SensOrch: QoS-Aware Resource Orchestration for Provisioning Sensors-as-a-Service

Aishwariya Chakraborty[†], Sudip Misra[†], Ayan Mondal[†], and Mohammad S. Obaidat[‡]
[†]Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India
[‡]College of Computing and Informatics, University of Sharjah, UAE, KAIST
University of Jordan, Jordan, and University of Science and Technology Beijing, China
Email: {[†]aishwariya.chakraborty, [†]sudipm, [†]ayanmondal}@iitkgp.ac.in, [‡]msobaidat@gmail.com

***Abstract*—In this work, we address the problem of efficient utilization of resource-constrained wireless sensor nodes (WSNs) in sensor-cloud for provisioning high quality of Sensors-as-a-Service (Se-aaS). In sensor-cloud, the sensor-cloud service provider (SCSP) obtains sensor nodes on rent from their respective sensor-owners and utilizes them to create virtual sensors. Thereby, the SCSP provisions these virtual sensors as Se-aaS to the end-users for serving their WSN-based applications and earns revenue in exchange. In order to ensure high quality-of-service (QoS) of Se-aaS while simultaneously ensuring profits for itself and the sensor-owners, the SCSP needs to optimally allocate physical sensor nodes to serve the virtual sensors, while considering their limited capacity and the fair distribution of service load among different sensor-owners. Although a few existing works focused on optimal resource allocation in sensor-cloud, none of them considered the possibility of sharing the same physical sensor node among multiple virtual sensors. Hence, in this work, we propose an optimal resource orchestration scheme for sensor-cloud, named SensOrch, which is based on coalition formation game with transferable utility. Using SensOrch, the SCSP ensures the optimal allocation of physical sensor nodes to virtual sensors while maintaining high QoS and high profitability of Se-aaS. Through simulations, we observe that, using SensOrch, the network lifetime increases by 25.31–59.6% along with a simultaneous increase in the profit of the SCSP by 23.64–29.49% as compared to the existing schemes. Additionally, SensOrch ensures fair distribution of profits among the sensor-owners.**

***Index Terms*—Sensor-Cloud, Se-aaS, Cooperative Game Theory, Resource Management, Virtual Sensor**

I. INTRODUCTION

With the rapid adoption of Internet-of-Things (IoT) technology, recent years have witnessed a tremendous upsurge in the number of IoT devices and applications which are being used widely in the highly-connected modern world. To support the growth of IoT technology, researchers have conceptualized several architectures that aim to improve its usability and accessibility for the common people. One such architecture is the sensor-cloud which was proposed with the aim of unifying the advantages of wireless sensor networks (WSNs) and cloud computing in the light of the Service-Oriented Architecture (SOA) principles [1]–[3]. Basically, sensor-cloud utilizes the concept of resource virtualization of cloud, thereby allowing us to envision ordinary WSNs in the form of service units, termed as Sensors-as-a-Service (Se-aaS). Thus, the end-users of WSN-based applications can use these simple and easily accessible service units as per the requirement of their applications, without being bothered about the complexities of purchasing, installing, and maintaining their own WSN hardware.

Primarily, the sensor-cloud architecture comprises of three main entities — sensor-owners, Sensor-Cloud Service Provider (SCSP), and end-users. The sensor-owners purchase, deploy, and maintain their own WSNs. The SCSP obtains these sensor nodes on a rental basis from their respective sensor-owners and utilizes them to create *virtual sensors* for provisioning Se-aaS with the help of cloud infrastructure. The end-users, on the other hand, utilize the provisioned Se-aaS for serving their WSN-based applications by paying to the SCSP a nominal service charge which is decided based on their service usage following the *pay-per-use* model [4].

Similar to other cloud-based SOAs, two significant deciding factors for the adaptation of sensor-cloud technology are quality-of-service (QoS) of Se-aaS delivered to the end-users and the profitability of Se-aaS for the SCSP and the sensor-owners. These factors, in turn, depend on the efficient and optimal utilization of the resource-constrained WSNs obtained from the sensor-owners by the SCSP. In sensor-cloud, to serve each service-request of the end-users, the SCSP provisions one or more virtual sensors. Each of these virtual sensors is composed of one or more physical sensor nodes based on the requirement of the end-users. Additionally, an inherent advantage of virtualization in sensor-cloud is that the same physical sensor node can be used to serve more than one virtual sensors having similar requirements. So, in order to ensure efficient resource utilization, it is essential to optimally allocate the physical sensor nodes to form virtual sensors, while considering the QoS requirements of the end-users and the capacity of the physical sensor nodes to serve them. Moreover, to ensure the profitability of sensor-owners, it is equally essential to ensure that the sensor nodes belonging to each sensor-owner have a fair chance to be allocated for serving the virtual sensors. In the existing literature, few schemes [5]–[7] are proposed for optimal virtual sensor formation in sensor-cloud. However, none of these schemes considered the possibility of allocating the same physical sensor nodes to serve multiple virtual sensors, which eventually ensures optimal resource utilization and an increase in network lifetime. Hence, there is a need to design a scheme for optimal orchestration of physical sensor nodes to provision QoS-aware Se-aaS in sensor-cloud.

