 doi: 10.1109/JIOT.2019.2899949

DVSP: Dynamic Virtual Sensor Provisioning in Sensor-Cloud based Internet of Things

Tamoghna Ojha, *Student Member, IEEE,* Sudip Misra, *Senior Member, IEEE,* Narendra Singh Raghuwanshi, and Hitesh Poddar

***Abstract*— Virtual sensor provisioning is an essential process in sensor-cloud based Internet of Things (IoT), and it is responsible for efficient utilization of physical resources in the system. However, the existing schemes for virtual sensor provisioning do not provide an optimal solution while considering overall demand of multiple users/services. As a result, redundant sensor nodes are provisioned, which leads to increased energy consumption and reduced network lifetime. In this paper, we present a dynamic virtual sensor provisioning scheme (DVSP) for sensor-cloud based IoT applications to maintain the energy-efficiency of the deployed physical sensor nodes while maintaining the QoS of the service requests. We model the interaction between the Cloud Service Provider (CSP) and the Sensor owners (SOs) using the Single-Leader Multi-Follower Stackelberg game. The players of the game exploit the spatial correlation among the on-field sensor nodes, and consequently, the *oligopoly* created between the players is dynamically updated. We show the existence of a *Stackelberg-Nash-Cournot* equilibrium in the game. We evaluated the performance of the proposed scheme through extensive simulations. The results depict improvement in the energy-efficiency of the nodes as well as increase in the lifetime of the deployed on-fields sensors in the proposed scheme compared to benchmark schemes. We also plot the average number of Quality of Service (QoS) violations in each iteration for the user requests.**

***Index Terms*—Virtual sensor, energy-efficiency, sensor-cloud, game theory**

I. Introduction

Internet of Things (IoT) is a futuristic paradigm which enables connectivity between any type of devices [1]–[3]. In this regard, the sensor-cloud framework is envisioned to provide a scalable architecture to manage this ecosystem of enormous number of sensing devices [4]. The sensor-cloud offers a collaboration of the service provides (such as CSP and SOs) and the users. The CSP and SOs provide services while gaining economical benefit in terms of price charged to end-users. A real-life example of such framework is weather service. A service provider, for example AccuWeather (https://www.accuweather.com/), provides weather services to end-users by utilizing the cloud services (such as Amazon Web Services) and the weather stations deployed by National Oceanic Atmospheric Administration (NOAA) or private TV stations. Thus, NOAA is one SO in this example. The advent of sensor-cloud framework empowers various application domains providing numerous advantages compared to the traditional Wireless Sensor Network (WSN) based infrastructure [5]–[9]. It enhances the real-time information processing and storage with the cloud-based framework where the on-field nodes are deployed covering a vast geographical area. The sensor-cloud architecture facilitates dynamic access and resource management of the physical sensory resources by providing a virtualized interface between the end-users and the sensory resources. In this infrastructure, using the technique of virtual sensor provisioning, these physical sensors are accessed by end-users using various different services offered by the CSP. Therefore, based on all these features, the sensor-cloud system provides a multi-user multi-application environment for designing decision support systems. Few potential applications of the sensor-cloud architecture to name in different domains are health-care, precision agriculture, environmental monitoring, and military.

The initial works [5], [10], [11] on sensor-cloud systems focused on defining the components of the infrastructure and middlewares (SenseWrap, ZeroConf) for enabling virtualization. Madria *et al.* [7] presented an architecture for sensor-cloud systems, which defines different parts of the protocol stack and interconnections with physical sensors as well as users. In another work, Misra *et al.* [8] presented the theoretical modeling of sensor-cloud, including the mathematical formulation of sensor virtualization. Typically, the sensor-cloud system framework consists of three layers – *client-centric*, *middleware*, and *sensor-centric*. The client-centric layer acts as the interaction layer between the users and the services offered by the sensor-cloud. The services running in the cloud request the middleware for specific resources from the deployed sensor networks. Thereby, the middleware performs the task of sensor virtualization, virtual sensor provisioning and maintenance. It also manages the accounts of different users and performs billing services. The registration (for new sensor owners or sensors) and maintenance of physical sensor network is performed by the sensor-centric layer. It is also responsible for the information routing by the deployed sensor nodes. The middleware reviews the received user queries, and resolves the query to find the physical sensors according to the queries. Therefore, after successfully finding the required physical sensor(s), the middleware creates a virtual sensor for the time period.

The existing schemes for virtual sensor provisioning consider activating redundant nodes (such as [12], [13]) or higher number of nodes (such as [14]) while considering the demand from the users. However, this technique is not energy-efficient for the deployed physical sensor nodes, as the nodes need to update their sensed information to the cloud periodically. Also, additional number of selection results in increased energy consumption. As a result, the lifetime of the deployed network will be reduced significantly. Additionally, increased energy consumption of the on-field nodes incurs additional cost of maintenance to either CSP or to the SO. Consequently, the usage price for the end-users also increases. Thus, considering the sensor nodes to be resource constrained, the objective is to minimize the energy consumption of these nodes to the extent possible, while maintaining the Quality of Service (QoS) of the running services. Here, the QoS of the requested services are *sensor node's availability* and *sensing*

T. Ojha, S. Misra and N. S. Raghuwanshi are with Indian Institute of Technology Kharagpur, WB, 721302 India. (e-mail: {tojha, smisra}@sit.iitkgp.ernet.in, nsr@agfe.iitkgp.ernet.in). H. Poddar is with Vellore Institute of Technology, Vellore, India (e-mail: hpoddar06@gmail.com)

information quality.

In reality, sensor readings exhibit correlation in temporal as well as spatial domain [15]. Spatial correlation refers to correlation among the sensor readings between two sensors placed at any particular distance. Similarly, temporal correlation refers to the correlation between the readings of a sensor at two different time instances. In on-field sensor deployment by multiple SOs, the correlation in spatial domain results in redundant nodes spatially distributed throughout different SOs. Thus, in any periodic data collection scheme governed by the CSP, there exists redundant information in each iteration of transmission. Consequently, the energy consumption of the deployed nodes increases, which, in turn, reduces the overall network lifetime. Similarly, it is evident that there will be multiple queries from various users to the sensor-cloud for various physical sensor nodes. Therefore, the provisioning manager has an opportunity to optimize the allocation of physical sensors to the virtual sensors while maintaining QoS demands of all the service requests from different users. Thus, in this situation, SOs and CSP, both suffer from the same problem – minimization of the energy consumption of individual nodes to enhance the overall network lifetime. Motivated by this problem, in this paper, we devise a dynamic virtual sensor provisioning scheme to optimize the selection of the deployed physical sensor nodes to the virtual sensors.

