Dynamic Optimal Pricing for Heterogeneous Service-Oriented Architecture of Sensor-Cloud Infrastructure

Subarna Chatterjee, *Student Member, IEEE*, Ranjana Ladia, and Sudip Misra, *Senior Member, IEEE*

Abstract—This paper proposes a dynamic and optimal pricing scheme for provisioning Sensors-as-a-Service (Se-aaS) [1] within the sensor-cloud infrastructure. Existing cloud pricing models are limited in terms of the homogeneity in service-types, and hence, are not compliant for the heterogeneous service oriented architecture of Se-aaS. We propose a new pricing model comprising of two components, applicable for Se-aaS architecture: *pricing attributed to Hardware* (*pH*) and *pricing attributed to Infrastructure* (*pI*). pH addresses the problem of pricing the physical sensor nodes subject to variable demand and utility of the end-users. It maximizes the profit incurred by every sensor owner, while keeping in mind the end-users' utility. pI mainly focuses on the pricing incurred due to the virtualization of resources. It takes into account the cost for the usage of the infrastructural resources, inclusive of the cost for maintaining virtualization within sensor-cloud. pI maximizes the profit of the sensor-cloud service provider (SCSP) by considering the user satisfaction. Simulation results depict improved performance of pH in comparison to the traditional hardware pricing algorithms, viz. PPM and Sprite, in terms of the residual energy, proximity to the base station (BS), received signal strength (RSS), overhead, and cumulative energy consumption. The results also show the tendency of the sensor-owners to converge to the end-user utility, but not exceed it. We also analyze the performance of pI. The results show the optimality in the profit incurred by SCSP and the user satisfaction.

Index Terms—Sensor-cloud infrastructure, wireless sensor network (WSN), sensor owners, cloud pricing

1 Introduction

Recent research [2], [3], [4] envisions sensor-cloud infrastructure as a potential substitute for traditional wireless sensor networks (WSNs). Sensor-cloud infrastructure thrives on the principle of virtualization of physical sensor nodes and is essentially an offshoot of conventional cloud computing [5], [6], [7], thereby rendering a powerful infrastructure that interfaces between the physical and cyber worlds. According to MicroStrain's,[1] who stands among one of the pioneers in inventing this technology, sensor-cloud infrastructure is defined as [2]:

> *A unique sensor data storage, visualization and remote management platform that leverage [sic] powerful cloud computing technologies to provide excellent data scalability, rapid visualization, and user programmable analysis.*

Unlike the usual WSNs, sensor-cloud disseminates the usability of the physical sensors to the common mass of end-users who do not have to own, maintain, or manage the physical sensor nodes. The end-users possess their own WSN-based applications which are fed by the sensed information, directly from the sensor-cloud service provider (SCSP), on-demand from the end-users. The underlying procedure of obtaining the raw sensed data from the physical networks and the complex processing of those data are completely abstracted from the end-users. Thus, virtualization of the physical sensor nodes enables the end-users to envision the *Sensors-as-a-Service*, commonly known as *Se-aaS* [1], [4]. Se-aaS breaks the conventional perception of the sensor nodes as typical hardwares and enables the users to envision it simply as a service, just like water or electricity.

As sensor-cloud infrastructure is the extension of cloud computing, it complies with the features that are intrinsic to the latter. A cloud platform generally conforms with a pay-per-use model [8], [9], in which the end-users pay only for those units of resources that they have utilized. Within sensor-cloud infrastructure, end-users utilize the physical sensors and the cloud infrastructure as per their demand and pay as per their usage, to the SCSP. Thus, it is necessary to develop a pricing scheme for Se-aaS to quantify the usage of the end-users and charge them accordingly. The profit incurred from the payment from the end-users is not only enjoyed by the SCSP, but is also shared among the several sensor owners whose physical sensors are registered within sensor-cloud [10].

This work focuses to design a dynamic and optimal pricing scheme, specifically for Se-aaS. Currently, different pricing models are suggested for the various service oriented architectures (SOAs), namely infrastructure-as-a-service (IaaS) [11], [12], platform-as-a-Service (PaaS) [13], and software-as-a-Service (SaaS) [14]. However, these pricing models have been designed for homogeneous types of services such as infrastructure, platform and software. On the

[1] http://www.sensorcloud.com/system-overview

• *The authors are with the School of Information Technology, Indian Institute of Technology, Kharagpur, Kharagpur, West Bengal, India. E-mail: {chatterjeesubarna, sudip_misra}@yahoo.com, ranjanaladia@gmail.com.*

Manuscript received 9 Oct. 2014; revised 23 June 2015; accepted 26 June 2015. Date of publication 8 July 2015; date of current version 7 Apr. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSC.2015.2453958

contrary, Se-aaS follows a heterogeneous SOA in which service is provisioned in the form of hardware as well as infrastructure to the end-users.

1.1 Motivation

As mentioned earlier, the existing pricing schemes follow homogeneous SOA, thereby rendering homogeneous services such as “infrastructure”, “platform” and “software”. Such pricing models incorporate the payment strategies which primarily focus on the parameters related to the specified services only. However, sensor-cloud infrastructure follows a heterogeneous SOA, thereafter providing a fusion of two distinct service types such as “hardware” and “infrastructure” [15], commonly known as Se-aaS. Naturally, the existing pricing models do not deem fit for Se-aaS. Thus, there persists an urgent need for designing a new and efficient pricing model, specifically for Se-aaS. For proper modeling and implementation of the sensor-cloud infrastructure, it is important to measure the usage of the end-users in terms of the infrastructural resources that are consumed, along with the involvement of the underlying physical sensor nodes for the purpose of data gathering and transmission. As the requirement of the end-users vary with time and application, there arises a strong need to design a dynamic pricing scheme that can balance and optimize the cash inflows and outflows among the SCSP, the sensor owners and the end-users. In order to satisfy the business requirements, the profit of the SCSP needs to be maximized while keeping in mind that the end-user is not overcharged.

1.2 Contribution

The work presents significant research contributions, as stated below:

1) In this paper, a pricing model is designed for heterogeneous SOA, Se-aaS, in which the end-users need to pay for utilizing physical sensor nodes and the sensor-cloud infrastructure, as per their application demand.
2) The proposed algorithm for pricing of the physical nodes is context-aware, and the price charged is purely based on the quality of information (QoI) that the end-user obtains finally.
3) The work takes into account the end-users’ satisfaction and their net utility as one of the factors to establish the optimality in the pricing. The objective is to maximize the expected individual profit made by the several registered sensor owners along with the profit made by the SCSP.
4) The proposed pricing model is energy-efficient, as computations are primarily performed at the sensor-cloud end, rather than at the physical sensor network, thereby, reducing the complexity of pricing computation among the physical sensor nodes.
5) The work presents a comparative study of the proposed algorithms with some of the traditional hardware pricing algorithms. The former clearly outperforms the latter in terms of residual energy, proximity with Base Station, received signal strength (RSS), and overhead.

1.3 Organization of the Paper

The rest of the paper is organized as follows. Section 2 discusses the prior work that has been done so far in this domain. Section 3 illustrates the problem scenario. The system model is depicted in Section 4. Section 5 presents an analysis of the results of simulation. Finally, we conclude the work in Section 6.

2 Related Work

Sensor-cloud infrastructure has recently been a major evolution in the field of research, as it is being envisioned as a substitute for the traditional WSNs [3], [16]. It is an extension of cloud computing that efficiently manages the physical sensor nodes which are widely spread across the several WSNs [10]. In our work, we design a dynamic and optimal pricing scheme for rendering Se-aaS to the end-users. The goal of the model is to maximize the profit incurred by the SCSP as well as by the sensor owners.

Some prior work has been done on network pricing [17], [18], [19], [20]. Ng and Seah [21] applied game theory analyzing for truthful cooperation of physical nodes in a sensor network. This work considered the behavior of the colluding nodes involved in data delivery and the message acknowledgment in a lossy, multihop wireless network. Buttyan and Hubaux [22] have proposed a secured pricing technique which encourages the physical node to cooperate in message delivery and prevents from network overloading. In fact some of the works [17], [18], [20] also focused on the energy-efficiency aspects in which the authors envisioned the problem of maintaining resource efficiency as a functional objective of pricing. However, such pricing considered only the network attributes to be shared among the sensor nodes. In our work, the goal is not to distribute the network parameters but to provision Se-aaS through routing and forwarding of data packets. In the process on involving intermediate sensor nodes, we aim to optimize the energy efficiency and maintain the user-satisfaction, simultaneously.

On cloud pricing, specifically, several schemes have already been proposed for utilizing various cloud computing resources [23]. Li and Li [24] proposed a hierarchical pricing model, which considers the issues related to quality of service (QoS) and the utility of both the users and the service providers thereby enforcing a fair approach for both the parties. The authors of [9], [25], [26] in their work have proposed dynamic pricing schemes adopting a revenue management framework from economics. The works suggest a pricing model in which the provider makes a profitable margin without affecting the customers’ demands in the near future. The major challenges of this work is to predict how the demands of the users change with the change in the price, based on which the dynamic pricing model is suggested. Sharma et al. [27] proposed a pricing model that mainly focuses on two constraints: (a) the QoS to provide greater service satisfaction from the user perspective and (b) profitability aspects from the cloud service provide perspective. Son and Sim [28] have studied on both the pricing and time-slot negotiation for the various usage of the cloud services.

Few works focus on dynamic resource pricing within a prespecified time-limit and fixed resource budget ensuring

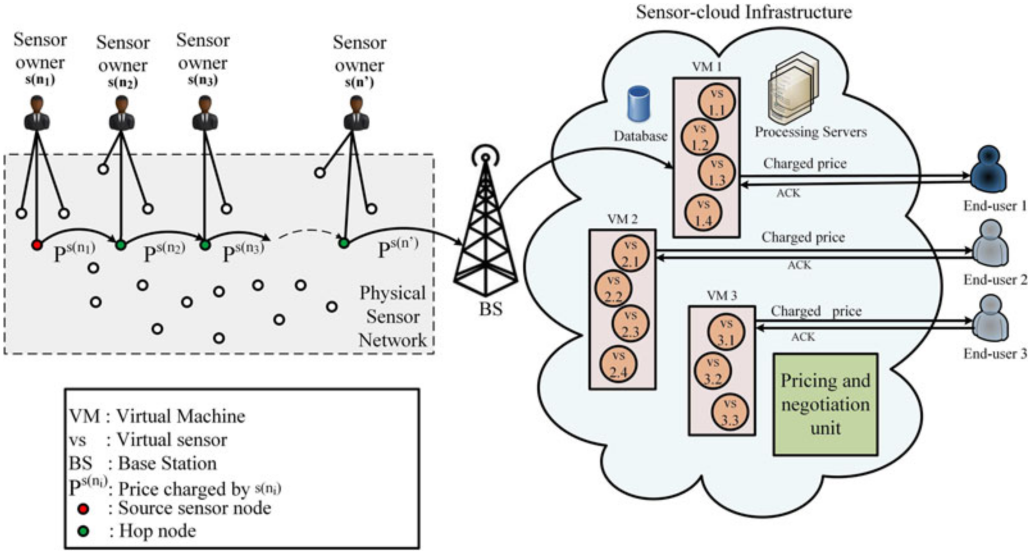


Fig. 1. Network architecture of sensor-cloud.

