iDVSP: Intelligent Dynamic Virtual Sensor Provisioning in Sensor-Cloud Infrastructure

Tamoghna Ojha*, Sudip Misra*, Narendra Singh Raghuwanshi†, Mohammad S. Obaidat‡ *Fellow of IEEE*

*Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, 721302, India
Email: {tojha, smisra}@sit.iitkgp.ernet.in
†Department of Agricultural and Food Engineering, Indian Institute of Technology Kharagpur, 721302, India
Email: nsr@agfe.iitkgp.ernet.in
‡ECE Department, Nazarbayev University, Astana, Kazakhstan, KASIT, University of Jordan, Amman, Jordan,
and University of Science and Technology Beijing (USTB), Beijing, China
Email: msobaidat@gmail.com; m.s.obaidat@ieee.org

***Abstract*— In sensor-cloud framework, the concept of virtual sensor provisioning is applied to serve the end-users, who requests sensing information from the deployed sensor network. In a multi-hop sensor-cloud framework, the information collection from the physical sensors to the virtual sensor needs to activate additional nodes for information forwarding to the Cloud Service Provider (CSP). The existing works mainly consider the activation of these nodes from the same sensor owner (SO) and exhibit higher energy consumption. Although, in a sensor-cloud framework, multiple SOs co-exist naturally, and consequently, the service area of these SOs overlap. In this paper, contrasting to the existing works, we argue that the collaboration between the CSP and SOs can improve dynamic virtual sensor provisioning. We propose a scheme named *Intelligent Dynamic Virtual Sensor Provisioning* (*iDVSP*) to enable optimal selection of nodes in a multi-hop path with different SOs. We employ *multi-unit single-item combinatorial reverse auction* to model the interaction between the CSP and SOs. The auction based scheme facilitates the CSP to dynamically negotiate with the SOs, and ensure cost-effective node selection for virtual sensor provisioning. Simulation based results indicate that the proposed scheme is 46.51% energy-efficient compared to existing literature. Furthermore, we observe that the proposed scheme employ fair policy for node selection from different SOs. Therefore, we can argue that the proposed scheme enforces cooperation between the SOs in the sensor-cloud framework.**

***Index Terms*—Virtual sensor, energy-efficiency, sensor-cloud, sensor owner cooperation, auction theory**

I. Introduction

Recent years saw the emergence of a new paradigm named sensor-cloud, which provides an improved infrastructure combining the benefits of cloud computing with traditional Wireless Sensor Networks (WSNs) [1]–[4]. The use of sensor-cloud based architecture provides multiple advantages over the traditional WSNs based deployment of sensors. Using the notion of sensor virtualization, the sensor-cloud infrastructure enhances the real-time information processing and storage abilities of the WSNs. The framework offers ubiquitous information sensing and access to multiple end-users from sensors deployed by different sensor owners over a vast geographical area. Few promising applications of sensor-cloud are in the area of precision agriculture, environmental monitoring, health-care and military.

Yuriyama *et al.* [1], Evenson *et al.* [5] defined the basic framework of sensor-cloud, and characterized the various requirements to enable sensing-as-a-service. Madria *et al.* [3] presented a three-layer protocol stack for sensor-cloud framework. In this protocol stack, the top and bottom layers, namely *client-centric* and *sensor-centric*, are responsible for connecting with the users and the deployed physical sensors, respectively. The *middleware* layer is responsible for provisioning of virtual sensors as per user's request. Additionally, the middleware layer performs the task of managing user accounts and billing. A mathematical model for sensor virtualization in sensor-cloud is presented by Misra *et al.* [6].

