 doi: 10.1109/TCC.2019.2950396

Dynamic Pricing for Sensor-Cloud Platform in the Presence of Dumb Nodes

Arijit Roy†, *Student Member, IEEE*, Sudip Misra‡, *Senior Member, IEEE*, and Prerona Dutta§
†Advanced Technology Development Centre, ‡Department of Computer Science and Engineering,
§Department of Information Technology,
†‡Indian Institute of Technology Kharagpur, India, §National Institute of Technology Durgapur, India
{†arijitroy, ‡sudipm}@iitkgp.ac.in, §prerona.dutta94@gmail.com

***Abstract*—The presence of dumb nodes in sensor-cloud environment leads to degraded system performance. In this paper, we consider the presence of dumb nodes in the sensor-cloud platform, and thereafter, propose a dynamic pricing scheme, while considering the existence of such nodes in the networks. The existing literature addresses the problem of pricing in sensor-cloud with the assumption of an ideal environment with normally functioning sensor nodes. The proposed pricing model considers the realistic existence of dumb nodes in sensor-cloud platforms. Further, the dumb behavior of a sensor node is dynamic in nature, as it is dependent on environmental conditions such as the occurrence of heavy rainfall, high temperature, and the presence of fog. However, in the absence of such adverse environmental conditions, the erstwhile dumb nodes resume normal behavior. The permanent removal of a dumb node from sensor-cloud is not always a feasible solution. When a dumb node is assigned to a virtual sensor, the existing pricing scheme in sensor-cloud charges same as other normal nodes. Thus, in such a situation, a user pays the normal price for a dumb node to the Sensor-Cloud Service Provider (SCSP). Consequently, the sensor owner of dumb node earn same profit as the owner of a normal node. Therefore, we formulate a scheme for *Dynamic pricing in sensor-cloud environment in the presence of dumb nodes* (*DISCLOUD*). As the presence of dumb nodes in sensor-cloud affects the Quality of Service (QoS), we propose a scheme considering QoS of the sensor-cloud. The proposed scheme, DISCLOUD, enables profit maximization of the SCSP, while considering the price required to be paid by end-user based on QoS.**

Keywords*—Dumb Node, Sensor-Cloud, Dynamic Pricing, Wireless Sensor Network, Quality of Service (QoS).

I. Introduction

A User procures a Wireless Sensor Network (WSN) solution targeting a specific application such as monitoring a particular area, gathering some raw data from an environment, and tracking a moving object. Traditionally, this renders WSNs to be used for single dedicated applications, typically in a single-user centric manner. However, the limitation of single-user centricity can be overcome with the help of *sensor-cloud* platform [1]–[3]. Typically, in sensor-cloud, the same sensor is sharable among multiple users for serving different applications. Among the deployed sensor nodes, homogeneous sensor nodes logically combine to form a virtual sensor (VS) and multiple VSs combine to form a virtual sensor group (VSG). In sensor-cloud, a set of heterogeneous sensor nodes are deployed over different geographical locations by a sensor owner. These sensor nodes are activated based on some application demand by forming distinct virtual sensors. Different sensor owners procure WSNs and lease their sensor nodes out to the SCSP. Further, the SCSP offers *Sensor-as-a-Service* (*Se-aaS*) to the common end-users based on requirements [4], through a Web portal by following pricing mechanism. The SCSP uses this mechanism for the end-users considering his/her own profit and the sensor owner's revenue. A sensor-cloud architecture is based on the pay-per-use model. Thus, an end-user pays for a service as per usage. On the other hand, a sensor owner receives the revenue as per the usage of his/her owned sensor node.

In sensor-cloud, the sensor owner deploys the sensor node once, and gets revenue from the services provided by WSN, to several users through SCSP. A WSN is resource-constrained in nature. Consequently, it is prone to misbehavior, selfishness, and faults. Thus, in a sensor-cloud environment, the sensor nodes may also be selfish, dead, and misbehaving, due to the lack of close observation by the sensor owner. The selfish behavior of a node is intentional, in which the node conserves energy by not forwarding other's data, but it transmits own sensed data to other sensor nodes. The misbehavior of a node may arise due to the occurrence of a fault in the internal circuitry or it is due to malicious programming to misbehave in the network. On the other hand, a dead node arises in a network due to some disaster or drainage of energy in the node. After a certain number of operations, the energy level of the node drops below a threshold value. Consequently, the node is unable to sense or transmit data until the power source is replaced. Therefore, dead nodes occur in the network unintentionally and these nodes are unable to resume their operations permanently. In this work, we focus on a specific and unique misbehavior of sensor nodes – "the dumb" behavior [5], which is taxonomically unintentional [6]. The dumb behavior is dynamic in nature. In the presence of adverse environmental affects (such as high temperature, heavy rainfall, and fog) the communication range of the sensor node shrinks, and in such a situation, all of its neighbor nodes become outside the communication range. Thus, in the presence of such adverse environmental affects a dumb node is unable to communicate with other nodes in the network, whereas in normal environmental conditions, the same node resume normal communication functions. Thus, the permanent removal of dumb nodes from the sensor-cloud is not a worthy solution, as it provides useful services when normal environmental conditions prevail. In this work, our goal is to propose a pricing policy that decides the price for end-users considering the QoS related to dumb behavior of sensor nodes in sensor-cloud.

A. Motivation

In sensor-cloud, the SCSP tries to maximize its profit by providing efficient services from WSNs. On the other hand, as per requirement, an end-user requests Se-aaS through the Web interface on payment basis. Consequently, the end-user expects desirable QoS from SCSP. The presence of dumb nodes in WSN leads to the degradation in performance of the sensor-cloud platform. A dumb node works well when favorable environmental conditions prevail. However, in the presence of adverse environmental conditions, it is unable to transmit data to other nodes. In such a situation, the end-user may not get desirable QoS after paying for the service. Consequently, an end-user must pay to SCSP as per QoS s/he receives from the sensor-cloud. Therefore, a trade-off is required between QoS and pricing, in the presence of dumb nodes in the sensor-cloud infrastructure. As dumb behavior is temporal in nature, permanent removal of a dumb node from the network is not a desirable solution. Thus, SCSP can use the service of dumb nodes when they work normally. The strong motivation behind this work is to design a pricing scheme, while allowing dumb nodes to participate in providing the services under normal environmental conditions, instead of eliminating them permanently from the network. Consequently, the proposed scheme offers a pricing model, while considering the QoS of sensor-cloud infrastructure in the presence of dumb nodes. This work ensures profit maximization of the SCSP with efficient usage of sensor nodes in the presence of dumb behavior.

B. Contribution

In this work, we propose a pricing scheme for sensor-cloud by considering a unique type of misbehavior known as dumb behavior [6], [7]. In the presence of dumb behavior, a node is unable to offer any significant service temporarily. However, when a node behaves normally, we can use its services for the end-user. Thus, we do not eliminate dumb nodes from the sensor-cloud. The specific *contributions* in this work are summarized as follows:

- Dumb behavior is relatively newly explored. Consequently, the existing literature do not consider the dumb behavior for determining QoS. We compute the QoS considering the significant effects of dumb node in sensor-cloud. Further, QoS is used to determine the price which needs to be paid by an end-user.
- Sensor-cloud typically consists of three classes of actors, *viz.*, sensors owners, end-users, and SCSP. Therefore, keeping in mind the business interests involved in the sensor-cloud environment, we have formulated the pricing scheme, which offers equal priority to all actors.
- A utility function is derived for the end-users, while the profit of SCSP and revenue of sensor owner is maximized. Further, we characterized the proposed problem mathematically and analyze the results rigorously.

II. Related Work

This Section discusses the prior work done in the domain of sensor-cloud, pricing, and dumb nodes.

A. Sensor-Cloud and Pricing

Sensor-cloud is a well explored area of research. Yuriyama *et al.* [2] proposed the use of sensor-cloud infrastructure for managing physical sensors through virtualization in cloud. In sensor-cloud, a dynamic group of sensors is formed virtually, as per the requirements of the users. The basic mechanism of virtualization of sensor-cloud was theoretically modeled by Misra *et al.* [1]. Madria *et al.* [3] architected the *Missouri S&T sensor-cloud*, which comprises of different layers. This work provides a detailed architecture, which can be used as a reference for sensor-cloud deployment. Roy *et al*. [8] introduced a caching mechanism for the destroyed virtual machine in sensor-cloud architecture. Kim [9] proposed a two-stage game theoretic scheme for providing on-demand sensing services in sensor-cloud architecture. Neiat *et al.* [10] proposed a sensor-cloud framework, which is composed of different dynamic features. The authors specifically considered the spatio-temporal aspects of sensor-cloud by proposing two algorithms – A^* and $3D$ R-Tree. Security is an essential element which is required for sensor-cloud in order to prevent data theft. Sen *et al.* [11] assessed the risk involved in the use of sensor-cloud infrastructure, and developed an attack graph-based risk assessment framework. Subsequently, the attacks are analyzed with the help of a Bayesian network. In order to avoid resource wastage, Rachkidi *et al*. [12] proposed a scheme for efficient distribution of shared virtual sensors in a sensor-cloud platform. Considering the overlapping sensor deployment region, Roy *et al*. [13] proposed a dynamic virtual sensor formation scheme for the sensor-cloud architecture.

Pricing using cloud is an important aspect. Several prior works are available in the domain of cloud pricing. Son and Sim [14] proposed a price and time-slot negotiation mechanism for cloud. In the proposed work, the authors define a time-slot utility function, for provisioning the preference for different time slots. Considering the flexibility of user requests, Divakaran and Gurusamy [15] proposed a scheme, which helps in designing bandwidth allocation with differential pricing. Prasad and Rao [16] proposed three resource procurement strategies, *viz.*, cloud-dominant strategy incentive compatible (C-DSIC), cloud-Bayesian incentive compatible (C-BIC), and cloud optimal (C-OPT). The authors claim that with the increase in the number of cloud vendors, the resource procurement cost decreases. Mashayekhy *et al.* [17] designed an online mechanism in cloud for Virtual Machine (VM) allocation along with the pricing problem. In the proposed online mechanism, there is no need of prior knowledge about the future demand for VMs. Finally, Chatterjee *et al.* [18] designed a dynamic pricing mechanism for sensor-cloud. The work concerned the pricing of hardware and infrastructure. The proposed solution maximizes sensor owner's profit along with the end-users' utility. Chakraborty *et al*. [19] designed a pricing scheme in sensor-cloud infrastructure, considering the oligopolistic sensor-owners. In the proposed scheme, the authors used Single-Leader-Multiple-Follower Stackelberg game to enforce the trust among the sensor-owner in sensor-cloud.