In this paper, we present a *dynamic virtual sensor provisioning scheme* (*DVSP*) for sensor-cloud applications to maintain the energy-efficiency of the deployed physical sensor nodes while maintaining the service requirements of the users. In the proposed scheme, we exploit the spatial correlation among the deployed nodes throughout various SOs. The CSP's decision to select a node depends on this parameter. Also, the middleware, which is responsible for provisioning of the sensor nodes, considers the QoS demands of each of the running services – *node availability* and *sensing information quality*. We model the interaction of the CSP and multiple SOs as a *Single-Leader Multi-Follower Stackelberg game* [16], where the CSP is the *single leader* and the SOs are *multiple followers*. In this oligopolistic environment, the CSP is referred to as the *Stackelberg firm*, which dynamically provisions sensor nodes for serving the user's QoS demand, while maintaining the energy-efficiency of the deployed nodes. As a result, optimal number of active nodes are decided at different time-instants. Consequently, the actual topology of the deployed nodes changes, leading to incomplete connectivity among the deployed nodes and their corresponding gateway. The SOs, or the *Cournot firms*, on the other hand, set their objective to fix the connectivity of these active nodes selected by the CSP. Thus, any SO optimizes the selection of additional nodes to minimize its overall energy consumption, while offering uninterrupted connectivity among the nodes. In sum, our specific *contributions* in this work are as follows.

- We frame the interaction between a CSP and the SOs as a *Single-Leader Multi-Follower Stackelberg game*. This game model articulates dynamic virtual sensor provisioning in the sensor-cloud systems.
- We devise a model for the CSP to find the optimal set of nodes to activate at any given time, while maintaining the QoS requirements of the requesting services and minimizing the overall network energy consumption. This fabric exploits the spatial correlation among the deployed nodes to enforce energy-efficiency throughout the network.
- We present a dynamic topology control model for the SOs enabling cost-effective selection of deployed nodes with the objective to construct a coherent topology.

The rest of the paper is organized as follows. We briefly review the existing literature in the area of sensor-cloud systems, and discuss in Section II. The proposed system architecture is depicted in Section III. Section IV presents the proposed game theory based scheme with algorithms for both CSP and SOs. The performance evaluation of the proposed scheme is presented in Section V. Finally, we conclude the paper in Section VI indicating few future research direction.

II. Related Works

The concept of sensor virtualization is the heart of the sensor-cloud system. In several recent works, the authors discussed various components of the sensor-cloud framework. For example, Madria *et al.* [7] discussed the various components of a three-layer protocol stack for supporting a sensor-cloud system. In another work, a theoretical model for sensor virtualization in sensor-cloud system was proposed by Misra *et al.* [8]. The authors discuss the composition of various components in the framework, and present a detailed performance evaluation with respect to various metrics. Abdelwahab *et al.* [17] proposed a cloud of things framework for distributed sensing resource discovery and in-network processing of the sensed data. The cloud of things platform virtualizes the deployed sensing resources, and thereby enhance the resource utilization by offering sensing-as-a-service.

Dinh *et al.* [18] proposed an information-centric model for sensor-cloud which helps in decoupling of *information producers* (*IPDs*) or the *physical sensors* and *information providers* (*IPVs*) or the virtual sensors. This decoupling helps in minimizing the energy consumption of the IPDs by keeping themselves in sleep mode while IPVs are able to provide the IPD data by predicting their values. In this way, the model provides a trade-off between the data accuracy requirement of the applications and energy efficiency of the sensor nodes. The issue of data delivery in sensor-cloud was studied by Zhu *et al.* [19]. In this work, the authors propose a Multi-Modal Data Delivery (MMDD) approach which covers four different types of data delivery – *cloud to subscribers*, *sensor network to subscribers*, *subscriber to subscriber*, and *cloudlet to subscribers*. A sensor-cloud architecture, named *Mils-Cloud* for military applications was proposed by Misra *et al.* [20]. Mils-Cloud facilitates integration of the military tri-services with the sensor-cloud framework, and thereby increasing the cooperation among the military units for decision making.

To enhance the energy-efficiency of the deployed sensor nodes, Ojha *et al.* [21] proposed a dynamic duty scheduling framework in a sensor-cloud system. The authors show how dynamic duty selection for on-field sensor nodes can prolong the lifetime of the overall network. However, this work did not consider the QoS for multiple service requests. In the context of WSNs, a virtual sensing framework was proposed by Sarkar *et al.* [22]. This framework presents a prediction based scheme, which uses the concepts of temporal and spatial correlation, to reduce a sensor node's sensing and communication tasks. Thereby, reducing the energy consumption of the nodes. However, the work did not consider the presence of a sensor-cloud system where multiple SOs can co-exist.

Kothari *et al.* [23] presented a Data Quality (DQ) aware Sensor-cloud (DQS-Cloud), and discussed the various components of the architecture. This architecture facilitates the customers to discover DQ-aware sensor services, and then, allows the customers to get the best sensor feed in terms of content and quality. The authors also present a technique for enabling DQ-aware fault-tolerant service availability. In [24], Lawson *et al.* discussed the trade-off between DQ and energy-efficiency in sensor-cloud services. In their work, the proposed cloud based architecture ranks the feeds according to DQ, and later assigns them to service requesting customers according to their need.

Chatterjee *et al.* [12] proposed a scheme for optimal composition of a virtual sensor from a set of physical sensor nodes. In this work, a node is provisioned for the virtual sensor if it satisfies certain level of *goodness*, and consequently, the total number of nodes are minimized for any particular application. However, the sensor selection scheme did not exploit inter-node correlation, and therefore, might select redundant nodes too. [13] presented an adaptive data caching scheme to achieve efficiency of sensor energy consumption and network lifetime in sensor-cloud system. Using this scheme, an optimal caching interval is decided, and nodes transmit new data after that time. This scheme is dynamic to the change in the physical sensor network. However, this scheme also did not consider the information similarity and inter-node correlation. Therefore, the energy consumption of the network can be optimized further. In [14], the authors proposed a middleware which aggregates the user requests, and consequently, minimizes the number of queries to the physical sensor nodes. However, these reduced number of queries are forwarded to all the nodes, and accordingly all nodes change their transmission interval.