QoS [29], [30], [31], [32]. Qin et al. [33] proposed a dynamic pricing model, which is flexible to the change in the demand of the end-users and accordingly, it adjusts the pricing of the cloud resources. Jangjaimon and Tzeng [34] have implemented an 'enhanced adaptive incremental checkpointing' (EAIC) meant to significantly reduce the effective monetary cost involved in the expected turnaround time of an end-user application. Kantere et al. [35] have designed a pricing scheme for the data cache of the cloud infrastructure. Few works [36], [37] have focused on price-based load balancing or resource sharing.

However, all of the above are designed only for a specific service. As Se-aaS is built on a heterogeneous SOA, serving both infrastructure and hardware, we design and implement a dynamic and optimal pricing scheme for Se-aaS. The proposed scheme considers issues that are inherent to the heterogeneity of services of sensor-cloud infrastructure.

3 Problem Scenario

This work focuses on determining the price to be charged by the SCSP from the end-users (based on his/her usage), to achieve the following goals:

1) Maximizing the profit made by the SCSP.
2) Maximizing the profit of the sensor owners whose physical sensor nodes either participate as the source sensor nodes or as the intermediate hop nodes.
3) Ensuring that the end-users are not overcharged, thereby achieving end-users satisfaction.

As per the requirement of the end-users, the SCSP determines the source sensor node, and the other participating physical sensor nodes that are to be activated. The source sensor node may not be within direct reach of the BS, thereby leading to a multi-hop transmission. The other nodes of the network are encouraged to participate as the intermediate hops, as they are offered incentives for their participation. The incentives are determined as per the policy to gain a net positive profit.

Some cost is also incurred for using and maintaining the sensor-cloud's infrastructural resources—the virtual machines, the virtual sensors, the IT resources, the processing ability of the cloud, and so on. Considering all these related aspects, the SCSP regulates the price to be paid by the end-users. Fig. 1 provides a pictorial illustration of the scenario.

4 System Model

There is a set of m physical sensor nodes, $N = \{n_1, n_2, n_3, \ldots, n_m\}$, within the physical sensor network of the sensor-cloud infrastructure, registered by their respective sensor owners. S represents the set of sensor owners. The owner of sensor node n_i is denoted by $s(n_i)$. $E = \{e_1, e_2, e_3, \ldots, e_l\}$, represents the set of end-users requesting for the data from the SCSP. We formally define the components of the proposed system as follows:

- $S' = \{s(n_1), s(n_2), s(n_3), \ldots, s(n')\}$, $n' < n$, where $S' \subset S$ represents the sensor owners whose physical sensor nodes are actually utilized during the data transmission for a particular end-user e.
- n_1 represents the source sensor node, n_i, $2 \leq i \leq n'$, represents the hop node.
- $P_t^{s(n_j)}$, $1 \leq j \leq n'$ represents the price charged by the sensor owner $s(n_j)$ for utilizing its physical sensor node at time instant t.
- VM_e represents the Virtual Machine created for the end-user e, $e \in E$.
- $VS_e = \{vs_1, vs_2, \ldots vs_{k(t)}\}$, where VS_e represents the set of virtual sensors created within VM_e at time instant t for e.
- $C_{VM_e}(t)$ represents the cost of VM_e at time t.
- P_{VM_e} represents the price charged by the SCSP from end-user e for using VM_e.
- $P_{vs_i}(t)$ represents the price charged by the SCSP to the end-user e for the virtual sensor vs_i at time instant t.
- $\lambda_{vs_i}^e(t)$ represents the demand by the end-user e for the virtual sensor vs_i at time instant t.

- c represents the criticality of the data per unit time.
- R represents the total number of requests made by all the end-users.
- w represents the service rate of the SCSP.

4.1 Assumptions of the Model

1) A single SCSP and multiple sensor owners are present in the system, i.e., the system is monopolized with respect to the SCSP, and oligopolized with respect to the sensor owners.
2) An end-user is allocated a single VM. However, allocation of multiple virtual sensor nodes within the VM is permitted.
3) An end-user continues to accept the prices charged at time t, until s/he is dissatisfied at time $t+1$.
4) The physical sensor nodes periodically transmit control packets to the cloud end to enable the SCSP to be aware of the health information of the nodes.
5) Every physical sensor node is static, and is aware of the location coordinates of itself, its neighbors and the corresponding BS.

In Fig. 1, the sensor owner $s(n_1)$ owns the source sensor node which generates the required data. $s(n_1)$ needs the help of any immediate physical sensor node in order to transmit the data. The SCSP encourages the neighboring physical nodes of the source sensor node to participate in the data transmission. The source sensor node n_1 chooses one of its neighbors as the next hop node n_2, based on a utility value. $s(n_1)$ charges $s(n_2)$ with price ${P_t}^{s(n_1)}$.[2] ${P_t}^{s(n_1)}$ is accepted by the hop node owned by the sensor owner $s(n_2)$. With the intention to make profit, $s(n_2)$ charges a price ${P_t}^{s(n_2)}$ greater than ${P_t}^{s(n_1)}$ to its next willing participant. This pricing scheme continues until the data finally reaches the last participating hop node. The last hop node owned by the sensor owner $s(n')$ charges a price ${P_t}^{s(n')}$ to the end-user e who actually requested the data. Furthermore, it is intuitive that ${P_t}^{s(n')} > {P_t}^{s(n'-1)} > \cdots > {P_t}^{s(n_2)} > {P_t}^{s(n_1)}$. In order to transfer the required data, the infrastructural resources of the SCSP are utilized. Based on the end-user demand, the virtual machines and the component virtual sensors are created for which the SCSP charges some amount of price. This profit is solely enjoyed by the SCSP for provisioning infrastructure as a service. We fragment the pricing scheme of Se-aaS into two distinct modules and propose two different algorithms:

a) *Pricing attributed to Hardware* (*pH*)
b) *Pricing attributed to Infrastructure* (*pI*).

4.2 pH: Pricing Attributed to Hardware

The pricing attributed to the usage of the physical sensor nodes concern the profit of the respective sensor owners. As the source sensor node n_1 generates the raw sensed data, it either directly transmits it to the BS in a single-hop, or follows a multi-hop route. Motivated by the pricing strategies mentioned in [38], [39], we design the proposed pricing model for the hardware usage. We propose a context aware optimal pricing scheme for the usage of the physical sensor nodes.

4.3 Selection of the Next Hop Node

In order to transmit data from the source sensor node n_1 to the Base Station BS,[3] n_1 selects the next hop node n_2 with the maximum utility η among all the nominated hop nodes, in set H_{n_1}. The transmission radius of n_1 at t is denoted as $r_{n_1}(t)$. The set of physical sensor nodes that are located within the transmission area $A_{n_1}(t) = \pi r_{n_1}(t)^2$ is considered to be the nominated hop nodes. Thus, $H_{n_1} = \{h_1, h_2, h_3, \ldots, h_b\} \mid \xi(h_j, n_1) \leq r_{n_1}(t)$, where $\xi()$ computes the inverse of the Euclidean distance between two nodes. The node h_i with the maximum utility η_{max} emerges as the winner hop node among all the participating physical nodes. The utility of node h_i at time instant t is dependent on the following factors.

- *Residual energy:* The utility $\eta_{h_i}(t)$ of h_i at t is dependent on its residual energy level, $Q_{h_i}(t)$, which expressed as,

$$Q_{h_i}(t) = \frac{E^{cur}_{h_i}}{E^{init}_{h_i}}, \tag{1}$$

  where E_{init} and E_{cur} are the initial and current energy level of h_i, respectively.
- *Proximity to the BS:* $\eta_{h_i}(t)$ is dependent on the euclidean distance $\xi(h_i, BS)$ between node h_i and BS.

$$\xi(h_i, BS) = \left(\sqrt{(BS_{x_i} - h_{x_i})^2 + (BS_{x_i} - h_{y_i})^2}\right)^{-1} \tag{2}$$

  h_{x_i}, h_{y_i}, BS_{x_i}, and BS_{y_i} being the abscissa and ordinate of h_i, and the BS, respectively.
- *Received signal strength:* The Received Signal Strength of h_i, RSS_{h_i}, is also one of the factors affecting its utility at time instant t. We have,

$$RSS_{h_i}(t) = \psi_{h_i} \frac{P^{tr}_{h_i}(t)}{\xi(h_i, n_1)^a}, \tag{3}$$

  where $P^{tr}_{h_i}$ is the transmitted power, ψ_{h_i} which comprises of all the other factors affecting RSS such as the antenna gain and antenna height, and a denotes the propagation constant [40]. In our problem scenario, $a = 2$.
- *State transition overhead:* Node h_i exists in either of the three states —active ($S^0_{h_i}$), passive ($S^1_{h_i}$), and sleep ($S^2_{h_i}$). For the data transmission, h_i needs to exist in the active state $S^0_{h_i}$. The state transition overhead in terms of energy dissipation while switching from $S^1_{h_i}$ or $S^2_{h_i}$ to $S^0_{h_i}$ is denoted by, $P_{pq}, p = \{S^0_{h_i}, S^1_{h_i}, S^2_{h_i}\}$, $q = \{S^0_{h_i}\}$. Quite intuitively, $P_{S^1_{h_i} S^0_{h_i}} \ll P_{S^2_{h_i} S^0_{h_i}}$. However, when h_i remains in the active state, there is ideally no overhead. We assume $P_{S^0_{h_i} S^0_{h_i}} \rightarrow 0$.

