests.

At last, we examine the system stability under our control algorithm. Different from the above simulations with fixed average arrival rate of each type sensing request, Fig. 8 plots the average request queue backlog on each smartphone over time with the average arrival rates of sensing requests varying in a bursty manner. We divide the simulation time into three phases, each of which is of 10,000 time slots. The arrival rates are varying according to Fig. 7. In the first phase, the average arrival rates are equal to original values. We let the average arrival rates of each type sensing request abruptly rise to $2\overline{r_i}$ in the second phase and suddenly drop to $0.5\overline{r_i}$ in the last phase. We see from Fig. 8 that the system are strong stable all the time despite abruptly changing in arrival rates.

V. Related Work

Due to the fast increasing of usage of smartphones, mobile crowd sensing is becoming more and more popular in recent years and has attracted extensive research attention from both academia and industry.

A great number of crowd sensing applications have been designed and implemented. But they all built for specific purpose and requirements. The sensing data generated by the application are usually only available for single use. To realise the greater vision of a mobile crowdsensing application, common platforms that facilitates easy development and deployment of crowd sensing applications are developed. For example, MOSDEN [7] is a collaborative mobile crowd sensing platform to develop and deploy opportunistic sensing applications. It can operate on smartphones capturing and sharing sensing data between multiple crowd sensing applications. CDAS system [12] enables deployment of various crowd sensing applications that require human involvement for simple verification tasks delivering high accuracy. MECA [8] provides a platform by which different applications can use data generated from diverse mobile data sources. But most of these platforms do not consider the problem of dynamic request arrival and utilize smartphone resources without limitation and payment to smartphone users.

Little existing work has studied the problem of maximizing the profit of the corwd sensing applications and platforms, subject to the constraints of system stability and sensing time availability. And most of the related work require sufficient statistical knowledge and perform in offline manner or prediction-based approach. For example, in [13] the authors seek a mechanism for user participation level determination and payment allocation which is most viable for the platform, that is, it minimizes the total cost of compensating participants, while delivering a certain quality of experience to service requesters. In [14] the authors develop a novel smart phone based vehicular crowd sensing system that achieves efficient utilization of limited 3G budgets to improve system performance. But they all work relying on a prediction-based approach. Their feasibility heavily depend on the accuracy of the prediction of future patterns, which is usually nonstationary and almost unpredictable in the context of crowd sensing. In comparison, our optimal online control algorithm does not require any priori knowledge of the future patterns and can achieve a time averaged profit that could be arbitrarily close to the optimum.

VI. Conclusion

In this paper, we focus on the problem of maximizing profit for crowd sensing platforms. We propose an online control framework based on stochastic Lyapunov optimization techniques in response to the dynamic and unpredictable sensing requests arrival and purchasable sensing time on participants which represent their willingness to participate. Our control framework can independently and concurrently make four important online control decisions including request admission control, dispatching control, sensing time purchasing and allocation control for crowd sensing platform, without any knowledge of the future patterns. Rigorous theoretical analysis as well as comprehensive simulation results indicate that our control framework can achieve a time averaged profit that could be arbitrarily close to the optimum, while still maintaining strong system stability.

References

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, 2010.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 32–39, 2011.

[3] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010, pp. 105–116.

[4] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 55–68.

[5] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 85–98.

[6] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *Proceedings of the 9th workshop on Mobile computing systems and applications*. ACM, 2008, pp. 11–16.

[7] P. P. Jayaraman, C. Perera, D. Georgakopoulos, and A. Zaslavsky, "Efficient opportunistic sensing using mobile collaborative platform mosden," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*. IEEE, 2013, pp. 77–86.

[8] F. Ye, R. Ganti, R. Dimaghani, K. Grueneberg, and S. Calo, "Meca: mobile edge capture and analysis middleware for social sensing applications," in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 699–702.

[9] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[10] L. Huang and M. J. Neely, "Utility optimal scheduling in processing networks," *Performance Evaluation*, vol. 68, no. 11, pp. 1002–1021, 2011.

[11] ——, "Utility optimal scheduling in energy harvesting networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2011, p. 21.

[12] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "Cdas: a crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, 2012.

[13] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 1402–1410.

[14] H. Liu, S. Hu, W. Zheng, Z. Xie, S. Wang, P. Hui, and T. Abdelzaher, "Efficient 3g budget utilization in mobile participatory sensing applications," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 1411–1419.