In this work, we propose a QoS-aware resource orchestration scheme, named *SensOrch*, to ensure efficient allocation of physical sensor nodes to virtual sensors for provisioning Se-aaS in sensor-cloud. The proposed scheme takes into consideration different factors such as the QoS requirements of the service requests, the capability of each physical sensor node to serve each virtual sensor, and the fair distribution of profits among the sensor-owners while allocating the resources optimally. The main contributions of this work are as follows:

(1) We propose a dynamic resource orchestration scheme, named SensOrch, for provisioning Se-aaS with high QoS while ensuring efficient utilization of sensor nodes, high profit of SCSP, and fair distribution of profits among the sensor-owners.

(2) Using cooperative coalition formation game, we model the problem of optimal allocation of sensor nodes to virtual sensors with an aim to achieve the aforementioned objectives.

(3) We propose two *online* algorithms, i.e., *merge* and *split*, using cooperative coalition formation game to ensure optimal service allocation.

(4) We evaluate the performance of SensOrch through simulations and compare its performance with two existing benchmark schemes.

II. RELATED WORK

In the existing literature, several research works focused on sensor-cloud architecture. The basic conceptualization and theoretical modeling of the sensor-cloud were proposed by Yuriama *et al.* [1], [2] and Misra *et al.* [3], in which the authors also demonstrated a few applications of the sensor-cloud architecture. Thereafter, several researchers proposed various schemes to improve the performance of sensor-cloud. Chatterjee *et al.* [8] proposed the cache-enabled architecture of sensor-cloud in which two caches — internal and external caches — were introduced in order to reduce the resource consumption due to redundant data transmissions. Ojha *et al.* [9] proposed a scheme for optimal duty scheduling in sensor-cloud in order to reduce the energy consumption of sensor nodes. Another scheme was proposed by Chatterjee *et al.* [10] for the optimal selection of intermediate nodes for data transmission in the presence of unintentional node failures. Kim [11] proposed an efficient sensor-cloud control scheme to select the most adaptable data-center for a request and to motivate sensor nodes to participate in Se-aaS provisioning through incentives. Researchers also proposed a few pricing schemes for sensor-cloud. For example, Chakraborty *et al.* [12] proposed a pricing scheme for cache-enabled sensor-cloud in order to ensure optimal distribution of service-requests among the two caches. In another work, Chatterjee *et al.* [4] proposed a dynamic pricing scheme comprising of two components – pricing due to hardware and pricing due to infrastructure – for sensor-cloud while considering high profit of SCSP and service satisfaction of the end-users. Chakraborty *et al.* [13] proposed a dynamic trust enforcing pricing scheme for sensor-cloud, in order to prevent misbehavior of sensor-owners and ensure high profits of SCSP. In another work, five pricing schemes were proposed by Zhu *et al.* [14] while considering different parameters such as service duration, type, and lease period.

The problem of optimal resource allocation in sensor-cloud has also been addressed by the researchers in the existing literature. Chatterjee *et al.* [5] proposed two schemes – CoV-I and CoV-II – for the optimal composition of virtual sensors using physical sensor nodes depending on the region of interest (RoI) of the service requests. Another optimal virtual sensor mapping scheme was proposed by Roy *et al.* [6] while considering overlapping deployment region of multiple sensor owners. Ojha *et al.* [15] proposed a virtual sensor provisioning scheme in order to ensure cooperation among sensor-owners for data transmission in sensor-cloud. Ojha *et al.* [7] proposed another virtual sensor composition scheme in order to improve network lifetime while maintaining high QoS of Se-aaS. However, none of these aforementioned schemes consider that the same physical sensor node can be used to serve multiple service-requests simultaneously based on the property of virtualization in sensor-cloud. Although Rachkidi *et al.* [16] considered the possibility of sharing physical sensor nodes among multiple end-users, the authors primarily addressed the problem of optimal sharing of virtual sensors and placement of virtual machines, instead of the allocation of the physical sensor nodes. Hence, we infer that it necessitates the designing of an efficient resource orchestration schemes for sensor-cloud which not only ensure high QoS but also maintain the profitability of the SCSP and sensor-owners.

