

On Theoretical Modeling of Sensor-Cloud: A Paradigm Shift From Wireless Sensor Network

Sudip Misra, *Senior Member, IEEE*, Subarna Chatterjee, *Student Member, IEEE*, and Mohammad S. Obaidat, *Fellow, IEEE*, *Fellow, SCS*

***Abstract*—This paper focuses on theoretical modeling of sensor-cloud – one of the first attempts in this direction. We endeavor to theoretically characterize virtualization, which is a fundamental mechanism for operations within the sensor-cloud architecture. Existing related research works on sensor-cloud have primarily focused on the ideology and the challenges that WSN-based applications typically encounter. However, none of the works has addressed theoretical characterization and analysis, which can be used for building models for solving different problems to be encountered in using sensor-cloud. We present a mathematical formulation of sensor-cloud, which is very important for studying the behavior of WSN-based applications in the sensor-cloud platform. We also suggested a paradigm shift of technology from traditional WSNs to sensor-cloud architecture. A detailed analysis is made based on the performance metrics – energy consumption, fault-tolerance, and lifetime of a sensor node. A thorough evaluation of the cost-effectiveness of sensor-cloud is also done by examining the cash inflow and outflow characteristics from the perspective of every actor of sensor-cloud. Analytical results show that the sensor-cloud architecture outperforms traditional WSN by increasing the sensor lifetime by 3.25% and decreasing the energy consumption by 36.68%. We also observe that technology shift to sensor-cloud reduces the expenditure of an end-user by 14.72%, on an average.**

***Index Terms*—Wireless Sensor Network, Sensor-cloud, Virtualization, Modeling and Simulation of sensor-clouds**

I. Introduction

The emergence of Wireless Sensor Networks (WSNs) [1], [2] has enhanced the standard of living of mankind with the touch of advanced technology. The manifestations of this fact are found in numerous real-life applications such as target-tracking [3], [4], battlefield monitoring [5], [6], telemonitoring [7], ubiquitous monitoring [8], [9], and several other applications [10], [11]. However, all of these WSN-based applications are *single-user centric*, in which a user-organization owns and deploys its personalized sensor network and typically does not share the accessed data to another party (user/organization). This holds true especially if the WSN application is physical security-centric, such as that involving target tracking, zone monitoring, and terrain surveillance. For applications that do not primarily involve security aspects, viz., environment monitoring, and telemonitoring, the administrator of a particular WSN may agree to share the sensed data in exchange of money. It is obvious that data sharing policies vary across organizations. However, an external user-organization is able to retrieve sensor information that is specific only to the region that is administered by the network administrator. Thus, generally, only user-organizations that own a sensor network have satisfactory access to sensor data. Recently, *sensor-cloud* architecture has been conceived as a potential solution for multi-organization WSN deployment and data access [12]–[14]. This work studies the performance enhancements that can be obtained using sensor-cloud platform over traditional WSN architecture for sensor-network.

Among the pioneers who promoted sensor-cloud based terrain/environment monitoring, IntelliSys (http://www3.ntu.edu.sg/intellisys/index.html), and MicroStrains (http://www.sensorcloud.com/system-overview) stand distinct. According to MicroStrains, sensor-cloud is formally defined as [14]:

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

The idea of sensor-cloud thrives on the principle of virtualization of physical sensor nodes. The sensor-cloud architecture is positioned as an intermediate stratum of processing between the physical sensor nodes and the end-user-organization. The user-organizations possess their own applications, and request the sensor-cloud for retrieval of sensed data. These requests are interpreted within the sensor-cloud environment and the physical sensor nodes are dynamically consorted to form virtual sensor groups, as per requirements. Data from the wireless physical sensor nodes reach the sensor-cloud through standard wireless multi-hop communication. On behalf of each virtual group, the aggregated data is transmitted to the end-user organization. The data obtained is then delivered to the application, followed by subsequent processing and analysis on part of the end-users.