B. Dumb Nodes

In WSNs, a large number of works explored misbehaviors, faults, and selfishness [20]–[22]. Specifically, dumb behavior [5], [6], a temporal misbehavior, has attracted research attention in the recent years. Kar *et al.* [5] studied the performance effects of WSNs in the presence of dumb nodes, while considering different parameters such as residual energy, percentage of dumb nodes, delivery ratio, and average end-to-end delay. The authors discovered that the performance of a WSN degrades significantly in the presence of dumb nodes in the network. Roy *et al.* [23] proposed an approach using a CUSUM-based technique for detecting dumb nodes in a WSN. A mobile agent-based dumb node detection scheme was proposed by Roy *et al.* [24]. In order to identify dumb nodes in a network, the authors used *Evidence Theory* on an abstracted mobile agent system. Connectivity is an important problem for maintaining the normal functioning of the network. Kar *et al.* [25] proposed a connectivity re-establishment scheme, *CoRAD*, in order to form a connected network in the presence of dumb nodes. The proposed scheme considers that the sensor nodes adjust their respective communication range, and thereafter, proposes a price-based scheme for achieving connectivity among nodes. Considering the fixed communication range of sensor nodes, a connectivity re-establishment scheme, *CoRD*, was proposed by Roy *et al.* [6]. Kar *et al.* [26] solved the problem of connectivity in the presence of dumb nodes in WSN, using learning automata. The authors claimed that the proposed scheme is energy-efficient. On sensor-cloud, specifically, a connectivity re-establishment scheme was designed by Roy *et al.* [7].

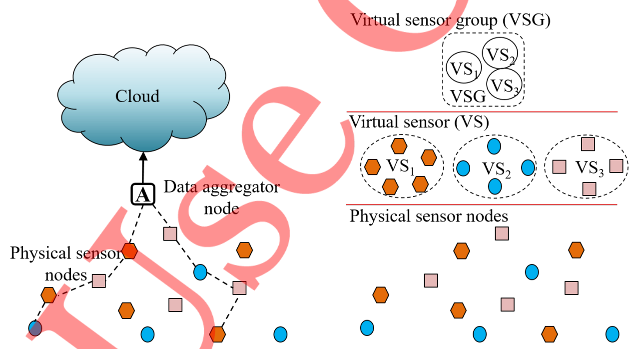
C. Summary of Related Work

There exist significant research efforts individually on sensor-cloud and dumb nodes. Dumb misbehavior of a sensor node is temporal and unintentional, which is different from traditional misbehaviors, faults, and selfish behavior of sensor nodes. However, none of the existing pieces of literature considers the presence of dumb nodes in the sensor-cloud environment. Authors in the existing literature assume that, in sensor-cloud, the nodes function normally. However, one of the fundamental challenges to be faced in the real-life deployment of sensor-cloud is the occurrence of dumb nodes. Additionally, the existing works discuss about the existence of dumb nodes only in the context of WSNs. However, the dynamic behavior of a dumb node may affect the QoS of the sensor-cloud temporarily. The permanent elimination of dumb nodes from sensor-cloud environment may increase the cost for re-deployment of the sensor nodes. Therefore, such a solution approach is impractical. Thus, the presence of dumb nodes in sensor-cloud, is inevitable, should be considered in the study, and accordingly, necessary step(s) should be executed, to maintain the normal operation of the sensor-cloud infrastructure.

III. Problem Description

A. Problem Scenario

We consider a sensor-cloud platform with a certain number of heterogeneous sensor nodes deployed at geographically distant locations. Each of the sensor nodes is owned by a sensor owner. A sensor owner may deploy multiple homogeneous or heterogeneous sensor nodes. The heterogeneous sensor nodes have different individual capabilities of sensing and communicating. In order to provide See-aaS, a sensor node transmits the sensed data to a centralized node through single/multi-hop connectivity. This centralized node is known as data agregator node as shown in Fig. 1(a). Among the deployed sensor nodes, a set of similar type of sensor nodes forms a virtual sensor (VS). Further, these virtual sensors combine to form virtual sensor group (VSG) to serve an end-user application [27]. In the process of forming VS and VSG, users remain unaware that which of the sensor nodes is used for serving his/her application. The mapping of physical sensor to VS and VS to VSG are logical, as depicted in Fig. 1(b). An end-user requests services through a Web interface, and depending on the type of service needed, the sensors are allocated to the end-users virtually. In the process, it illudes that the sensor is dedicated to serve a particular end-user. However, in reality, the sensor may serve the requirement of the other user at the same time. The communication among the nodes takes place through multi-hop mode in order to transmit the sensed data to the server. Due to the presence of adverse



(a) Network view (b) Logical view

Fig. 1: Views of sensor-cloud

environmental conditions, one or multiple sensor nodes becomes dumb. Consequently, these nodes are unable to communicate with other nodes in the network temporarily. The presence of dumb nodes in a network creates hindrance to provide satisfactory QoS to the end-user by SCSP. In such a situation, the SCSP needs to consider the presence of dumb nodes in the network, and thereafter, formulate a mechanism for offering the service. Thus, in this work, our goal is to design a pricing scheme in such a way that the users pay for the service, while the network subsumes the appearance of dumb nodes at the network level. At the same time, the sensor owner as well as the SCSP gain profit out of the payment obtained from the end-users. The system architecture is depicted in Fig. 2.

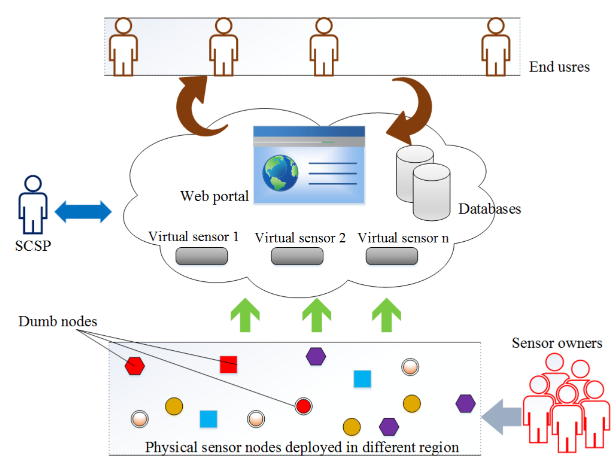


Fig. 2: System Architecture

$$\text{Therefore, } s_{\tau}^{n_i} = \alpha(\nu_{\tau-1}^{n_i} + (1-\alpha)\nu_{\tau-2}^{n_i} + (1-\alpha)^2\nu_{\tau-3}^{n_i} + \cdots + (1-\alpha)^k\nu_{\tau-(k+1)}^{n_i}) + (1-\alpha)^{k+1}s_{\tau-(k+1)}^{n_i} \quad (5)$$

$$s_{\tau(max)}^{n_i} = \alpha\tau(1 + (1-\alpha) + (1-\alpha)^2 + \cdots + (1-\alpha)^k) + (1-\alpha)^{k+1}s_{\tau-(k+1)}^{n_i} \quad (6)$$

B. Assumptions

The list of assumptions in the proposed work are as follows:

- Different sensor nodes are deployed randomly in a large terrain owned by respective sensor owners.
- We assume that at time $t = 0$, all the nodes work under perfect conditions and none is dumb. Here, $t = 0$ signifies the time when the nodes are deployed.
- All the sensor nodes are heterogeneous in nature, and thus, they have different sensing and communication ranges.

C. Problem Formulation

In our system, the sensor nodes are heterogeneous in nature, which indicates that all the sensor nodes may not have the same capability of sensing and communicating. These nodes are deployed randomly over different geographical locations. The components included in the sensor-cloud system are formally defined as follows:

- *Sensor nodes*: Let there be N physical sensor nodes in the system, where S_N represents the set of physical sensor nodes. A physical sensor node is denoted by n_i, such that $n_i \in S_N = \{n_1, n_2, n_3 \cdots, n_N\}$. These nodes are able to provide maximum k distinct services. The available set of services from the deployed sensor nodes are represented by $\mathbb{W} = \{w_1, w_2, w_3, \cdots, w_k\}$. Thus, $w_a \neq w_b \; \forall \; w_a, w_b$.
- *Sensor owners*: The number of sensor owners registered with the sensor-cloud system is denoted by m. Each sensor node n_i is owned by its respective owner o_j. The set of sensor owners is represented as $\mathcal{O} = \{o_1, o_2, o_3, \cdots, o_j, \cdots, o_m\}$. However, one sensor owner can own multiple physical sensor nodes, but one physical sensor node cannot be owned by multiple owners. Therefore, we infer that the mapping from set S_N to set $\mathcal{O}$ is *surjective* or *many-to-one onto*, as depicted in Fig. 3.
- *End user*: At time instant t, the available end-users, who desire to utilize the services offered by the sensor-cloud, is taken to be x. Each end-user is denoted by u_i, who can use maximum k services. Mathematically, the set of end-users is depicted as $\mathbb{U}$, such that $u_i \in \mathbb{U} = \{u_1, u_2, u_3 \cdots, u_x\}$.
- *Virtual sensor*: Among the deployed physical sensor nodes, S_N, a subset of homogeneous sensor nodes form a virtual sensor, Vs_i.

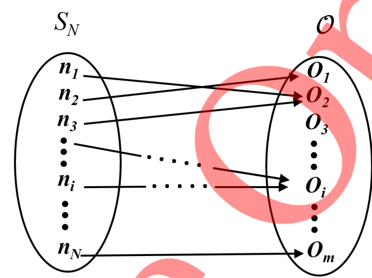


Fig. 3: Mapping of sensor nodes to sensor owners

***Proposition* 1.** *If, at a particular time instant t, the set of total number of virtual sensors that exist in the network is $\mathbb{V}_{max}$, and the sum of all the physical sensor nodes that comprise all the virtual sensors is N_{max}, then $N_{max} \leq N$.*

Proof: For the proof of Proposition 1, refer to the supplementary file. ∎

IV. Solution Approach

Our work focuses on the pricing of the services, considering the QoS of the system. In our system, the QoS, $\mathbb{Q}$, is the function of *Fitness* ($\mathcal{H}_{n_i}$) of any physical sensor node, n_i, and the *Discounting factor* (w_{ni}). Thus, we have,

$$\mathbb{Q} = f(\mathcal{H}_{n_i}, w_{ni}) \quad (1)$$

A. Calculation of QoS Parameters

We introduce the factor, *Reputation*, in order to incorporate the effect of dumb behavior of a node. Reputation, $\mathcal{R}_{n_i}$, of a physical sensor node, n_i is dependent on its *Status Factor*, $\mathcal{F}_{n_i}$, and the mean probability of being dumb, $\Phi_{dumb}^{n_i}$.