[25], [26] presented virtual sensor provisioning by selecting sensors based on similarity of heterogeneous sensors. In this way, the overall energy consumption of the nodes reduces and network lifetime enhances. The node selection is done by similarity of measurements between nodes, and not just by the distance between the nodes. However, the authors did not consider the presence of node deployments by multiple SOs. Also, the scheme did not present the information routing in a large-scale multi-hop deployment. In our proposed scheme, we consider selection of an optimal set of nodes while serving multiple user's query considering their QoS.

A smart parking solution to enable the traffic officers find parking violations quickly and efficiently was proposed by Dinh *et al.* [27]. In contrast to the earlier studies, which focus mainly on finding parking locations for drivers, the authors in [27] propose a location-centric smart parking violation system using IoT-cloud framework. This system facilitates the government officials to maximize the fine collected by issuing tickets to parking violators, while minimizing the travel cost for the officials. In another work, Dinh *et al.* [28] presented an efficient interactive model for enabling the multiple services with different requirements to work in an sensor-cloud framework. The proposed system intelligently aggregates the requests from various services to optimize the workload, traffic bandwidth, and resource requirements for the physical nodes. An on-demand interactive sensing model for sensor-cloud was presented in [29]. The proposed scheme minimizes the energy consumption of the deployed sensor nodes by on-demand location-based sensor information collection method. The framework also facilitates the users with custom settings of sensing service quality.

Synthesis: It is evident from the existing works that the provisioning schemes activate redundant nodes (such as [12], [13]) or higher number of nodes (such as [14]). Thus, the issue of maintaining the energy-efficiency of the deployed nodes require further attention, specifically in a multi-SO deployment environment. In Table I, we present the related works in different categories.

TABLE I: Categories of the related works

Main focus	**Related works**
Architecture, Protocol stack	[7], [8], [17], [20]
Energy-efficiency	[18], [21], [12], [13], [25], [26]
Request/Data aggregation	[19], [22], [14], [28]
Data quality	[23], [24]

III. Proposed System Architecture

We consider a sensor-cloud system, consisting of $|N|$ number of sensor nodes partitioned among m SOs, serviced by a CSP χ. The sensors are deployed in a 2D area for periodic monitoring of specific events. Here, $\Theta = \{\theta_1, \theta_2, \cdots, \theta_m\}$ denotes the set of all SOs, where $m = |\Theta|$. The nodes associated with any SO θ_i are denoted as N_θ^i. We consider each SO to have separate gateways ($G_i \in \mathcal{G}$) to offer connectivity between its on-field nodes ($\forall j \in N_\theta^i$) and the CSP χ. The set of neighbors for any node $j \in N_\theta^i$ at any time t is denoted as $Nbr_\theta(i,j,t)$, such that any node present in $Nbr_\theta(i,j,t)$ is present in N_θ^i. In other words, $Nbr_\theta(i,j,t) \subseteq N_\theta^i$. On the other hand, the CSP also computes the neighbors of a node, regardless of the SO. We denote this function as $Nbr_\chi(j,t)$. Here, $Nbr_\chi(j,t) = \sum_{\forall \theta_i \in \Theta} Nbr_\theta(i,j,t)$.

Typically, any general user requests for the sensed information of any location. The user query (s_k) mentions the location and the required QoS for the required sensed information. The CSP supports two QoS parameters – *node availability* ($NA(i,t)$) and *sensing information quality* ($\varrho_{th}^{s_k}$). *Node availability* ($NA(\cdot)$) refers to the percentage of up-time for a node. It is defined as, $NA(i,t) = (100 * t_i^{life})/t$, where i is the node id, t is the total time elapsed, and t_i^{life} is the up-time of node i. Whereas, we define *sensing information quality* as inversely proportional to the distance ($\bar{d}_j$) between the event's location and sensor's location, and the distance threshold for service s_k ($d_{s_k}^{th}$). Therefore, $\varrho_j^{s_k} = (1 - \bar{d}_j/d_{s_k}^{th})$. However, the parameter *sensing information quality* can be modeled using multiple parameters such as accuracy, frequency, freshness, validity [23], [24]. Also, these parameters can be user defined for different applications. The CSP, on the other hand, collates all such received requests ($\forall s_k \in \mathcal{S}$), and finds the nodes which can serve to each demand. Here, $\mathcal{S}$ is the set of all services offered by the sensor-cloud. For example, node j is selected if $\varrho_j^{s_k} \geq \varrho_{th}^{s_k}$). All such nodes are added to the set of possible nodes (N_ω) to consider for that iteration. Here, $\varrho_j^{s_k}$ denotes the quality offered by this node relating to the query s_k. This parameter ($\varrho_j^{s_k}$) can be computed by exploiting the correlation between nodes.

The proposed system architecture is shown in Figure 1. In this figure, we depict the scenario where the active nodes are connected to different gateways provided by the SOs. Each SO, in addition to the active nodes, activates few additional nodes which help in providing the connectivity between the active nodes and the gateway. In the following sections, we describe the procedure for the selection of active nodes and the additional nodes in detail.

