Dynamic Sensor Renting in RF-powered Crowdsensing Service Market with Blockchain

Shaohan Feng[1], Wenbo Wang[1], Dusit Niyato[1], Dong In Kim[2], and Ping Wang[3]
[1] School of Computer Engineering, Nanyang Technological University (NTU), Singapore
[2] School of Information and Communication Engineering, Sungkyunkwan University (SKKU), Korea
[3] Department of Electrical Engineering and Computer Science, York University, Canada

***Abstract*—Embedding sensors on wireless devices for collaborative environment sensing has been envisioned as a cost-effective solution for IoT applications. However, existing IoT platforms faces challenges, e.g., unsustainablility due to the limited on-device battery and tremendous cost of deploying middlewares for centralized task dispatching. In this paper, we employ wireless energy transfer and permissionless blockchains to construct a sustainable and decentralized IoT crowdsensing platform. Therein, IoT sensing cloud composed of multiple co-located sensors is wirelessly powered by RF-energy beacons for data sensing and transmission. The data is then forwarded to the blockchain for distributed data/transaction verification and trading. The data users access the crowdsensing service by renting sensors from the sensing clouds. Both the sensing clouds and data users are self-interested and aim to maximize their individual profits. The sensing clouds handle the interference of concurrent wireless transmissions and the on-chain transaction cost. Meanwhile, each user distributes its limited budget over the sensing clouds to optimize the service quality. We formulate a Stackelberg differential game to analyze the interaction among the sensing clouds and data users. Then, we investigate the Stackelberg equilibrium by capitalizing on Pontryagin's maximum principle. Furthermore, we provide a series of insightful numerical results about the Stackelberg equilibrium.**

***Index Terms*—Crowdsensing, blockchain, energy harvesting, Stackelberg differential game, evolutionary game.**

I. Introduction

With the accelerated development of 5G and Internet of things (IoT) technologies, a plethora of scenarios now sees the trend of massive deployment of smart devices for data collection/analysis and cyber-physical system control. Gartner Inc. predicts that 14.2 billion IoT devices will be installed world-wide by the end of 2019, and the number will increase twofold, reaching more than 25 billion by 2021 [1]. Such wide use of IoT devices has created unprecedented opportunities of access to ubiquitous sensing data about the environment context in concern (e.g., urban environment such as pollution and traffic condition) for both real-time use and big data-based analysis. However, it also imposes great challenges to network operation and data processing. Compared with the conventional Wireless Sensor Networks (WSNs), existing IoT models are characterized by the feature of high-level decentralization, hence lacking cooperation due to user deployment and non-existence of infrastructure operators. Furthermore, the data generated by the same sensors may be consumed by different data services, which require various levels of data quality and sampling frequency for different purposes. For those reasons, the existing architectures of WSNs are limited in proliferation due to the high cost of maintaining connections in massive networks as well as the rigidness of the task-specified data processing/dispatching frameworks.

In this paper, we study an RF-powered IoT network, where the user-deployed IoT sensing clouds powering themselves with the RF-energy transferred from the operator-deployed power beacons. Unlike conventional WSNs, a number of non-supplementary sensing tasks are posted by different data users. The sensing clouds can freely choose different sensing tasks to which the sensing clouds respond. Due to the nature of decentralization and task-demand heterogeneity, a series of challenges are to be addressed in the considered RF-powered IoT sensing system:

- *Connectivity*: Facing multiple sensing tasks, the RF-powered sensing clouds need to choose a suitable data user to which the sensing clouds respond without the help from a central coordinator.
- *Reliability*: Unlike the conventional WSNs, it is required that the turstworthiness among different parties can be established without the help of a central intermediary.
- *Adaptivity*: The considered RF-powered IoT sensing system is expected to not only support multiple data sensing tasks, but also be extensible to serve new data sensing demands of other users in the future.

To address these challenges without introducing centralization at the data processing stage, we resort to the emerging technology of permissionless blockchains [2] to ensure that the task assignment, the data collection, storage and trading are all performed in a decentralized but trusted manner. The introduction of the blockchain removes the intermediate layer while enhancing user privacy and data security with the embedded cryptographic functionalities [2], and the data integrity is publicly verifiable thanks to the special data structure built upon the chain of data blocks. In brief, a permissionless blockchain system can be seen as a replicated database maintained by a number of pseudonymous nodes over P2P connections. Blockchains use the public key infrastructure (PKI) mechanism and the data structure of hash linked list to ensure that the time order and the content of the corresponding digest of a data record (also known as a transaction) cannot be tampered without being discovered once confirmed in the local replica [2]. The blockchain adopts a certain Byzantine fault-tolerate (BFT) mechanism, e.g., Nakamoto protocol [3] or Practical Byzantine Fault Tolerance (PBFT) protocol [4],

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

to coordinate the Byzantine agreement (i.e., peer consensus) about the state of the transaction storage among the nodes.

In this paper, we propose a novel resource management framework of RF-powered wireless crowdsensing, where the wireless network operator is only responsible for providing the infrastructure of data communication and power beacons at flat prices [5]. Our designed crowdsensing system involves four parties, i.e., the massive-scale (wireless) sensors working as multiple sensing clouds, the wireless network operator working as a utility provider with a flat-rate pricing scheme for wireless power transfer, a permissionless blockchain to handle the data trading, task dispatching and data storage, and the sensing application schedulers, i.e., data users. As shown in Fig. 1(a), the IoT sensing clouds are connected to the permissionless blockchain through the backhaul provided by the wireless network operator. The blockchain enables smart contracts in the form of programmable automata on the chain [6], with which the sensing clouds and the data users are able to deploy their own sensing cloud service access contracts or choose some of them to answer to. The sensing data is delivered by the sensing clouds to the data users via the logical links established on the blockchain, and the data is encapsulated as transactions over the blockchain. Note here that the service access and data delivery processes are completely self-organized in the form of smart contract, without the need of any help from an intermediary controlled by the operator. Then, from the perspective of the sensing clouds and the data users, the blockchain can be seen as a decentralized platform as a service (see Fig. 1(b)) and the details about the blockchain network organization are completely hidden behind the smart contract interface. With our proposed crowdsensing system, the following key performance properties are satisfied:

- The data (i.e., transaction) throughput of the blockchain scales well such that the massive data volume from the sensing clouds is handled smoothly.
- The sensing clouds are able to work in a self-orgainzed manner with limited coordination among themselves.
- The sensing clouds are rational and each of them independently decides on its own sensing/transferring schedule for revenue optimization.