***Definition* 1.** *Behavior of a sensor node: Behavior of a sensor node is a boolean value that determines whether a physical sensor node n_i is found to be behaving as dumb at a particular time instant t. The behavior of a physical sensor node n_i, at a time instant t, is denoted by $\mathcal{B}_{n_i}(t)$. Thus, we have,*

$$\mathcal{B}_{n_i}(t) = \begin{cases} 1, & \text{if the node is dumb} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

In order to compute the *Frequency of Occurrence*, ν^{n_i}, the behavior of a physical sensor node, $\mathcal{B}_{n_i}(t)$ is used. $\nu_y^{n_i}$ denotes the number of times a physical sensor node, n_i, is found to be dumb in the y^{th} time slot. To keep the computation simple, we divide the time domain into slots. Each time slot comprises of T time instants. We have,

$$\nu_y^{n_i} = \sum_{t=1}^{T} \mathcal{B}_{n_i}(t) \quad (3)$$

The dumb behavior of a sensor node is temporal in nature. Thus, in order to emphasize on frequency of a node being dumb in previous time instants, we use the *Exponential Moving Average* (*EMA*). Consequently, the parameter,*Status*, is introduced, as shown in Equation (4). However, $s_{\tau}^{n_i}$ represents the status of a physical sensor node n_i over τ time slots, which is obtained using Equation (5).

$$s_0^{n_i} = \nu_0^{n_i} = 0 \qquad\qquad s_t^{n_i} = \alpha\nu_t^{n_i} + (1-\alpha)s_{t-1}^{n_i} \tag{4}$$

Considering the maximum frequency of occurrence in a time-slot to be τ times, the maximum status at any time instant, $s_{\tau(max)}^{n_i}$, is expressed as Equation (6).

***Definition* 2.** *Status Factor $\mathcal{F}_{n_i}$: The Status Factor of a physical sensor node is the ratio of its obtained status, $s_{\tau}^{n_i}$, to its maximum status, $s_{\tau(max)}^{n_i}$, over τ time slots.*

$$\mathcal{F}_{n_i} = \frac{s_{\tau}^{n_i}}{s_{\tau(max)}^{n_i}} \tag{7}$$

We know that the dumb behavior of a node depends on its shrinkage in communication range. Therefore, the probability of being dumb, $\mathcal{P}_{dumb}(t)$, is obtained as:

$$\mathcal{P}_{dumb}(t) = 1 - \frac{r}{R} \tag{8}$$

where r is the current communication range and R is the maximum communication range of a sensor node. Therefore, for generalization, we define the *mean probability of being dumb* $\Phi_{dumb}^{n_i}$ over τ time slots as:

$$\Phi_{dumb}^{n_i} = \frac{\sum_{t=1}^{T.\tau} \mathcal{P}_{dumb}(t)}{\sum_{t=1}^{T.\tau} t} \tag{9}$$

***Definition* 3.** *Reputation: Reputation of any physical sensor node n_i is obtained by subtracting both its mean probability, $dumb^{n_i}$ and the status factor, $\mathcal{F}_{n_i}$, individually, from unity, and thereafter, obtaining their mean. Mathematically,*

$$\mathcal{R}_{n_i} = \frac{(1-\Phi_{dumb}^{n_i}) + (1-\mathcal{F}_{n_i})}{2} \tag{10}$$

Lemma 1. *Reputation of a physical sensor node at a particular time instant t lies within [0,1].*

Proof: For the proof of Lemma 1, refer to the supplementary file. ∎

Reputation of a virtual sensor $\mathcal{R}_{Vs_i}$: The reputation of a virtual sensor node Vs_i is a dynamic parameter, which is obtained by computing the mean of the *Reputations* of all its component physical sensor nodes n_j. $\mathcal{R}_{Vs_i}$ is updated at every time instant (considering that the number of physical sensor nodes in virtual sensor Vs_i is i_n).

$$\mathcal{R}_{Vs_i} = \frac{\sum_{n_j \in Vs_i} \mathcal{R}_{n_j}}{i_n} \tag{11}$$

Motivated by the system model definition of [28] and [29], we include the following parameter.

Signal-to-Interference-and-Noise Ratio $\zeta_{i,i+1}(\rho_i)$:

$$\zeta_{i,i+1}(\rho_i) = \frac{d_{i,i+1}^{-\beta}.\rho_i}{\gamma + \sum_{n_k \in \mathcal{NS}_{n_i}} d_{i,k}^{-\beta}.\rho_k} \tag{12}$$

where $\mathcal{NS}_{nh_i}$ is the set of all neighbor nodes of node nh_i, ρ_i is the transmission power from node n_i, β is the fading factor which results in path loss, $d_{i,j}$ is the Euclidean distance between nodes n_i and n_j, and γ is the parameter that accounts for all the other noise factors that creep in during transmission in the network.

Frame Success Rate $FSR_{i,i+1}(\rho_i)$:

$$FSR_{i,i+1}(\rho_i) = (1 - BER(\zeta_{i,i+1}(\rho_i)))^M \tag{13}$$

where M: frame size or the number of bits in a frame.

B. Fitness Parameters

When a user requests service from the SCSP, the data are transmitted from source to the gateway, through multi-hop communication. Thus, to decide the fittest or the most suitable next intermediate node, every hop node nh_i judges the *Fitness* of its neighbor nodes. Each neighbor node of a node nh_i is denoted by ng_j, such that $ng_j \in \mathcal{NS}_{ng_i}$. The fitness of each neighboring physical sensor node is dependent on the following fitness parameters:

- *Reputation* of the neighbor node ng_j, $\mathcal{R}_{ng_j}$: As defined in Definition 3.
- *Residual Energy* of the neighboring physical sensor node ng_j $\mathcal{RE}_{ng_j}(t)$: It is the ratio of the remaining energy in the neighboring physical sensor node ng_j to its initial energy at the particular time t.

$$\mathcal{RE}_{ng_j}(t) : \frac{E_{remaining}(t)}{E_{initial}} \tag{14}$$

- *Effective Distance* of the neighbor node Ed_{ng_j}:

$$Ed_{ng_j} = \frac{d_{ng_j}}{d_{max}} \tag{15}$$

  where d_{ng_j} is the Euclidean distance between node ng_j and the gateway, and d_{max} is the Euclidean distance between the gateway node and the farthest physical sensor node in the network from the gateway.

Reputation of a sensor node depends on the probability of occurrence of a node to be dumb. Moreover, with the increment of this probability, the reputation of a node decreases. Additionally, the lifetime of a node depends on its residual energy. On the other hand, if the Euclidean distance between two nodes is less, the delay in data delivery between them decreases. Therefore, it is better to select a node, with higher reputation, higher residual energy, and lesser distance as the next hop node. Considering the reputation, residual energy, and the Euclidean distance, we formally define the fitness of a node as:

***Definition* 4.** *Fitness of node ($\mathcal{H}_{ng_j}(t)$): It is a parameter that determines the extent to which a node ng_j at a time instant t is deemed suitable to be the next hop node for service transmission. It is a function of Reputation, $\mathcal{R}_{ng_j}$, Residual Energy, $\mathcal{RE}_{ng_j}(t)$, and Effective Distance, Ed_{ng_j}. We have,*

$$\mathcal{H}_{ng_j}(t) = \mathcal{R}_{ng_j} + \frac{\mathcal{RE}_{ng_j}(t)}{Ed_{ng_j}} \tag{16}$$

Consequently, the neighbor node with the maximum *Fitness* is selected as the next hop node n_{hi}

$$nh_{i+1} = \arg \max_{\forall ng_j \in \mathcal{NS}_{nh_i}} [\mathcal{H}_{ng_j}(t)] \tag{17}$$

As explained in Section III-C, the system model comprises of three broad actors – *the sensor owners* (the ones at the

$$\left(\mathcal{Y}_{n_{h'}} - \mathcal{Y}_{n_{h'-1}}\right)P\left(\mathcal{U} \geq \mathcal{X}^{max}_{u_j,n_{h'}}\right) = \left(\mathcal{Y}_{n_{h'}} - \mathcal{Y}_{n_{h'-1}}\right)\left[\mu_{min} + (1-\mu_{min})\left(\frac{\mathcal{X}^{max}_{u_j,n_{h'}} - \mathcal{Y}_{n_{h'}}}{\mathcal{X}^{max}_{u_j,n_{h'}} - \mathcal{X}^{min}_{u_j,n_{h'}}}\right)\right] \tag{26}$$

$$\mathcal{Y}_i = \begin{cases} \mathcal{X}^{min}_{ui,nj}, & \text{if } \frac{\mu_{min}(\mathcal{Y}_{i-1}+\mathcal{X}^{min}_{ui,nj})-(\mathcal{X}^{max}_{ui,nj}+\mathcal{Y}_{i-1})}{2(\mu_{min}-1)} < \mathcal{X}^{min}_{ui,nj} \\ \frac{\mu_{min}(\mathcal{Y}_{i-1}+\mathcal{X}^{min}_{ui,nj})-(\mathcal{X}^{max}_{ui,nj}+\mathcal{Y}_{i-1})}{2(\mu_{min}-1)}, & \text{if } \frac{\mu_{min}(\mathcal{Y}_{i-1}+\mathcal{X}^{min}_{ui,nj})-(\mathcal{X}^{max}_{ui,nj}+\mathcal{Y}_{i-1})}{2(\mu_{min}-1)} \in [\mathcal{X}^{min}_{ui,nj}, \mathcal{X}^{max}_{ui,nj}] \\ \mathcal{X}^{max}_{ui,nj}, & \text{otherwise} \end{cases} \tag{28}$$

$$\left(\mathcal{Y}_{n_{i-1}} - \mathcal{Y}_{n_{i-2}}\right)P\left(\mathcal{U} \geq \mathcal{X}_{ui,nj}\right) = \left(\mathcal{Y}_{n_{i-1}} - \mathcal{Y}_{n_{i-2}}\right)\left[\mu_{min} + (1-\mu_{min})\left(\frac{\mathcal{X}^{max}_{ui,nj} - \frac{\mathcal{X}^{max}_{ui,nj}+\mathcal{Y}_{n_{i-1}}-\mu_{min}(\mathcal{X}^{min}_{ui,nj}+\mathcal{Y}_{n_{i-1}})}{2(1-\mu_{min})}}{\mathcal{X}^{max}_{ui,nj} - \mathcal{X}^{min}_{ui,nj}}\right)\right] \tag{29}$$

---

service producing end), *the end-users* (the ones at the service consuming end), and the SCSP, keeping this in mind, we have modeled the entire pricing strategy under two sub-strategies:

- End-to-End Pricing
- Service Providing Infrastructure Pricing

C. End-to-End Pricing

The selection of the next-hop node occurs every time a source or an intermediate physical sensor node is active and has to transmit a data packet. Once the next hop node is decided, the sensor node transmits the data to that selected hop node. This process occurs at every intermediate node until and unless the data packet reaches its intended destination gateway node. In order to formulate the end-to-end pricing strategy, we focus on the QoS offered provided by each of the intermediate hop nodes along with the source node. Thus, by adopting the result from [30], we express the QoS, $\mathbb{Q}_{n_i}$, of a node n_i, as:

$$\mathbb{Q}_{n_i} = w_{n_i}.\mathbb{Q}_{n_{i-1}} + \mathcal{H}_{n_i}(t) \tag{18}$$

where, w_{n_i} is a discounting factor, such that,

$$w_{n_i} = \frac{\mathcal{R}_{n_i} FSR_{i,i+1}}{\mathfrak{d}_{n_i}} \tag{19}$$

and $w_{n_i} \in [0,1]$. The effective latency for transmitting the data to its next hop is represented by $\mathfrak{d}$, which is derived as:

$$\mathfrak{d}_{n_i} = \frac{d_{n_i}}{d_{max}} \tag{20}$$

where d_n is the total average delay between node, n_i, to its one-hop neighbor for the last t^{th} instants and d_{max} represents the maximum allowable delay in the network for one-hop packet transmission.