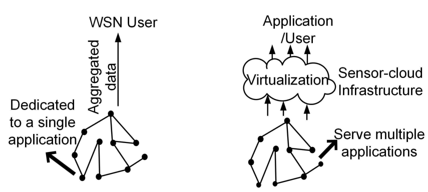
The motivation behind selecting sensor-cloud as the operational platform for WSN-based applications is discussed in the following subsection. Table I briefly outline the responsibilities of the actors for both WSN and sensor-cloud.

Table I: A comparison of the roles of actors. CSP refers to the Cloud Service Provider.

Actors and Roles		
	WSN	Sensor-cloud
Ownership	WSN-user	Sensor-owner
Deployment	WSN-user	Sensor-owner
Redeployment	WSN-user	CSP
Maintenance	WSN-user	CSP
Overhead	WSN-user	CSP
Usage	WSN-user	End-user

A. Motivation

Sensor-cloud is essentially a cloud platform for retrieval, storage and analysis of huge amount of heterogeneous sensed data. The main motivation behind introducing the idea of sensor-cloud is to allow a user-organization to remain unaware of the actual physical location of sensor nodes through the process of virtualization. Virtualization creates a complete abstraction of the underlying physical sensor nodes, independent of the physical topology, as shown in Fig. 1(a). Fig. 1(b) considers the same topology of sensor nodes, but is based on the cloud architecture. Thus, the architecture for sensor-cloud is independent of topology or the orientation of resources. The paramount success of operating applications in a sensor-cloud environment is the widespread dissemination of the usability of sensor nodes to every user (with an application), even without owning sensor nodes. The use of sensor-cloud also reduces the additional responsibilities that a user of a WSN has to bear due to maintenance, replacement, redeployment, and other hardware-management overheads. Thus, unlike WSN, every user-organization envisions sensors as a service, rather than as a conventional physical hardware.



(a) Wireless Sensor Network (b) Sensor-cloud

Figure 1: Analysis of topology independence

Sensor-as-a-Service (*Se-aaS*) is also highly cost-effective compared to the traditional WSNs. Thus, a user-organization is relieved from initial high deployment costs and auxiliary management costs. From a billing point of view, Se-aaS is quantified into measurable units, and user-organizations are charged for the consumable units only. This pay-per-use model contributes to the overall prosperity of sensor-cloud.

The sensor-cloud architecture also increases the utility of a physical sensor nodes by enabling them to serve multiple applications. As the sensor nodes of a WSN comprise of operating systems with monolithic kernel (in which an application remains compiled), the sensor nodes are application specific to a particular application only. Generally, it is infeasible to schedule and load multiple applications (of the same type) within a single sensor node. However, in some cases, a two-stage bootloader is used to support switching between multiple applications. But such operations involve manual intervention, and significantly high overhead cost due to memory management, operating system independence, and complicated process management [15].

B. Contribution

In this work, we mathematically justify the necessity for a paradigm shift for all WSN-based applications to a sensor-cloud platform. Performance evaluation is performed for both applications running both on a traditional WSN and sensor-cloud platform. The major contributions of this work are catalogued as follows.

- The current state-of-the-art generally does not allow the users to access WSN-based applications without owning the sensor nodes, and deploying the same. Our work significantly contributes towards dissemination of the access of such applications to multiple persons/organizations. For this purpose, the work focuses on a theoretical modeling of virtualization of physical sensor nodes.
- In contrast to traditional WSN-based technology, sensor-cloud remarkably improves on the pricing scheme. Sensor-cloud enables a user-organization to scale (up/down) its demands, and pay only for the service it seeks/receives. The user-organization is relieved from the deployment, and maintenance overheads associated with a typical WSN. This work illustrates the flow of revenue for each of the actors associated with the sensor-cloud architecture.
- The work also suggests a framework for performance analysis of sensor-cloud based on few chosen metrics such as *fault-tolerance*, *lifetime of a sensor node*, and *energy consumption*, in contrast to that of a WSN.
- Finally, this work endeavors to conceive the idea of using physical *sensors as a service* (*Se-aaS*). Unlike a WSN that realizes sensor nodes as mere hardware components, sensor-cloud facilitates the end-users to render sensor nodes as cost-effective on-demand service.