The rest of the paper is organized as follows. Section II describes the system model. Section III presents the formulation of a Stackelberg differential game. Section IV analytically analyzes the equilibrium of the proposed Stackelberg differential game. Section V presents the numerical performance evaluation. Section VI concludes the paper.

II. System Description

We consider an RF-powered IoT crowdsensing service market as shown in Fig.1. Specifically, a set of sensing clouds, denoted by $\mathcal{N}$, harvest energy from the beamforming-enabled RF-energy beacons deployed by the network operator to support data transfer from the sensing clouds to the access point. Each sensing cloud $i \in \mathcal{N}$ receives the power of $p_i \in \mathcal{D}_{p_i} = [0, p_i^{\mathrm{u}}]$ from the network operator through wireless power transfer, where p_i^{u} is the upper bound of the power

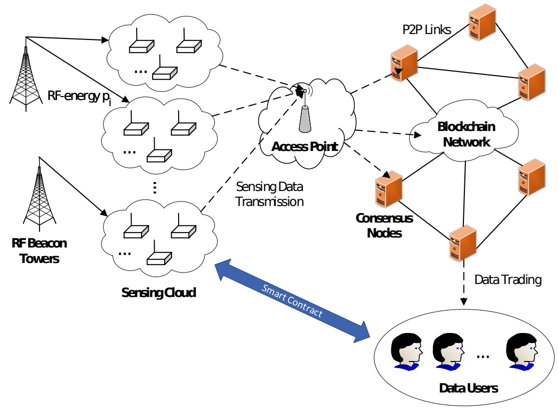

Figure 1: System model.

Table I: Notations

Symbol	Description
i, j, k, h	Sensing clouds i, j and users k, h.
$\mathcal{N}$, N	The set of sensing clouds and its cardinality.
$\mathcal{K}$, K	The set of users and its cardinality.
p_i, r_i	The received power and transmission rate of sensing cloud i, respectively.
x_{ik}	The ratio of user k's budget paid to sensing cloud i.
w_k	User k's budget for renting the sensors.
g_i, h_i, σ	The channel gain and bandwidth of sensing cloud i, and the noise, respectively.
c_i	The power used for circuit maintenance and data sensing of sensing cloud i.
d_i, α_i	The distance between sensing cloud i and the access point connected to the blockchain, and the path loss coefficient of wireless communication, respectively.
d_i^{t}, η	The distance between the RF-energy beacons and sensing cloud i, and the path-loss exponent of wirelessly power transfer, respectively.
$\mathbf{x}_i$, $\mathbf{x}_k$, $\mathbf{X}$, $\mathbf{x}_{i_{-k}}$, $\mathbf{X}_{-k}$	$\mathbf{x}_i = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iK}]$, $\mathbf{x}_k = [\mathbf{x}_{1k}, \ldots, \mathbf{x}_{Nk}]$, $\mathbf{x}_{i_{-k}} = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ik-1}, \mathbf{x}_{ik+1}, \ldots, \mathbf{x}_{iK}]$, $\mathbf{X} = [\mathbf{x}_k]_{k\in\mathcal{K}}$, $\mathbf{X}_{-k} = [\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_K]$.
$\mathcal{D}_{\mathbf{r}}$, $\mathcal{D}_{\mathbf{p}}$, $\mathcal{D}_{\mathbf{X}}$	The domains of definition for $\mathbf{r}$, $\mathbf{p}$, and $\mathbf{X}$, respectively.
Π_i, U_{ik}	The profit of sensing cloud i, and the utility of user k from sensing cloud i, respectively.

that sensing cloud i can receive. Let c_i be the power of sensing cloud i used for circuit maintenance and data sensing, the power used by sensing cloud i to transmit data to the blockchain for trading is therefore $p_i - c_i$. The propagation model of sensing cloud i follows (1) by the free-space propagation model and consider the path loss as a function of the distance between sensing cloud i and the access point connected to the blockchain, i.e., d_i. Let $\mathbf{r} = [r_i]_{i\in\mathcal{N}}$ be the vector of the transmission rates and $\mathbf{p} = [p_i]_{i\in\mathcal{N}}$ be the vector of received powers. Likewise, let $\mathbf{p}_{-i}$ be the vector of the received powers except sensor i, the transmission rate of sensing cloud i can be derived as follows:

$$r_i = R_i\left(p_i, \mathbf{p}_{-i}\right) = h_i \log_2\left(1 + \frac{g_i \frac{p_i - c_i}{(d_i)^{\alpha_i}}}{\sum\limits_{j\neq i} g_j \frac{p_j - c_j}{(d_j)^{\alpha_j}} + \sigma^2}\right), \forall i \in \mathcal{N} \quad (1)$$

where σ^2 is the noise, g_i is the channel gain of sensing cloud i, and h_i is the bandwidth of sensing cloud i. We define $\mathbf{R}(\mathbf{p}) = [R_i(p_i, \mathbf{p}_{-i})]_{i\in\mathcal{N}}$, and $\mathbf{R}(\mathbf{p}) : \mathcal{D}_{\mathbf{p}} = \underset{i\in\mathcal{N}}{\times} \mathcal{D}_{p_i} \mapsto \mathcal{D}_{\mathbf{r}}$ is a continuous closed mapping.