Definition 1. *The utility $\eta_{h_i}(t)$ of a hop node h_i, $\forall i = \{1, 2, 3, \ldots, b\}$, at time instant t, is a function of its residual energy*

[2] Although it appears that the price charged by one sensor owner is paid by another, the net price is essentially paid by the end-user.

[3] It is to be noted that to ensure fault tolerance and efficiency, in practice, the system model may support multiple BSs. However, for the sake of simplicity and understandability, we consider a single BS in this paper.

$Q_{h_i}(t)$, its received signal strength $RSS_{h_i}(t)$, its proximity to the BS $\xi(h_i, BS)$, and its state transition overhead P_{pq}. $\eta_{h_i}(t)$ is expressed as,

$$\eta_{h_i}(t) = \left(Q_{h_i}(t) + g \times \frac{RSS_{h_i}(t)\xi(h_i, BS)}{P_{pq}}\right)$$

g being a normalization factor with the same unit as that of ξ.

Having computed the utility of every nominated hop node, the node with the maximum utility emerges as the winner hop node. Thus, without the loss of generality we can infer,

$$n_i = \max_{\forall h_k \in H_{n_{i-1}}} \eta_{h_k}(t). \tag{4}$$

4.4 Context-Aware Pricing

Having decided the next hop node, we now propose a context-aware pricing scheme. Initially, we determine the expected price to be charged by the last hop node $s(n')$, which is denoted by ${P_t}^{s(n')}$. The context of the data is examined in terms of few parameters which we describe below.

Definition 2. *Transmission confidence of the data between a pair of nodes $< a, b >$ at time t, $f_{a,b}(t)$, is expressed in the form of profit/loss factor based on the difference of the raw sensed data between the sender and the receiver nodes [41]*

$$f_{a,b}(t) = \begin{cases} \frac{1}{N} f_{a,b}(t-1)e^{(\rho\delta)(t)}, & \rho = |D_a - D_b| \ < \ \rho_{th} \\ \frac{1}{N} f_{a,b}(t-1)e^{-(\rho\delta)(t)}, & \text{otherwise,} \end{cases} \tag{5}$$

where N is a factor for normalization, ρ is the absolute deviation of the transmitted data D_a from the received data D_b, and δ is the profit/loss factor.

Definition 3. *The temporal relevance of the data $\mathcal{T}$ at time t is defined as the tolerable time interval, beyond which the data is assumed to be insignificant. Thus, $\mathcal{T}(t) = \frac{t_d}{t_r}, 0 \leq t_r - t_d \leq k$, where t_d and t_r are the time instants of detecting an event at n_i and receiving the data at n_{i+1}, respectively. If $t_r - t_d$ exceeds k, $\mathcal{T}$ is considered to be negligible, i.e., $\mathcal{T} \sim 0$.*

Motivated by the general design for the metric QoI [42], we model the QoI of node n_i at time t as, $\mathcal{Q}_{n_i} = \omega_{n_i}\mathcal{Q}_{n_i - 1} + \eta_{n_i}, \mathcal{Q}_{n_1} = 1$, where ω_{n_i} is the discounting factor at time t, which is expressed as $\omega_{n_i} = Q_{n_i} f_{n_{i-1},n_i}\mathcal{T}$. Thus, we get,

$$\mathcal{Q}_{n_i} = \prod_{j=2}^{n_i} \omega_{n_j}\mathcal{Q}_{n_1} + \sum_{k=2}^{n_i - 1}\prod_{l=k+1}^{n_i} \omega_{n_l}\eta_{n_k} + \eta_{n_i}. \tag{6}$$

Definition 4. *The price ${P_t}^{s(n')}$ charged by sensor owner $s(n')$ of the last hop node n', is directly proportional to the QoI of the data of n' at time t,*

$${P_t}^{s(n')} \propto \mathcal{Q}_{n'}(t) \Rightarrow {P_t}^{s(n')} = c_1(t)\mathcal{Q}_{n'}(t), \tag{7}$$

where c_1 is a multiplicative factor that accounts for both the signal attenuation in terms of the nodal signal to noise ratio (NSNR) [43] and the total number of transmission attempts for the corresponding packet. Thus, we have,

$$c_1(t) = g\frac{P_{signal}(t)}{P_{noise}(t)}, \tag{8}$$

where P_{signal}, P_{noise}, and g are the power of signal and the background noise, and the count of the transmission attempts, respectively.

Definition 5. *The utility $\mathcal{U}$ of the end- user e is defined as the amount of data received per virtual sensor vs_i per unit time. Thus, $\mathcal{U} \sim U(\gamma_1, \gamma_2)$.*

Motivated by the works of Lam et al. [39], and Fudenberg and Tirole [44], we illustrate the strategy profile of the proposed system as follows.

Strategy profile:

- The end-user e obtains data from a virtual sensor vs_i for a time period, τ. The end-user follows a myopic strategy: it retains a virtual sensor vs_i at time t, if $(t \leq \tau)$ and $(\mathcal{U} \geq {p_t}^{s(n')})$ i.e., within the time period τ, the end-user accepts the service if and only if the utility $\mathcal{U}$ is higher than the price to be payed by the end-user.
- The sensor owner $s(n_i)$, $\forall i = \{2, 3, \ldots, n'\}$, of a participating hop node, charges a price $p^{*s(n_i)}(p^{s(n_{i-1})})$ which is dependent on the price charged by the previous sensor owner $s(n_{i-1})$.

$$p^{*s(n_i)}(p^{s(n_{i-1})}) \in \arg\max_{p^{s(n_i)}} \left[\left(p^{s(n_i)} - p^{s(n_{i-1})}\right) P\left(\mathcal{U} \geq m^{s(n_i)}(p^{s(n_i)})\right)\right] \tag{9}$$

where $m^{s(n_i)}(p^{s(n_i)})$ is the mark up function, as defined in Definition 6. As depicted in Equation (9), a sensor-owner $s(n_i)$ strategically claims his/her price by probabilistically determining the effective price payable (by the end-user) to the stream of sensor-owners $s(n_i)$ to $s(n')$. For the strategy to be effective, it also considers the probability of the end-user to be willing to pay the price, $(P(\mathcal{U} \geq m^{s(n_i)}(p^{s(n_i)})))$.

- The owner of the last hop node $s(n')$ charges a decreasing price sequence $\{p_t^{s(n')}\}$. We have,

$$p_t^{s(n')} = c_1(t)\left[\prod_{j=2}^{n'} \omega_{n_j}(t)\mathcal{Q}_{n_1} + \sum_{k=2}^{n'-1}\prod_{l=k+1}^{n'} \omega_{n_l}(t)\eta_{n_k}(t) + \eta_{n'}(t)\right]. \tag{10}$$

Equation (10) considers the QoI of the final data received at the cloud-end.

Definition 6. *The mark-up function $m^{s(n_i)}(p^{s(n_i)})$ of the proposed system is defined as the price that an end-user has to pay for the stream of nodes from node n_i to n' after the price is fixed at n_i. Thus, $m^{s(n_i)}(p^{s(n_i)})$ is expressed as,*

$$p^{*s(n_i)} = \begin{cases} (2^{n'-i})\gamma_1 - (2^{n'-i}-1)\gamma_2, & \text{if } \frac{p^{s(n_{i-1})}+\gamma_2}{2} < (2^{n'-i})\gamma_1 - (2^{n'-i}-1)\gamma_2 \\ \frac{p^{s(n_{i-1})}+\gamma_2}{2}, & \text{if } \frac{p^{s(n_{i-1})}+\gamma_2}{2} \in [(2^{n'-i})\gamma_1 - (2^{n'-i}-1)\gamma_2, \gamma_2] \\ \gamma_2, & \text{otherwise} \end{cases} \tag{11}$$

$$m^{s(n_i)}(p^{s(n_i)}) = \begin{cases} p^{*s(n')}(p^{*s(n'-1}(...(p^{*s(n_{i+1})}(p^{s(n_i)}))...)), & \\ \qquad i = \{1, ..., n'-1\} & \\ p^{s(n')}, \qquad i = n'. & \end{cases} \tag{12}$$

With the assumption that γ_1, γ_2 are known, we determine the optimal price $p^{*s(n')} \in [\gamma_1, \gamma_2]$ charged by $s(n')$. We have,

$$\begin{aligned} &(p^{s(n')} - p^{s(n'-1)})P(\mathcal{U} \geq m^{s(n')}(p^{s(n')})) \\ &= (p^{s(n')} - p^{s(n'-1)})\left(\frac{\gamma_2 - p^{s(n')}}{\gamma_2 - \gamma_1}\right). \end{aligned} \tag{13}$$

On differentiating Equation (13) w.r.t $p^{s(n')}$ and equating to zero, we get the optimal price $p^{*s(n')}$ as,

$$p^{*s(n')} = \begin{cases} \gamma_1, & \text{if } \frac{p^{s(n'-1)}+\gamma_2}{2} < \gamma_1 \\ \frac{p^{s(n'-1)}+\gamma_2}{2}, & \text{if } \frac{p^{s(n'-1)}+\gamma_2}{2} \in [\gamma_1, \gamma_2] \\ \gamma_2, & \text{otherwise.} \end{cases} \tag{14}$$

On iterating the above process, we derive the optimal price $p^{*s(n_i)}$ charged by the owner, $s(n_i)$ of any participating hop node $\forall i = \{2, 3, \ldots, n'\}$, as shown in Equation (11).