$\mathbb{Q}_{n_1}$ is the QoS of source node. We consider,

$$\mathbb{Q}_{n_1} = w_{n_1} + \mathcal{H}_{n_1}$$

In order to take into account the QoS provided by all the intermediate hop nodes along with the source node, Equation (18) is written as,

$$\mathbb{Q}_{n_i} = \prod_{j=2}^{n_i} w_{n_j}\mathbb{Q}_{n_1} + \sum_{k=2}^{n_i-1}\prod_{l=k+1}^{n_i} w_{n_l}\mathcal{H}_{n_k} + \mathcal{H}_{n_i} \tag{21}$$

Theorem 1. *The maximum attainable value of QoS provided by a physical sensor node depends on the Effective Distance (Ed_{n_i}) of the previous hop nodes. Mathematically,*

$$\mathbb{Q}^{max}_{n_i} = f(Ed_{n_1}, Ed_{n_2}, \cdots, Ed_{n_i}) \tag{22}$$

Proof: For the proof of Theorem 1, refer to the supplementary file. ■

Corollary 1. *The minimum attainable value of QoS, $\mathbb{Q}^{min}_{n_i}$, provided by a physical sensor node depends on the Effective Distance (Ed_{n_i}) of the previous hop nodes.*

Proof: For the proof of Corollary 1, refer to the supplementary file. ■

Motivated by the works of [31], we take the normalized value of QoS as:

$$\mathbb{Q}^{norm}_{n_i} = \frac{\mathbb{Q}^{max}_{n_i} - \mathbb{Q}_{n_i}}{\mathbb{Q}^{max}_{n_i} - \mathbb{Q}^{min}_{n_i}} \tag{23}$$

Pricing for the source node is formulated as a function of the normalized value of the QoS provided by the hop node. Let, $\mathcal{Y}^{max}_{o_j,n_i}$ be the most preferred price desired by the sensor owner o_j for rendering out the service by his/her physical sensor node n_i. Therefore, the price charged by the owner, o_{hj}, of the source node n_{hi}, is denoted by $\mathcal{Y}_{o_j,n_{hi}}$ (or simply $\mathcal{Y}_{n_{hi}}$), and is defined as:

$$\mathcal{Y}_{o_j,n_i} = \mathcal{Y}^{max}_{o_j,n_i}.(1 - \mathbb{Q}^{norm}_{n_i}) \tag{24}$$

Let $\mathcal{X}^{max}_{u_j,n_i}$ and $\mathcal{X}^{min}_{u_j,n_i}$ denote the maximum and minimum prices, respectively, that a user u_j wills to pay for a particular service from a sensor node, n_i. The maximum price that a user should will to pay for a hop node is defined to be related to the QoS provided by the node, and the price charged by the owner of the previous hop node. Thus, we have,

$$\mathcal{X}^{max}_{u_j,n_i} = (1 - \mathbb{Q}^{norm}_{n_i}).(\mathcal{Y}^{max}_{o_j,n_i} - \mathcal{Y}_{n_{i-1}}) + \mathcal{Y}_{n_{i-1}} \tag{25}$$

Therefore, the user accepts the service if and only if the utility of service $\mathcal{U}$ obtained by the end-user is more than or equal to $\mathcal{X}^{max}_{u_j,n_i}$, i.e., $\mathcal{U} \geq \mathcal{X}^{max}_{u_j,n_i}$. Furthermore, on adopting the concept in [32], we propose that, in our model, the price charged by the owner of hop node, $\mathcal{Y}_{o_j,n_i}$ (or simply $\mathcal{Y}_{n_i}$), is also affected by the price charged by the previous hop node, $\mathcal{Y}_{o_j,n_{i-1}}$ (or simply $\mathcal{Y}_{n_{i-1}}$). Thus, motivated by the works of [14] and [32], and considering the number of hop nodes h', we arrive at the mathematical formulation for the price charged for the last hop node, n_{hi}, as in Equation (26). Where μ_{min} is the minimum utility received by an end-user and sensor owner for the deal at their least preferred price [14].

On differentiating Equation (26) w.r.t. $\mathcal{Y}_{n_{h'}}$, we get Equation (27), which represents the optimal price.

$$\mathcal{Y}_{n_{h'}} = \frac{\mu_{min}(\mathcal{Y}_{n_{h'-1}} + \mathcal{X}^{min}_{u_j,n_{h'}}) - (\mathcal{X}^{max}_{u_j,n_{h'}} + \mathcal{Y}_{n_{i-1}})}{2(\mu_{min}-1)} \tag{27}$$

An end-user will be willing to accept a service if and only if the price charged for the services by the sensor owner lies between the most preferred and the least preferred prices. Consequently, considering this fact, the price charged, $\mathcal{Y}_i$, by a sensor owner of node n_i can be expressed as the

$$\text{Maximize } f_2 = \left(\sum \mathcal{D}^u_{vsi} P_{vsi}(t) + BP_{ui}(t) + \sum Reg(t) - \mathcal{E}_{total}(t) - \mathcal{Y}_i^{effective} - c \right) \quad (36)$$

$$Obj = \sum \mathcal{D}^u_{vsi} P_{vsi}(t) + BP_{ui}(t) + \sum Reg(t) - \mathcal{E}_{total}(t) - \mathcal{Y}_i^{effective} - c - \sum_{n_i \in VG_i} m_{ui}^{n_i}(\mathcal{Y}_i) + \lambda \left(\sum_{n_j \in VG_i} \left(\frac{\mathcal{X}^{max}_{ui,nj} - \mathcal{Y}_{n_i}}{\mathcal{X}^{max}_{ui,nj} - \mathcal{X}^{min}_{ui,nj}} \right) - \Delta |VG_i| \right) \quad (39)$$

Equation (28). Thus, the optimal price for the last hop node is represented in Equation (29).

D. Service Providing Infrastructure Pricing

A SCSP recognizes each physical sensor node in the sensor-cloud environment as a 4-tuple entity (p_id, o_j, (x_{n_i},y_{n_i}), ϱ_{n_i}), where p_id is the unique identity number for each physical sensor node, o_j is the owner of n_i, (x_{n_i},y_{n_i}) is the geographical coordinates of its position, and ϱ_{n_i} is the date of registration into the cloud service network.

When an end-user requests services, the SCSP creates a virtual sensors group (also referred to as virtual group), dedicated to serve the request of a single user. Each virtual sensor group contains a number of virtual sensors according to the requirement of the particular user. The cost incurred by the SCSP for each virtual group, $\mathcal{E}_{VG_i}$, includes the creation cost, $\mathcal{E}_{creation_{vg}}$, and maintenance cost of the virtual group, $\mathcal{E}_{Mvg_i}(t - t_0)$. Thus, inspired from the works of [18], $\mathcal{E}_{VG_i}$ is expressed as

$$\mathcal{E}_{VG_i} = \mathcal{E}_{Mvg_i}(t - t_0) + \mathcal{E}_{creation_{vg}} \quad (30)$$

Further, the maintenance cost is expressed as:

$$\mathcal{E}_{Mvg_i}(t - t_0) = \sum_{n_i \in VG_i} \left(CC_{vsi}(t) + M_{vsi}(t - t_0) \right) \quad (31)$$

where $CC_{vsi}(t)$ the creation cost, and $M_{vsi}(t - t_0)$ refers to the maintenance cost of each virtual sensor node. The miscellaneous expenditure that takes into account all other expenditures incurred by the SCSP, at time t, for serving the user's request, is denoted by $\mathcal{E}_{misc}(t)$. It includes expenditures such as deployment of any newly registered sensor node at time t and expenditure for any damage repair in the sensor network, if required, per user, at time t. $\mathcal{E}_{security}$ represents the expenditure for maintaining security of the data transmitted for a user. Therefore, the overall expenditure that the SCSP bears with, for serving a single user's request, is denoted by $\mathcal{E}_{total}(t)$. We have,

$$\mathcal{E}_{total}(t) = \mathcal{E}_{VG_i} + \mathcal{E}_{misc}(t) + \mathcal{E}_{security}(t, w) \quad (32)$$

Considering the price paid by the end-users and the price charged by the sensor owners, the SCSP gauges its own profit.

***Definition* 5.** *Net Profit (NetP): The net profit of the SCSP is obtained by deducting the total expenditure of the SCSP at time t, and the price to be paid to the respective sensor owners for rendering their respective services, $\mathcal{Y}_i^{effective}$, from the total income of the SCSP. We have,*

$$NetP = \sum \mathcal{D}^u_{vsi} P_{vsi}(t) + BP_{ui}(t) + Reg(t) - \mathcal{E}_{total}(t) - \mathcal{Y}_i^{effective} \quad (33)$$

where $BP_{ui}(t)$ is the basic constant price every end-user has to pay for obtaining service at a particular time t and $Reg(t)$ is the price that a new end-user pays for being a registered member user for obtaining services from the sensor cloud environment. Both are decided by the SCSP in a way that maximize its own profits. The overall revenue generated by the SCSP is expressed as the product of the demand, $\mathcal{D}^u_{vsi}$, and the price of the services, $P_{vsi}(t)$. However, the expenditure of the SCSP, due to a virtual sensor group, at time t, is defined to be directly dependent on its demand function $\mathcal{D}^u_{vsi} P_{vsi}(t)$ at time t.

$$P_{vsi}(t) = f(\mathcal{D}^u_{vsi}, \frac{\partial \mathcal{D}^u_{vsi}}{\partial t}, \mathcal{R}_{Vsi},) \quad (34)$$

Therefore, as explained in [18], we expressed Equation (34) mathematically as Equation (35)

$$P_{vsi}(t) = \left[\alpha \frac{\partial \mathcal{D}_{vsi}}{\partial t} + \beta \mathcal{D}_{vsi} + \frac{\partial \mathcal{R}}{\partial t} \right] e^{\Gamma \mathcal{R}(t)} \quad (35)$$

where, α, β and Γ are system modeled constants.