There is a set of users, denoted by $\mathcal{K}$, who want to rent sensors from the sensing clouds. The users rent a portion of the sensors in each sensing cloud by setting the smart contract with the sensing clouds through blockchain. The rented sensors collect data and send the data to its corresponding sensing cloud. Then, the sensing cloud transmits the data to the users by wireless transmission. Due to the limited transmission rates of the sensing clouds, the sensing cloud determines the data transmission rate for each user based on the portion of its sensors rented by the user. The portion of the sensors rented by the user in each sensing cloud depends on the percentage of the sensing cloud's total revenue from the user. Each user $k \in \mathcal{K}$ has a budget of w_k for renting the sensors. Let x_{ik} denote the ratio of user k's budget paid to sensing cloud i, sensing cloud i will accordingly receive the revenue of $x_{ik}w_k$ from user k and provides the data at the rate of $\frac{x_{ik}w_k}{\sum_{h\in\mathcal{K}} x_{ih}w_h} R_i(p_i, \mathbf{p}_{-i})$ to user k, where $\sum_{h\in\mathcal{K}} x_{ih}w_h$ is the total revenue of sensing cloud i. Therefore, the utility of user k obtained from the data collected by its rented sensors in sensing cloud i is

$$U_{ik}\left(x_{ik}, \mathbf{x}_{i_{-k}}, \mathbf{p}\right) = \Psi_{ik}\left(\frac{x_{ik}w_k}{\sum\limits_{h\in\mathcal{K}} x_{ih}w_h} R_i(p_i, \mathbf{p}_{-i})\right) - x_{ik}w_k, \quad (2)$$

where $\Psi_{ik}(\cdot)$ is a continuous and increasing concave function representing the benefit that the user k can obtain from the data provided by sensing cloud i [7]. $\Psi_{ik}(\cdot)$ depends not only on the transmission rate of the data but also on the quality and type of the data. The higher the quality of the data is, the higher utility that the user can achieve. Also, the type of the data can affect the users' utilities. For example, if the user wants to estimate the road traffic jam in a city while the sensing cloud provides the data concerning the weather, the user will obtain less utility from the data concerning weather than that from the data of road traffic.

We assume that the smart contract service is implemented based on a framework of sharded blockchain. According to the protocol in [8], by assuming a fixed transaction data size, the transaction throughput of the sharded blockchain is linearly related to the transmission rate. Moreover, the computational power admitted by the sharded blockchain is a linear function of the transaction throughput [8] and hence is a linear function of the transmission rate. Without loss of generality, the computational power of the sharded blockchain required to support the total transmission rate from all the sensors, i.e., $\sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})$, is $m \sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})$, where $m > 0$. Furthermore, since the energy consumption of the sharded blockchain is convexly related to its computational power [9], a quadratic functional form can be adopted to describe the relationship between the computational power and its corresponding energy consumption [10]. Therefore, the energy consumption of the sharded blockchain is $\Upsilon\left(m \sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})\right)$, where $\Upsilon(\cdot)$ is a quadratic function and hence an increasing convex function.

To enjoy the data service (e.g., smart contracts and transaction recording) provided by the sharded blockchain, sensing cloud i needs to pay a transaction fee to compensate the cost of the consensus nodes, e.g., incurred from energy consumption. We assume that the sensing clouds proportionally share the blockchain maintenance cost among themselves according to the volume of data that they propose to the blockchain. Then, the payment of sensing cloud i based on its fraction of the total transmit rate is $\frac{R_i(p_i,\mathbf{p}_{-i})}{\sum_{j\in\mathcal{N}} R_j(p_j,\mathbf{p}_{-j})} \Upsilon\left(m \sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})\right)$. Let ϕ denote the price of unit power transferred from the RF-energy beacons. To support the wirelessly received power level p_i for sensing cloud i, the wirelessly transferred power level of the RF-energy beacons i is $P^{\mathrm{t}}(p_i, d_i^{\mathrm{t}}) = p_i (d_i^{\mathrm{t}})^{\eta}$ based on the Slivnyak-Meckes theorem [11], where η is the path-loss exponent of wirelessly power transfer and d_i^{t} is the distance between the RF-energy beacons and sensing cloud i. The total cost of sensing cloud i for providing crowdsensing service is $\phi P^{\mathrm{t}}(p_i, d_i^{\mathrm{t}}) + \frac{R_i(p_i,\mathbf{p}_{-i})}{\sum_{j\in\mathcal{N}} R_j(p_j,\mathbf{p}_{-j})} \Upsilon\left(m \sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})\right)$ while its revenue is $\sum_{k\in\mathcal{K}} x_{ik}w_k$. Accordingly, the profit of sensing cloud i is

$$\begin{aligned} \Pi_i(p_i, \mathbf{p}_{-i}, \mathbf{x}_i) &= \sum_{k\in\mathcal{K}} x_{ik}w_k - \phi P^{\mathrm{t}}(p_i, d_i^{\mathrm{t}}) \\ &- \frac{R_i(p_i, \mathbf{p}_{-i})}{\sum\limits_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})} \Upsilon\left(m \sum_{j\in\mathcal{N}} R_j(p_j, \mathbf{p}_{-j})\right) \end{aligned} \quad (3)$$

For the convenience of analyzing the Stackelberg equilibrium of the Stackelberg differential game in Section IV-B, we let the sensing clouds maximize their profits by deciding on their transmission rates instead of their received powers. This is due to the fact that the vector of the functions describing the relationship between $\mathbf{r}$ and $\mathbf{p}$, i.e., $\mathbf{r} = \mathbf{R}(\mathbf{p})$, is a continuous closed injective operator as shown in Theorem 1. This means that there exists an inverse operator of $\mathbf{R}(\mathbf{p})$ denoted by $\mathbf{R}^{-1}(\mathbf{r}) : \mathcal{D}_{\mathbf{r}} \mapsto \mathcal{D}_{\mathbf{p}}$ such that the power consumption of the sensors, i.e. $\mathbf{p}$, can be determined by their transmission rates, i.e., $\mathbf{r}$.

THEOREM 1. *There exists an inverse operator of $\mathbf{R}(\mathbf{p})$, i.e., $\mathbf{R}^{-1}(\mathbf{r}) : \mathcal{D}_{\mathbf{r}} \mapsto \mathcal{D}_{\mathbf{p}}$, such that $\mathbf{p} = \mathbf{R}^{-1}(\mathbf{r})$, where $\mathbf{R}^{-1}(\mathbf{r}) = \left[R_i^{-1}(r_i, \mathbf{r}_{-i})\right]_{i\in\mathcal{N}}$.*

Proof: Let $\gamma_i(r_i) = e^{\frac{r_i \ln 2}{h_i}} - 1$, $\beta_i(p_i) = g_i \frac{p_i - c_i}{(d_i)^{\alpha_i}}$, and $\boldsymbol{\beta}(\boldsymbol{p}) = [\beta_1(p_1), \ldots, \beta_N(p_N)]^{\top}$, according to (1), we have

$$\gamma_i(r_i) = \Lambda_i(\boldsymbol{\beta}(\boldsymbol{p})) = \frac{\beta_i(p_i)}{\sum\limits_{j\neq i} \beta_j(p_j) + \sigma^2}. \quad (4)$$

Since both $\gamma_i(r_i)$ and $\beta_i(p_i)$, $\forall i \in \mathcal{N}$ are continuous closed injective operators, the injective properties of $\mathbf{R}(\mathbf{p})$ can be ensured if and only if $\boldsymbol{\Lambda}(\boldsymbol{\beta}(\boldsymbol{p})) = [\Lambda_1(\boldsymbol{\beta}(\boldsymbol{p})), \ldots, \Lambda_N(\boldsymbol{\beta}(\boldsymbol{p}))]^\top$ is injective.