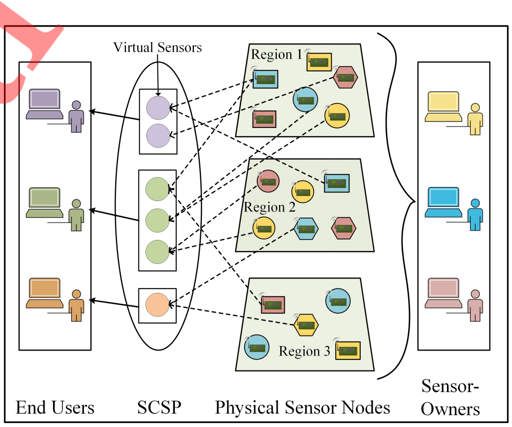


Fig. 1. Schematic Diagram of Sensor-Cloud

III. SYSTEM MODEL

We consider that the sensor-cloud comprises of a single SCSP and multiple registered sensor-owners as shown in Figure 1. Each sensor-owner $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of sensor-owners, deploys a subset of sensor nodes $\mathcal{N}_s \subseteq \mathcal{N}$, where $\mathcal{N}$ denotes the set of sensor nodes deployed. Each sensor-owner s registers his/her sensor nodes with the SCSP. Thereafter, the SCSP takes control over the registered sensor nodes, and provisions Se-aaS to the end-users as per their requirements. Here, we consider that, each end-user specifies his/her requirements for a particular service-request a in terms of the type of data $\tau_a(t)$, required data-rate $r_a(t)$, and region of interest $i_a(t)$. Based on these service requirements, the SCSP decides the number of nodes $\eta_a(t)$ and the amount of memory $m_a(t)$ per node required to serve the request and the price P per unit service to be charged from the end-user.

Moreover, in this work, we consider that each of the registered sensor nodes is capable of serving multiple applications, simultaneously. However, due to data-rate constraint, each sensor node $n \in \mathcal{N}$ can serve limited number of services at once. The maximum data-rate capacity of sensor node n is denoted as R_n^{max}. Considering that sensor node n serves $\mathcal{A}_n(t)$ set of applications at time instant t, where each application $a \in \mathcal{A}_n(t)$ has a data-rate requirement of $r_a(t)$, the following condition needs to be satisfied:

$$\sum_{a \in \mathcal{A}_n(t)} r_a(t) \leq R_n^{max} \tag{1}$$

On the other hand, in order to enhance the performance of sensor-cloud and increase the network-lifetime, the number of active sensor nodes in the network needs to be reduced, as each sensor node consumes significant amount of energy while being in *active* state. Thereby, we define a threshold value R_n^{th} for the minimum serviceable data-rate for each sensor node n. Thus, the services being served by sensor node n are migrated to other active nodes and it transits into *sleep* mode, if the following condition holds:

$$\sum_{a \in \mathcal{A}_n(t)} r_a(t) < R_n^{th} \tag{2}$$

Therefore, using the proposed scheme, SensOrch, we aim to reduce the number of active nodes while ensuring high QoS of Se-aaS. Additionally, we try to ensure that the overall service-load is properly distributed among the active physical sensor nodes and each sensor-owner gets an equal opportunity to earn profit from the SCSP.

Assumptions: The assumptions considered while designing the proposed scheme are — (i) sensor nodes are heterogeneous in nature and are capable of serving all types of requests; (ii) the SCSP is in charge of controlling resource management in sensor-cloud; and (iii) each service-request of the end-users is served using a single virtual sensor, which is maintained by the SCSP. However, each virtual sensor can be served using multiple physical sensors.

IV. SENSORCH: THE PROPOSED RESOURCE ORCHESTRATION SCHEME

A. *Game formulation*

In order to decide the optimal mapping of the virtual sensors to the physical sensor nodes, we use a *dynamic coalition-formation cooperation game with transferable utility* [12] in SensOrch. In SensOrch, each sensor node n represents a *coalition*, and the set of incoming service-requests or the virtual sensors served by node n defines the population in the coalition. We consider that the sequentially arriving service-requests of the end-users are not known to the SCSP *a priori* in sensor-cloud. Thus, for each incoming request, the SCSP needs to decide the optimal coalitions that it should join, i.e., the optimal sensor nodes that need to be allocated, for serving it. The objectives of the SCSP are — (1) to minimize the number of activated nodes in the system and (2) to distribute the service load properly among the activated nodes, while ensuring that the QoS requirements of the service-requests are satisfied. Additionally, the SCSP aims to ensure that the overall profit is evenly distributed among the sensor-owners. Hence, we argue that *coalition formation cooperative game* appropriately models the aforementioned problem scenario. Thus, in this work, we propose SensOrch, which is an online scheme, using the dynamic coalition-formation game theoretic approach. The components of the scheme are as follows:

(i) The SCSP acts as the centralized coordinator. Considering that the sensor nodes are cooperative in nature, the SCSP tries to maximize the network lifetime by activating an optimal set of physical sensor nodes.