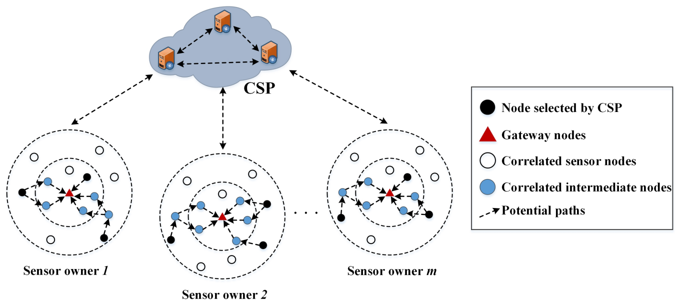

Fig. 1: The proposed system architecture

IV. DVSP: Dynamic Virtual Sensor Provisioning

In DVSP, we model the interaction between the CSP and the SOs using the *Single-Leader Multi-Follower Stackelberg game* model [16]. In the sensor-cloud framework, the interaction between the CSP and SOs creates an *oligopoly*. In this scenario, the CSP (or the *leader*) takes the first decision, and accordingly the decision of SOs (or the *followers*) change. Thereafter, the decisions of SOs are also considered by the CSP in further decision making. In our proposed scheme, the choice of applying the *Single-Leader Multi-Follower Stackelberg game* model is motivated by such observation. For example, in the proposed scheme, the CSP initially selects optimal set of nodes which minimizes the energy consumption of the nodes while maintaining the QoS of the requesting applications. We name these nodes as CSP selected Duty Node (CDN). However, as a result of this selection, the topology of the deployed nodes changes, and creates incomplete connectivity between the selected nodes and the corresponding gateway. Therefore, the information collection from the on-field nodes may be disrupted. To mitigate this problem, each SO selects an optimal set of nodes for connectivity maintenance. These nodes are named as Optimal Connectivity Nodes (OCNs). This decision by SOs are considered by the CSP in further decision making. The individual decision making process for CSP and SOs are facilitated by computing the *utility* for any decision. The CSP and SOs exchange the information about the selected nodes between themselves using standard APIs defined in the sensor-cloud framework, where the sensors are referred by their unique registration ID with the CSP [7]. Also, in the proposed model, we have considered that the information of activation is communicated to the physical sensors by the corresponding SOs. Please see the Appendix A for a supplementary material which depicts the overall process followed in the proposed scheme.

A. Utility for the CSP

In the following, we mathematically define the rules for utility computation of the CSP.

Definition 1. ***Potential*** *($P(j, \varrho_j^{s_k}, \omega)$) of a node $j \in N$ is denoted by the number of service requests ($|N_s^k|$) for which this node can be selected in that iteration (ω) – having higher value of sensed information quality $\varrho_j^{s_k} \geq \varrho_{th}^{s_k}$. Mathematically,*

$$P(j, \varrho_j^{s_k}, \omega) = |N_s^k| \qquad \forall \varrho_j^{s_k} \geq \varrho_{th}^{s_k}, \forall \theta_i \in \Theta \tag{1}$$

As explained in Section III, to calculate the node potential for each node, the CSP exploits the spatial correlation among the nodes of different SOs.

(i) *Node Potential* (*NP*): For the CSP, selecting a node with higher *potential* minimizes the number of duty nodes for any iteration. Thus, the utility of the CSP ($\mathcal{U}_\chi(\cdot)$) is considered to be increasing with the selection of a node with higher NP.

$$\frac{\delta \mathcal{U}_\chi(j, \omega)}{\delta P(j, \varrho_j^{s_k}, \omega)} \geq 0 \tag{2}$$

Definition 2. *Previous selection (PS_j^ω) of any node ($j \in N$) at iteration ω refers to the number of times the node was activated either by CSP (χ) or by its SO ($\theta_i; \forall j \in N_\theta^i$) till iteration ($\omega - 1$).*

(ii) *Previous Selection* (*PS*): As discussed in Definition 2, this parameter is a counter that helps the CSP determine the number of times any particular node $j \in N$ was selected for transmission. Consequently, the utility of the CSP ($\mathcal{U}_\chi(\cdot)$) is considered to be non-increasing with respect to the selection of nodes with higher PS count.

$$\frac{\delta \mathcal{U}_\chi(j, \omega)}{\delta PS_j^\omega} \leq 0 \tag{3}$$

Definition 3. *Consecutive selection (CS_j^ω) of any node ($j \in N$) at iteration ω refers to the number of* ***consecutive iterations*** *the node was activated either by CSP (χ) or by its SO ($\theta_i; \forall j \in N_\theta^i$) till iteration ($\omega - 1$). For example, $CS_j^\omega = k$, iff, node j was selected in all iterations in between $\omega - k$ to $\omega - 1$.*

(iii) *Consecutive Selection* (*CS*): The CS value of a node reflects the information based on which nodes are selected in the recent past iterations. It is straightforward to infer that with the selection of nodes with higher CS value, the overall network's energy consumption becomes higher, and these few selected nodes become prone to quicker energy depletion. Thus, the utility of the CSP ($\mathcal{U}_\chi(\cdot)$) is non-increasing with respect to the increase in CS value of any selected node. Hence,

$$\frac{\delta \mathcal{U}_\chi(j, \omega)}{\delta CS_j^\omega} \leq 0 \tag{4}$$

Therefore, from Equations (2), (3) and (4), the overall utility for the CSP is formulated as,

$$\mathcal{U}_\chi(j, \omega) = w_1 \times \frac{P(j, \varrho_j^{s_k}, \omega)}{\sum_{\forall j \in N} P(j, \varrho_j^{s_k}, \omega)} + w_2 \times \left(1 - \frac{PS_j^\omega}{\omega - 1}\right) + w_3 \times \left(1 - \frac{CS_j^\omega}{CS_{th}}\right) \tag{5}$$

where w_1, w_2, w_3 are the weight factors for each of the three parameters, i.e., $P(j, \varrho_j^{s_k}, \omega)$, PS_j^ω, CS_j^ω, for the utility calculation. CS_{th} refers to a threshold value, which limits the maximum allowable CS value for any node. These parameters can be user-defined.

B. Utility for the SOs

After the node selection by the CSP, each SO selects a set of nodes which ensure connectivity between the CSP selected nodes and the specific gateway.

(i) *Previous Selection* (*PS*): This parameter has the same functionality for both CSP and SOs. Therefore,

$$\frac{\delta \mathcal{U}_{\theta_i}(\lambda_{p,i}, \omega)}{\delta \sum_{s \in \lambda_{p,i}} PS_s^\omega} \leq 0 \qquad p > 0, \exists \lambda_{p,i} \tag{6}$$

(ii) *Consecutive Selection* (*CS*): CS also offers the same functionality for both the CSP and the SOs. Therefore,

$$\frac{\delta \mathcal{U}_{\theta_i}(\lambda_{p,i}, \omega)}{\delta \sum_{s \in \lambda_{p,i}} CS_s^{\omega}} \leq 0 \qquad p > 0, \exists \lambda_{p,i} \tag{7}$$

Definition 4. *Let $\lambda_p(j,k)$ be the p^{th} path between node j to k consisting of the set of intermediate nodes, along with the starting and destination nodes (e.g. j and k). Clearly, multiple such paths ($p > 0$) may exist for any specific j,k. Also, $|\lambda_p(j,k)| - 2$ denotes the number of intermediate nodes present in any path (2 subtracted for the start and end node). We also use $\lambda_{p,i}$ to denote any path which connects any two nodes, say j and k, such that $j,k \in N_\theta^i$. Thus, $\lambda_{p,i} \simeq \lambda_p(j,k)$, $\forall j,k \in N_\theta^i$. In the following, we use $\lambda_{p,i}$ and $\lambda_p(j,k)$ interchangeably.*