C. Organization of the paper

The rest of the paper is organized as follows. Section II discusses the related work in this area. Section III focuses on the details of different views or perspectives of sensor-cloud architecture for various actors (users). In Section IV, we present a mathematical model for virtualization within the sensor-cloud architecture. Section V illustrates and evaluates the performance of sensor-cloud in comparison to conventional WSNs. Section VI presents a comparative case study of practical application scenarios in the context of sensor-cloud and WSNs. Finally, Section VII concludes and discusses the future scope for this work.

II. Related Work

In this Section, we thoroughly discuss, and analyze the work that has been done so far on sensor-cloud. Before the concept of sensor-cloud was actually proposed, quite a good number of works explored the real-time communication aspects of cloud computing [16], [17]. Some works focused on the integration of sensors to a cloud framework. In [18], Misra *et al.* considered an integration of sensors with cloud from a perspective of health monitoring. The work focuses on an optimal selection of gateway in order to obtain the maximum bandwidth required for health data transmission. However, our work does not focus only on the integration of sensor networks and cloud computing. The work also contributes by formally modelling the virtualization of sensors within the sensor-cloud environment.

Some of the fundamental works was also addressed by Eugster *et al.* [19]. They proposed a publish/subscribe model that demonstrates the interaction between a publisher and subscriber based on notification of an event. This work is considered to form the basis of integrating sensor nodes in a cloud environment, as it focuses on data transfer between dissimilar entities of a system. Hassan *et al.* [12] projects the challenges normally encountered while integrating WSN with cloud. The work proposes a sensor-cloud framework focusing mainly on Software-as-a-Service (SaaS) applications. The work also proposed Statistical Group Index Matching (SGIM) scheme, which can be used to transfer data to cloud applications, and evaluates it to exemplify its remarkable performance, compared to the existing algorithms. A similar effort has been put by Eggert *et al.* in [20]. The work highlights the challenges that will be faced due to the difficulty in understanding the diverse nature, implementation of the varied and scalable functionalities, and ensuring privacy in sensor-cloud. Additionally it draws a baseline for addressing the aforesaid issues. In another work, Kumar *et al.* [21] devised a mechanism for transferring large volume of sensed data from the local memory of sensor nodes to a cloud storage. The authors also proposed to transfer the responsibility of data processing to the cloud gateways, thereby achieving high energy efficiency. The authors exercised the algorithm for back propagation networks within the cloud-gateways to execute data filtration. We see that most of the aforesaid works have primarily enlisted and discussed the benefits of sensor-cloud and the challenges involved with the same. Few of the above works have focused on designing an application-specific framework and the data transmission methodologies. However, our work focuses on a theoretical characterization of sensor-cloud. We also present a comparative study of sensor-cloud, and analyze the performance in contrast to WSNs.

Alamri *et al.* [14] presented a thorough survey on sensor-cloud, its definition, the intrinsic concepts and the benefits of using it. The paper also presents a comparison of the type of message flows for different algorithmic approaches. Eventually, the authors have also briefed about the possible technical challenges in this aspect. Another recent work that has proved to be highly advantageous and constructive towards sensor-cloud research is by Yuriyama and Kushida in [13]. This work has clearly carved out the constructive and opportunistic aspects of sensor-cloud architecture to a great extent. Few works are also focused on virtualization in sensor networks. Olariu *et al.* [22] contributed in this domain by proposing a very simple and general-purpose virtual infrastructure for WSNs. It is a protocol independent work that can be used by the existing routing or data aggregation protocols. Ojha *et al.* [23] has dealt with topology virtualization by self-organization of nodes in Underwater Sensor Networks. Thus, [22] and [23] have focused more on the designing aspects, whereas, our work concentrates on the theoretical characterization of the virtualization model and a comparative illustration of sensor-cloud and WSN.

Evidently, despite the upsurge in research on sensor-cloud, there lacks mathematically-based theoretical works that can help in supporting performance evaluation and analysis of sensor-cloud based systems. This work proposes a detailed formalization of the mathematical model behind *virtualization*, a key enabler of the sensor-cloud technology. In [24], an idea for a high-level model for virtualuization is proposed. Our main focus, in this specific work, is to justify the necessity for a shift of technology from the conventional WSN to a sensor-cloud platform in the near future.