(ii) Each service-request is treated as an independent service, and the end-users pay accordingly. Hence, the SCSP aims to consolidate the requirements of the service-requests for ensuring optimal resource management.

(iii) While allocating a service-request to a subset of activated sensor nodes $\mathcal{Q}_a(t)$, the SCSP needs to ensure that the following constraints are satisfied:

$$|\mathcal{Q}_a(t)| \geq \eta_a(t); \;\; M_n^{rem+} \geq 0; \text{ and } E_n^{res+} \geq E_{th} \tag{3}$$

where M_n^{rem+} denotes the effective memory space available at each node n, as defined in Definition 1; E_n^{res+} and E_{th} are the effective residual energy of node n and the threshold energy requirement for serving a service-request, respectively, as defined in Definition 2.

***Definition** 1:* The effective memory space available M_n^{rem+} at each node n is defined as free memory available after considering the memory space requirement of the allocated service requests including the existing service-requests, which are currently getting served by the sensor node n, and the newly arrived service request.

$$M_n^{rem+} = M_n^{rem-} - m_{\tilde{a}}(t) \tag{4}$$

where $M_n^{rem-} = M_n^{max} - \sum_{a \in \mathcal{A}_n(t)} m_a(t)$; M_n^{max} is the maximum memory space available; and $\tilde{a}$ is the newly arrived service-request.

***Definition** 2:* The effective residual energy E_n^{res+} of each node n is defined as the predicted remaining energy of node n after a fixed time duration Δt, if the newly arrived service-request is allocated to sensor node n along with the existing service requests.

$$E_n^{res+} = \begin{cases} E_{res} - [E_n^{res-} + r_{\tilde{a}}(t)\Delta t], & \text{if } \exists a \in \mathcal{A}_n(t) : \\ & \quad \text{GCD}(r_a(t), r_{\tilde{a}}(t)) = r_a(t) \\ E_{res} - E_n^{res-}, & \text{otherwise} \end{cases} \tag{5}$$

where $E_n^{res-} = \Delta t \sum_{a \in \mathcal{A}_n(t)} r_a(t)$; E_n^{res} is the current residual energy; and $\tilde{a}$ is the newly arrived service-request.

Thus, in SensOrch, the incoming service-requests, which are considered to be the players of the proposed game, need to be allocated optimally among the set of coalitions, i.e., the set of activated nodes. To achieve this aim, the SCSP tries to maximize the overall payoff of the coalitions, while ensuring that an optimal set of sensor nodes are activated, and each sensor node has an optimal service load. While incorporating the aforementioned attributes, we define the utility function of each coalition, as mentioned in Section IV-B.

B. Utility Function of Each Coalition

The utility function $\mathcal{U}_n(t)$ of sensor node n, i.e., coalition n, signifies a trade-off between the resource utilization of the node and the profit earned using the node by its owner for serving the set of already-allocated service requests along with the newly arrived request. In SensOrch, each sensor node tries to obtain the optimal allocation of service requests for maximizing the payoff of utility function of its coalition. The utility function $\mathcal{U}_n(t)$ of each coalition n needs to satisfy the following properties:

(i) The payoff of $\mathcal{U}_n(t)$ increases with increase in the effective residual energy E_n^{res+}.

(ii) The payoff of $\mathcal{U}_n(t)$ decreases with increase in the effective data-rate allocated R_n^{eff+} to node n, as defined in Definition 3.

(iii) The payoff of $\mathcal{U}_n(t)$ increases with increase in the effective remaining memory space M_n^{rem+} of node n.

(iv) The payoff of $\mathcal{U}_n(t)$ increases with increase in the profit earned P_n by the sensor-owner of node n, where $P_n = (p-c)\sum_{a\in\mathcal{A}_n(t)} r_a(t)$, and p and c denote the price charged and the maintenance cost incurred per unit data-rate, respectively.

***Definition** 3:* The effective data-rate R_n^{eff+} of each node n is defined as the data-transmission rate of node n. If the newly arrived service-request is allocated to sensor node n along with the existing service-requests, we have:

$$R_n^{eff+} = \begin{cases} R_n^{eff-} + r_{\tilde{a}}(t), & \text{if } \exists a \in \mathcal{A}_n(t) : \\ & \quad \text{GCD}(r_a(t), r_{\tilde{a}}(t)) = r_a(t) \\ R_n^{eff-}, & \text{otherwise} \end{cases} \tag{6}$$

where $\tilde{a}$ is the newly arrived service-request, and R_n^{eff-} denotes the effective data-rate with existing service-requests.