We apply the proof by contradiction method to prove that $\boldsymbol{\Lambda}(\boldsymbol{\beta}(\boldsymbol{p}))$ is injective. We assume that $\boldsymbol{\Lambda}(\boldsymbol{\beta}(\boldsymbol{p}))$ is not injective. Then, there exist $\mathbf{p}' \neq \mathbf{p}$ and $\mathbf{p}'$, $\mathbf{p} \in \mathcal{D}_\mathbf{p}$ such that $\mathbf{r}' = \mathbf{r}$. Without loss of generality, we assume $p_i' > p_i$. As such, to ensure $r_i' = r_i$, $\mathbf{p}'$ should satisfy

$$\frac{\beta_i(p_i')}{\sum\limits_{j\neq i}\beta_j\left(p_j'\right)+\sigma^2} = \frac{\beta_i(p_i)}{\sum\limits_{j\neq i}\beta_j(p_j)+\sigma^2}, \tag{5}$$

and hence

$$\sum_{j\neq i}\beta_j(p'_j) = \frac{\beta_i(p'_i)}{\beta_i(p_i)}\sum_{j\neq i}\beta_j(p_j) + \left[\frac{\beta_i(p'_i)}{\beta_i(p_i)} - 1\right]\sigma^2. \tag{6}$$

For $\forall l \in \mathcal{N}$,

$$\begin{aligned}&\frac{\beta_l(p'_l)}{\sum\limits_{j\neq l}\beta_j(p'_j)+\sigma^2} = \frac{\beta_l(p'_l)}{\beta_i(p'_i)+\sum\limits_{j\neq i}\beta_j(p'_j)-\beta_l(p'_l)+\sigma^2}\\ &\overset{(6)}{=}\frac{\beta_l(p'_l)}{\beta_i(p'_i)+\frac{\beta_i(p'_i)}{\beta_i(p_i)}\sum\limits_{j\neq i}\beta_j(p_j)+\left[\frac{\beta_i(p'_i)}{\beta_i(p_i)}-1\right]\sigma^2-\beta_l(p'_l)+\sigma^2}\\ &=\frac{\beta_l(p'_l)}{\frac{\beta_i(p'_i)}{\beta_i(p_i)}\left[\sum\limits_{j\in\mathcal{N}}\beta_j(p_j)+\sigma^2\right]-\beta_l(p'_l)}.\end{aligned} \tag{7}$$

Since $r_l' = r_l$, $\forall l \in \mathcal{N}$ also needs to be ensured, according to (7), we need to ensure the following equality equation: $\frac{\beta_l(p_l)}{\sum\limits_{j\neq l}\beta_j(p_j)+\sigma^2} = \frac{\beta_l\left(p'_l\right)}{\sum\limits_{j\neq l}\beta_j\left(p'_j\right)+\sigma^2} = \frac{\beta_l\left(p'_l\right)}{\frac{\beta_i(p'_i)}{\beta_i(p_i)}\left[\sum\limits_{j\in\mathcal{N}}\beta_j(p_j)+\sigma^2\right]-\beta_l(p'_l)} \Leftrightarrow \beta_l(p'_l) = \frac{\beta_i\left(p'_i\right)}{\beta_i(p_i)}\beta_l(p_l)$. We take the summation of $\beta_l(p'_l)$ with respect to l over $\mathcal{N}$, i.e., $\sum\limits_{l\in\mathcal{N}}\beta_l(p'_l) = \sum\limits_{l\in\mathcal{N}}\frac{\beta_i\left(p'_i\right)}{\beta_i(p_i)}\beta_l(p_l)$. This is contradicted with the result derived from (6), i.e., $\beta_i(p'_i) + \sum\limits_{j\neq i}\beta_j\left(p'_j\right) = \sum\limits_{j\in\mathcal{N}}\beta_j\left(p'_j\right) = \frac{\beta_i\left(p'_i\right)}{\beta_i(p_i)}\sum\limits_{j\in\mathcal{N}}\beta_j(p_j) + \left[\frac{\beta_i\left(p'_i\right)}{\beta_i(p_i)}-1\right]\sigma^2$, the assumption that $\boldsymbol{\Lambda}(\boldsymbol{\beta}(\boldsymbol{p}))$ is not injective is false, and $\boldsymbol{\Lambda}(\boldsymbol{\beta}(\boldsymbol{p}))$ is injective. Moreover, due to the injective property of $\gamma_i(r_i)$ and $\beta_i(p_i)$, $\forall i \in \mathcal{N}$, $\mathbf{R}(\mathbf{p})$ is an injective operator. Furthermore, since $\mathbf{R}(\mathbf{p})$ is continuous closed operator, its inverse operator, i.e., $\mathbf{R}^{-1}(\mathbf{r})$, exists and the proof is completed. ■

Then, according to Theorem 1, the profit of sensing cloud i as shown in (3) is rewritten as follows:

$$\begin{aligned}&\Pi_i(r_i, \mathbf{r}_{-i}, \mathbf{x}_i) = \sum_{k\in\mathcal{K}} x_{ik}w_k -\\ &\phi P^{\mathrm{t}}\left(R_i^{-1}(r_i, \mathbf{r}_{-i}), d_i^{\mathrm{t}}\right) - \frac{r_i}{\sum\limits_{j\in\mathcal{N}} r_j}\Upsilon\left(m\sum_{j\in\mathcal{N}} r_j\right).\end{aligned} \tag{8}$$

Furthermore, the utility of user k obtained from the data collected by its rented sensors in sensing cloud i as shown in (2) can be accordingly rewritten as follows:

$$U_{ik}\left(x_{ik}, \mathbf{x}_{i_{-k}}, r_i\right) = \Psi_{ik}\left(\frac{x_{ik}w_k}{\sum\limits_{h\in\mathcal{K}} x_{ih}w_h} r_i\right) - x_{ik}w_k. \tag{9}$$

III. A Stackelberg Differential Game Formulation

In this section, we model the interaction between the sensing clouds and the users as a Stackelberg differential game. In the lower level of the game, the interplay among the users is formulated as an evolutionary game by using replicator dynamics. In the upper level, the sensing clouds determine their transmission rates in a continuous time space.