The optimal price $p^{*s(n_1)}$ charged by the owner of the source sensor node $s(n_1)$ is,

$$p^{*s(n_1)} = \begin{cases} \frac{\gamma_2}{2}, & \text{if } (2^{n'-1})\gamma_1 - (2^{n'-1}-1)\gamma_2 \leq \frac{\gamma_2}{2} \\ (2^{n'-1})\gamma_1 - (2^{n'-1}-1)\gamma_2, & \text{otherwise.} \end{cases} \tag{15}$$

Theorem 4.1. *The theoretical maximum of an end-user utility, γ_2, is dependent on the price charged by the last hop node, $p^{*s(n')}$ and the price charged by the second last hop node, $p^{s(n'-1)}$.*

Proof. We obtain the optimal price charged by $s(n')$ from Equation (14). Thus, to maintain the optimality in price, the utility provisioned to an end-user has an upper bound γ_{2max}. We observe that, as $\frac{p^{s(n'-1)}+\gamma_2}{2} \in [\gamma_1, \gamma_2]$, $\gamma_2 = 2p^{*s(n')} - p^{s(n'-1)}$, and as $\frac{p^{s(n'-1)}+\gamma_2}{2} > \gamma_1$, $\gamma_2 = p^{*s(n')}$. Now,

$$\begin{aligned} \gamma_2 &= 2p^{*s(n')} - p^{s(n'-1)} \\ &= p^{*s(n')} + p^{*s(n')} - p^{s(n'-1)}. \end{aligned}$$

Since $(p^{*s(n')} - p^{s(n'-1)})$ is the net profit of $s(n')$, it is expected to be a positive quantity. Thus, we infer $\gamma_{2max} = 2p^{*s(n')} - p^{s(n'-1)}$. This implies,

$$\gamma_{2max} = \begin{cases} 2p^{*s(n')} - p^{s(n'-1)}, \\ if\ \gamma_2 \in \max(2\gamma_1 - p^{s(n'-1)}, p^{s(n'-1)}) \\ p^{*s(n')}, \text{otherwise.} \end{cases} \tag{16}$$

This concludes the proof. □

Corollary 4.1. *The maximum utility γ_2 obtained by an end-user e, at a particular time instant, is dependent on the number of hop nodes n'.*

Justification: Ideally, every $s(n_i), \forall i \in \{1...n'\}$, makes a net positive profit. Therefore, $\{p^{*s(n_i)}\}$ is a non-decreasing sequence. Hence as n' increases, $p^{*s(n')}$ also increases. This justifies the statement.

Proposition 4.1. *For a single hop case, i.e., when the source node n_1 behaves as the only hop node, the maximum utility that can be provisioned is twice the price charged by $s(n_1)$.*

Proof. For a single hop case, n_1 directly connects to BS, i.e., $n' = 1$. From Equation (15), we infer that, as $\gamma_2 \geq 2\gamma_1$, $\gamma_2 = 2p^{*s(n_1)}$, and when $\gamma_2 < 2\gamma_1$, $\gamma_2 < 2p^{*s(n_1)}$. Thus, without the loss of generality we can say, $\gamma_{2max} \leq 2p^{*s(n_1)}$. This completes the proof. □

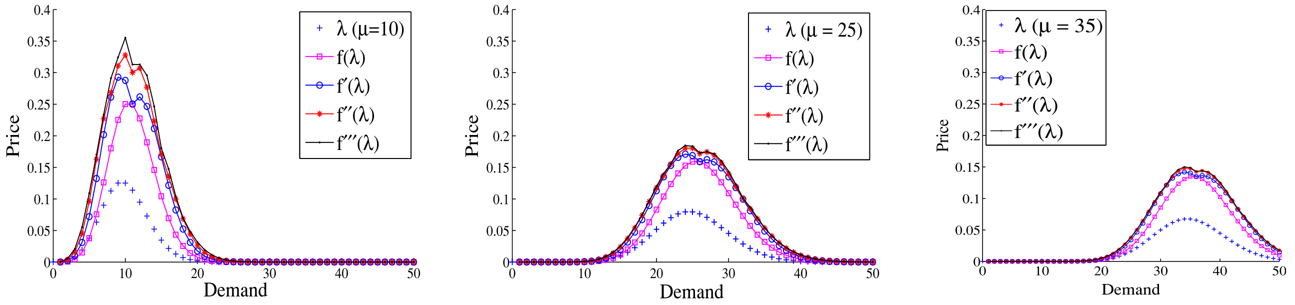
4.5 pI: Pricing Attributed to Infrastructure

In terms of the usage of infrastructure within the sensor-cloud platform, whenever end-user e requests the SCSP for some data to be fed into his/her application, the SCSP creates a VM dedicated to e, VM_e. The number of virtual sensors within VM_e that are created and deleted depends upon the requirement of e, and, thereby, being time dependent, and is denoted by $k(t)$. Based on the demand $\lambda^e_{vs_i}(t)$ of e for virtual sensor vs_i, the price charged by the SCSP is $P_{vs_i}(t)$ at time instant t. $C_{VM_e}(t)$ is the cost of creating VM_e within the cloud platform, inclusive of the initial cost for creating the instance of VM_e, B_{VM_e}, and the cost for maintaining it over time. The maintenance cost of a VM_e is charged from the time it is built (t_{built}) till it is discarded. The maintenance cost of a VM_e per unit time, M_{VM_e}, comprises of the cost for creating its component virtual sensors $vs_i \in VS_e$, in addition to the maintenance cost per unit time, for each of them. Thus,

$$C_{VM_e}(t) = B_{VM_e}(t) + M_{VM_e}(t - t_{built}) \tag{17}$$

$$M_{VM_e}(t - t_{built}) = \sum_{i=1}^{k(t)} \left(B_{vs_i} + M_{vs_i}(t - t_{0i}) \right) \tag{18}$$

where t_{0i} represents the time instant at which the virtual sensor vs_i is created. The final equation of the cost incurred by the SCSP for the creation and maintenance of VM_e and its corresponding virtual sensors, at time t is,



(a) $\lambda_{vs_i} \sim P(50, 10)$ (b) $\lambda_{vs_i} \sim P(50, 25)$ (c) $\lambda_{vs_i} \sim P(50, 35)$

Fig. 2. Analysis of price-demand relationship.

$$C_{VM_e}(t) = B_{VM_e}(t) + \sum_{i=1}^{k(t)} B_{vs_i} + \sum_{i=1}^{k(t)} M_{vs_i}(t - t_{0i}). \tag{19}$$

A virtual sensor comprises of a set of homogeneous (with respect to sensing hardware) physical sensors serving a particular application. The creation and deletion of the virtual sensors is completely dependent on the end-user's requirement. However, if a virtual sensor is unused for a long time duration, the maintenance cost exceeds the cost of creating the same. In such cases, it is preferred to delete a virtual sensor and create it when required.

Proposition 4.2. *The optimum time interval $\triangle t$ between two consecutive demands for a particular virtual sensor vs_i is $\frac{B_{vs_i}}{M_{vs_i}}$.*

Proof. We assume that the last time instant at which the maintenance cost equals the cost of creation of vs_i is t_{max} and t represents the current time instant. Thus, $M_{vs_i}(t_{max} - t) = B_{vs_i}$. Thus, for all $t' > t_{max}$,

$$M_{vs_i}(t_{max} - t') > B_{vs_i} \Rightarrow t_{max} = \frac{B_{vs_i}}{M_{vs_i}} + t \tag{20}$$

$$\text{Thus, } \triangle t = t_{max} - t = \frac{B_{vs_i}}{M_{vs_i}}. \tag{21}$$

□

Corollary 4.2. *The instantaneous cost incurred at the cloud end, for a virtual sensor vs_i, at time t', $(C^{inst}_{vs_i}(t'))$, is dependent on the time instant when the last demand was placed.*

Proof. We assume that the last demand for vs_i was placed at t_{last}. From Proposition 4.2, it follows that, at current time instant t', if $t' - t_{last} < \triangle t$, then the instantaneous cost for vs_i will be only due to maintenance at t'. Otherwise, it includes both the creation and maintenance cost. Thus,

$$C^{inst}_{vs_i}(t') = \begin{cases} M_{vs_i}, & t' - t_{last} < \frac{B_{vs_i}}{M_{vs_i}} \\ B_{vs_i} + M_{vs_i}, & otherwise. \end{cases} \tag{22}$$

This completes the proof. □

Definition 7. *The net profit of the SCSP at time t, $r(t)$, is defined as the difference of the total price charged from the end-user and the sum of the cost incurred in creating and maintaining the VM for a particular end-user e and the overall price charged through pH for e $(p^{s(n')}_e(t))$. Thus, $r(t)$ is expressed as,*

$$r(t) = \left(\sum_{i=1}^{k(t)} \lambda^e_{vs_i}(t) P_{vs_i}(t) \right) + P_{VM_e} - C_{VM_e}(t) - p^{s(n')}_e(t), \tag{23}$$

where the price charged for each virtual sensor is a function of the rate of change of demand for each $vs_i(t)$.

Theorem 4.2. *The price charged for a virtual sensor, vs_i, is based on the memory of demand: the price $P_{vs_i}(t)$ charged at a particular time instant t, is based on the previous demands $\lambda_{vs_i}(t-1)$, and the jth order of rate of change of demands over time, $\frac{d^j \lambda_{vs_i}}{dt^j}$, $1 \leq j \leq n, n \in \mathbb{N}$.*

Proof. As in Corollary 4.2, the instantaneous cost of the virtual sensor depends upon the time instant at which the demand was last placed. Thus, as the rate of demand increases within $\triangle t$, the cost decreases accordingly. Therefore, for a vs_i,

$$C_{vs_i}(t) = f\left(\lambda^e_{vs_i}, \frac{d\lambda^e_{vs_i}}{dt}, \ldots, \frac{d^{n-1}\lambda^e_{vs_i}}{dt^{n-1}}, \frac{d^n \lambda^e_{vs_i}}{dt^n} \right). \tag{24}$$

From Equation (23), we see that an increase in the cost, increases the price of vs_i, for the SCSP to make positive net profit, i.e., price and cost are linearly connected. Thus,

$$P_{vs_i}(t) = f\left(\lambda^e_{vs_i}, \frac{d\lambda^e_{vs_i}}{dt}, \ldots, \frac{d^{n-1}\lambda^e_{vs_i}}{dt^{n-1}}, \frac{d^n \lambda^e_{vs_i}}{dt^n} \right). \tag{25}$$

Fig. 2 shows the relationship between demand and price. We have assumed that in Figs. 2a, 2b, and 2c, demand follows a Poisson distribution $(n = 50)$ with varying mean ($\mu = 10, \mu = 25, \mu = 35$), respectively. We observe that the change in price is significant with the first order derivative of the demand. However, there is not much effective change reflected from the higher order derivatives of the demand. Therefore, for the sake of simplicity, in this work, we have,

$$P_{vs_i}(t) = \alpha \frac{d\lambda^e_{vs_i}(t)}{dt} + \beta \lambda^e_{vs_i}(t), \tag{26}$$

where the parameters α, β are assumed to be system-modeled coefficients. This completes the proof. □

At a particular time t', we have, R representing the total number of requests made by all the end-users $\in E$

$$R = \sum_{j=1}^{l}\sum_{i=1}^{k(t')} \lambda_{vs_i}^{e_j}, \quad \forall e_j \in E, \forall vs_i \in VS_{e_j}. \tag{27}$$