Therefore, we define the utility function $\mathcal{U}_n(t)$ of each coalition n as follows:

$$\mathcal{U}_n(t) = \frac{E_n^{res+}}{E_n^{max}} + \frac{R_n^{max}}{R_n^{eff+}} + \frac{M_n^{rem+}}{M_n^{max}} + \frac{P_n}{p\sum_{a\in\bigcup_n \mathcal{A}_n(t)} r_a(t)} \tag{7}$$

C. Utility Function of the SCSP

The SCSP aims to obtain that an optimal distribution of service load among the activated nodes. Hence, the SCSP tries to maximize the overall utility of the activated nodes, i.e., the payoff value of the coalitions. Thereby, the utility function $\mathcal{B}(t)$ of the SCSP is as follows:

$$\mathcal{B}(t) = \prod_{n\in\mathcal{N}_a(t)} \mathcal{U}_n(t) \tag{8}$$

where $\mathcal{N}^{'}(t) \subseteq \mathcal{N}$ denotes the set of activated nodes in the region of interest. The SCSP tries to maximize the payoff of utility function $\mathcal{B}(t)$, while satisfying the constraints mentioned in Equations (1), (2), and (3).

D. Equilibrium in SensOrch

The SCSP aims to ensure Pareto optimal resource orchestration using SensOrch by obtaining a preference relation among the elements of the superset of possible partition combinations. The preference relation among two partitions X and Y is defined in Definition 4. Thus, in sensor-cloud, as the SCSP has the centralized view of the entire system, it is able to ensure the existence of Pareto optimal solution [17] using the proposed scheme, SensOrch, while considering that the sensor nodes are cooperative in nature.

***Definition** 4:* Considering that an incoming service request $\tilde{a}$ can be associated with nodes x and y, we get two partition combinations X and Y, respectively. We prefer partition X over partition Y, which is represented mathematically as $X \triangleright Y$, iff the following condition is satisfied:

$$\left[\mathcal{U}_x^{'}(t)\mathcal{U}_y(t) \prod_{n\notin\{x,y\}} \mathcal{U}_n(t)\right] \geq \left[\mathcal{U}_x(t)\mathcal{U}_y^{'}(t) \prod_{n\notin\{x,y\}} \mathcal{U}_n(t)\right] \tag{9}$$

where $\mathcal{U}_x^{'}(t)$ and $\mathcal{U}_y^{'}(t)$ denote the payoff of nodes x and y, while considering that service request $\tilde{a}$ is associated with node x and node y, respectively.

Algorithm 1 MERGE Algorithm for SensOrch

```
INPUTS: $\tilde{a}$, $\mathcal{Q}_{\tilde{a}}(t)$, $\{r_a(t)|a \in \mathcal{A}_n(t)\}$, $M_n^{rem+}$, $M_n^{max}$, $E_n^{res+}$, $E_n^{max}$,
    $R_n^{max}$, $R_n^{eff+}$, $P_n$, $\forall n \in \mathcal{N}^{'}$
OUTPUT: $\mathcal{N}^{'}$ and $\{\mathcal{A}_n(t)|\forall n \in \mathcal{N}^{'}\}$
PROCEDURE:
$\mathcal{K} \leftarrow \{\emptyset\}$
count $\leftarrow 0$
for $i := 1$ to $|\mathcal{Q}_{\tilde{a}}(t)|$ do
    for each $n \in \mathcal{N}^{'}$ do
        if conditions in Equations (1)-(3) are true then
            $\mathcal{A}_n^{'}(t) \leftarrow \mathcal{A}_n(t) \cup \tilde{a}$
            count $\leftarrow$ count $+ 1$
            Calculate the payoff $\mathcal{B}(t)$ using Equation (8)
            $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{B}(t)\}$
        end if
    end for
end for
if count $< |\mathcal{Q}_{\tilde{a}}(t)|$ then
    for $i :=$ count $+ 1$ to $|\mathcal{Q}_{\tilde{a}}(t)|$ do
        Activate a sensor node $n$ from set $(\mathcal{N}/\mathcal{N}^{'})$
        $\mathcal{A}_n^{'}(t) \leftarrow \mathcal{A}_n(t) \cup \tilde{a}$
        Calculate the payoff $\mathcal{B}(t)$ using Equation (8)
        $\mathcal{K} \leftarrow \mathcal{K} \cup \{\mathcal{B}(t)\}$
    end for
end if
Select $|\mathcal{Q}_{\tilde{a}}(t)|$ number of partitions having higher values in $\mathcal{K}$
for each selected partition $k$ do
    $\mathcal{A}_k(t) \leftarrow \mathcal{A}_k(t) \cup \tilde{a}$
end for
return $\mathcal{N}^{'}$ and $\{\mathcal{A}_n(t)|\forall n \in \mathcal{N}^{'}\}$
```

E. Algorithms

In order to obtain the optimal service load distribution, we propose two algorithms based on *Merge-and-Split* [17] in this work. On receiving each service-request from the end-users, the SCSP executes Algorithm 1, i.e., the *Merge* algorithm, to consolidate the service request among the set of activated nodes. Additionally, if required, the SCSP activates a subset of sleep nodes to meet the requirements of the requested service, for which it follows a *round-robin* scheme. In other words, the SCSP activates a single node from each sensor-owner, sequentially, given that the sensor nodes are within the concerned region of interest. On the other hand, at the completion of each service request, the SCSP executes Algorithm 2, i.e., the *SPLIT* algorithm, to distribute the datarate associated with sensor node n to other active nodes, while ensuring Equation (2) holds.