A. Motivation

In sensor-cloud, the end users are relieved from the task of network deployment, management and maintenance, which are typically executed by the sensor owners. The Cloud Service Providers (CSPs), using the WSN deployed by different sensor owners (SOs), offer various services for the end users. For each user query, the CSP creates a virtual sensor based on the required parameters mentioned in the query. The middleware layer provisions the allocation of the physical sensors to the virtual sensor. In the literature, the existing schemes [7], [8] consider the allocation of redundant nodes present in region of interest mentioned by the user query. However, such selection of physical sensor nodes is not energy-efficient for the deployed nodes. This technique will increase the maintenance cost of the deployed nodes, and thereby, increasing the price to the end-users. In addition to this, in a multi-hop sensor deployment, the information collection from the selected physical sensor node to the virtual sensor needs to activate additional nodes for information forwarding to the CSP. The existing schemes [9]–[11] mainly consider activation of the nodes from one sensor owner only. [12] proposes a pricing model for sensor-cloud framework which considers a multi-hop topology of sensor deployment with different owners. However, in this work, any deployed node itself select the node for the consecutive hop for transmission of sensed information from the source node to the base station. Thus, such technique increases the energy consumption of the deployed sensor nodes, and increases the overall communication overhead involved in the virtual sensor provisioning. Therefore, it is evident from the existing works, that the virtual sensor provisioning process faces challenges in terms of energy consumption of the deployed nodes. Motivated by this problem, in this paper, we propose a virtual sensor provisioning scheme which exploits the cooperation between the sensor owners and computes an optimal selection of the sensor nodes for virtual sensor provisioning.

B. Contributions

In this paper, we present a scheme named *Intelligent Dynamic Virtual Sensor Provisioning* (*iDVSP*) to enable optimal selection of nodes in a multi-hop path consisting of multiple SOs in sensor-cloud framework. We model the interaction of the CSP and the SOs using *multi-unit single-item combinatorial reverse auction* [13]. In this scenario, the CSP and the SOs act as the buyer and sellers respectively. The CSP has the knowledge about the node deployments by these different owners. Thereby, based on the information of the provisioned sensors, the CSP can easily find out different route options to aid information collection from these provisioned sensors to the cloud. However, these routes are formed with the nodes from different sensor owners. As the cost of the route is private information to the corresponding SO, the overall route's cost cannot be computed by the CSP single-handedly. In our proposed scheme, the final decision is made by the CSP based on the information from the various SOs involved. The proposed scheme exploits the cooperation between the SOs, as the selected route may constitute of nodes from different SOs. Therefore, the *commodities* in the auction are these routes, and the sellers (the SOs) are asked to quote the price for the corresponding route segment only. Therefore, in our model, for each item of commodity there are multiple units. Finally, the CSP, after receiving quotation from different SOs, selects the optimal route. Consequently, the SOs are paid according to the amount (or quantity) of commodity taken from them. In the following, we enlist the *contributions* made in this work.

- We model the interaction between the CSP and the SOs using *multi-unit single-item combinatorial reverse auction*. This model facilitates the CSP to dynamically negotiate with the SOs for virtual sensor provisioning to the end-users.
- Our proposed scheme enables energy-efficient node selection for the CSP and SOs. Thereby, ensuring prolonged network lifetime and long term service availability.
- We device the scheme in such way that the SOs are assured with fair node selection policy.

The rest of the paper is organized as follows. Section II discusses the related works in the literature. We discuss the considered system model for our proposed work in Section III. In Section IV, we present the proposed combinatorial reverse auction based scheme. The performance evaluation of the proposed scheme is presented in Section V. Finally, we conclude the paper in Section VI, citing directions for future works.

II. Related Works

To enable energy-efficient sensor selection in sensor-cloud framework, a dynamic duty scheduling scheme was proposed by Ojha *et al.* [14]. By employing dynamic duty scheduling, the authors show that the lifetime of the deployed nodes enhance. A scheme for enabling optimal composition of virtual sensor from a set of deployed physical sensor nodes, was proposed by Chatterjee *et al.* [8]. The authors employ a *goodness* factor, which is used to map the selection of a physical node to a virtual sensor. An adaptive data caching scheme for sensor-cloud was presented by [7]. In this scheme, the objective was to achieve energy-efficiency of sensor nodes, and thereby enhancing the network lifetime. With the change in physical sensor nodes, the scheme can adaptively select an optimal data caching interval. However, all these schemes did not consider a multi-hop deployment of sensor nodes in a sensor-cloud framework with multiple SOs.