A. A Lower-Level Evolutionary Subgame Formulation

In the proposed evolutionary game framework, the users evolve their sensing cloud rental strategies to rent the sensors from the sensing clouds through blockchain. The reason that the users are in the same level, i.e., lower-level is that they have the same set of information and make decisions simultaneously by observing each other's strategies. This is different from the sensing clouds that usually rent out the sensors first and then collect data for the users. Hence, the sensing clouds are considered to be the leaders and their problems are defined in the upper level.

With the utility of user k from sensing cloud i defined in (9), the average utility of user k from all the sensing clouds is expressed by

$$\begin{aligned}&\overline{U}_k(\mathbf{x}_k, \mathbf{X}_{-k}, \mathbf{r}) = \sum_{i\in\mathcal{N}} x_{ik}U_{ik}\left(x_{ik}, \mathbf{x}_{i_{-k}}, r_i\right) =\\ &\sum_{i\in\mathcal{N}} x_{ik}\left[\Psi_{ik}\left(\frac{x_{ik}w_k}{\sum\limits_{h\in\mathcal{K}} x_{ih}w_h} r_i\right) - x_{ik}w_k\right].\end{aligned} \tag{10}$$

Then, by the pairwise proportional imitation protocol [12] and given the sensing clouds' transmission rates, i.e., r_i, the replicator dynamics yields the following lower-level evolutionary subgame, i.e., a system of Ordinary Differential Equations (ODEs):

$$\begin{aligned}&\nabla_t x_{ik}(t) = x_{ik}(t) \times\\ &\left[U_{ik}\left(x_{ik}(t), \mathbf{x}_{i_{-k}}(t), r_i(t)\right) - \overline{U}_k(\mathbf{x}_k(t), \mathbf{X}_{-k}(t), \mathbf{r}(t))\right],\end{aligned} \tag{11}$$

with initial strategies, i.e., Dirichlet boundary condition, $x_{ik}(0) = x^0{}_{ik}$, $\forall i \in \mathcal{N}$, $\forall k \in \mathcal{K}$.

B. A Stackelberg Game Formulation

According to (8), the instantaneous profit of sensing cloud i at time t is given by

$$\begin{aligned}&\Pi_i(r_i(t), \mathbf{r}_{-i}(t), \mathbf{x}_i(t)) = \sum_{k\in\mathcal{K}} x_{ik}(t)w_k -\\ &\phi P^{\mathrm{t}}\left(R_i^{-1}(r_i(t), \mathbf{r}_{-i}(t)), d_i^{\mathrm{t}}\right) - \frac{r_i(t)}{\sum\limits_{j\in\mathcal{N}} r_j(t)}\Upsilon\left(m\sum_{j\in\mathcal{N}} r_j(t)\right)\end{aligned} \tag{12}$$

and the corresponding cumulative profit of sensing cloud i over the time space $\mathcal{T} = [0, T]$ is expressed as follows:

$$\begin{aligned}&\int_{\mathcal{T}}\Pi_i(r_i(t), \mathbf{r}_{-i}(t), \mathbf{x}_i(t))\,\mathrm{d}t = \int_{\mathcal{T}}\sum_{k\in\mathcal{K}} x_{ik}(t)w_k -\\ &\phi P^{\mathrm{t}}\left(R_i^{-1}(r_i(t), \mathbf{r}_{-i}(t)), d_i^{\mathrm{t}}\right) - \frac{r_i(t)}{\sum\limits_{j\in\mathcal{N}} r_j(t)}\Upsilon\left(m\sum_{j\in\mathcal{N}} r_j(t)\right)\mathrm{d}t.\end{aligned} \tag{13}$$

With the lower-level evolutionary subgame defined in (11), the Stackelberg differential game can be formulated as follows:

$$\begin{aligned}&\max_{r_i(t)} \int_{\mathcal{T}} \Pi_i\left(r_i\left(t\right), \mathbf{r}_{-i}\left(t\right), \mathbf{x}_i\left(t\right)\right) \mathrm{d}t \\ &\text{s.t. } \nabla_t x_{ik}\left(t\right) = x_{ik}\left(t\right)\Big[U_{ik}\Big(x_{ik}\left(t\right), \mathbf{x}_{i_{-k}}\left(t\right), r_i\left(t\right)\Big) \\ &\qquad -\overline{U}_k\left(\mathbf{x}_k\left(t\right), \mathbf{X}_{-k}\left(t\right), \mathbf{r}\left(t\right)\right)\Big], \\ &\qquad x_{ik}\left(0\right) = x^0{}_{ik}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}.\end{aligned} \tag{14}$$

Note that the Stackelberg differential game in (14) is to maximize the cumulative profit of the sensing clouds taking the strategy evolution processes of the users' strategies as the constraint. In other words, these strategies indicate the states of the sensing clouds.

IV. Equilibrium Analysis

We analyze the equilibriums of the lower-level evolutionary subgame and the Stackelberg differential game in this section. We first use the Cauchy-Lipschitz theorem in [13] to prove the uniqueness of the solution to the lower-level evolutionary subgame in Theorem 2. Then, by using the Lyapunov's second method for stability, i.e., Lemma 1, we prove the stability of the solution to the lower-level evolutionary subgame in Theorem 3. At last, we prove that there exists a Stackelberg equilibrium in Theorem 4.

A. Solution to Lower-Level Evolutionary Subgame

Firstly, we investigate the uniqueness of the solution to the lower-level evolutionary subgame defined in (11), i.e., the uniqueness of the evolutionary equilibrium.