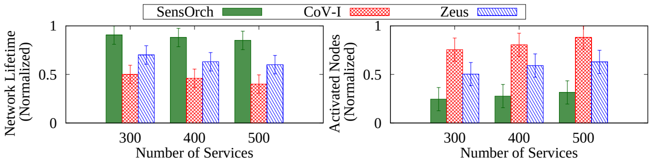
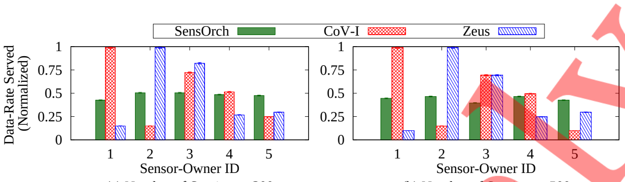


Fig. 2. Network Lifetime and Activated Sensor Nodes



(a) Number of Services = 300
(b) Number of Services = 500

Fig. 3. Data-Rate Served by Each Sensor-Owner

Algorithm 2 SPLIT Algorithm for SensOrch

INPUTS: $\mathcal{Q}_a(t)$, $\{r_a(t)|a \in \mathcal{A}_n(t)\}$, M_n^{rem+}, M_n^{max}, E_n^{res+},E_n^{max}, R_n^{max}, R_n^{eff+}, $P_n, \forall n \in \mathcal{N}'$

OUTPUT: $\mathcal{N}'$ and $\{\mathcal{A}_n(t)|\forall n \in \mathcal{N}'\}$

PROCEDURE:

```
$\mathcal{K} \leftarrow \{\emptyset\}$
for each $n \in \mathcal{Q}_a(t)$ do
    if condition in Equation (2) or (3) is false then
        $\mathcal{N}' \leftarrow (\mathcal{N}'/n)$
        $\mathcal{K} \leftarrow \mathcal{K} \cup \mathcal{A}_n(t)$
    end if
end for
for each $\tilde{a} \in \mathcal{K}$ do
    Call MERGE Algorithm
end for
return $\mathcal{N}'$ and $\{\mathcal{A}_n(t)|\forall n \in \mathcal{N}'\}$
```

V. PERFORMANCE EVALUATION

We evaluate the performance of the proposed scheme, SensOrch, through simulations. The details of the simulations and the analysis of the results obtained are presented in the following subsections.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation area	1000 $m \times$1000 m
Number of sensor owners	5
Number of sensor nodes per sensor owner	20
Number of service requests	$300 - 500$
Number of sensor nodes per service requests	1-5
Data-rate requirement per service request	30-50 kbps
Maximum data-rate per node	250 kbps
Maximum memory per node	512 kb [18]
Communication protocol	IEEE 802.15.4
Initial energy of each node	20 J [13]
Tx energy consumption	50 nJ/bit [13]
Rx energy consumption	50 nJ/bit [13]
Energy consumption at amplifier	100 pJ/bit-m^2 [13]

A. Simulation Parameters

We simulated the proposed scheme, SensOrch, in a MATLAB-based simulation platform. We considered a single rectangular geographical region in which 5 sensor-owners have deployed 20 heterogeneous sensor nodes each. These sensor owners have registered their sensor nodes with a single SCSP. Additionally, we considered that the end-users request for the service of the SCSP, sequentially, and the service requests vary in terms of data-rate and memory requirements. Each service request is considered to be served using a single virtual sensor, which is composed using multiple physical sensor nodes. For simulations, the service requirements and the node requirement for each virtual sensor were determined randomly. The detailed simulation parameters are presented in Table I.

B. Benchmarks

We compared the performance of the proposed scheme, SensOrch, with two existing benchmark schemes – optimal composition of virtual sensors (CoV-I) [5] and resource allocation algorithm for the cloud of sensors (Zeus) [19]. In CoV-I, Chatterjee *et al.* [5] proposed an optimal virtual sensor formation scheme while considering that physical sensor nodes are deployed in the same geographic region. The authors defined two parameters of sensor nodes – goodness which is measured based on its physical parameters, and quality of information – based on which the nodes are selected for composing the virtual sensors. In Zeus, Santos *et al.* [19] proposed an optimal resource allocation algorithm for virtual sensors in order to minimize the overall resource consumption. The authors considered that the requests common to multiple applications are executed only once and the result is shared among them. Although both of these works focus on optimal resource allocation in sensor-cloud, neither of them considered the fair distribution of profits among the sensor-owners. Additionally, the existing schemes considered that the set of service-requests are known to the SCSP *a priori*.