Lemos *et al.* [10], [11] propose virtual sensor provisioning by enabling selection of physical sensors based on similarity of heterogeneous sensors. As a result of such selection, the energy consumption of the nodes reduces. The node selection scheme is based on similarity of measurement between the nodes and not just the inter-node distance. However, these schemes do not consider multi-hop deployment of sensors in a multi-SO sensor-cloud scenario. The "pricing for Hardware (pH)" scheme proposed by Chatterjee *et al.* [12] considers multi-hop topology of sensor deployment with different owners. In this work, the deployed nodes participate in the node selection for the consecutive hop to enable transmission of sensed information from the source node to the base station. However, this technique increases the energy consumption of the deployed sensor nodes, and increases the overall communication overhead involved in the virtual sensor provisioning.

Therefore, it is evident from the existing works, that virtual sensor provisioning faces challenges in terms of energy consumption of the deployed nodes.

III. System Model

Our proposed system is a sensor-cloud framework with a CSP χ having total N number of sensor nodes deployed by m sensor owners represented by the set $\Theta = \{\theta_1, \theta_2, \cdots, \theta_m\}$. We denote the nodes associated with any sensor owner θ_i as N_θ^i. In this framework, each SO has different gateway $G_i \in \mathcal{G}$ for providing connectivity between the deployed nodes and the CSP. The deployed nodes for any SO form a multi-hop topology among themselves. The CSP has the knowledge of the whole topology which comprises of nodes deployed by all SOs. Therefore, the CSP has a global view of the topology while each SO can only have the knowledge of the topology consisting of nodes deployed by itself. Although, the cost related information remain private to each SO only.

The user request to the CSP for sensor information is considered as a query (s_k). The CSP processes the query, creates a virtual sensor for this query, and assigns the physical sensor which can serve the demand. Consequently, for all such user requests $\forall s_k \in \mathcal{S}$, the CSP selects a set of nodes from various SOs. We name this set as CSP selected Duty Nodes (CDNs). However, being a multi-hop topology, the information collection from the CDNs need additional nodes to route the information to the CSP. Typically, such routes are considered to be computed by the corresponding SO. In our proposed scheme, we consider cooperation between the SOs, allowing the CSP to select routes with nodes involved from different SO. The set of all routes for the node i at time t, as computed by the CSP, is denoted by $\lambda_i^t = \{\lambda_{i,1}^t, \lambda_{i,2}^t, \cdots, \lambda_{i,r_i}^t\}$. Here, $r_i = |\lambda_i^t|$ denote the number of routes for the node i.

In Figure 1, we depict the problem scenario discussed in this paper. Here, a provisioned node has three different routes available to send the information to the CSP. Each of these routes consist of nodes owned by different SOs. Also, the SOs can have different number of nodes in different routes. In the proposed reverse auction framework, such routes are the *multiple units* for any *single item* or the corresponding provisioned node. The CSP, using the proposed framework, computes the total cost for each route with the price information quoted by the involved SOs.