Theorem 2. *For any given $\mathbf{r}\left(t\right) = \mathbf{r}^{\text{given}} \in \mathbf{D}_{\mathbf{r}}$, the lower-level evolutionary subgame defined in (11) is uniquely solvable and hence admits a unique evolutionary equilibrium.*

Proof: The uniqueness of the solution to the lower-level evolutionary subgame can be validated by using Cauchy-Lipschitz theorem in [13]. Due to the limited space, we omit the proof. ■

Lemma 1. Lyapunov's second method for stability *[14]: Consider the system of ODEso*

$$\nabla_t \mathbf{y}\left(t\right) = \mathbf{F}\left(\mathbf{y}\left(t\right), t\right), \forall t \in \mathcal{T}, \mathbf{y}\left(0\right) = \mathbf{y}^0. \tag{15}$$

Provided that there exists a function $V\left(\mathbf{y}\left(t\right)\right) : \mathbb{R}^n \mapsto \mathbb{R}$ such that

- $V\left(\mathbf{y}\left(t\right)\right) = 0$, *if* $\mathbf{y}\left(t\right) = \mathbf{0}$,
- $V\left(\mathbf{y}\left(t\right)\right) > 0$, *if* $\mathbf{y}\left(t\right) \neq \mathbf{0}$,
- $\nabla_t V\left(\mathbf{y}\left(t\right)\right) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{y}(t)} V\left(\mathbf{y}\left(t\right)\right) \times \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}\left(t\right) = \left(\nabla_{\mathbf{y}} V\right)^\top \times \nabla_t \mathbf{y}\left(t\right) = \left(\nabla_{\mathbf{y}} V\right)^\top \cdot \mathbf{F}\left(\mathbf{y}\left(t\right), t\right) \leq 0$, *i.e., negative semidefinite, for all values of* $\mathbf{y}\left(t\right) \neq \mathbf{0}$,

then the solution to the system of ODEs defined in (15) is stable.

Theorem 3. *For any given $\mathbf{r}\left(t\right) = \mathbf{r}^{\text{given}} \in \mathcal{D}_{\mathbf{r}}$, the lower-level evolutionary subgame defined in (11) admits a stable evolutionary equilibrium.*

Proof: By adopting the Lyapunov's second method for stability introduced in Lemma 1, we design a Lyapunov function as follows:

$$V\left(\mathbf{X}\left(t\right)\right) = \left[\sum_{i \in \mathcal{N}, k \in \mathcal{K}} x_{ik}\left(t\right)\right]^2, \tag{16}$$

where $V\left(\mathbf{X}\left(t\right)\right) : \mathcal{D}_{\mathbf{X}} \mapsto \mathbb{R}$ satisfies

$$V\left(\mathbf{X}\left(t\right)\right) \begin{cases} = 0, \text{ if } \mathbf{X}\left(t\right) = \mathbf{0} \in \mathcal{D}_{\mathbf{X}}, \\ > 0, \text{ otherwise.} \end{cases} \tag{17}$$

With any given $\mathbf{r}\left(t\right) = \mathbf{r}^{\text{given}} \in \mathcal{D}_{\mathbf{r}}$, the first derivative of $V\left(\mathbf{X}\left(t\right)\right)$ with respect to t is given in (18). This means that $\nabla_t V\left(\mathbf{u}\left(t\right), \mathbf{v}\left(t\right), \mathbf{x}\left(t\right)\right)$ is negative semidefinite, i.e., ≤ 0. According to Lemma 1, the lower-level evolutionary subgame defined in (11) admits a stable evolutionary equilibrium. The proof is then completed. ■

B. Solution to Stackelberg Differential Game

For the formulated Stackelberg differential game, Pontryagin's maximum principle [15] can be used as necessary optimality conditions to find the optimal strategies. With Pontryagin's maximum principle, a Hamiltonian system could be established for the Stackelberg differential game. In this case, we denote the right side of the first constraint in our formulated Stackelberg differential game, i.e., (14), as $\theta_{ik}\left(\mathbf{X}\left(t\right), \mathbf{r}\left(t\right)\right)$. Accordingly, the present value Hamiltonian function can be defined as follows:

$$\begin{aligned}&H_i\left(r_i\left(t\right), \mathbf{r}_{-i}\left(t\right), \mathbf{X}\left(t\right), \boldsymbol{\mu}_i\left(t\right)\right) = \\ &\Pi_i\left(r_i\left(t\right), \mathbf{r}_{-i}\left(t\right), \mathbf{x}_i\left(t\right)\right) + \sum_{k \in \mathcal{K}} \mu_{ik}\left(t\right) \theta_{ik}\left(\mathbf{X}\left(t\right), \mathbf{r}\left(t\right)\right),\end{aligned} \tag{19}$$

where $\boldsymbol{\mu}_i\left(t\right) = \left[\mu_{ik}\left(t\right), \ldots, \mu_{ik}\left(t\right)\right]^\top$ and $\mu_{ik}\left(t\right)$ is the costate variable of sensing cloud i associated with $x_{ik}\left(t\right)$. With the definition of the Hamiltonian function, the corresponding maximized Hamiltonian function is defined as follows:

$$\begin{aligned}&H_i^*\left(\mathbf{r}_{-i}\left(t\right), \mathbf{X}\left(t\right), \boldsymbol{\mu}_i\left(t\right)\right) \\ &= \max_{r_i(t)} \left\{ H_i\left(r_i\left(t\right), \mathbf{r}_{-i}\left(t\right), \mathbf{X}\left(t\right), \boldsymbol{\mu}_i\left(t\right)\right) \middle| \, r_i\left(t\right) \in \mathcal{D}_{r_i} \right\}.\end{aligned} \tag{20}$$

As proven in the optimal control theory, the optimal strategy of the original problem must also maximize the corresponding Hamiltonian function. Therefore, all candidate optimal strategies have to satisfy the following necessary optimality conditions:

$$0 = \frac{\partial H_i\left(r_i\left(t\right), \mathbf{r}_{-i}\left(t\right), \mathbf{X}\left(t\right), \boldsymbol{\mu}_i\left(t\right)\right)}{\partial r_i\left(t\right)}, \tag{21}$$

$$\dot{\mu}_{ik}\left(t\right) = \mu_{ik}\left(t\right) - \frac{\partial H_i^*\left(\mathbf{r}_{-i}\left(t\right), \mathbf{X}\left(t\right), \boldsymbol{\mu}_i\left(t\right)\right)}{\partial x_{ik}\left(t\right)}, \tag{22}$$

where (22) is the adjoint equation to describe the dynamics of a costate variable.