(iii) *Number of Hops*: This parameter helps a SO to select the most optimal path, in terms of the number of hops, between a duty node (k) and the gateway (G_i). Such selection of a path having minimum number of intermediate nodes also indirectly minimizes the overall network energy consumption. Hence, the utility ($\mathcal{U}_{\theta_i}(\cdot)$) of a SO is non-increasing with the selection of a path, between any duty node and the gateway, with increased hop-count. Thus,

$$\frac{\delta \mathcal{U}_{\theta_i}(\lambda_{p,i}, \omega)}{\delta |\lambda_{p,i}|} \leq 0 \qquad p > 0, \exists \lambda_{p,i} \tag{8}$$

From Equations (6), (7) and (8), we formulate the overall utility for a SO as,

$$\mathcal{U}_{\theta_i}(\lambda_{p,i}, \omega) = \bar{w}_1 \times \left(1 - \frac{1}{|\lambda_{p,i}|} \sum_{s \in \lambda_{p,i}} \frac{PS_s^{\omega}}{\omega - 1}\right) + \bar{w}_2 \times \left(1 - \frac{1}{|\lambda_{p,i}|} \sum_{s \in \lambda_{p,i}} \frac{CS_s^{\omega}}{CS_{th}}\right) + \bar{w}_3 \times \left(1 - \frac{|\lambda_{p,i}| - 2}{\sum_p (|\lambda_{p,i}| - 2)}\right) \tag{9}$$

C. Existence of Stackelberg-Nash-Cournot Equilibrium

In the equilibrium state, both the CSP and SOs cannot increase their individual utility values by merely changing their individual actions single-sidedly. In this case, the game achieves the Stackelberg-Nash-Cournot Equilibrium, when each of the SOs selects the optimal path ($\lambda_p(j, G_i)$) for each of the CDNs ($\forall j \in N_\theta^i$) available in their service area, and the following inequality holds.

$$\mathcal{U}_f(\lambda_{p,i}^*, \boldsymbol{\lambda}_{p,-i}^*) \geq \mathcal{U}_f(\lambda_{p,i}, \boldsymbol{\lambda}_{p,-i}^*) \qquad \forall \omega, f \in \Theta \tag{10}$$

where $\boldsymbol{\lambda}_{p,-i}^* = \{\lambda_{p,1}^*, \lambda_{p,2}^*, \cdots, \lambda_{p,i-1}^*, \lambda_{p,i+1}^*, \cdots, \lambda_{p,m}^*\}$.

Theorem 1. *The SOs achieve Stackelberg-Nash-Cournot Equilibrium in any iteration ω by selecting an optimal path ($\lambda_{p,i}^*$) with minimum number of nodes from $N_\theta^i \cup \mathcal{N}_{CSP}^{\omega}$, while the utility is maximized. Therefore,*

$$\frac{\Delta \mathcal{U}_f}{\Delta \lambda_{p,i}} = \frac{\mathcal{U}_f(\lambda_{p,i}^*) - \mathcal{U}_f(\lambda_{p,i})}{\lambda_{p,i}^* - \lambda_{p,i}} \leq 0 \qquad \forall \omega, f \in \Theta$$

Proof: Please see the Appendix B for a supplementary material for the detailed proof. ■

D. Duty Scheduling Models for the CSP and SOs

In Algorithms 1 and 2, we present the algorithm followed by the CSP and the SOs, respectively. In the oligopoly, the CSP acts first and the SOs follow thereafter. First, the CSP chooses $\mathcal{N}_{CSP}^{\omega}$, the duty nodes (CDNs) for any iteration ω. The rest of the active nodes or the OCNs, for any iteration are selected by the different SOs $\theta_i \in \Theta$. The CSP looks to optimize the total number of CDNs such that the overall network lifetime is enhanced. On the other hand, the SOs have the responsibility to set up the connectivity between any CDN $j \in (N_\theta^i \cup \mathcal{N}_{CSP}^{\omega})$ and its corresponding gateway G_i. At the same time, the SOs look to maximize their profit. Thus, each SO (θ_i) needs to guarantee an optimal connectivity between any duty node and the gateway.

In the proposed scheme, the objective of the CSP is to activate minimum number of nodes (as in CDNs) such that the activated nodes cover the whole area covered by all the nodes. Thus, in each iteration, each node ($\forall j \in N$) present in CDNs targets to minimize $|\mathcal{N}_{CSP}^{\omega} \cup N|$ while maximizing $\mathcal{U}_\chi(j, \omega)$.

On the other hand, the objective of any SO is to minimize the number of nodes selected in the OCNs to provide connectivity between the nodes in CDNs and the corresponding gateway. The SOs calculate the utility for all the possible paths between the nodes in CDNs and the gateway. Therefore, each SO has the objective to minimize $|\mathcal{N}_{\theta_i}^{\omega} \cup N_\theta^i|$, while maximizing its own utility ($\mathcal{U}_{\theta_i}(\lambda_{p,i}, \omega)$), for $p > 0$.

1) Algorithm for CSP: The CSP collates all the service requests from various users for each iteration. For each such request, the CSP resolves the query and finds the set of nodes (N_ω) which satisfies $\varrho_j^{s_k} \geq \varrho_{th}^{s_k}$, $\forall j \in N, s_k \in \mathcal{S}$. Thereafter, the utility for each node $j \in N_\omega$ are calculated. Consequently, the node (say l) with maximum utility value is selected as 'Transmitting', and marked as 'Visited'. All neighbors of l, which are also present in N_ω are also marked as visited. This process is repeated until all nodes of N_ω are 'Visited'. At the end, the set of nodes denoted as 'Transmitting' are the CDNs. Algorithm 1 outlines the steps followed by the CSP.

2) Algorithm for Sensor Owners: As discussed in Section IV-D, each SO needs to minimize the number of OCNs or the additional nodes to be activated ($\mathcal{N}_{\theta_i}^{\omega}$) for any iteration ω. The SO finds the utility for all the possible paths ($\lambda_p(j, G_i)$) from the duty nodes to its gateway. Thereafter, for each duty node j, an optimal path to gateway is computed such that the utility for the SO is maximized. The set of OCNs is populated with each such node present in the selected optimal path ($\lambda_p^*(j, G_i)$). The detailed steps of the process followed is explained in Algorithm 2.