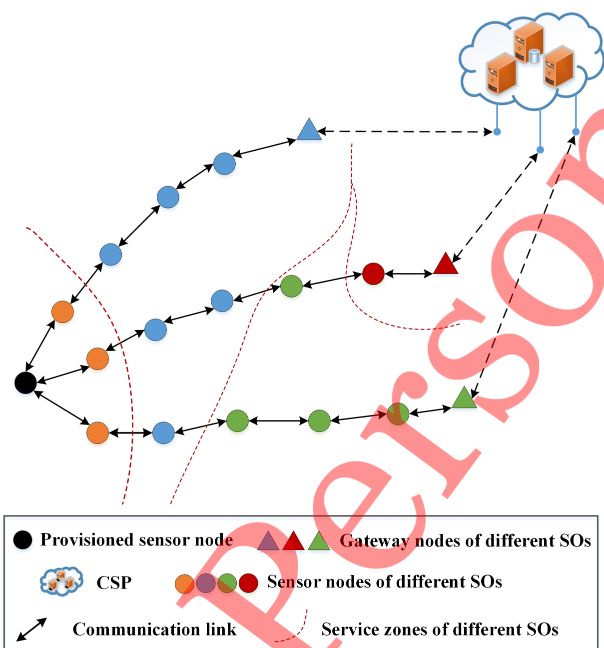


Fig. 1: The problem scenario

IV. iDVSP: Intelligent Dynamic Virtual Sensor Provisioning

In this section, we discuss the proposed *combinatorial reverse auction* [13] based scheme in detail. In the combinatorial reverse auction process, the CSP is the *buyer* and the SOs are the *sellers*. As mentioned previously, the CSP processes the user requests, and provisions the deployed sensor information to them. Consequently, for each such selected node, multi-hop routes are to be formed from the rest of the nodes. Therefore, the *commodity* in this marketplace are these routes formed by nodes from different SOs, and for each *item* (provisioned node) there will be multiple *units* (routes).

The reverse auction process is performed in the following steps. *First*, the CSP finds the nodes requested by the users in each round. For each such node, the CSP finds different route options with nodes from various SOs. These SOs are termed as *potential sellers*. Consequently, it prepares *request for quotation* (*RFQ*) for each SO, filled with information on different route options for each *items*, mentioning the required units of the item for each route. The SOs, prepare *quotation table* (*QT*) for each RFQ for different requested units of the item. Based on the received QTs from different SOs, the CSP decides the units of the items to be procured from them, with the objective of minimizing the cost paid to the SOs.

A. Price Charged by Sensor Owners

Each RFQ ($RFQ_{n,i}$) sent to a SO (θ_i) consists of different options for the routes ($\lambda_{n,j}^t \in \lambda_n^t$) for any sensor node ($n$) which needs to be provisioned. Therefore, each such RFQ is for a specific *item*, and its multiple options contains the different *units* of the *item* requested. These different units are the different nodes required for forming these paths. Each SO finds out the different nodes asked in the RFQ $RFQ_{n,i}$ for each item. Then, the SO also computes the energy consumption status of the requested nodes. In the following, we present the set of regulations for the SO to compute the prices and prepare the QT.

Let the price for k^{th} request in $RFQ_{n,i}$ is $\mathcal{P}_{n,k}^i$, and the SO sells the nodes represented by $\mathcal{N}_{i,n,k}^{rfq}$, where $\mathcal{N}_{i,n,k}^{rfq} = N_\theta^i \cap \lambda_{n,k}^t$, $\forall k \in RFQ_{n,i}$.

- The price charged by the SO increases with higher number of nodes required from this SO ($|\mathcal{N}_{i,n,k}^{rfq}|$). Therefore,

$$\frac{\delta \mathcal{P}_{n,k}^i}{\delta \mathcal{N}_{i,n,k}^{rfq}} > 0 \tag{1}$$

Definition 1. *Previous selection (PS_j^ω) of any node ($j \in N_\theta^i$) at round ω refers to the number of times the node was activated till round ($\omega - 1$).*

- The SO charges higher price for requested nodes which has higher usage than that of the others. The SO can compute the PS (PS_j^ω) value of the nodes requested in this round ($\forall j \in \mathcal{N}_{i,n,k}^{rfq}$). Therefore,

$$\frac{\delta \mathcal{P}_{n,k}^i}{\delta PS_j^\omega} > 0 \qquad \forall j \in \mathcal{N}_{i,n,k}^{rfq} \tag{2}$$

Definition 2. *Request uncertainty factor ($\mu_{i,n,k}$) for any SO is defined as the difference between the maximum and minimum value of the nodes requested in the received*

RFQ from the CSP.