Theorem 4. *For the Stackelberg differential game defined in (14), the candidate strategy of the sensing clouds, i.e., $\mathbf{r}^*\left(t\right)$, is a Stackelberg equilibrium.*

Proof: We use the Pontryagin's maximum principle [15] to construct the candidate equilibrium strategy $\mathbf{r}_i\left(t\right)$. Moreover, $\mathcal{D}_{\mathbf{X}}$ is a convex compact set, and Theorem 2 has proven the existence and uniqueness of the solution to the lower-level evolutionary game in (11) with any given $\mathbf{r}\left(t\right) \in \mathcal{D}_{\mathbf{r}}$.

$$\begin{aligned}\nabla_t V\left(\mathbf{X}(t)\right) &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\sum_{i\in\mathcal{N},\,k\in\mathcal{K}} x_{ik}(t)\right]^2 = 2\left[\sum_{i\in\mathcal{N},\,k\in\mathcal{K}} x_{ik}(t)\right]\left[\sum_{i\in\mathcal{N},\,k\in\mathcal{K}} \nabla_t x_{ik}(t)\right] \\ &=2\left[\sum_{i\in\mathcal{N},\,k\in\mathcal{K}} x_{ik}(t)\right]\left\{\sum_{i\in\mathcal{N},\,k\in\mathcal{K}} x_{ik}(t)\left[U_{ik}\left(x_{ik}(t),\mathbf{x}_{i_{-k}}(t), r_i(t)\right)-\overline{U}_k\left(\mathbf{x}_k(t),\mathbf{X}_{-k}(t),\mathbf{r}(t)\right)\right]\right\}\le 0.\end{aligned} \tag{18}$$

According to these conditions, we can state that the obtained strategy of the sensing cloud together with the obtained users' sensing cloud rental strategies constitute the Stackelberg equilibrium. ∎

V. Performance Evaluation

In this section, we conduct extensive numerical studies to evaluate the performance of the players in the Stackelberg differential game. For ease of illustration, we consider a crowdsensing platform containing 5 sensing clouds and 3 users, i.e., $N = 5$ and $K = 3$, over the time space of $\mathcal{T} = [0, 20]$. The vector of the distances between the access point connected to the blockchain and sensing clouds is $\mathbf{d} = [5.1, 5.2, 5.3, 5.4, 5.5]$ while the vector of the path loss coefficients is $\boldsymbol{\alpha} = [2.6, 2.7, 2.8, 2.9, 3]$ and the noise is $\sigma^2 = 1$. Moreover, the vector of the channel gains is $\boldsymbol{g} = [7.65, 7.8, 7.95, 8.1, 8.25]$ and the vector of the bandwidths is $\boldsymbol{h} = [40.1, 40.2, 40.3, 40.4, 40.5]$. The price of the power received by the sensing clouds is $\phi = 0.01$, the vector of the distance between the RF-energy beacons and sensing clouds is $\mathbf{d}^{\mathrm{t}} = [d_i^{\mathrm{t}}] = [1.1, 1.2, 1.3, 1.4, 1.5]$, and the path-loss exponent of wireless power transfer is $\eta = 2$. Without loss of generality, the quadratic functional form for the energy consumption of the sharded blockchain, i.e., $\Upsilon(\cdot)$, is $\Upsilon(z) = az^2 + bz + c$, where the coefficients are $a = 0.01$, $b = 0.01$, and $c = 0.01$. The coefficient describing the linear relationship between the computational power of the sharded blockchain and its supported total transmission rate $m = 0.1$. For the utility function $\Upsilon(z)$, we adopt $\Psi_{ik}(z) = A_{ik} - B_{ik} \times exp(-C_{ik} z)$ from [7]. The coefficient matrices of the users' utility functions are $\mathbf{A} = \mathbf{B} = \mathbf{C} = [3.2, 3.5, 3.3; 3.5, 3.4, 3.4; 3.4, 3.3, 3.5; 3.1, 3.1, 3.1; 3.3, 3.2, 3.2]$. The initial strategies of the users' are $\mathbf{x}^0{}_k = [0.325, 0.175, 0.275, 0.125, 0.1]$, $\forall k \in \mathcal{K}$. The vector of the budget is $\mathbf{w} = [5.1, 5.2, 5.3]$.

A. Numerical Results

1) Demonstration of the stability of evolutionary equilibrium: The stability of the evolutionary equilibrium can be verified by using the direction fields of the replicator dynamics. We use the evolutionary process of the users' sensing cloud rental strategies on sensing clouds 1 and 3 as examples. The direction fields between x_{21} and x_{11}, x_{31} and x_{11} are shown in Figs. 2(a) and (b), respectively. As shown in Fig. 2, any unstable strategy will follow the blue arrows to reach its corresponding evolutionary equilibrium, which is marked by the black circle. This demonstrates the convergence and stability of the evolutionary equilibrium and is consistent with Theorem 3.

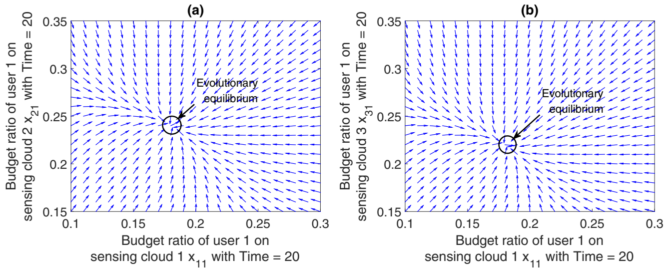

Figure 2: Direction Field of the replicator dynamics showing the stability of the evolutionary equilibrium