C. Performance Metrics

We evaluated the performance of the proposed scheme, SensOrch, based on the following performance metrics.

Network Lifetime: Network lifetime is calculated as the total duration between the time of initial deployment of the network and the time at which the last node in the network dies. It is dependent on the energy consumption of nodes which, in turn, depends on the service load of each node.

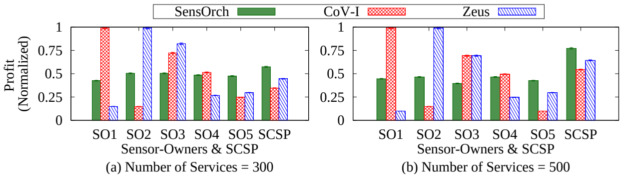
Data-rate served by each sensor-owner: The average data-rate served by the set of sensor nodes belonging to a particular sensor-owner is directly proportional to the revenue earned by the sensor-owner. This is due to the fact that the sensor-cloud follows a pay-per-user model as mentioned in Section I.

Profit of sensor-owners: The profit earned by each sensor-owner is the difference between the price received by him/her from the SCSP and the cost incurred for maintaining his/her sensor nodes. We consider that the profit of each sensor-owner is dependent on the data-rate served by his/her nodes and is calculated as mentioned in Section IV-B.

Profit of SCSP: The profit earned by the SCSP is calculated as the difference between the price paid by the end-users and the total service provisioning cost incurred by the SCSP. The service provisioning cost includes the cloud infrastructure maintenance cost, which is considered to be fixed per unit service, and the price charged by the sensor-owners.

D. Results and discussions

From Figure 2, we observe that, using SensOrch, the network lifetime increases by 50-59.6% and 25.31-33.11% compared to using CoV-I and Zeus, along with a corresponding decrease in the number of activated nodes. This is due to



(a) Number of Services = 300 (b) Number of Services = 500

Fig. 4. Profit of Each Sensor-Owner and the SCSP

the fact that in CoV-I, each physical sensor node is used to serve only a single request at a time, unlike SensOrch in which multiple requests can be served simultaneously using the same sensor node. Thus, at a given time instant, a higher number of sensor nodes are activated in the network using CoV-I than using SensOrch, thereby increasing resource consumption and reducing network lifetime. On the other hand, although Zeus aims to serve the service requests having similar requirements using the same sensor node, it attempts to do so only if the requirements of service-requests are known *a priori*, which is not possible in case of sensor-cloud. Thus, we observe that Zeus also allocates a single service request to each sensor node, thereby resulting in a higher number of activated sensor nodes and lower network lifetime compared to SensOrch.

Figure 3 shows the variation of the data-rates served by each sensor owner with the increase in the number of service requests. We observe that, using SensOrch, the total data-rate requirement of the service-requests is almost equally distributed among the sensor owners. However, using CoV-I and Zeus, the distribution of the datarate, which is equivalent to the service load, among each sensor-owner varies randomly. This is due to the fact that SensOrch achieves a trade-off between the resource utilization and the service load on each sensor node, unlike the other two schemes which choose the nodes based only on their physical parameters. Since the profit earned by each sensor owner is considered to be proportional to the datarate served by his/her nodes, a similar trend is observed in the variation of profits with the increase in the number of service requests, as shown in Figure 4. We observe that SensOrch ensures fair distribution of profits among the sensor-owners, unlike the other two existing schemes.

Additionally, from Figure 4, we observe that the profit earned by the SCSP increases by 29.49% and 23.64% using SensOrch than using the existing schemes — CoV-I and Zeus, respectively. This is due to the fact that the consolidation of services and increased network lifetime enables the SCSP to support a higher number of services using SensOrch than using the existing schemes. Thus, we argue that SensOrch outperforms the existing benchmark schemes – CoV-I and Zeus.

VI. CONCLUSION

In this work, we proposed SensOrch, which is an optimal resource orchestration scheme for provisioning QoS-aware Se-aaS in sensor-cloud. We used cooperative coalition formation game to model the aforementioned problem and proposed a merge-and-split-based online algorithm to obtain the optimal allocation of physical sensor nodes to virtual sensors and ensure efficient resource utilization. We evaluated the proposed scheme through simulations and compared its performance to two existing benchmark schemes – CoV-I and Zeus. We observed that SensOrch outperforms the existing schemes in terms of increased network lifetime, increased profits of SCSP, and ensures even distribution of service load among sensor nodes belonging to different sensor-owners.