$$\mu_{i,n,k} = \frac{\mathcal{N}^{rfq}_{i,n,k}|_{max} - \mathcal{N}^{rfq}_{i,n,k}|_{min}}{\mathcal{N}^{rfq}_{i,n,k}}$$

where $\mathcal{N}^{rfq}_{i,n,k}|_{max} = \max_{\forall k \in RFQ_{n,i}} \{\mathcal{N}^{rfq}_{i,n,k}|\}$ *and* $\mathcal{N}^{rfq}_{i,n,k}|_{min} = \min_{\forall k \in RFQ_{n,i}} \{\mathcal{N}^{rfq}_{i,n,k}|\}$

- The price is increased when uncertainty ($\mu_{i,n,k}$) is higher. Such model incorporates lower risk for SOs.

$$\frac{\delta \mathcal{P}^i_{n,k}}{\delta \mu_{i,n,k}} > 0 \tag{3}$$

Based on the Equations (1), (2), and (3), we device the price $\mathcal{P}^i_{n,k}$ as,

$$\mathcal{P}^i_{n,k} = |\mathcal{N}^{rfq}_{i,n,k}| \times e_c(i,t) + \sum_{j \in \mathcal{N}^{rfq}_{i,n,k}} \frac{PS^{\omega}_j}{PS^{i,\omega}_{\theta}} + \mu_{i,n,k} \times \eta_{i,t} \tag{4}$$

where $e_c(i,t)$ and $\eta_{i,t}$ denote the unit price for energy cost per node and compensation factor for SO θ_i, respectively. Any SO increases the value of the compensation factor, if the units (of commodity) bought from this SO are less than the average number of units requested during previous t_η times.

Each SO prepares a QT $QT_{n,i}$ for each received RFQ $RFQ_{n,i}$ and send it back to CSP. For any k^{th} request in $RFQ_{n,i}$, the elements are $< \mathcal{P}^i_{n,k}, d^i_{n,k} >$. Here, $d^i_{n,k}$ refers to the delay of the part of route estimated by the cloud. The computation of the delay is presented below.

$$d^i_{n,k} = d \times \frac{\sum_j \rho_j}{|\mathcal{N}^{rfq}_{i,n,k}|} \qquad \forall j \in \mathcal{N}^{rfq}_{i,n,k} \tag{5}$$

where ρ_j is the number of RFQs associated with this node, and d is the unit processing delay for a single request.

B. Route Selection by CSP

The CSP, analyzes the received QTs ($\forall QT_{n,i} \in QT_n$), and decides the route ($\lambda^{t,*}_n$) to select for each of the user requested nodes by procuring optimal units of items from different SOs. For each item or the node $n \in \mathcal{N}^{\omega}_{CSP}$, the CSP computes cost ($\mathcal{C}(\lambda^t_{n,k})$) for each route $\lambda^t_{n,k} \in \lambda^t_n$. The cost calculation of the CSP is governed by the following set of rules.

- The cost ($\mathcal{C}(\lambda^t_{n,k})$) is non-decreasing with increased price ($\mathcal{P}^i_{n,k}$) for any route $\lambda^t_{n,k}$. Therefore,

$$\frac{\delta \mathcal{C}(\lambda^t_{n,k})}{\delta \mathcal{P}^i_{n,k}} \geq 0 \qquad \forall \theta_i \in \Theta \tag{6}$$

- With the increase in the delay ($d^i_{n,k}$) of any route $\lambda^t_{n,k}$, the cost of the CSP increases,

$$\frac{\delta \mathcal{C}(\lambda^t_{n,k})}{\delta d^i_{n,k}} > 0 \qquad \forall \theta_i \in \Theta \tag{7}$$

Based on the Equations (6) and (7), the cost for the CSP is computed as,

$$\mathcal{C}(\lambda^t_{n,k}) = \sum_{\forall \theta_i \in \Theta : N^i_{\theta} \cap \lambda^t_{n,k} \neq \{\emptyset\}} w_i \times \left(\mathcal{P}^i_{n,k} + d^i_{n,k}\right) \tag{8}$$

where $w_i = \frac{|N^i_\theta \cap \lambda^t_{n,k}|}{\cap_{\theta_i \in \Theta} |N^i_\theta \cap \lambda^t_{n,k}|}$ is the weight associated with the price quoted by the corresponding SO, and signifies the number of nodes from this SO compared to the total number of nodes in this route.