III. Sensor-cloud Architecture

This Section presents the details of the architectural aspects of sensor-cloud from two different points of view: (a) *User-organization's view* or the *logical view*, and (b) *Algorithmic view* or the *real view*.

Sensor-cloud architecture is essentially a three-tier architecture [13], as shown in Fig. 2.

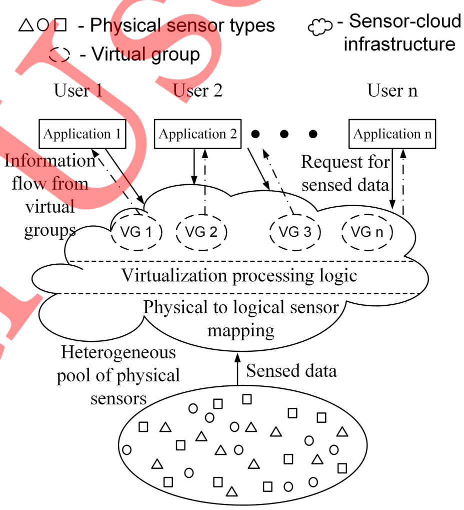


Figure 2: Architecture of Sensor-cloud

Initially, we present the architectural design of the logical view, i.e., the user-organization's view of obtaining Se-aaS. The communication interface of a user-organization is primarily a Web interface running at the site of the Cloud Service Provider (CSP). It is a Web portal through which the user-organization requests for Se-aaS [13]. After the user-organization logs into the portal, the CSP presents some specific templates that collect information relevant to the type of application such as the type of sensor nodes that the user is expecting, and the region that the user is interested in.

Having specified the relevant details, the user-organization is kept abstracted from the underlying complex processing logic required due to physical sensor node allocation, application-specific aggregation, and virtualization. Following the consolidated data processing, the user-organization retrieves the sensed data from the CSP, which, in turn, is fed into the intended application. Fig. 3 depicts the logical view of the architecture from the viewpoint of the end-user-organization.

We now discuss the real view of the architecture for the actual processing required within the sensor-cloud architecture. Instructions obtained from the end of a user-organization are extracted from the template data. As sensor-cloud architecture deals with sensor nodes with heterogeneous specifications, the

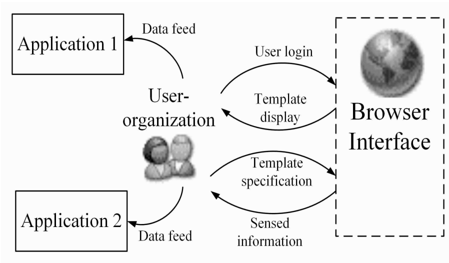


Figure 3: User-organization's view

sensor nodes are standardized using Sensor Modeling Language (SensorML), defined by the Open Geospatial Consortium [13], [25]. To make the processing flexible, manageable, and platform-independent, SensorML uses XML encoding while maintaining the sensor metadata [26]. The directive of the user-organization for the virtual sensor group is interpreted in terms of the physical sensor nodes, and, thereby, scheduling the physical sensor nodes in an on-demand and application-specific manner.

Every physical sensor node reports its sensed data to the sensor-cloud storage. Within the cloud environment, the sensed data are efficiently aggregated in real-time. These data from a consolidated group of sensor nodes are transmitted to the end-user-organization. An end-user anticipates the source of the data to be virtual, which is served from an infinite pool of resources. Fig. 4 shows the diagrammatic representation of the philosophy behind sensor node virtualization.