Theorem 2. *The decision of any SO θ_i is based on zero conjecture variation, i.e., other SOs $\theta_x \in \Theta$ ($\theta_x \neq \theta_i$) hold their strategies as in the existing level.*

Proof: Please see the Appendix C for a supplementary material for the detailed proof. ■

V. Performance Evaluation

A. Simulation Settings

The evaluation of the proposed scheme was done using discrete event simulation in NS-3 (http://www.nsnam.org/). In the simulation, we consider a sensor-cloud framework consisting of 1 CSP and 4 SOs. Each SO has 50 nodes (N_θ^i) randomly deployed over

Algorithm 1: Algorithm for CSP

Inputs: N_θ^i, $Nbr_\theta(i,j,t)$, $Nbr_\chi(j,t)$, $N_\chi^{corr}(j,t,\varrho_j^{s_k})$, PS_j^ω, CS_j^ω.
Output: CSP selected Duty Nodes ($\mathcal{N}_{CSP}^\omega$).
for *each node* $j \in N$ **do**
  if $NA(j,t) \geq NA_{th}^{s_k}$ ***and*** $\varrho_j^{s_k} \geq \varrho_{th}^{s_k}$ **then**
    $N_\omega \longleftarrow N_\omega \bigcup j$;
for *each node* $j \in N_\omega$ **do**
  Compute Utility $\mathcal{U}_\chi(j,\omega)$ for iteration ω;
while $|Visited| < |N_\omega|$ **do**
  Select a node l such that $l \longleftarrow \arg\max_{l \in N_\omega} \mathcal{U}_\chi(\cdot)$;
  if *node* l *is NOT present in* $Visited$ **then**
    $Transmitting \longleftarrow Transmitting \bigcup \{l\}$;
    $Visited \longleftarrow Visited \bigcup \{l\}$;
    $Temp \longleftarrow Nbr_\chi(l,t) \cap N_\omega$;
    $Duplicates \longleftarrow Visited \bigcap Temp$;
    $Temp \longleftarrow Temp - Duplicates$;
    $Visited \longleftarrow Visited \bigcup Temp$;
  else
    Move to next node;
Update ω, PS_j^ω, CS_j^ω;
Return $Transmitting$;

Algorithm 2: Algorithm for any SO $\theta_i \in \Theta$

Inputs: ω, N_θ^i, $Nbr_\theta(i,j,t)$, PS_j^ω, CS_j^ω, $\mathcal{N}_{CSP}^\omega$.
Output: Optimal Connectivity Nodes ($\mathcal{N}_{\theta_i}^\omega$).
Find out duty nodes, $DutyNodes \longleftarrow N_\theta^i \bigcup \mathcal{N}_{CSP}^\omega$;
for *each node* $j \in DutyNodes$ **do**
  Find all paths ($\lambda_p(j,G_i)$) from j to Gateway G_i;
  for *each possible path* $\lambda_p(j,G_i)$ **do**
    Compute Utility $\mathcal{U}_{\theta_i}(\lambda_p,\omega)$ for iteration ω;
for *each node* $j \in DutyNodes$ **do**
  Find p^{th} optimal path between j and G_i, $\lambda_p^*(j,G_i) \longleftarrow \arg\max_{p>0} \mathcal{U}_{\theta_i}(\lambda_p,\omega)$;
  for *each node* $\bar{j} \in \lambda_p^*(j,G_i); \bar{j} \in N_\theta^i, \bar{j} \neq j$ **do**
    if $\bar{j}$ *is NOT present in* $IntrNodes$ **then**
      $IntrNodes \longleftarrow IntrNodes \bigcup \{\bar{j}\}$;
Update ω, PS_j^ω, CS_j^ω;
Return $IntrNodes$;

an area of 500 $m \times 500$ m. We consider equally weighted factors for utility calculation, thereby $\forall i \in \{1,2,3\}$, $w_i = \bar{w}_i = 0.33$. In the simulations, we randomly generate user queries requesting for different nodes with $\varrho_{th}^{s_k} = 0.9$. We use iteration number to generalize the results according to the occurrence of events rather than that of over time. In each iteration of the simulation, the users query for some of the nodes, which are randomly selected. Such distribution is used to reflect the random nature of the user queries in real scenarios. In Table II, we list all the simulation parameters.

TABLE II: Simulation Parameters

Parameter	Value
Number of nodes	200
Simulation area	500 m × 500 m
Transmission range of a sensor node (r)	100 m
Power for transmission, reception	24.75, 13.5 mW [30]
Data rate	40 $kbps$ [30]
Initial energy of a node	1 J
Number of iterations	300

B. Evaluation Metrics

The following performance metrics were used to study the performance of the proposed model. We briefly present their definition and relevance in performance evaluation of our proposed scheme.

- *Average number of nodes selected*: The average number of nodes selected in each iteration by either by the CSP or any SO. Using this metric, we can evaluate the optimality of the schemes in node selection through different iterations.
- *Average energy consumption*: The average energy consumed by any deployed node (from any SO) over the whole simulation time. This metric helps in evaluating the energy-efficiency of the schemes for the deployed nodes.
- *Network lifetime*: We measure the network lifetime as the percentage of remaining energy of the overall network. This metric helps us in evaluating the effectiveness of the schemes for a long term real on-field application.
- *Average number of QoS violations*: We measure the number of QoS violations, i.e., the number of requests not served, occurred in each iteration. This metric shows the possible chances of failure in the provisioning process.
- *Communication overhead*: This denotes the number of additional communication required for node selection, and shows the effectiveness of the provisioning process.

C. Benchmark

We compare the performance of our scheme with the 'flood' approach and the 'Pricing for Hardware' (pH) scheme [31]. In 'flood' scheme, each requested sensor node broadcasts its data to its neighbors and the neighbors again broadcasts the information to their neighbors, till the data reach the gateway of the corresponding SO. On the other hand, in 'pH', the information from the source node to the corresponding gateway is transmitted using multiple hops. In each hop, the next hop is selected from the neighbors of the current hop node. Thus, this scheme provides a comparatively finer selection of nodes from the deployed nodes. For all schemes, we consider similar simulation settings, and the users request the data of same nodes. The major difference between the proposed scheme and benchmarks is in the technique of selecting the nodes for information transmission between the source and gateway node. We plot the simulation results till the 300 iterations, as after this the nodes deplete energy for the 'flood' scheme.