The optimal route is selected as,

$$\lambda^{t,*}_n = \arg\min \mathcal{C}(\lambda^t_{n,k}) \qquad \forall n \in \mathcal{N}^{\omega}_{CSP}, \lambda^t_{n,k} \in \lambda^t_n \tag{9}$$

C. Algorithms for Sensor Owners and CSP

The procedures followed by the SOs and CSP are presented in Algorithm 1 and 2, respectively. The CSP first computes the different route options ($\lambda^t_{n,k} \in \lambda^t_n$) for the node ($n$) to be provisioned as per user request. Then, for each route ($\lambda^t_{n,k}$), the CSP creates a list ($\mathcal{L}_{i,t}$) for each SO (θ_i) and add the nodes for which the information is needed. Thereafter, $RFQ_{n,i}$ is populated with information for all route options for any SO θ_i, and send to the SO requesting for QTs. Based on the QTs received from all SOs, the CSP selects the optimal route which provide lower cost among all such options. On the other hand, the SO computes the price ($\mathcal{P}^i_{n,k}$) and delay ($d^i_{n,k}$) for each RFQ.

Algorithm 1: Algorithm for any SO $\theta_i \in \Theta$

Inputs: N^i_θ, PS^ω_j, $\mathcal{L}_{i,t}$.
Output: $QT_{n,i}$.
Receive request for quotations $RFQ_{n,i}$ and $\mathcal{L}_{i,t}$;
for *each* k^{th} *request in* $RFQ_{n,i}$ **do**
  Find out nodes requested $\mathcal{N}^{rfq}_{i,n,k} = N^i_\theta \cap \lambda^t_{n,k}$;
  for *each node* $j \in \mathcal{N}^{rfq}_{i,n,k}$ **do**
    Compute PS^ω_j;
    Compute the number of requests (ρ_j) for j;
  Compute price $\mathcal{P}^i_{n,k}$;
  Compute delay $d^i_{n,k}$;
  Add entry $< \mathcal{P}^i_{n,k}, d^i_{n,k} >$ to $QT_{n,i}$;
Send $QT_{n,i}$ to CSP;
Wait for notification on selected nodes ($N^\theta_i \cap \lambda^{t,*}_n$) from CSP;
Update $PS^\omega_j, \forall j \in \{N^\theta_i \cap \lambda^{t,*}_n\}$;
$\omega \longleftarrow \omega + 1$;

V. Performance Evaluation

A. Simulation Settings

The proposed scheme was simulated using NS-3 (http://www.nsnam.org/). Table I lists the simulation parameters used in our experiments. We considered a single CSP and 5 SOs each with 20 nodes randomly deployed

Algorithm 2: Algorithm for CSP

Inputs: N_θ^i, $\mathcal{N}_{CSP}^\omega$.
Output: Optimal route $\lambda_n^{t,*}$.
Get CDN information;
for *each node* $n \in \mathcal{N}_{CSP}^\omega$ **do**
    Find out routes options for node n, λ_n^t;
    for *each route* $\lambda_{n,k}^t \in \lambda_n^t$ **do**
        for *each SO* $\theta_i \Theta$ **do**
            if $N_\theta^i \cap \lambda_{n,k}^t \neq \{\emptyset\}$ **then**
                Add $N_\theta^i \cap \lambda_{n,k}^t$ to $\mathcal{L}_{i,t}$;
for *each SO* $\theta_i \in \Theta$ **do**
    Send $\mathcal{L}_{i,t}$ information in $RFQ_{n,i}$;
for *each node* $n \in \mathcal{N}_{CSP}^\omega$ **do**
    for *each route* $\lambda_{n,k}^t \in \lambda_n^t$ **do**
        Compute $\mathcal{C}(\lambda_{n,k}^t)$;
    Compute optimal route for node n,
    $\lambda_n^{t,*} \longleftarrow \arg\min \mathcal{C}(\lambda_{n,k}^t)$;
    for *each SO* θ_i **do**
        if $N_\theta^i \cap \lambda_n^{t,*} \neq \{\emptyset\}$ **then**
            Notify about selected nodes $N_\theta^i \cap \lambda_n^{t,*}$;
$\omega \longleftarrow \omega + 1$;