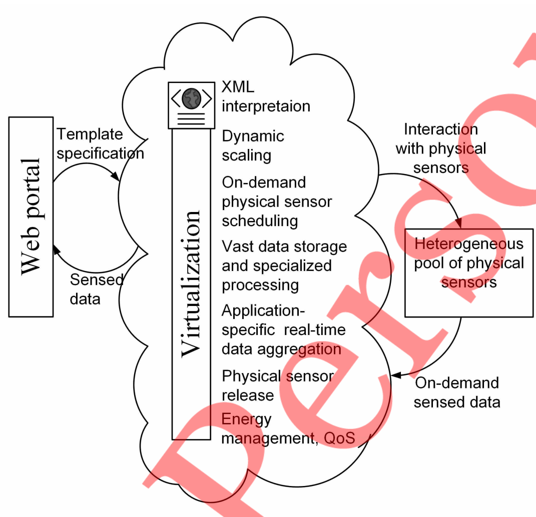


Figure 4: Real view of complex processing

Advantages of Sensor-cloud Architecture

From Figs. 2 and 3, the usefulness of sensor-cloud architecture is well perceived. The end-users of sensor-cloud can be any naive person/organization possessing its own WSN application(s). Unlike conventional WSNs, the end-users can obtain *Se-aaS*, just as water or electricity that can be obtained on-demand, in no time. Thus, sensor-cloud brings in a revolutionary change by enabling the dissemination of WSN technology to the common mass of people/organizations who do not really own WSNs.

As mentioned previously, by virtue of the property of virtualization, sensor-cloud enables run time switching of applications, and real-time data and resource provisioning without the user being aware of the complex processing logic. On the contrary, in a WSN, the nodes are statically configured for a fixed set of applications. From the architectural aspects of sensor-cloud infrastructure, it can be inferred that only the virtualization aspect of it makes it so convenient, accessible, beneficial, and adaptable for public interests. The pay-per-use policy within sensor-cloud also adds on to the benefits of the end-users by diminishing the huge expenditure incurred for setup, maintenance, and management of WSNs.

Difference With Virtual Sensor Network

It is important to differentiate sensor-cloud from a Virtual Sensor Network (VSN). As stated by Jayasumana *et al.* [27], a VSN is a logical subset of a WSN, dedicated for a specific application. The necessity of such a network arises, when two or more sensor nodes serving the particular application are non-adjacent in terms of connectivity. The naming behind VSN is justified by the fact that the network user is abstracted from the complexities involved to set up communication link between multiple zones of the same VSN. The underlying problem is addressed in the said work by involving other nodes (of a different WSN) to provide support in multi-hop communication. Nodes of a VSN can be an intermediate hop for communication with another VSN. Thus, a VSN is distinctly divergent from a sensor-cloud. Sensor-cloud tries to virtualize physical resources and render Se-aaS.

IV. Virtualization Model

This Section describes the mathematical model for virtualization of sensor node resources. As already mentioned in Section III, every sensor node is standardized with an XML encoding. Prior to mathematically formulating the virtualization model, we define the entities and the sub-entities which play active roles in the process of virtualization.

Definition 1. The type of a physical sensor node, along with its specification, T_i, is interpreted to be an element from the set $T = \{T_1, T_2, ..., T_\alpha\}$, where α is the number of distinctly registered sensor types.

For example, T_1 may represent a ADXL345 3-Axis 3g accelerometer, whereas T_2 may be the type indicator of a Laser Doppler Vibrometer.

Definition 2. Every sensor owner is denoted by O_i, such that, $O_i \in O = \{O_1, O_2, ..., O_\beta\}$, where β is the total number of sensor owners who contribute towards the sensor-cloud architecture.

A sensor owner can voluntarily register into or deregister from the sensor-cloud.

Definition 3. The location of a physical sensor node is denoted by a 2-tuple $Loc = < l_1, l_2 >$, where l_1 and l_2 represent the latitude and longitude of the position of the sensor node, correctly, upto a negotiated precision value. The location of a physical sensor node is stored within the cloud storage at the time of its registration, following its deployment.

Definition 4. The state of a sensor is denoted by a Boolean variable $st = \{1, 0\}$, to indicate whether the sensor is active (serving any user-organization), or inactive, respectively.

Although the CSP is generally visualized as a centralized authority for provisioning cloud services, the realistic scenario involves a role-specific or region-specific distribution of service providers under a common roof. Thus, distributed cloud service providers are expressed as, $CSP = \{CSP_1, CSP_2, ..., CSP_\gamma\}$, where a total of γ number of cloud service providers are authorized. The Quality of Service (QoS) of a physical sensor node is also a significant component to identify it. It is a composite tuple that includes several sensor node parameters such as sensing range, transmission range, energy status, and sensing accuracy. We denote the set of currently running applications, the set of physical sensor nodes and the set of virtual sensor nodes available within the sensor-cloud as A, S, and V, respectively.