Since the service rate of SCSP is w, the expected time to finish serving a request, inclusive of the waiting time and the time being served is, $\frac{1}{w-R}$ [12], [45]. Therefore, the time spent for waiting is $\frac{1}{w-R} - \frac{1}{w}$ [12]. Thus, the expected finishing time for e is,

$$\frac{1}{w-R} - \frac{1}{w} + \frac{\sum_{i=1}^{k(t')} \lambda_{vs_i}^{e}}{w} = \frac{R}{w(w-R)} + \frac{\sum_{i=1}^{k(t')} \lambda_{vs_i}^{e}}{w}. \tag{28}$$

Definition 8. *The user satisfaction $u_e(t)$ for a particular end-user e at any time instant t, is a function of the total demand made by e for all the virtual sensors within VM_e, the total cost incurred at the sensor-cloud end for serving the demand, and the total price charged by the SCSP.*

$$\begin{aligned} u_e(t) = \sum_{i=1}^{k(t)} \lambda_{vs_i}^{e} - c\left[\frac{R}{w(w-R)} + \frac{\sum_{i=1}^{k(t)} \lambda_{vs_i}^{e}}{w}\right] \\ - \left[\sum_{i=1}^{k(t)} P_{vs_i}(t) + P_{VM_e}\right]. \end{aligned} \tag{29}$$

The main objective of our work is to maximize the total profit of the SCSP over time T, while considering the user satisfaction, i.e.,

$$\mathcal{F}(T) = \sum_{t=0}^{T} r(t) \tag{30}$$

$$\text{subjected to,} \quad \lambda_{vs_i}^{e} \geq 0, \forall i = 1, 2, 3, \ldots, k(t) \tag{31}$$

$$\begin{aligned} \sum_{i=1}^{k(t)} \lambda_{vs_i}^{e} - c\left[\frac{R}{w(w-R)} + \frac{\sum_{i=1}^{k(t)} \lambda_{vs_i}^{e}}{w}\right] \\ - \left[\sum_{i=1}^{k(t)} P_{vs_i}(t) + P_{VM_e}\right] > v_{opt}^{e}, \end{aligned} \tag{32}$$

where v_{opt} is the threshold value, below which the values of $u_e(t)$ are not allowed. From the Equation (30), we observe that $r(t)$ can be maximized for every time instant t. Accordingly, $\mathcal{F}(T)$ can be maximized. $r(t)$ is simplified as,

$$\begin{aligned} r(t) &= \sum_{i=1}^{k(t)} \Big(\lambda_{vs_i}^{e}(t) P_{vs_i}(t)\Big) + P_{VM_e} - C_{VM_e}(t) - p_e^{s(n')}(t) \\ &= F\Big(\lambda_{vs_1}, \lambda_{vs_2}, \ldots, \lambda_{vs_{k(t)}}, t_{01}, t_{02}, \ldots, t_{0k(t)}\Big). \end{aligned}$$

TABLE 1
Testbed Information for pH and pI

Parameters	Values
Processor	Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz
RAM	4 GB, DDR3
Disk Space	320 GB
Operating System	Ubuntu 14.04 LTS
Application Software	MATLAB R2013a

We aim to maximize F using the approach of lagrange multiplier. Thus,

$$\begin{aligned} &\nabla F\Big(\lambda_{vs_1}, \lambda_{vs_2}, \ldots, \lambda_{vs_{k(t)}}, t_{01}, t_{02}, \ldots, t_{0k(t)}\Big) \\ &= \theta \nabla u\Big(\lambda_{vs_1}, \lambda_{vs_2}, \ldots, \lambda_{vs_{k(t)}}\Big), \end{aligned} \tag{33}$$

where θ is the Lagrangian multiplier. We have,

$$\frac{\partial F}{\partial \lambda_{vs_i}^{e}} = \alpha \frac{d\lambda_{vs_i}^{e}}{dt} + \beta \lambda_{vs_i}^{e} + \lambda_{vs_i}^{e}\left(\alpha \frac{d}{d\lambda_{vs_i}^{e}} \frac{d\lambda_{vs_i}^{e}}{dt} + \beta\right) \tag{34}$$

$$\begin{aligned} \frac{\partial u_e}{\partial \lambda_{vs_i}^{e}} &= 1 - c\left(\frac{w^2 - 2wR}{w^2(w-R)^2} + \frac{1}{w}\right) \\ &\quad - \left[\alpha \frac{d\lambda_{vs_i}^{e}}{dt} + \beta \lambda_{vs_i}^{e} + \lambda_{vs_i}^{e}\left(\alpha \frac{d}{d\lambda_{vs_i}^{e}} \frac{d\lambda_{vs_i}^{e}}{dt} + \beta\right)\right]. \end{aligned} \tag{35}$$

Using Equations (33) through (35), we get,

$$\begin{aligned} &\alpha \frac{d\lambda_{vs_i}^{e}}{dt} + \beta \lambda_{vs_i}^{e} + \lambda_{vs_i}^{e}\left(\alpha \frac{d}{d\lambda_{vs_i}^{e}} \frac{d\lambda_{vs_i}^{e}}{dt} + \beta\right) \\ &= \theta\left[1 - c\left(\frac{w^2 - 2wR}{w^2(w-R)^2} + \frac{1}{w}\right)\right. \\ &\quad \left. - \left(\alpha \frac{d\lambda_{vs_i}^{e}}{dt} + \beta \lambda_{vs_i}^{e} + \lambda_{vs_i}^{e}\left(\alpha \frac{d}{d\lambda_{vs_i}^{e}} \frac{d\lambda_{vs_i}^{e}}{dt} + \beta\right)\right)\right]. \end{aligned} \tag{36}$$

At a particular time instant t', $\frac{d\lambda_{vs_i}^{e}}{dt'} = 0$. Assuming

$$\alpha \frac{d\lambda_{vs_i}^{e}}{dt} + \beta \lambda_{vs_i}^{e} + \lambda_{vs_i}^{e}\left(\alpha \frac{d}{d\lambda_{vs_i}^{e}} \frac{d\lambda_{vs_i}^{e}}{dt} + \beta\right) = K \tag{37}$$

we get, $K = 2\beta\lambda_{vs_i}^{e}$. Using K and Equation (36), we get,

$$\theta = \frac{2\beta\lambda_{vs_i}^{e}}{1 - c\left(\frac{w^2-2wR}{w^2(w-R)^2} + \frac{1}{w}\right) - 2\beta\lambda_{vs_i}^{e}}. \tag{38}$$

5 Simulation Results

In this Section, we present and analyze the results of simulation. Followed by this, the complexity analysis of both pH and pI are provided. The generic test-bed information for pH and pI is provided in Table 1. Although this work is one of the first attempts to design a pricing scheme for sensor-cloud, some of the hardware pricing solutions that are found similar (but not exact), are discussed in Section A. The simulation setup for pH is shown in Table 2. Some comparative analysis of the proposed solutions with the benchmark approaches are also performed.

5.1 Analysis of pH

We initially compare *pH* with few identified benchmark solution approaches. Followed by this, we also evaluate the performance of *pH* separately. The performance metrics that are considered for comparison are:

TABLE 2
Simulation Setup for pH

Parameters	Values
Deployment Area	500 m × 500 m
Deployment	Uniform, random
Number of nodes	100
Communication range	[100, 200] m
Channel overhead	[1, 5]%
Transmission energy	7 nJ/bit
Computation energy	5 nJ/sec
Number of end-users	10
Average user utility	10000
% of C.I.	95 %

- Mean residual energy
- Mean proximity with BS
- Mean RSS
- Mean state transition overhead
- Cumulative energy consumption
- Packet delivery rate.

The first four are already defined in Section 4. The simulation metric for cumulative energy consumption is discussed later in this section, with the corresponding results. The packet delivery rate is defined below.

Definition 9. *Packet delivery rate is defined as the percentage of the total packets successfully delivered from any source sensor node to the BS.*

5.2 Benchmark Solutions

In order to find the solution for the proposed model, the following existing benchmark solutions are used as the basis for comparison,

- The packet purse model (PPM) [22]
- Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks [17]

In PPM, the sender bears the total cost of transmitting sensed data from the source sensor node to the BS. This cost is calculated in terms of the virtual currency called nuglets. If the amount is under estimated by the sender, then the packet is dropped mid-way, and if it is over estimated, then the sender suffers a loss of nuglets. Moreover, this model requires a tamper- proof hardware established at each node for proper deduction and addition of nuglets. Also, the size of the Packet Purse Header increases than the actual packet size resulting in slow inefficient packet transmission.

In Sprite, a central authority, known as credit clearance service (CCS), is implemented. It evaluates the amount of nuglet to be charged or credited to each node involved in the packet transmission, based on the submitted receipts of a message. For message authentication, the sender transfers a signed message to the next immediate hop node, which accepts the message only after proper verification of the signature of the sender nodes. The digital signature and verification procedure involves a significant processing overhead. Moreover, the CPU processing time exceeds an acceptable limit if any node attempts to send huge number of messages. The storage and the bandwidth requirement increases due to the addition of the authentication header with each message packet.

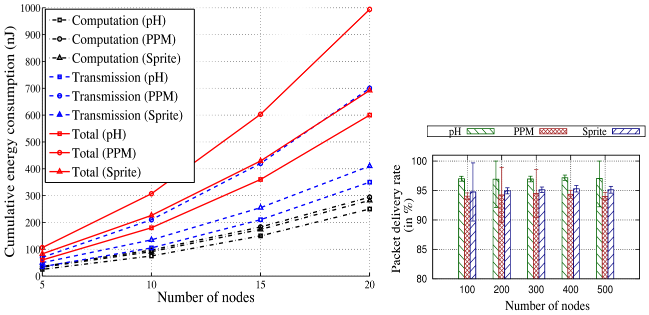
TABLE 3
Comparative Study of pH with PPM and Sprite

	Mean Residual energy (in %)	Mean Proximity with BS (in metre)	Mean RSS (in units)	Mean state transition overhead (in units)
pH	72.15	203.91	36.6	1.05
PPM, Sprite	37.33	223.67	35.9	1.93

In pH, the selection of the next-hop node is evaluated using Equation (4), whereas in PPM [22] and Sprite [17], the standard selection of next hop node is based on simple dynamic source routing (DSR) [46], [47] protocol, in which the physical sensor node closest to the source sensor node is expected to emerge as the next hop node under ideal channel conditions. The experiment is repeated 50 times and the mean of several node parameters is compared for both the approaches, and is shown in Table 3.