TABLE I: Simulation Parameters

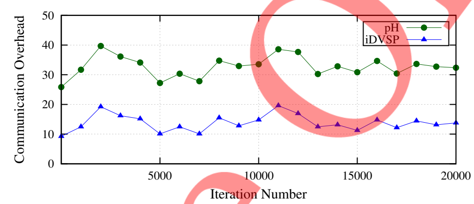
Parameter	Value
Number of nodes	100
Simulation Area	$500\ m \times 500\ m$
Transmission power	$24.75\ mW$ [15]
Reception power	$13.5\ mW$ [15]
Idle power	$13.5\ mW$ [15]
Data rate	$40\ kbps$ [15]
Initial energy of a node	$10\ J$

over the simulation area. The users randomly request for provisioning of any deployed node's information.

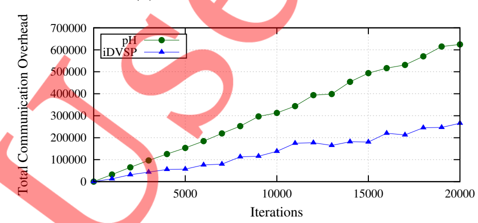
We compare the performance of the proposed scheme with 'Pricing for Hardware' (pH) scheme [12]. We discuss the results for both the schemes with respect to the following metrics – communication overhead, energy consumption, network lifetime, and fairness for SOs.

B. Results

1) Communication overhead: Figure 2(a) presents the results for the communication overhead occurred in individual iterations for both pH and iDVSP. In pH, a node selects the next hop node among its neighbors such that the utility of the sender node is maximum. On the other hand, in iDVSP, the CSP and the SOs jointly participate in node and route selection. In such scenario, the communication overhead for the node selection refers to the number of communications required by the deployed sensor nodes. In our proposed scheme, iDVSP, the total number of such communications is lower compared to the pH scheme. Thereby, the communication overhead in the proposed scheme in 58.56% lower than pH. In Figure 2(b), we present the cumulative communication over the iterations. It is evident from the results that communication overhead for the pH scheme increases than that of the iDVSP in the long run.



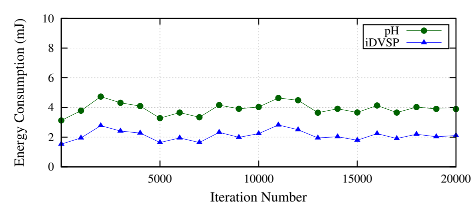
(a) In individual iterations



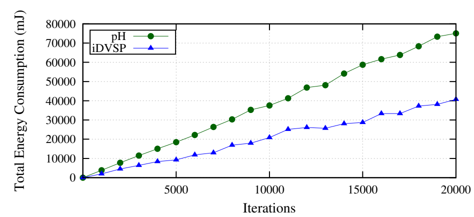
(b) Cumulative over iterations

Fig. 2: Communication overhead

2) Energy consumption: We present the results for energy consumption of the deployed nodes in Figure 3(a) and 3(b). The results signify that iDVSP is 46.51% energy-efficient compared to pH, on an average. In both the schemes, the main reason of energy consumption is the communication overhead, i.e., the additional number of message transmission for enabling sensor provisioning. Consequently, the energy consumption of the nodes vary in different iterations. In long-term, the energy consumption of the nodes increase significantly for pH compared to iDVSP.