D. Results and Analysis

1) Average number of nodes selected: We measured the average number of nodes selected in each iteration for all three schemes – DVSP, 'flood', and 'pH'. Figure 2(a) and 2(b) show the results for this metric in each individual iterations and cumulative

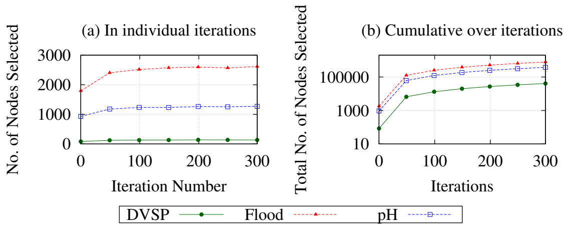

Fig. 2: Number of nodes selected

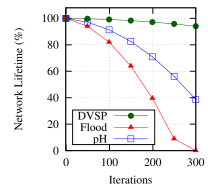

Fig. 4: Network lifetime

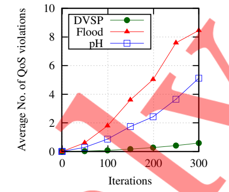

Fig. 5: Avg. no. of QoS violations

over the iterations, respectively. It is evident from the results that compared to both the benchmark schemes, DVSP activates 89.39% less number of deployed nodes in each iteration. In DVSP, the virtual sensor provisioning process is aided by the game-theoretic interaction between the CSP and SOs. Such interaction facilitates both CSP and SOs to optimally select nodes while considering the changes in the network. Also, in both 'flood' and 'pH', the process of node selection requires increased number of communication between the nodes. As a result, in the benchmark schemes, more number of nodes are selected in each iteration. Therefore, in any iteration, the average number of activated nodes in DVSP is significantly lower than both 'flood' and 'pH' schemes.

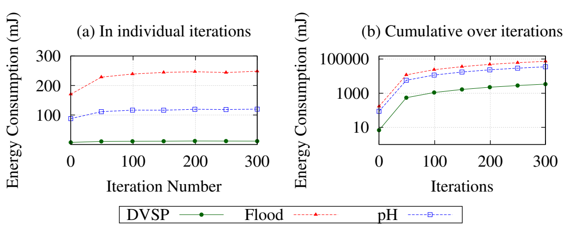

Fig. 3: Energy consumption

2) Average energy consumption: In Figure 3, we present the results for average energy consumption of the deployed nodes in the sensor-cloud system. Figure 3(a) presents the results for individual iterations, whereas, the results for the total energy consumption over cumulative iterations are presented in Figure 3(b). The results indicate that, on an average, the proposed DVSP scheme is 90.63% energy-efficient compared to the benchmark schemes. In all the schemes, the average energy consumption for any iteration is due to the communication occurred for activation of the deployed nodes. The benchmark schemes require 89.39% higher number of active deployed nodes, and thereby, require higher number of communications between the nodes. On the other hand, DVSP applies game-theoretic optimal node selection procedure, where the CSP and SOs select nodes according to the changed network conditions. Consequently, the number of activated nodes (both CDNs and OCNs) in any iteration is very few compared to that of the benchmarks. Due to this, in DVSP, the deployed nodes achieve energy-efficiency.

3) Network lifetime: In Figure 4, we depict the change in lifetime of the nodes with cumulative iterations. Here, we consider the network lifetime decrease due to the increase in energy consumption because of node selection in the provisioning process. In DVSP, the deployed nodes attain energy-efficiency compared to the benchmark schemes. The game-theory based algorithm enables optimal number of nodes to remain active in each iteration. As a result, in DVSP, the nodes remain functional for an increased period of time. It is evident from the results that the proposed scheme is more energy-efficient compared to the benchmark scheme, for a long-term deployment.

4) Average number of QoS violations: In Figure 5, we show the results for the average number of QoS violations occurred in different iterations. In higher iterations, the number of QoS violations increase significantly, as nodes run out of energy. The provisioning of nodes is done while considering the QoS requirements of each particular request. QoS violations are considered if the requested QoS can not be matched among the available nodes. Therefore, the number of nodes selected depends on the QoS requests of that iterations. If nodes are available and satisfying the QoS requirements, then QoS violations will not be reported. In the benchmark schemes, the average number of QoS violations are 88.9% higher, compared to DVSP, due to the higher energy depletion rate in those schemes.

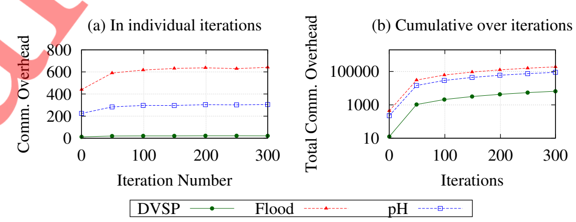

Fig. 6: Communication Overhead

5) Communication Overhead: We present the results for the overhead of node selection for the three schemes in Figure 6. This metric represent the average number of additional communication occurred per sensor owner for the sensor provisioning in that iteration. The game-theoretic provisioning process ensures optimal selection of nodes in DVSP, compared to the existing schemes. In each iteration, the node selection process considers the changes in the network. Accordingly, the communication overhead required in DVSP is lower than the existing schemes. Typically, the communication overhead and the number of active nodes in any iteration posses a linear relationship. On the other hand, in the benchmark schemes, the node selection process requires 92.99% increased number of communication between themselves.

VI. Conclusion

In this paper, we present DVSP – a dynamic virtual sensor provisioning scheme for sensor-cloud based IoT applications. Provisioning of virtual sensor is one of the basic requirements in sensor-cloud framework. Typically, the schemes in the existing literature consider selection of physical nodes which includes redundant nodes too. In our proposed scheme, we design the scheme such that the optimal node selection excludes the redundant nodes.