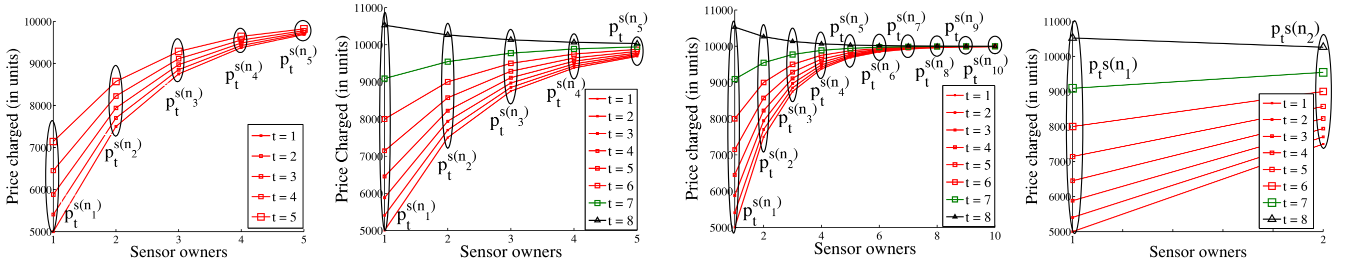
From Table 3, it is evident that the pH selects a better node, compared to PPM or Sprite, in terms of the mean residual energy, mean proximity with the BS, mean RSS, and mean state transition overhead. As the hop selection algorithm in DSR does not consider the other node parameters, e.g., energy level of a node, RSS intensity, and state transition overhead, the hop nodes in PPM, and Sprite are likely to have poor residual energy, or a low RSS intensity. pH outperforms the other approaches in this regard, thereby choosing the nodes with the maximum utility.

Fig. 3a illustrates the cumulative energy consumption of the 10 end-users with the increase in the number of hop nodes. For every end-user, any source sensor node is subjected to identical sensing phenomenon for pH, PPM, and Sprite. Hence, the performance comparison is significant in terms of energy consumption due to transmission, and computation, only. As shown in Fig. 3a, PPM incurs the maximum computation due to repeated estimation of nuglets for every round of transmission. In Sprite, the node maintains a receipt after every transmission. The computation overhead is less, and is mainly because of the processing and generation of the receipt. Unlike PPM, and Sprite, the energy consumption due to computation in pH is primarily handled at the cloud-end. The computational parameters are periodically fed to the sensor-cloud end through control packets (as per the assumptions of the model). Thus, the energy consumption due to computation within the physical sensor



(a) Cumulative energy consumption (b) Packet delivery rate

Fig. 3. Comparative study of performance in terms of network parameters.



(a) $n = 5, t = 5$ (b) $n = 5, t = 8$ (c) $n = 10, t = 8$ (d) $n = 2, t = 8$

Fig. 4. Analysis of price charged (due to hardware) with time.

nodes is the least in pH. As observed in Fig. 3a, Sprite leads in terms of the energy expenditure due to transmission. This is because Sprite periodically communicates with the CCS, sending packets containing the receipts of the currency to be obtained by every physical node. For both PPM, and pH, the energy expended due to transmission is significantly low. However, in PPM, retransmission of packet is required sometimes because of under-estimation of nuglets, thereby incurring an additional energy overhead. The overall effect is indicated by the line-plots for the total energy expenditure.

Fig. 3b compares pH, PPM, and Sprite in terms of packet delivery rate. For 10 end-users, $n = \{100, 200, 300, 400, 500\}$ number of nodes, every node is allowed to transmit data to the BS, under identical channel conditions using pH, PPM, and Sprite. PPM estimates the nuglets before start of packet transmission. However, sometimes due to underestimation of the nuglets, the packets are dropped midway. On the other hand, Sprite, periodically transmits the receipt of the messages from each node to the CCS, thereby overloading the network, and reducing the packet delivery rate. However, for pH, pricing does not affect the network load at all. The prices charged are transmitted along with the data packets to the cloud-end. The computation, and the monetary transactions are handled outside the network, which increases the chance of the packet delivery rate. Fig. 3b depicts the variation of the packet delivery rate with the increase in the number of nodes. For every iteration, the experiment is repeated for 50 times and the data plot is shown within a 95 percent Confidence Interval (CI).

The price charged at various time instants by different sensor owners for a single end-user is also shown. Fig. 4 highlights the sequence of the price charged and the point at which the optimality is reached. Fig. 4a demonstrates a five-hop scenario ($n = 5$) involving five different sensor owners, where $s(n_1)$ is the owner of the source sensor node. As indicated in the figure, $s(n_i)$ initially charges a price, based on which the price charged by $s(n_{i+1})$ depends. The price charged at $t = 1$ increases with time. However, it does not exceed the equivalent user utility, that we have assumed to be 10,000. Thus, the tendency is to reach the user-utility as close as possible, but not exceed it. For the sake of simulation, we define a new metric defined below.

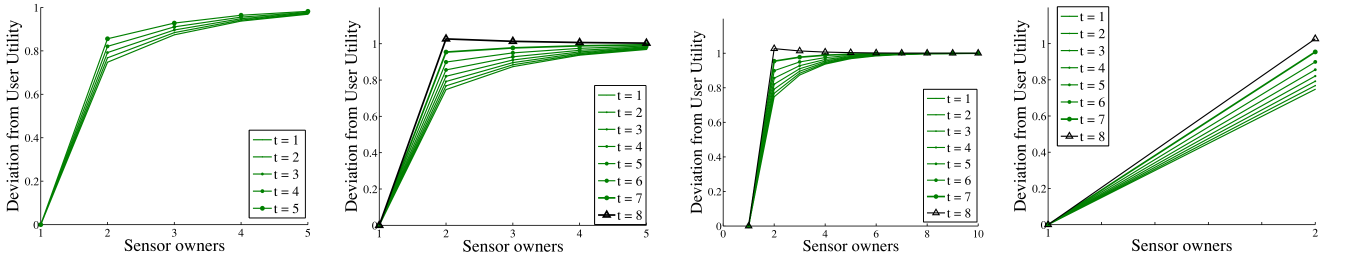
Definition 10. *Deviation from the user utility (d) is a metric in the scale of 0 to 1 that indicates the degree of convergence of the price charged by the sensor owners to the utility. It is computed as,*

$$d = 1 - \frac{\mathcal{U} - p^{s(i)}}{\gamma_2 - \gamma_1}. \tag{39}$$

Practically, $d \rightarrow 1$, but $d \neq 1$. Corresponding to Fig. 4a, Fig. 5a shows the tendency of convergence of the price charged with the user utility. In Fig. 4b, the experiment was done for $n = 5$, $t = 8$. At $t = 8$, we find that $s(n_1)$ exceeds the user-utility. From this, we conclude that the price charged by the sensor owners attains optimality at $t = 7$, for this simulation setup. Fig. 5b indicates the asymmetry of the pattern at $t = 8$, as $d > 1$. To infer with generality, we performed the same experiment over a different setup, where $n = 10$, $t = 8$, as shown in Fig 4c. Even with the increase in the number of hops, it is found that that equilibrium is reached at $t = 7$. Fig. 5c supports the equilibrium at $t = 7$. Figs. 4d and 5d demonstrate the same effect with a setup of $n = 2$, $t = 8$. Thus, the system attains its equilibrium at $t = 7$. Hence, without the loss of generality, it can be inferred that for a particular network setup, the system attains equilibrium after a finite period of time t_f, after which the sequence $\{p_t^{s(n_i)}\}$ stabilizes, i.e., $\forall t \geq t_f, p_{t+1}^{s(n_i)} = p_t^{s(n_i)}$, and $p_t^{s(n_i)} = p_{t_f}^{s(n_i)}$.

5.3 Analysis of pI

This section puts forth the performance analysis of pI. pI primarily provides the pricing scheme for the infrastructure of virtualization. The simulation setup for pI is illustrated in Table 4.

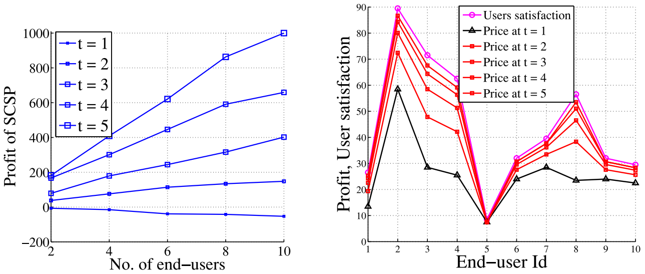


(a) $n = 5, t = 5$ (b) $n = 5, t = 8$ (c) $n = 10, t = 8$ (d) $n = 2, t = 8$

Fig. 5. Analysis of the tendency of the charged price to converge with the user utility.

TABLE 4
Simulation Setup for pI

Parameters	Values
Building cost of VM	4 unit
Building cost of vs	3 unit
Price of VM per unit	5 unit
Price of vs per unit	4 unit
Maintenance Cost of vs per time slot	2 unit
Number of end-users	[1, 10]
Number of VMs per user	1
Service rate of SCSP	15 demand/sec
β	0.5



(a) Number of end-users vs profit of SCSP

(b) Increment in price towards user satisfaction

Fig. 7. Overall analysis of the profit made by the SCSP.

Fig. 6 shows the demand and the user satisfaction $u_e(t)$ provided by the cloud for 10 end-users. As per Definition 8, and also following Fig. 6, it is evident that the $u_e(t)$ varies with the demand λ_{vs_i}. The increase in $u_e(t)$ is clearly reflected by the increase in demand for the virtual sensors. However, if the demand is too small, as in the case of end-user 5, the processing overhead at the sensor-cloud end increases, thereby reducing the user satisfaction. Fig. 7a illustrates the variation of the profit acquired by the SCSP, with time. Initially, at $t = 1$, the SCSP runs at a loss for serving the requests of 10 end-users, indicated by the negative y axis. Only after a period of time, i.e., $t = 2$ onwards, significant profit is incurred with the increase in the number of end-users. Fig. 7b illustrates the timely increment of the price charged by the SCSP, for a fixed user satisfaction, and a fixed demand for five time instants. Clearly, end-users 2 and 3 face 4 increments in the charged price, whereas the price charged from end-users 9, and 10 are incremented only thrice. This is because, the SCSP has a tendency to charge a price close to $u_e(t)$, but not exceed it. This strategy ensures that the end-users are not overcharged with time. The user satisfaction value of end-user five is significantly low (because of low demand and low data urgency), and hence, the SCSP does not get the opportunity to increase the charged price with time. To examine the stability of the proposed system, we simulated for a longer period of time, i.e., for 50 time units, for a single end-user, as shown in Fig. 8. The increasing demand λ_{vs_i} of the end-user, and the corresponding satisfaction $u(t)$ are shown. The price charged by the SCSP increases with the increase in λ_{vs_i}. However, at $t = 44$, it can be seen that the price remains constant, i.e., $p_{vs_i}(45) = p_{vs_i}(44) = p_{vs_i}(43)$, although the demand increases ($\lambda(44), \lambda(45) > \lambda(43)$, and $u(43) = u(44) = u(45)$), mainly to prevent the price from exceeding the user satisfaction.