(a) In individual iterations



(b) Cumulative over iterations

Fig. 3: Energy consumption

3) Network lifetime: The results for the network lifetime for the deployed nodes in the network is presented in Figure 4. In the experiments, we measure the energy consumption of the nodes for communication overhead in route selection and during information transmission for the provisioned sensors. Consequently, we compute the network lifetime in each iteration for both pH and iDVSP. As evident from the energy consumption profile of the nodes, the remaining network lifetime in iDVSP is higher compared to pH in long-term.

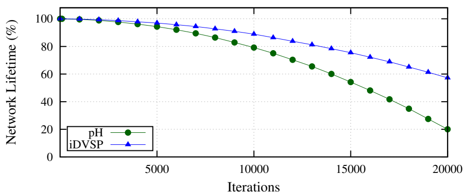
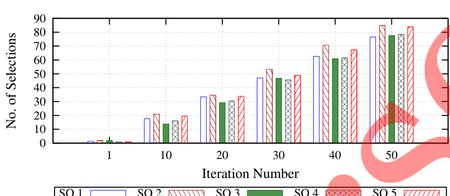
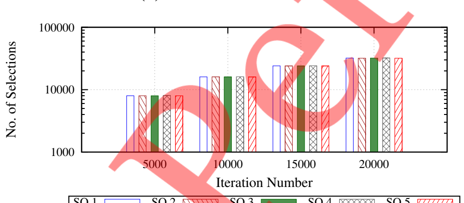


Fig. 4: Network lifetime

4) Fairness for Sensor Owners: We measure the fairness for SOs in the proposed scheme. It is defined as the average number of selection for each SO over the iterations. Figure 5(a) and 5(b) present the number of selections for each SO for the first 50 iterations and in the higher iterations, respectively. The results show that the number of selection for each SO vary nearly 10.84–51.45% in the initial iterations. However, in the higher iterations, the number of selection for each SO becomes nearly equal (difference $<1\%$). Therefore, it is observable that the proposed scheme employ fair policy for the SOs. Furthermore, we can argue that the proposed scheme enforces cooperation between the SOs in the sensor-cloud framework.



(a) In initial iterations



(b) In higher iterations

Fig. 5: Fairness for sensor owners

VI. Conclusion

In this paper, we present a scheme named *Intelligent Dynamic Virtual Sensor Provisioning* (*iDVSP*) to enable optimal selection of nodes in a multi-hop path with different SOs. To enable virtual sensor provisioning, the existing works consider the activation of nodes from the same SO only. Although, in a sensor-cloud framework, multiple SOs co-exist naturally. In iDVSP, we exploit this information. In contrast to the existing works, we argue that the collaboration between the CSP and SOs can improve the virtual sensor provisioning. In the proposed scheme, we model the interaction between the CSP and SOs using the *multi-unit single-item combinatorial reverse auction* [13]. The auction based scheme facilitates the CSP to dynamically negotiate with the SOs, and ensure cost-effective node selection for virtual sensor provisioning. NS-3 based simulation results show the effectiveness of the proposed scheme. Compared to pH [12], iDVSP is 46.51% energy-efficient. Additionally, iDVSP implements fair policy for node selection from different SOs in long-term. In future, we plan to expand the proposed scheme with detailed mathematical model and evaluate the performance by considering various deployment scenarios.

References

[1] M. Yuriyama and T. Kushida, “Sensor-cloud Infrastructure – Physical Sensor Management with Virtualized Sensors on Cloud Computing,” in *Proc. of IEEE Intl. Conf. on Network-Based Information Systems*, Takayama, Japan, 2010, pp. 1–8.

[2] 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 Distributed Sensor Networks*, vol. 2013, pp. 1–18, 2013.

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

[4] T. Ojha, S. Misra, and N. S. Raghuwanshi, “Sensing-cloud: Leveraging the benefits for agricultural applications,” *Computers and Electronics in Agriculture*, vol. 135, pp. 96–107, 2017.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] S. Pal, S. Kundu, M. Chatterjee, and S. Das, “Combinatorial reverse auction based scheduling in multi-rate wireless systems,” *IEEE Trans. on Computers*, vol. 56, no. 10, pp. 1329–1341, 2007.

[14] 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.

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