Lemma 2. *Visualizing an initial ideal situation, at one particular time instant, if only any node exists in the network, the profit earned by the SCSP depends only on the price charged for the services rendered and the QoS of the network.*

Proof: For the proof of Lemma 2, refer to the supplementary file. ■

The objective is to maximize the profit of the SCSP, while minimizing the price paid by the end-user. Thus, the maximization for profit is represented in Equation (36), where $BP_{ui}(t)$ is the base price a user pays if it is newly registered into the system and $Reg(t)$ is the registration fees that a sensor node owner pays if it is newly deployed into the system. Again, the price paid by the user for the series of sensor nodes needs to be minimized. Thus, the minimization function of the price paid by the user:

$$\text{Minimize } f_1 = \sum_{n_i \in VG_i} m_{ui}^{n_i}(\mathcal{Y}_i) \quad (37)$$

Therefore,

$$\text{Maximize } f = (-f_1, f_2) \quad (38)$$

subject to, $\sum_{n_j \in VG_i} \left(\frac{\mathcal{X}^{max}_{ui,nj} - \mathcal{Y}_{n_i}}{\mathcal{X}^{max}_{ui,nj} - \mathcal{X}^{min}_{ui,nj}} \right) \geq \Delta |VG_i|.$

where Δ is a constant $[0, 1]$ decided by the SCSP. Therefore, using *Lagrange's multiplier*, the complete objective function is represented in Equation (39).

On differentiating Equation (39) with respect to $\mathcal{D}_{vsi}$ and $\mathcal{Y}_i$ and equating both the results to zero, we observe that, at time instant t^0, when both $\frac{\partial \mathcal{D}_{vsi}}{\partial t^0}$ and $\frac{\partial \mathcal{R}}{\partial t^0}$ are equal to zero, the value of the Lagrange's multiplier λ is:

$$\lambda = \sum_{n_j \in VG_i} \left(\mathcal{X}_{ui,nj}^{max} - \mathcal{X}_{ui,nj}^{min}\right)[i_n + \frac{\partial m_{ui}^{n_i}(\mathcal{Y}_i)}{\partial \mathcal{Y}_i} - 2\beta\mathcal{D}_{vsi} - \Gamma\mathcal{R}(t^0)] \tag{40}$$

Thus, the value of the Lagrange's multiplier enables us to obtain the optimal pricing strategy, constrained by the aforementioned three goals.

Algorithm 1 DISCLOUD

Require:
–*comrange*: Communication range of individual physical sensor node
–$\mathcal{P}_{dumb}$: Probability of being dumb for every individual physical sensor node for all the past $T.\tau$ time instants
–ν: Frequency of occurrence of being dumb for every individual physical sensor node for all the past τ time slots

1: **for** j=1 to last physical sensor node **do**
2: comrange $\leftarrow$ communication range of j;
3: **for** i=1 to last physical sensor node **do**
4: **if** (distance between i and j node $<$ comrange) **then**
5: neighbor of $j \leftarrow$ node i;
6: **end if**
7: **end for**
8: **end for**
9: presenthop $\leftarrow$ source;
10: **while** nexthop $\neq$ destination **do**
11: maximum fitness $\leftarrow$ 0 ;
12: **for** all neighbors of presenthop **do**
13: presenthop send the HELLO packets using multicast;
14: **if** (ACK received from neighbor) **then**
15: Calculate Fitness Parameters of neighbor:
16: $\Phi temp$ of neighbor $\leftarrow$ 1;
17: **for** $timeslot$ = 1 to τ **do**
18: $\nu \leftarrow$ frequency of dumb occurences in 1 $timeslot$;
19: $s_{timeslot} \leftarrow \alpha * s_{timeslot} + (1-\alpha) * s_{timeslot-1}$;
20: **for** time t=1 to T **do**
21: $\mathcal{P}_{dumb} \leftarrow$ dumb probability at time t;
22: $\Phi temp \leftarrow \Phi temp * (1 - \mathcal{P}_{dumb})$;
23: **end for**
24: **end for**
25: $\Phi_{dumb} \leftarrow 1 - \Phi temp$;
26: $\mathcal{R} \leftarrow \frac{(1-\Phi_{dumb})+(1-\mathcal{F})}{2}$;
27: $\zeta_{i,i+1}(\rho_i) \leftarrow \frac{d_{i,i+1}^{-\beta} \cdot \rho_i}{\gamma + \sum_{n_k \in \mathcal{NS}_{n_i}} d_{i,k}^{-\beta} \cdot \rho_k}$;
28: $FSR_{i,i+1}(\rho_i) \leftarrow (1 - BER(\zeta_{i,i+1}(\rho_i)))^M$;
29: $\mathcal{RE}_{n_i}(t) \leftarrow \frac{E_{remaining}(t)}{E_{initial}}$;
30: $Ed_{n_i} \leftarrow \frac{d_{n_i}}{d_{max}}$;
31: Calculate Fitness of neighbor:
32: $\mathcal{H}_{n_i}(t) \leftarrow \mathcal{R}_{n_i} + \frac{\mathcal{RE}_{n_i}(t)}{Ed_{n_i}}$;
33: **if** $(\mathcal{H}_{n_i}(t) >$ maximum fitness) **then**
34: maximum fitness $\leftarrow \mathcal{H}_{n_i}(t)$;
35: nexthop-temp $\leftarrow \mathcal{H}_{n_i}(t)$;
36: **end if**
37: **end if**
38: **end for**
39: **if** (nexthop-temp is not visited earlier) **then**
40: nexthop $\leftarrow$ nexthop-temp;
41: hopcount $\leftarrow$ hopcount + 1;
42: **end if**
43: **end while**

Algorithm 1 depicts the simulation scenario of our proposed scheme, DISCLOUD. In this algorithm, initially, we define different parameters involved in the simulations, such as communication range of the sensor nodes, dumb probability and frequency of occurrence of a node to be dumb. Steps 1-8 identify the neighbor nodes of the j^{th} sensor node. Further, in Steps 9-26, the j^{th} node sends HELLO packet to all its neighbor nodes using multicast and receives the acknowledgment message, ACK. After receiving ACK message, node j identifies the frequency of occurrence of the dumb behavior, probability of being dumb and reputation of its neighbor nodes. In Steps 27-43, the frame success rate is computed with the help of signal-to-interference-and-noise ratio. Thereafter, the j^{th} node calculates the fitness of all its neighbor nodes using frame success rate, residual energy, and the Euclidean distance. The node with the highest fitness is chosen as next hop-node

V. Performance Evaluation

A. Simulation Design

In this section, we evaluate the performance of the proposed algorithm, *DISCLOUD*. Both the concepts of dumb node and sensor-cloud are new. Consequently, due to the lack of availability of literature, which consists the dumb behavior in sensor-cloud environment, it is difficult to compare with other schemes. However, for the sake of generosity, we compare the proposed scheme, *DISCLOUD*, with other pricing schemes – [18] and [33]. *Dynamic Optimal Pricing for Heterogeneous Service-Oriented Architecture of Sensor-cloud Infrastructure* was proposed by Chatterjee *et al.* [18], which, for simplicity, we refer to as *DOPH*. The authors propose a pricing scheme in sensor-cloud without focusing on dumb nodes. We use another scheme, *Packet Trade model* (*PTM*) [33], in order to compare with *DISCLOUD*. *PTM* is a virtual currency-based system used for transmitting the packets in a network. The virtual currency is commonly known as *nugget*. In *PTM* every subsequent hop node has to trade the packet from the previous hop node, in exchange of some number of nuggets in order to transmit the packet.

TABLE I: Simulation Parameters

Parameter	Value
Number of nodes	$25,000$
Number of types of sensor node	10
Simulation area	$5Km \times 5Km$
Communication range	$20 - 80m$
Initial Energy	$2KJ$
μ_{min}	0.01 [14]
Drain efficiency (η)	15.7% [34], [35]
Path-loss exponent (α)	2 [34], [35]
Constant value (ξ)	0.0005 [34], [35]
Transmitting circuitry power (P_{T_0})	$15.9mw$ [34], [35]
Receiving circuitry power (P_{R_0})	$22.2mw$ [34], [35]
Transmitting power (P_T)	$47.75mw$ [34], [35]

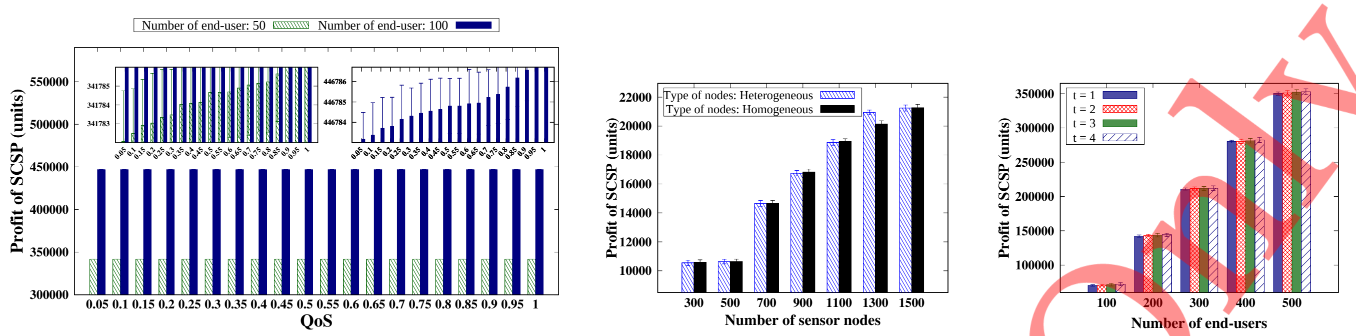
TABLE II: System Configuration

Parameter	Value
Processor	Intel(R) Core(TM) i5-6500 @ 3.20 GHz
RAM	8 GB
Disk Space	500 GB
Operating System	Windows 10
Application Software	MATLAB R2018a

Definition 6. Owner of hop *nodes refers to the owner of intermediate nodes between the physical sensor node, which senses some phenomena and the end-user.*

We evaluated *DISCLOUD* based on the following parameters:

- *Percentage of activated nodes*: The number of active physical sensor nodes in the network per 100 physical



(a) With QoS

(b) With number of sensor nodes

(c) With number of users

Fig. 4: Profit of SCSP

sensor nodes in the entire sensor-cloud environment. Mathematically, $\mathbb{P} = \frac{\mathbb{N}}{N} \times 100$, where $\mathbb{N}$ is the number of activated physical sensor nodes and N is the total number of physical sensor nodes in the sensor-cloud environment..