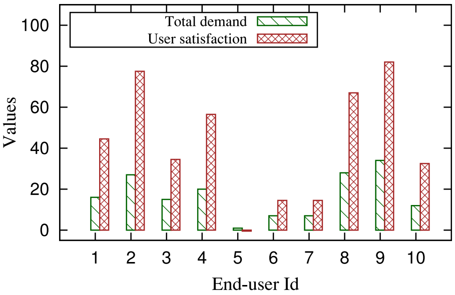


Fig. 6. Analysis of demand and user satisfaction.

Scalability analysis. Motivated by the works of [5], [6], for analysis of the system scalability, we perform an experiment on an increased set of end-users, as shown in Fig. 9. The experiment involves 10,000 to 50,000 end-users, denoted by e_{tot}. The total demand (λ_{tot}) for the end-users are also varied in terms of the number of virtual sensors allocated and is computed as, $\lambda_{tot} = \sum_{j=1}^{e_{tot}} \sum_{i=1}^{n_{e_j}} \lambda_{vs_i}^{e_j}$ where, n_{e_j} is the total number of component physical sensor nodes of vs_i for e_j. With the change in the request for the vs_i, the number of the allocated physical sensors is altered and by changing the number of allocated vs_i, $\lambda_{vs_i}^{e_j}$ is altered for multiple end-users. As depicted in Fig. 9a, with the increase in λ_{tot}, the profit of the SCSP $r(t)$ increases as per Equation (23). Therefore, the cumulative profit over all the end-users also increases and is evaluated as $\sum_{j=1}^{e_{tot}} r(t)_{e_j}$. However, as illustrated through Fig. 9b, we observe that the average user satisfaction $u_e(t)$ is above the threshold v_{opt} and remains almost unchanged with the increase in demand. We also observe that with 10,000 and 20,000 end-users, the $u_e(t)$ has a mean of approximately 47,500 and lies within the interval of [48,900, 45,600] with 95 percent confidence. However, at larger demands, the mean user satisfaction tends to lie at a slightly wider interval of [44,000, 51,000] with 95 percent confidence, but, the mean satisfaction stands at 47,400. From this we infer that even with the increase in larger demands for a greater and varying number of end-users, the SCSP incurs an increasing positive profit and the user satisfaction is simultaneously maintained. This justifies the scalability of the system.

5.4 Complexity Analysis

In this section, we analyze the asymptotic computational complexity of pH, and pI to examine its real-time processing ability. The complexity of computation is measured in terms of the simulation time required for the execution of the

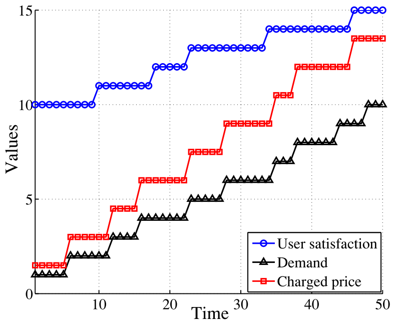
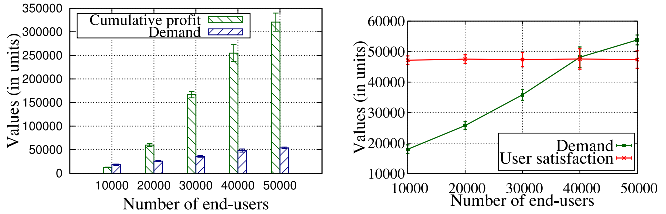


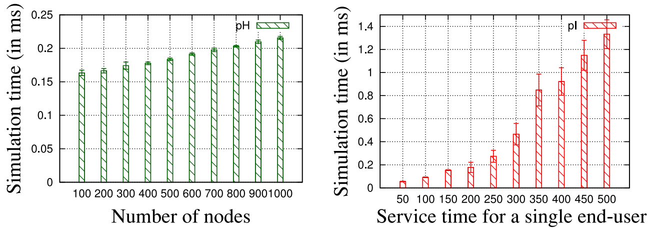
Fig. 8. Analysis of the correlation of price, demand, and user satisfaction



(a) Study of demand and cumulative profit

(b) Study of demand and user satisfaction

Fig. 9. Analysis of scalability of the system.



(a) Number of physical nodes vs simulation time

(b) Number of physical nodes vs simulation time

Fig. 10. Overall analysis of the profit made by the SCSP.

algorithms. Fig. 10a demonstrates the variation in the computational time with the increase in the number of the underlying physical sensor nodes. The mean simulation time is observed to be within the interval $[0.27, 0.82]$ with 95 percent confidence. Thus, we find that the increase in the number of the physical sensor nodes has significantly low impact of the computational complexity of pH.

Fig. 10b depicts the computational complexity of pI. The experiment is executed for serving the requests of a single end-user with varying demands for a varied period of time, from 50 to 500 time instants, and the corresponding execution time is calculated and analyzed to examine the computational complexity of pH. The mean simulation time was found to lie between $[0.21, 0.81]$ with 95% confidence. Therefore, both pH and pI are suitable for real-time implementation.

5.5 Real-Life Applicability: A Case Study

In this Section, we discuss the real-life applicability of such pricing schemes. As sensor-cloud is a new dimension of cloud computing, it has to follow the pay-as-you-go model of the cloud markets. Thus, it is imperative for the end-users of sensor-cloud to quantify their usage so that they can be charged to pay accordingly. A case study for an environment monitoring application is shown in Fig. 11.

The usage of resources in provisioning Se-aaS is quite significant to charge the payment from an end-user. The sensor-owners are also payed on a rental basis. Thus, for any application fed with data from sensor-cloud has to undergo through a pricing scheme. From our previous study [1], [4], [48], we observe that based on the template specifications, an end-user is allocated one or more virtual sensor. In Fig. 11, the end-user requests for Se-aaS to serve an environment monitoring application. During the entire tenure of obtaining Se-aaS, s/he is liable to pay for his usage. Herein, comes the motivation of pricing within sensor-cloud. For usage of every physical sensor, for sensing or communication, some price is charged through pH and for availing any cloud component over time, price is charged through pI. The final price payed by the end-user is distributed to the involved sensor-owners and the SCSP. Similar to such application, the proposed pricing scheme can be made applicable to any sensor-based application served through sensor-cloud.

6 Conclusion

In this paper, we have proposed a dynamic pricing model for rendering Se-aaS through the sensor-cloud infrastructure into two sections: pH and pI. pH deals with the pricing scheme for hardware with the aim to maximize the profit of several sensor owners involved in the data transmission. It presents the pricing scheme for maximizing the profit of the SCSP, by considering the user satisfaction at different time instants. A comparative study of the next hop selection is done for pH with PPM, and Sprite. It is observed that pH outperforms the aforesaid models in terms of residual energy, proximity with BS, RSS, and overhead. Moreover, pH reduces the cumulative energy consumption, and increases the packet delivery rate. The analysis of pI shows how SCSP incurs profit and the user satisfaction is also met, simultaneously. Finally, the complexity analysis of pH and pI are also performed and is analyzed to justify their real-time processing ability.

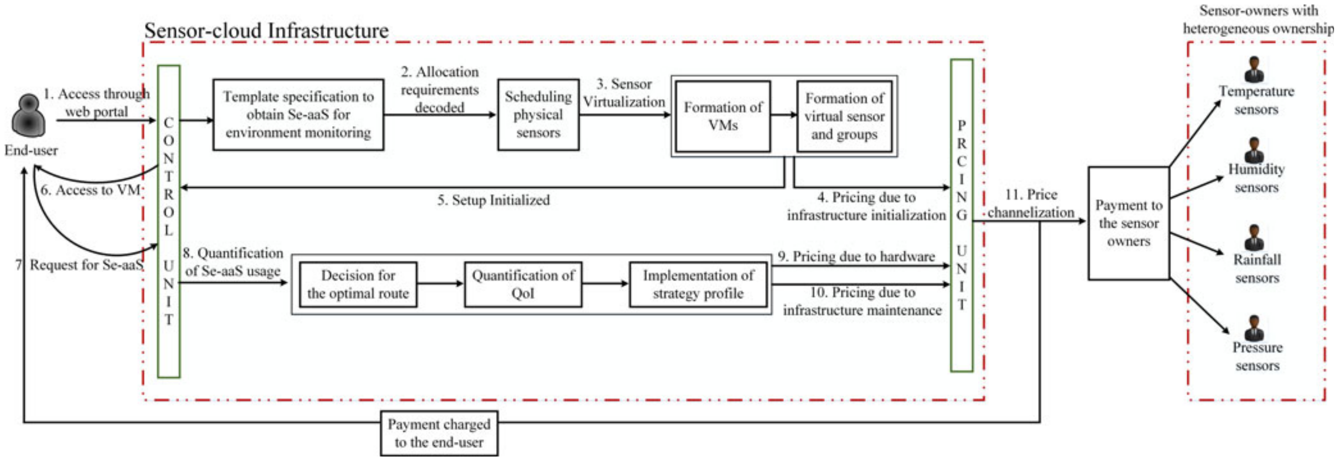

Sensor-cloud Infrastructure
Sensor-owners with heterogeneous ownership
End-user
1. Access through web portal
6. Access to VM
7. Request for Se-aaS
CONTROL UNIT
Template specification to obtain Se-aaS for environment monitoring
2. Allocation requirements decoded
Scheduling physical sensors
3. Sensor Virtualization
Formation of VMs
Formation of virtual sensor and groups
5. Setup Initialized
4. Pricing due to infrastructure initialization
8. Quantification of Se-aaS usage
Decision for the optimal route
Quantification of QoI
Implementation of strategy profile
9. Pricing due to hardware
10. Pricing due to infrastructure maintenance
PRICING UNIT
11. Price channelization
Payment to the sensor owners
Temperature sensors
Humidity sensors
Rainfall sensors
Pressure sensors
Payment charged to the end-user

Fig. 11. Applicability of pricing within an environment monitoring application.