The objective of this scheme is to maximize the lifetime of the deployed nodes by using intelligent node selection algorithms by the CSP and SOs, while maintaining the QoS of the incoming user requests. We used Single-Leader Multi-Follower Stackelberg game to model the interaction between the CSP and SOs. In our model, we consider the cloud as the *single leader*, and the SOs as the *multiple followers*. In DVSP, only an optimal set of nodes are selected by the CSP and the SOs. The existence of a *Stackelberg-Nash-Cournot* equilibrium in the game was also shown. The detailed steps followed by the CSP and SOs were also presented. Simulation-based results depict that, in the proposed scheme, the average number of activated nodes remain 89.39% low compared to the benchmark schemes, and consequently, the average energy consumption of the nodes is reduced by 90.63%. The lifetime of the deployed on-fields sensors is also enhance. Also, we measure the average number of QoS violations in the proposed scheme and the benchmarks. In future, we plan to extend the work for a deployment of heterogeneous nodes by various SOs.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[2] P. Sarigiannidis, E. Karapistoli, and A. A. Economides, "Modeling the internet of things under attack: A G-network approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1964–1977, 2017.

[3] Q. Ye and W. Zhuang, "Distributed and adaptive medium access control for internet-of-things-enabled mobile networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 446–460, 2017.

[4] Y. Xu and A. Helal, "Scalable cloud–sensor architecture for the Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 285–298, 2016.

[5] M. Yuriyama and T. Kushida, "Sensor-cloud Infrastructure – Physical Sensor Management with Virtualized Sensors on Cloud Computing," in *Proc. of IEEE NBiS*, Takayama, Japan, 2010, pp. 1–8.

[6] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: Architecture, applications, and approaches," *Intl. J. of Distr. Sens. Netw.*, vol. 2013, pp. 1–18, 2013.

[7] S. Madria, V. Kumar, and R. Dalvi, "Sensor Cloud: A cloud of virtual sensors," *IEEE Softw.*, vol. 31, no. 2, pp. 70–77, 2014.

[8] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor-cloud: A paradigm shift from wireless sensor network," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1084–1093, 2017.

[9] C. Delgado, M. Canales, J. Ortín, J. R. Gállego, A. Redondi, S. Bousnina, and M. Cesana, "Joint application admission control and network slicing in virtual sensor networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 28–43, 2018.

[10] P. Evensen and H. Meling, "SenseWrap: A service oriented middleware with sensor virtualization and self-configuration," in *Proc. of IEEE ISSNIP*, Melbourne, VIC, Australia, 2009, pp. 261–266.

[11] J. Ibbotson, C. Gibson, J. Wright, P. Waggett, P. Zerfos, B. Szymanski, and D. J. Thornley, "Sensors as a service oriented architecture: Middleware for sensor networks," in *Proc. of Intl. Conf. on Intelli. Env.*, Kuala Lumpur, 2010, pp. 209–214.

[12] S. Chatterjee and S. Misra, "Optimal composition of a virtual sensor for efficient virtualization within sensor-cloud," in *Proc. of IEEE ICC*, London, UK, 2015, pp. 448–453.

[13] S. Chatterjee and S. Misra, "Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud," in *Proc. of IEEE ANTS*, New Delhi, India, 2014, pp. 1–6.

[14] T. Dinh and Y. Kim, "An efficient sensor-cloud interactive model for on-demand latency requirement guarantee," in *Proc. of IEEE ICC*, Paris, France, 2017, pp. 1–6.

[15] M. C. Vuran, Özgür B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Comp. Netw.*, vol. 45, no. 3, pp. 245–259, 2004.

[16] H. D. Sherali, A. L. Soyster, and F. H. Murphy, "Stackelberg-Nash-Cournot Equilibria: Characterizations and computations," *Operations Research*, vol. 31, no. 2, pp. 253–276, 1983.

[17] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case," *IEEE Internet of Things J.*, vol. 3, no. 3, pp. 1099–1112, 2016.

[18] T. Dinh and Y. Kim, "Information centric sensor-cloud integration: An efficient model to improve wireless sensor networks lifetime," in *Proc. of IEEE ICC*, Paris, France, 2017.

[19] C. Zhu, V. C. M. Leung, K. Wang, L. T. Yang, and Y. Zhang, "Multi-method data delivery for green sensor-cloud," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 176–182, 2017.

[20] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-Cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Syst. J.*, vol. 10, no. 2, pp. 628–636, 2016.

[21] T. Ojha, S. Bera, S. Misra, and N. S. Raghuwanshi, "Dynamic duty scheduling for green sensor-cloud applications," in *Proc. of IEEE CloudCom*, Singapore, Dec 2014, pp. 841–846.

[22] C. Sarkar, V. S. Rao, R. V. Prasad, S. N. Das, S. Misra, and A. Vasilakos, "VSF: An energy-efficient sensing framework using virtual sensors," *IEEE Sens. J.*, vol. 16, no. 12, pp. 5046–5059, 2016.

[23] A. Kothari, V. Boddula, L. Ramaswamy, and N. Abolhassani, "DQS-Cloud: A data quality-aware autonomic cloud for sensor services," in *Proc. of IEEE CollaborateCom*, Miami, FL, USA, 2014, pp. 295–303.

[24] V. Lawson and L. Ramaswamy, "Data quality and energy management tradeoffs in sensor service clouds," in *Proc. IEEE Intl. Cong. on Big Data*, New York, NY, USA, 2015, pp. 749–752.

[25] M. Lemos, C. de Carvalho, D. Lopes, R. Rabelo, and R. H. Filho, "Reducing energy consumption in provisioning of virtual sensors by similarity of heterogenous sensors," in *Proc. of IEEE AINA*, Taipei, Taiwan, 2017, pp. 415–422.

[26] M. Lemos, R. H. Filho, R. Rabelo, C. de Carvalho, D. Lopes, and V. da Gama Costa, "An energy-efficient approach to enhance virtual sensors provisioning in sensor clouds environments," *Sensors*, vol. 18, no. 689, pp. 1–26, 2018.

[27] T. Dinh and Y. Kim, "A novel location-centric IoT-cloud based on-street car parking violation management system in smart cities," *Sensors*, vol. 16, no. 6, p. 810, 2016.

[28] T. Dinh and Y. Kim, "An efficient interactive model for on-demand sensing-as-a-services of sensor-cloud," *Sensors*, vol. 16, no. 7, 2016.

[29] T. Dinh, Y. Kim, and H. Lee, "A location-based interactive model of internet of things and cloud (IoT-Cloud) for mobile cloud computing applications," *Sensors*, vol. 17, no. 3, p. 489, 2017.

[30] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "On balancing energy consumption in wireless sensor networks," *IEEE Trans. on Vehicular Tech.*, vol. 58, no. 6, pp. 2909–2924, 2009.

[31] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Trans. on Serv. Comp.*, vol. 10, no. 2, pp. 203–216, 2017.