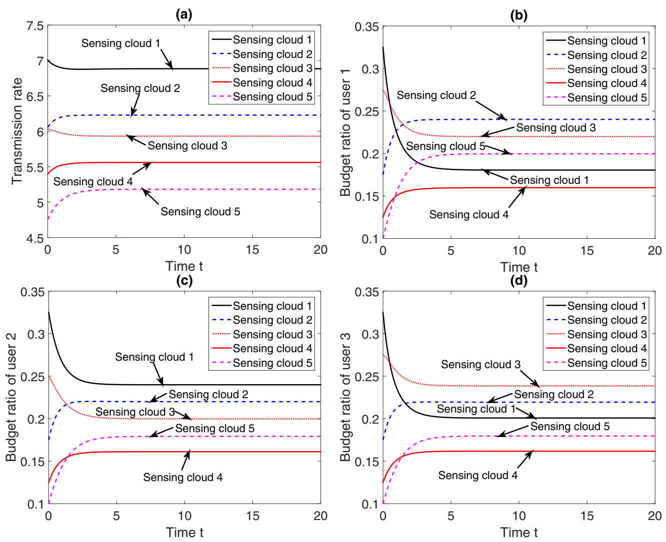

Figure 3: Evolutionary Equilibrium

2) Dynamic strategies: We next investigate the dynamic strategies of the lower-level evolutionary subgame from the initial strategies, i.e., $\mathbf{x}^0{}_k = [0.325, 0.175, 0.275, 0.125, 0.1]$, $\forall k \in \mathcal{K}$. The dynamic strategies of the lower-level evolutionary subgame are subject to the control of the sensing clouds' transmission rate strategies. The dynamic change of the strategies in the lower-level evolutionary subgame indicates the adaptation of the users' sensing cloud rental strategies. As shown in Figs. 3(b), (c), and (d), the strategies of the lower-level evolutionary subgame converge to the evolutionary equilibrium at which none of the users wants to change their strategies unilaterally. Although the sensing cloud 1's initial transmission rate, i.e., r_1, is at a high value due to the shortest distance between sensing cloud 1 and the access point connected to the blockchain, i.e., $d_1 = 5.1$, as well as the smallest path loss coefficient, i.e., $\alpha_1 = 2.6$, among these five sensing clouds, the budget ratios on sensing cloud 1 of both users 1 and 3, i.e., x_{11} and x_{13}, decrease from the initial large values to the small values at the evolutionary equilibrium. The reason is that the users' preferences of the sensing cloud depend not only on their data transmission rate but also on their data type. As a result, the profit of sensing cloud 1 decreases. In this case, sensing cloud 1 decreases its transmission rate to lower its cost and hence achieve its cumulative profit maximization. In contrast, due to a small initial transmission rate of sensing cloud 2, i.e., r_2, as shown in Fig. 3(b) and the users' preference of its data type, sensing cloud 2 cannot provide the data optimally. In this case, the transmission rate of sensing cloud 2 increases from the initial value to the larger value at the evolutionary equilibrium. Accordingly, all the rational users' budget ratios on sensing cloud 2, i.e., x_{21}, x_{22}, and x_{23}, increase from the initial small values to the large values at the evolutionary equilibrium as shown in Figs. 3(b), (c), and (d).

VI. Conclusion

We have presented a hierarchical dynamic game framework to analyze the strategies of the sensing clouds and the users in the crowdsensing market under the permissionless blockchain as a decentralized but trusted trading platform. The interaction among the sensing clouds, the blockchain, and the users has been modeled as a two-stage Stackelberg differential game. Specifically, the dynamics of the users' sensor rental strategies are captured by the replicator dynamics in the lower-level evolutionary subgame. The stability and uniqueness of the evolutionary equilibrium have been validated both analytically and numerically. The stable and unique evolutionary equilibrium has been obtained as the solution to the lower-level evolutionary subgame. Taking into account the lower-level evolutionary subgame, the transmission rate determination problem of the sensing cloud has been formulated as a Stackelberg differential game. Moreover, we have provided a series of insightful analytical and numerical results on the equilibrium of the game. For the future work, we will further include the blockchain as a player.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government under Grant 2014R1A5A1011478, WASP/NTU M4082187 (4080), Singapore MOE Tier 1 under Grant 2017-T1-002-007 RG122/17, MOE Tier 2 under Grant MOE2014-T2-2-015 ARC4/15, NRF2015-NRF-ISF001-2277, and EMA Energy Resilience under Grant NRF2017EWT-EP003-041.

References

[1] "Gartner identifies top 10 strategic iot technologies and trends," https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends, Nov 2018.

[2] W. Wang, D. T. Hoang, Z. Xiong, D. Niyato, P. Wang, P. Hu and Y. Wen, "A survey on consensus mechanisms and mining management in blockchain networks," *arXiv preprint arXiv:1805.02707*, 2018.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Self-published Paper*, May 2008.

[4] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002.

[5] S. Feng, W. Wang, D. Niyato, D. I. Kim and P. Wang, "Competitive data trading in wireless-powered internet of things (iot) crowdsensing systems with blockchain," *arXiv preprint arXiv:1808.10217*, 2018.

[6] C. D. Clack, V. A. Bakshi and L. Braine, "Smart contract templates: foundations, design landscape and research directions," *arXiv preprint arXiv:1608.00771*, 2016.

[7] B. Gu, F. Hu and H. Liu, "Modelling classification performance for large data sets," in *International Conference on Web-Age Information Management*. Springer, 2001, pp. 317–328.

[8] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 17–30.

[9] L. Bahri and S. Girdzijauskas, "When trust saves enegry-a reference framework for proof-of-trust (pot) blockchains," in *The Web Conference 2018*. ACM Digital Library, 2018, pp. 1165–1169.

[10] R. Teodorescu and J. Torrellas, "Variation-aware application scheduling and power management for chip multiprocessors," in *ACM SIGARCH computer architecture news*. IEEE Computer Society, 2008, vol. 36, pp. 363–374.

[11] F. Baccelli, B. Błaszczyszyn et al., "Stochastic geometry and wireless networks: Volume ii applications," *Foundations and Trends® in Networking*, vol. 4, no. 1–2, pp. 1–312, 2010.

[12] W. H. Sandholm, "Potential games with continuous player sets," *Journal of Economic theory*, vol. 97, no. 1, pp. 81–108, 2001.

[13] E. of Mathematics, "Cauchy-lipschitz theorem," https://www.encyclopediaofmath.org/index.php/Cauchy-Lipschitz_theorem.

[14] S. Sastry, "Lyapunov stability theory," in *Nonlinear Systems*, pp. 182–234. Springer, 1999.

[15] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *Journal of Optimization Theory and Applications*, vol. 3, no. 3, pp. 184–206, 1969.