ACKNOWLEDGMENTS

This work was partially supported by a fellowship sponsored by the Tata Consultancy Services (TCS), 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 Syst. J.*, 2015, Doi: 10.1109/JSYST.2014.2362617.

[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," *Int. J. Distrib. Sensor Netw.*, vol. 2013, p. 917923, Nov. 2013.

[3] K.-L. Tan, "What's next?: Sensor + cloud!?" in *Proc. 7th Int. Workshop Data Manage. Sensor Netw.*, 2010, p. 1.

[4] S. Chatterjee and S. Misra, "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1435–1438, Aug. 2014.

[5] L. Ferretti, F. Pierazzi, M. Colajanni, and M. Marchetti, "Scalable architecture for Multi-user encrypted SQL operations on cloud database services," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 448–458, Oct. 2014.

[6] X. Zhang, L. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using map-reduce on cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 363–373, Feb. 2014.

[7] K.-W. Park, J. Han, J. Chung, and K. H. Park, "Themis: A mutually verifiable billing system for the cloud computing environment," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 300–313, Jul.–Sep. 2013.

[8] A. Zhou and B. He, "Transformation-based monetary cost optimizations for workflows in the cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 85–98, Mar. 2014.

[9] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 158–171, Jul. 2013.

[10] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure—Physical sensor management with virtualized sensors on cloud computing," in *Proc. 13th Int. Conf. Netw.-Based Inf. Syst.*, Sep. 2010, pp. 1–8.

[11] J. Shin, M. Jo, J. Lee, and D. Lee, "Strategic management of cloud computing services: Focusing on consumer adoption behavior," *IEEE Trans. Eng. Manage.*, vol. 61, no. 3, pp. 419–427, Aug. 2014.

[12] Y. Feng, B. Li, and B. Li, "Price competition in an oligopoly market with multiple IaaS cloud providers," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 59–73, Jan. 2014.

[13] H. Lu, X. Wu, W. Zhang, and J. Liu, "Optimal pricing of multi-model hybrid system for PaaS cloud computing," in *Proc. Int. Conf. Cloud Service Comput.*, Nov. 2012, pp. 227–231.

[14] D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized nash equilibria for the service provisioning problem in cloud systems," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 429–442, Oct.–Dec. 2013.

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

[16] S. Misra and S. Sarkar, "Priority-based Time-slot allocation in wireless body area networks during medical emergency situations: An evolutionary Game-theoretic perspective," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 2, pp. 541–548, Mar. 2015.

[17] S. Z. Yale and S. Zhong, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, 2002, pp. 1987–1997.

[18] P. Chavali and A. Nehorai, "Managing multi-modal sensor networks using price theory," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4874–4887, Sep. 2012.

[19] J. Elias, F. Martignon, L. Chen, and E. Altman, "Joint operator pricing and network selection game in cognitive radio networks: Equilibrium, system dynamics and price of anarchy," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4576–4589, Nov. 2013.

[20] S. Li and J. Huang, "Price differentiation for communication networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 703–716, Jun. 2014.

[21] S.-K. Ng and W.-G. Seah, "Game-theoretic approach for improving cooperation in wireless multihop networks," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 40, pp. 559–574, Jun. 2010.

[22] L. Buttyan and J.-P. Hubaux, "Enforcing service availability in mobile Ad-hoc wans," in *Proc. 1st Annu. Workshop Mobile Ad Hoc Netw. Comput.*, Aug. 2000, pp. 87–96.

[23] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[24] Z. Li and M. Li, "A hierarchical cloud pricing system," in *Proc. IEEE 9th World Congr. Services*, Jun. 2013, pp. 403–411.

[25] W. Wang, B. Liang, and B. Li, "Revenue maximization with dynamic auctions in IaaS cloud markets," in *Proc. IEEE/ACM 21st Int. Symp. Quality Service*, Jun. 2013, pp. 1–6.

[26] H. Xu and B. Li, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 2929–2933.

[27] B. Sharma, R. Thulasiram, P. Thulasiraman, S. Garg, and R. Buyya, "Pricing cloud compute commodities: A novel financial economic model," in *Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2012, pp. 451–457.

[28] S. Son and K. M. Sim, "A Price-and-time-slot-negotiation mechanism for cloud service reservations," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 42, no. 3, pp. 713–728, Jun. 2012.

[29] A. Prasad and S. Rao, "A mechanism design approach to resource procurement in cloud computing," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 17–30, Jan. 2014.

[30] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Services Comput.*, vol. 5, no. 2, pp. 164–177, Apr. 2012.

[31] S. Ren and M. van der Schaar, "Joint design of dynamic scheduling and pricing in wireless cloud computing," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 185–189.

[32] H. Roh, C. Jung, W. Lee, and D.-Z. Du, "Resource pricing game in Geo-distributed clouds," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1519–1527.

[33] H. Qin, X. Wu, J. Hou, H. Wang, W. Zhang, and W. Dou, "Self-adaptive cloud pricing strategies with Markov prediction and data mining method," in *Proc. Int. Conf. Cloud Service Comput.*, Nov. 2012, pp. 219–226.

[34] I. Jangjaimon and N. Tzeng, "Effective cost reduction for elastic clouds under spot instance pricing through adaptive checkpointing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 396–409, Feb. 2015.

[35] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki, "Optimal service pricing for a cloud cache," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1345–1358, Sep. 2011.

[36] N. Edalat, W. Xiao, C.-K. Tham, E. Keikha, and L.-L. Ong, "A price-based adaptive task allocation for wireless sensor network," in *Proc. IEEE 6th Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2009, pp. 888–893.

[37] Z. Hui, Q. Zhi-hong, S. Da-yang, and Z. Ding-guo, "Study on price-driven load balance in wireless sensor network," *ICIE Int. Conf. Inf. Eng.*, vol. 1, pp. 355–358, Jul. 2009.

[38] J. Musacchio and J. Walrand, "Wifi access point pricing as a dynamic game," *IEEE/ACM Trans. Netw.*, vol. 2, no. 2, pp. 289–301, Apr. 2006.

[39] R. K. Lam, D.-M. Chiu, and J. C. Lui, "On the access pricing and network scaling issues of wireless mesh networks," *IEEE Trans. Comput.*, vol. 56, no. 11, pp. 1456–1469, Nov. 2007.

[40] K. Cheung, H. So, W.-K. Ma, and Y. Chan, "Received signal strength based mobile positioning via constrained weighted least squares," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, vol. 5, pp. 137–140.

[41] M. Hossain, P. Atrey, and A. Saddik, "Context-aware qoi computation in Multi-sensor systems," in *Proc. 5th IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, Sep. 2008, pp. 736–741.

[42] E. Ciftcioglu, A. Yener, and M. Neely, "Maximizing quality of information from multiple sensor devices: The exploration vs exploitation tradeoff," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 883–894, Oct. 2013.

[43] S. Sheng and R. Gao, "Structural dynamics-based sensor placement strategy for high quality sensing," in *Proc. IEEE Sensors*, Oct. 2004, vol. 2, pp. 642–645.

[44] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, USA: MIT Press, 1991, ch. 8.

[45] R. Wolff, *Stochastic Modelling and the Theory of Queues*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[46] X. Yu, “Distributed cache updating for the dynamic source routing protocol,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 609–626, Jun. 2006.

[47] S. Adibi and G. Agnew, “Multilayer flavoured dynamic source routing in mobile ad-hoc networks,” *IET Comms*, vol. 2, pp. 690–707, May 2008.

[48] S. Chatterjee and S. Misra, “Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud,” in *Proc. IEEE Int. Conf. IEEE Adv. Netw. Telecommun. Syst.*, 2014, pp. 1–6.

Subarna Chatterjee is a TCS Research Scholar and a Google Anita Borg Fellow, pursuing her PhD from the School of Information Technology, Indian Institute of Technology, Kharagpur, India. She received her BTech degree in Computer Science and Technology from West Bengal University of Technology, India in 2012. Her current research interests include networking and communication aspects of Cloud Computing in Wireless Sensor Networks. He is student member of the IEEE.

Ranjana Ladia is working toward the BTech degree in computer science from National Institute of Technology, Durgapur. She has also been an intern from the Indian Institute of Technology, Kharagpur, India. Her current research interests include cloud computing, networking and telecommunications, and service oriented architectures (SOAs).

Dr. Sudip Misra is an Associate Professor in the School of Information Technology at the Indian Institute of Technology Kharagpur. Prior to this he was associated with Cornell University (USA), Yale University (USA), Nortel Networks (Canada) and the Government of Ontario (Canada). He received his PhD degree in Computer Science from Carleton University, in Ottawa, Canada, and the masters and bachelors degrees respectively from the University of New Brunswick, Fredericton, Canada, and the Indian Institute of Technology, Kharagpur, India. 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. His current research interests include algorithm design for emerging communication networks. He is the author of over 260 scholarly research papers (including 160+ papers, of which 40+ papers are in ACM/IEEE Transactions, Journals, & Magazines). He has won *nine research paper awards* in different conferences. He was awarded the 3rd Prize in the *Samsung Innovation Award* (2014) at IIT Kharagpur, and also the *IEEE ComSoc Asia Pacific Outstanding Young Researcher Award* at IEEE GLOBECOM 2012, Anaheim, California, USA. He was also the recipient of several academic awards and fellowships such as the *Young Scientist Award* (National Academy of Sciences, India), *Young Systems Scientist Award* (Systems Society of India), *Young Engineers Award* (Institution of Engineers, India), (*Canadian*) *Governor General’s Academic Gold Medal* at Carleton University, the *University Outstanding Graduate Student Award* in the Doctoral level at Carleton University and the *National Academy of Sciences, India – Swarna Jayanti Puraskar* (Golden Jubilee Award). He was also awarded the Canadian Government’s prestigious *NSERC Post Doctoral Fellowship* and the *Humboldt Research Fellowship* in Germany. He has published 8 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, published by reputed publishers such as Springer, Wiley, and World Scientific. He is senior member of the IEEE.