- *Price charged by the sensor owners*: This is the amount charged by the respective owners of the hop nodes which sense some physical phenomena and the end-user, for the services rendered by their owned physical sensor nodes.
- *Energy consumption*: We divide the energy consumption broadly in two folds:
  - *Computational energy*: The energy consumed to execute the algorithm, *DISCLOUD*, for determining the path along which the data would be transmitted from the source node to the end-user.
  - *Transmission energy*: The energy required to transmit data from the source node to the end-user along the predetermined computed path.

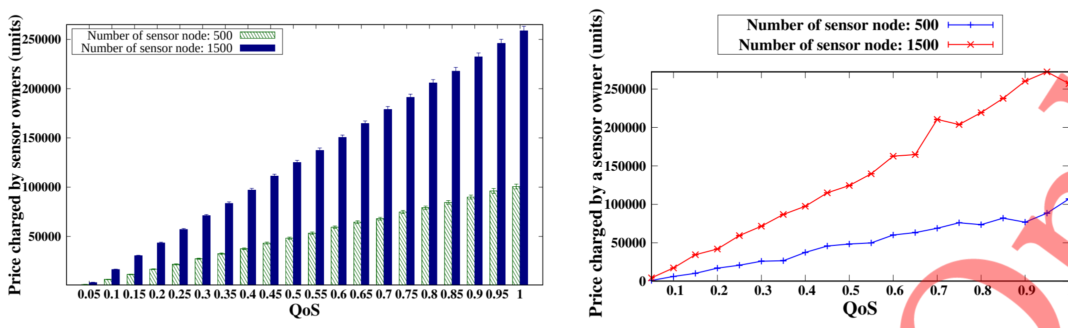
  For executing the scheme, we use the same energy models as the onces used in [34] and [35].
- *Profit of SCSP*: This is the amount of money that is gained by the SCSP, after subtracting the amount to be paid to the respective sensor owners and the maintenance costs of the system, from the total amount received from the end-users as payment for the services.
- *Revenue generated*: It is the gross amount of revenue that is earned by the SCSP and the sensor owners for rendering their respective services via the sensor-cloud environment.

The simulation of our algorithm is done using a realistically occurring scenario. The application area has an impact of heavy rainfall due to which the communication range of the sensor nodes reduce and these nodes become dumb. Every node is given an initial energy of $2nJ$. In this system, we assume that the nodes are non-rechargeable. All the nodes have a normal communication range lying uniformly in the range 20-80m, irrespective of the function or type of sensing they perform. Initially, at time $t = 0$, all nodes are assumed to be working in the perfect state, and therefore, their probability of being dumb and the frequency of occurrence (FoO) are taken to be zero. For the purpose of simulating the energy of computation, the size of the *HELLO* and *Acknowledgment* (*ACK*) packets are taken to be $8bits$ and $2bits$, respectively. The list of simulation parameters used is shown in Table I. Additionally, Table II depicts the system configuration on which performance of the proposed scheme were evaluated.

B. Results

In Fig. 4(a), we consider the presence of 50 and 100 end-users with the variations of the value of QoS from $0 - 1$, for analyzing the profit of SCSP. We observe that the average profit of the SCSP increases with the value of the QoS of sensor nodes. Moreover, profit of the SCSP is always higher in the presence of 100 end-users than in the presence of 50 end-users. Thus, from Fig. 4(a), we infer that the profit of the SCSP increases with an increase in the number of end-users and QoS. Fig. 4(b) depicts the variations in profit with the number of sensor nodes in the network. We vary the number of sensor nodes from 300-1500 with a step of 100 nodes. Also, we consider both the presence of homogeneous and heterogeneous sensor nodes in the networks. Unlike homogeneous sensor nodes, the heterogeneous sensor nodes have variable communication range. Thus, in certain cases, the communication range of a heterogeneous sensor node is significantly less as compared to a homogeneous sensor node. Additionally, in such a situation, the effect of adverse environment results in the shrinkage in their communication range. Therefore, in this figure, we observe that the profit of SCSP in the presence of homogeneous nodes is higher, as compared to the heterogeneous nodes. Fig. 4(c) shows the trend of the profit gained by the SCSP with varying number of users in the network. The increase in profit is steady with the increase in the number of end-users. Therefore, more the number of end-users, the higher are the renderable services. Consequently, higher amount of payments for services is possible. On looking closely, it can also be noticed that, although gradually, the profit of the SCSP increases steadily with time. The possible reason behind the pattern of the curve is linked to the increasing demand of the sensor nodes with time.

Fig. 5(a) depicts the average price charged by the sensor owners in the presence of 500 and 1500 sensor nodes, while varying the QoS from $0 - 1$. In this figure, the price charged by the sensor owners increases with the increasing value of QoS. Further, we observe for all the values of QoS, the charged price by the sensor owner is higher in case of 1500 sensor nodes than 500 sensor nodes. Fig. 5(b) depicts the variations in the price charged by a sensor-owner, considering the presence of 500 and 1500 sensor nodes in the network. In this figure, we notice for every value of QoS, the price



(a) With QoS (considering a set of sensor owners)

(b) With QoS (Considering a sensor owner)

Fig. 5: Change in price charged by sensor owners with QoS

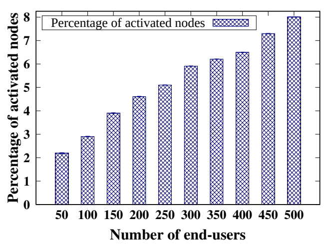


Fig. 6: Activated nodes

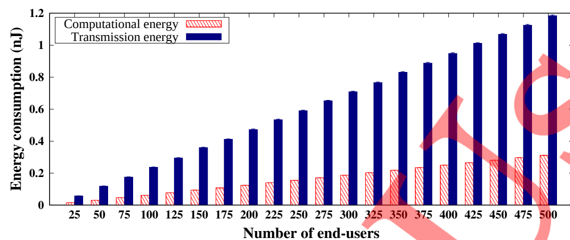


Fig. 7: Energy consumption

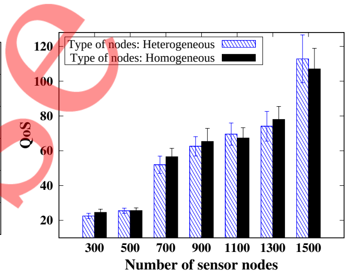
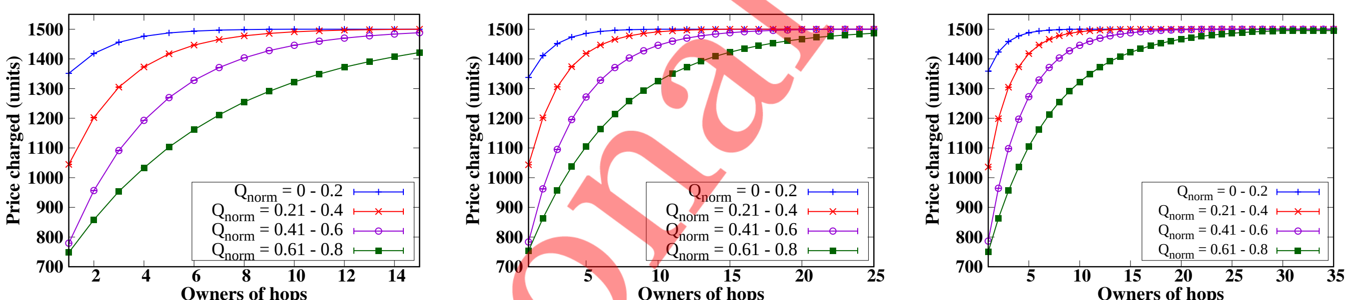


Fig. 8: Change in QoS



(a) Number of hop nodes=15

(b) Number of hop nodes=25

(c) Number of hop nodes=35

Fig. 9: Convergence of price charged with different number of hop nodes

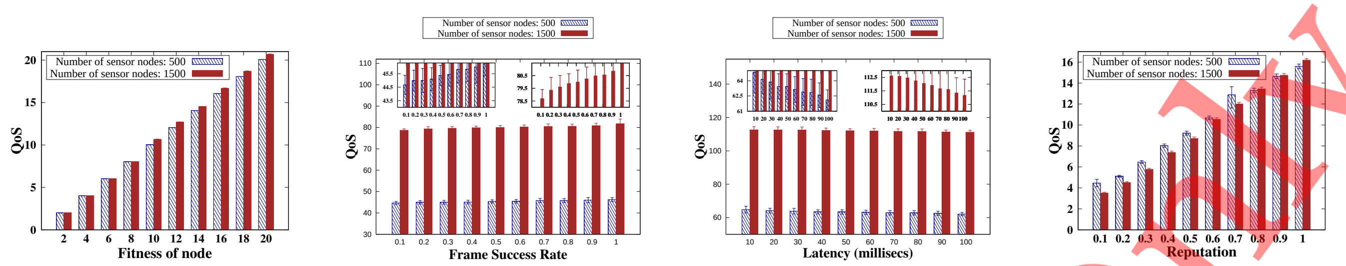
charged by the sensor owner is higher in case of 1500 sensor nodes than that of 500 sensor nodes.

Fig. 6 depicts the variation in the total percentage of activated sensor nodes in the network, with the variation in the number of active users in the network. We observe that, though not linear, it is a steadily increasing curve. With the increase in the number of end-users, the types of application also increases. Consequently, more number and types of sensor nodes are required to be participate for proving the services.

One of the most important factors in assessing the viability of an algorithm is its energy cost. Fig. 7 depicts the energy consumed by the proposed scheme, DISCLOUD, with the varying number of users in the interval $25 - 500$. We consider two types of energy consumption in this figure – *computational energy* and *transmission energy*. It is visible that the energy consumed to compute the optimal path for data transfer is significantly lower than the actual total energy that is consumed to transfer the data from the source node to the destination. Along with this, we observe that, with the increase in the number of end-users, the energy consumption also increases. Moreover, for each case, the computational energy consumption is much lower than the energy consumption for transmission.

In Fig. 8, we depict the result of the effects on QoS in the presence of 300-1500 homogeneous and heterogeneous nodes. In this analysis, we observe a general trend of higher QoS in the presence of homogeneous sensor nodes than heterogeneous sensor nodes. The possible reason, for the trend in this analysis, is similar as discussed for Fig. 4(b).