This work can be extended in future to study the network dynamics as well as the economic aspects of sensor-cloud while considering link quality of the data transmission paths. It can also be extended to study the effects of the geographically distributed cloud data-centers on the performance of sensor-cloud.

ACKNOWLEDGEMENT

The authors acknowledge the funding support received from INAE (Grant no. INAE/121/AKF, Dt. 13-02-2019), and SERB/IMPRINT-II (Grant no SERB/F/12680/2018-2019;IMP/2018/000451, Dt. 25-03-2019) for executing parts of this work.

REFERENCES

[1] M. Yuriyama and T. Kushida, “Sensor-Cloud Infrastructure – Physical Sensor Management with Virtualized Sensors on Cloud Computing,” in *Proc. of Int. Conf. on Net.-Based Inf. Syst.*, September 2010, pp. 1–8.
[2] M. Yuriyama, T. Kushida, and M. Itakura, “A New Model of Accelerating Service Innovation with Sensor-Cloud Infrastructure,” in *Ann. SRII Glob. Conf.*, Mar. 2011, pp. 308–314.
[3] 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, Jun. 2017.
[4] 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, Mar. 2017.
[5] S. Chatterjee and S. Misra, “Optimal Composition of a Virtual Sensor for Efficient Virtualization within Sensor-Cloud,” in *Proc. of IEEE ICC*, Jun. 2015, pp. 448–453.
[6] C. Roy, A. Roy, and S. Misra, “DIVISOR: Dynamic Virtual Sensor Formation for Overlapping Region in IoT-based Sensor-Cloud,” *Proc. of IEEE WCNC*, pp. 1–6, 2018.
[7] T. Ojha, S. Misra, N. S. Raghuwanshi, and H. Poddar, “Dvsp: Dynamic virtual sensor provisioning in sensor-cloud based internet of things,” *IEEE IoT J.*, pp. 1–8, 2019.
[8] S. Chatterjee and S. Misra, “Dynamic and Adaptive Data Caching Mechanism for Virtualization within Sensor-Cloud,” in *Proc. of IEEE ANTS*, Dec. 2014, pp. 1–6.
[9] T. Ojha, S. Bera, S. Misra, and N. S. Raghuwanshi, “Dynamic Duty Scheduling for Green Sensor-Cloud Applications,” in *Proc. of IEEE CloudCom*, Dec. 2014, pp. 841–846.
[10] S. Chatterjee, S. Sarkar, and S. Misra, “Energy-Efficient Data Transmission in Sensor-Cloud,” in *Proc. of App. and Innov. in Mob. Comp.*, Feb. 2015, pp. 68–73.
[11] S. Kim, “An Effective Sensor Cloud Control Scheme Based on a Two-Stage Game Approach,” *IEEE Acc.*, vol. 6, pp. 20 430–20 439, 2018.
[12] A. Chakraborty, A. Mondal, and S. Misra, “Cache-Enabled Sensor-Cloud: The Economic Facet,” *Proc. of IEEE WCNC*, pp. 1–6, 2018.
[13] A. Chakraborty, A. Mondal, A. Roy, and S. Misra, “Dynamic trust enforcing pricing scheme for sensors-as-a-service in sensor-cloud infrastructure,” *IEEE Trans. on Serv. Comp.*, pp. 1–12, 2018.
[14] C. Zhu, X. Li, V. C. M. Leung, L. T. Yang, E. C. H. Ngai, and L. Shu, “Towards Pricing for Sensor-Cloud,” *IEEE Trans. on Cloud Comp.*, vol. PP, no. 99, pp. 1–1, 2017.
[15] T. Ojha, S. Misra, N. S. Raghuwanshi, and M. S. Obaidat, “idvsp: Intelligent dynamic virtual sensor provisioning in sensor-cloud infrastructure,” in *Proc. of IEEE GLOBECOM*, Dec 2018, pp. 1–6.
[16] E. El Rachkidi, N. Agoulmine, N. Chendeb, and D. Belaid, “Resources optimization and efficient distribution of shared virtual sensors in sensor-cloud,” in *IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.
[17] W. Saad, Z. Han, M. Debbah, and A. Hjorungnes, “A Distributed Merge and Split Algorithm for Fair Cooperation in Wireless Networks,” in *Proc. of ICC Workshops*, May 2008, pp. 311–315.

[18] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov 2002.

[19] I. L. Santos, L. Pirmez, F. C. Delicato, G. M. Oliveira, C. M. Farias, S. U. Khan, and A. Y. Zomaya, "Zeus: A Resource Allocation Algorithm for the Cloud of Sensors," *Future Gen. Comp. Sys.*, vol. 92, pp. 564 – 581, 2019.