Fig. 9 shows how the QoS provided by the sensor node primarily determines the price charged by the owners of hop nodes. We consider two cases as shown in Figs 9(b) and 9(c). In both the cases we observe that the pattern of the curves remain similar. In Figs. 9(a), 9(b), and 9(c), the total number of hop nodes considered are 15, 25, and 35, respectively. We observe that when the QoS lies in the lowest range, i.e $\mathbb{Q}^{norm} = [0.61, 8]$, the price charged by the owner of the hop



(a) With fitness (b) With FSR (c) With latency (d) With reputation

Fig. 10: Effect on QoS

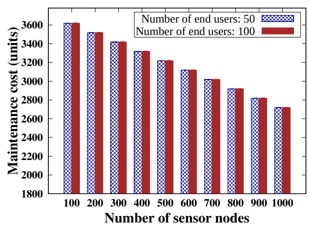


Fig. 11: Maintenance cost

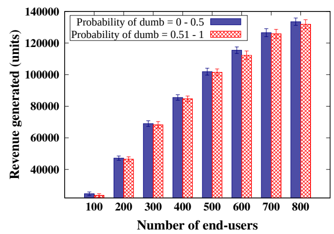


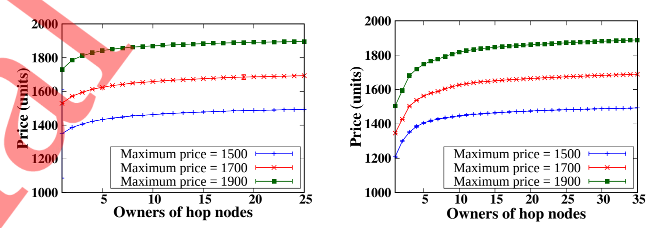
Fig. 12: Revenue

node is lesser as compared to the other three $\mathbb{Q}^{norm}$ ranges. Similarly, for the higher QoS cases, the curve grows steeper. Thus, it can be inferred from the plots that a fair price can be charged to the end-users as per the QoS requirements. Also, it can be noted that, with the increase in the number of intermediate hop nodes, the price charged tends towards the maximum or most preferred price, $\mathcal{Y}^{max}_{o_j,n_i}$. Further, we infer from the figure that after certain number of participating sensor owners (owner of hops), the price charged by them remains constant and converges at the same value.

Fig. 10 represents the effect on QoS by the *fiteness*, *FSR*, *delay*, and *reputation* in the presence of 500 and 1500 sensor nodes. In Figs. 10(a) and 10(b), we observe that the value of QoS are increasing with the increasing value of finess and FSR. However, the delay has a negative effect on the QoS. Therefore, we see that with the increasing amount of delay, the QoS decreases. Finally, we observe that the QoS is increases with increasing value of reputation.

We also evaluate the maintenance cost with the variations in the number of sensor nodes, in the presence of 50 and 100 end-users, as shown in Fig. 11. For maintaining the QoS of a VS, different expenditures such as cost for changing of power source and rectifying the fault in the physical sensor nodes are involved. In this figure, along the x-axis, the total number of sensor nodes varies from $100 - 1000$, with a step of 100. We observe that there is a decreasing trend in the plot with increasing number of sensor nodes. The possible reason for this trend is–when the number of sensor nodes in a network is more, the options for serving an application also increases. Consequently, a particular sensor node does not serve during a long time, which decreases the maintenance cost.

From Fig. 12, we are able to get a graphical representation of how the total revenue generated in the sensor-cloud environment is affected by the mean probability of being dumb, $\Phi^{n_i}_{dumb}$, of the sensor nodes. It is observed for every case that when $\Phi^{n_i}_{dumb}$ of all the active sensor nodes lies in the range $[0, 0.5]$, the revenue generated is higher than when $\Phi^{n_i}_{dumb}$ values lie in the range $[0.51, 1]$. The possible reason behind this is that, in our proposed scheme, the price charged by the sensor owners is dependent of the QoS provided by the sensors. $\Phi^{n_i}_{dumb}$ plays a major role in determining the QoS. Therefore, if $\Phi^{n_i}_{dumb}$ is low, then the QoS provided is high, and the price charged by the owners is high. Consequently, the end-users have to pay higher price for the services they receive, and the SCSP also gains higher profit. As a result, the overall revenue generated increases. It is also obvious that the revenue is positively affected by the swell in the number of end-users, since it would lead to higher request of services and hence greater money earned.

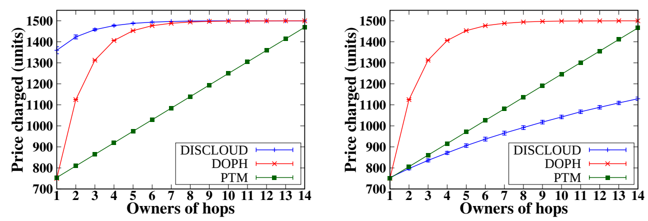


(a) Number of hop node=25 (b) Number of hop node=35

Fig. 13: Amount earn by the sensor owners

The strictly non-decreasing nature of the curve is clearly portrayed in Fig. 13. This figure depicts the analysis of amount earned by a sensor owner, considering the number of sensor nodes owned by them. It can be observed that irrespective of the total number of hop nodes, the nature of the curve of the price charged by the owners of the subsequent hop nodes is a strictly non-decreasing curve. It can be observed that with increasing values of the maximum or most preferred price, $\mathcal{Y}^{max}_{o_j,n_i}$, the curve shifts upwards. Consequently, the prices charged by the respective sensor owners increase.

The pricing schemes in the sensor-cloud environment are not much explored. Ours is a fairly new concept, specifically considering the case of the presence of dumb nodes. Yet, in order to portray its viability and applicability, in Fig. 14, we have compared our scheme, DISCLOUD, with two premier concepts in the field of network pricing, viz., DOPH and PTM, as already mentioned in Section V-A. DOPH deals with the pricing mechanism of traditional sensor-cloud architecture, in which the authors consider the presence of normal sensor nodes. In DOPH, the authors used received signal strength, residual energy of a sensor node, and the Euclidean distance between the sensor node and base station for calculating QoS. On the other hand, PTM is based on a virtual currency system. This scheme enforces forwarding the packet to its next hop.



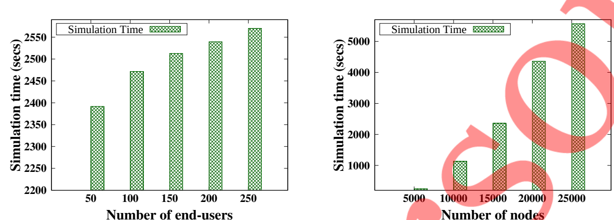
(a) High QoS

(b) Low QoS

Fig. 14: Pricing in different schemes

For simplicity, we illustrated two scenarios – the case when the QoS of the services provided is high ($Q_{norm} = 0.61-8$), as in Fig. 14(a), and the other when the QoS of the services provided is low ($Q_{norm} = 0-0.2$), as in Fig. 14(b). It can be distinctly observed that, in both the cases, the price charged as per DISCLOUD is more in accordance with the QoS. This means, in Fig. 14(a), when the QoS provided is high, the price charged by the owners of the hop nodes is greater than or equal to that of both PTM and DOPH. Similarly, in case of low QoS, the owner of the hops charges lower prices, in case of DISCLOUD, than charged by both DOPH and PTM. Therefore, it is evident from our analysis that DOPH and PTM charge for the service, irrespective of QoS of the sensor nodes. Hence, we conclude that, keeping in mind the QoS of the services provided, the DISCLOUD pricing scheme is a fairer scheme.

The configuration of the system, in which the simulation was executed, is depicted in Table II. The performance of the proposed scheme may vary with the configuration of different machines. Thus, it is very essential to analyze the performance in terms of time in this context. Figs 15(a) and



(a) With respect to number of end-users

(b) With respect to number of nodes

Fig. 15: Simulation Time

15(b) depict the simulation time required for DISCLOUD. Fig. 15(a) shows the variation in the simulation time with the variation in the number of users. Fig. 15(b) shows the change in simulation with time with increase in the number of sensor nodes in the network, while keeping the number of end-user constant. In these figures, we observe that the simulation is increased by 6.9% with varying number of users from $50-250$, and 95% with varying number of sensor nodes from $5000-25000$.

Example: ***Sensor-cloud for Agricultural Application***
In this example, we discuss about an agricultural sensor-cloud service in the presence of dumb nodes. Let there is a large agricultural field, which is divided among few farmers. The field is deployed with a sensor-cloud architecture which is owned by an SCSP, Mr. S. Further, 10 sensor owners, $so_1, so_2, so_3, \cdots, so_n$ deployed different types of sensor nodes over the field, in order to monitor different agricultural parameters. Let there be a farmer, Mr. F, who has subscribed to the service of agricultural intrusion detection from Mr. S. The intrusion detection system includes long-range proximity sensor (owned by the sensor owner so_3), long-range ultrasonic sensor (owned by the sensor owner so_5), and the camera sensor (owned by the sensor owner so_9). Let us assume that the proximity sensor node, p_1, had experienced dumb behavior twice in the last 10 days, and accordingly, the QoS is calculated for p_1. Farmer, Mr. F, pays a rent of x units for p_1. At time instant, t, p_1 becomes dumb, and therefore, it is unable to provide the service to the intrusion detection system. However, in order to continue the service of agricultural intrusion detection system, another proximity sensor node, p_3, is assigned. The history of occurrence of dumb behavior in p_3 is thrice in the past 10 days. Consequently, the QoS of p_3 is less than the QoS of p_1. Thus, the current payment for the proximity sensor node should be reduced, such that $y < x$, where y is the current payment. QoS affects the payment made by the end-users, profit of SCSP, and the sensor owners. Therefore, it is essential to design a scheme, which is capable to handle the pricing in sensor-cloud, considering the QoS, in the presence of dumb sensor nodes. Our proposed scheme, DISCLOUD, is explicitly designed to handle the pricing for sensor-cloud architecture, in the presence of dumb sensor nodes.

VI. Conclusion

In this work, we considered the presence of dumb nodes in the sensor-cloud environment. A dumb node works normally in the absence of adverse environmental effects. However, on the onset of adverse environmental conditions such as high temperature, heavy rainfall, and fog, a dumb node is unable to communicate with other nodes. In the sensor-cloud environment, the users pay as per the use of services from SCSP. The presence of dumb nodes affects the normal performance of sensor-cloud. Consequently, the sensor-cloud, which contains the dumb nodes, does not provide the desirable services even after paying a normal amount by the users. Thus, by taking the issue of the presence of dumb nodes in a sensor-cloud environment into account, we proposed a pricing scheme, DISCLOUD, which considers the users' prices, the sensor owner revenue, and the profit of SCSP. Finally, we compared the proposed scheme, DISCLOUD, with the existing schemes PTM and DOPH.

In the future, we plan to extend our work by establishing the connectivity between a dumb and other nodes in the sensor-cloud environment. This scheme of temporary dumb node replacement will possibly result in offering improved performance and QoS of sensor-cloud. Initially, voting theory is required to identify a dumb node in the sensor-cloud environment. Thereafter, in order to replace dumb nodes with other physical sensor nodes, the problem can be formulated as a normal optimization problem. Additionally, we plan to design a scheme, in future, for estimating the maximum and minimum payments, optimally, by an end-user in the sensor-cloud architecture.

VII. Acknowledgment

The first author of this work is partially funded by project file no. 9/81(1293)/17 sponsored by the Council of Scientific and Industrial Research (CSIR), Govt. of India.

REFERENCES

[1] S. Misra, S. Chatterjee, and M. S. Obaidat, "On Theoretical Modeling of Sensor Cloud: A Paradigm Shift From Wireless Sensor Network," *IEEE Systems Journal*, no. 99, pp. 1–10, 2014.

[2] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing," in *Proceedings of the* 13^{th} *International Conference on Network-Based Information Systems*, Sept 2010, pp. 1–8.

[3] S. Madria, V. Kumar, and R. Dalvi, "Sensor Cloud: A Cloud of Virtual Sensors," *IEEE Software*, vol. 31, no. 2, pp. 70–77, Mar 2014.

[4] S. Chatterjee, S. Misra, and S. Khan, "Optimal Data Center Scheduling for Quality of Service Management in Sensor-cloud," *IEEE Transactions on Cloud Computing*, no. 99, 2015.

[5] S. Misra, P. Kar, A. Roy, and M. S. Obaidat, "Existence of dumb nodes in stationary wireless sensor networks," *Journal of Systems and Software (Elsevier)*, vol. 91, pp. 135 – 146, 2014.

[6] A. Roy, S. Misra, P. Kar, and A. Mondal, "Topology Control for Self-Adaptation in Wireless Sensor Networks with Temporary Connection Impairment," *ACM Transactions Autonomous and Adaptive System*, vol. 11, no. 4, pp. 21:1–21:34, 2017.

[7] A. Roy, A. Mondal, and S. Misra, "Connectivity Re-establishment in the Presence of Dumb Nodes in Sensor-Cloud Infrastructure: A Game Theoretic Approach," in *Proceedings of the* 6^{th} *IEEE International Conference onCloud Computing Technology and Science (CloudCom)*, Dec 2014, pp. 847–852.

[8] A. Roy, S. Misra, and S. Ghosh, "QoS-Aware Dynamic Caching for Destroyed Virtual Machines in Sensor-Cloud Architecture," in *Proceedings of the 19th International Conference on Distributed Computing and Networking*, ser. ICDCN '18. ACM, 2018, pp. 28:1–28:7.

[9] S. Kim, "An Effective Sensor Cloud Control Scheme Based on a Two-Stage Game Approach," *IEEE Access*, vol. 6, pp. 20 430–20 439, 2018.

[10] A. G. Neiat, A. Bouguettaya, T. Sellis, and Z. Ye, "Spatio-temporal Composition of Sensor Cloud Services," in *IEEE International Conference on Web Services (ICWS)*, June 2014, pp. 241–248.

[11] A. Sen and S. Madria, "Risk Assessment in a Sensor Cloud Framework Using Attack Graphs," *IEEE Transactions on Services Computing*, no. 99, 2016.

[12] E. E. Rachkidi, N. Agoulmine, N. Chendeb, and D. Belaid, "Resources optimization and efficient distribution of shared virtual sensors in sensor-cloud," in *Proceedings of IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[13] C. Roy, A. Roy, and S. Misra, "DIVISOR: Dynamic virtual sensor formation for overlapping region in IoT-based sensor-cloud," in *IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–6.

[14] S. Son and K. M. Sim, "A Price- and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 713–728, June 2012.

[15] D. M. Divakaran and M. Gurusamy, "Towards Flexible Guarantees in Clouds: Adaptive Bandwidth Allocation and Pricing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1754–1764, June 2015.

[16] A. S. Prasad and S. Rao, "A Mechanism Design Approach to Resource Procurement in Cloud Computing," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 17–30, Jan 2014.

[17] L. Mashayekhy, M. M. Nejad, D. Grosu, and A. V. Vasilakos, "An Online Mechanism for Resource Allocation and Pricing in Clouds," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1172–1184, April 2016.

[18] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic Optimal Pricing for Heterogeneous Service-Oriented Architecture of Sensor-cloud Infrastructure," *IEEE Transactions on Services Computing*, no. 99, 2015.

[19] A. Chakraborty, A. Mondal, A. Roy, and S. Misra, "Dynamic Trust Enforcing Pricing Scheme for Sensors-as-a-Service in Sensor-Cloud Infrastructure," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[20] E. Soltanmohammadi, M. Orooji, and M. Naraghi-Pour, "Decentralized Hypothesis Testing in Wireless Sensor Networks in the Presence of Misbehaving Nodes," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 205–215, Jan 2013.

[21] V. Chatzigiannakis and S. Papavassiliou, "Diagnosing Anomalies and Identifying Faulty Nodes in Sensor Networks," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 637–645, May 2007.

[22] S. K. Das, B. J. Saha, and P. S. Chatterjee, "Selfish node detection and its behavior in WSN," in 5^{th} *International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, July 2014, pp. 1–6.

[23] A. Roy, P. Kar, S. Misra, and M. S. Obaidat, "D3: Distributed Approach for the Detection of Dumb Nodes in Wireless Sensor Networks," *International Journal of Communication Systems*, 2014.

[24] A. Roy, P. Kar, and S. Misra, "Detection of dumb nodes in a stationary wireless sensor network," in *Annual IEEE India Conference (INDICON)*, Dec 2014, pp. 1–6.

[25] P. Kar, A. Roy, and S. Misra, "Connectivity Reestablishment in Self-Organizing Sensor Networks with Dumb Nodes," *ACM Trans. Auton. Adapt. Syst.*, vol. 10, no. 4, pp. 28:1–28:30, Jan 2016.

[26] P. Kar, A. Roy, S. Misra, and M. S. Obaidat, "Energy-efficient connectivity re-establishment in WSN in the presence of dumb nodes," in *Proceedings of IEEE International Conference on Communication Workshop*, June 2015, pp. 1485–1490.

[27] S. Chatterjee and S. Misra, "Optimal composition of a virtual sensor for efficient virtualization within sensor-cloud," in *IEEE International Conference on Communications (ICC)*, June 2015, pp. 448–453.

[28] H. E. Hammouti, L. Echabbi, and Y. Benmaissa, "Maximizing QoS in heterogeneous Wireless Sensor Networks: Memory and time considerations," in *the* 10^{th} *International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2015.

[29] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, Feb 2002.

[30] E. N. Ciftcioglu, A. Yener, and M. J. Neely, "Maximizing Quality of Information From Multiple Sensor Devices: The Exploration vs Exploitation Tradeoff," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 883–894, Oct 2013.

[31] L. Taher, R. Basha, and H. E. Khatib, "Establishing association between QoS properties in service oriented architecture," in *International Conference on Next Generation Web Services Practices (NWeSP'05)*, Aug 2005.

[32] R. K. Lam, D. M. Chiu, and J. C. S. Lui, "On the Access Pricing and Network Scaling Issues of Wireless Mesh Networks," *IEEE Transactions on Computers*, vol. 56, no. 11, pp. 1456–1469, Nov 2007.

[33] L. Buttyan and J. P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *the* 1^{st} *Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2000, pp. 87–96.

[34] S. Misra, G. Mali, and A. Mondal, "Distributed topology management for wireless multimedia sensor networks: exploiting connectivity and cooperation," *International Journal of Communication Systems (Wiley)*, vol. 28, no. 7, 2015.

[35] Q. Wang, M. Hempstead, and W. Yang, "A Realistic Power Consumption Model for Wireless Sensor Network Devices," in *the* 3^{rd} *Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, Sept 2006, pp. 286–295.

Arijit Roy is a prestigious Council of Scientific & Industrial Research Senior Research Fellow (CSIR-SRF) and pursuing his PhD from the Indian Institute of Technology Kharagpur, India. He received his MS degree in Information Technology from Indian Institute of Technology Kharagpur, India in 2015. Prior to that, he has completed the B. Tech degree in Information Technology from the West Bengal University of Technology, India in 2010. He worked as a Network Engineer in IntecInfonet Pvt. Ltd. Mr. Roy received several academic awards such as Smasung Innovation Award and IBM Day Award. One of his research projects was selected among the top 9 worldwide in IEEE ComSoc Student competition in 2014. The current research interests of Mr. Roy include Internet of Things, Cloud Computing, and Wireless Body Area Networks. He is student member of the IEEE.

Dr. Sudip Misra received the bachelors degree from the Indian Institute of Technology, Kharagpur, India; the masters degree from the University of New Brunswick, Fredericton, NB, Canada; and the Ph.D. degree in computer science from Carleton University, Ottawa, ON, Canada. He is an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. Prior to this, he was with Cornell University, Ithaca, NY, USA; Yale University, New Haven, CT, USA; Nortel Networks, Canada; and the Government of Ontario, Canada. He has several years of experience working in the academia, government, and the private sectors in research, teaching, consulting, project management, architecture, software design, and product engineering roles. Currently, Dr. Misra is the Associate Editor of the IEEE Transactions Mobile Computing. He served as TPC member in several IEEE conferences. He is the author of over 290 scholarly research papers (including 160 journal papers). His current research interests include algorithm design for emerging communication networks and Internet of Things. Dr. Misra has been a recipient of nine research paper awards in different conferences. He was awarded the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at IEEE GLOBECOM 2012. He was also awarded the Canadian Government's prestigious NSERC Post Doctoral Fellowship and the Humboldt Research Fellowship in Germany. He has edited eight books in the areas of wireless ad hoc networks, wireless sensor networks, wireless mesh networks, communication networks and distributed systems, network reliability and fault tolerance, and information and coding theory, which were published by reputed publishers such as Springer, Wiley, and World Scientific.

Prerona Dutta is working as a Software Developer in AMDOCS – a telecommunications company. Prior to that, she has completed her B.Tech in Information Technology from National Institute of Technology Duragpur, India in 2017. Her current research interests include cloud computing, networking and telecommunications.