Definition 5. A physical sensor node is represented as a 7-tuple:

$$s =< id, t, o, Loc, st, csp, QoS >, \; t \in T, o \in O, csp \in CSP$$

where, $s.id$ is a sensor identification number, locally unique under $s.csp$.

Definition 6. An application *App* running at the end of a user-organization is a 4-tuple notion expressed as,

$$App =< A_{id}, A_{type}, A_{sec}, A_{span} >$$

where A_{id} is a system generated unique identification for the application, A_{type} is the type of the application, A_{sec} is a metric to measure the extent of expectation of data confidentiality, and A_{span} is the span of the application, as defined in Definition 7.

Definition 7. The span of an application, A_{span}, is a 2-tuple expressed as,

$$A_{span} =< Loc_1, Loc_2, Loc_3, Loc_4 >$$

where Loc_1, Loc_2, Loc_3, and Loc_4, respectively, indicate the location attributes of the four vertices (in sequence) of a rectangular region that is of interest to the application.

Based on the A_{type} and A_{sec}, a compatibility function f_1 is introduced to select a subset of sensor types ($T' \subset T$) and expressed as, $f_1(App.A_{type}, App.A_{sec}) = \{T_i : T_i \in T\} = T'$. After the types of sensor nodes are decided for an application, the selection of sensor nodes is done using a simple allocation function, $f_{alloc}()$.

The allocation function, defined as $f_{alloc} : A \rightarrow S_1$, maps the set of applications to a subset of physical sensor nodes S_1, such that, $S_1 \in 2^S$. The principle of $f_{alloc}()$ involves a sequence of other intermediate functions $f_1()$, $g_1()$, and $g_2()$. the functionality of g_1 is to select a subset of sensor nodes of one or more given types. Thus we have, $g_1 : T \rightarrow 2^S$. $g_1()$ is defined as,

$$g_1(T_j) = \{s_i | s_i \in S, s_i.t = T_j\} \tag{1}$$

The principle of g_2 is to choose the physical sensors based on their physical locations. The chosen sensor nodes comply with the span of an intended running application. It is expressed as, $g_2 : S_1 \rightarrow S_2, S_1, S_2 \in 2^S$.

Combining the definitions of $g_1()$, and $g_2()$, we arrive at the working model of $f_{alloc}()$, which is shown below.

$$\begin{aligned} f_{alloc}(App) &= g_2(g_1(f_1(App.A_{type}, App.A_{sec}))) \\ &= g_2(g_1(T')) \\ &= g_2(\hat{s}, |\hat{s} \in S', S' \subset S, \hat{s}.t \in T') \\ &= \{s \in S_1, S_1 \subseteq S', circ(s.Loc, R_s) \subset App.A_{span}, \\ &\quad\; s.st = 0, s.QoS \geq \delta\} \end{aligned}$$

where R_s is the sensing radius of the sensor node, and δ is a pre-negotiated QoS threshold value with the CSP and a user-organization. After defining a physical sensor node resource and an application, mathematically, we now introduce a mapping $f_{vir} : S \rightarrow V$ expressed as,

$$f_{vir}(f_{alloc}(App_i)) = v_{App_i} \tag{2}$$

A user-organization visualizes that each of its applications running through sensor-cloud, is mapped to a virtual sensor. Thus, $f(App) = v_{App}$. Our model considers an application *App* as input. After computing $f_{alloc}(App) = S_1$, f_{vir} takes S_1 as input. We have,

$$f_{vir}(S_1) = v_{App} | x \in S_1 \wedge x.st = 1 \tag{3}$$

Also, $f(App)$ is defined mathematically as,

$$f(App) = y | y \in G, f_{vir}(f_{alloc}(App)) = G = v_{App} \tag{4}$$

We now present some interesting characteristics of the functions of the virtualization model in Propositions 1 and 2.

Proposition 1. *The mapping $f(\cdot)$ from an application App_i to a virtual sensor v is injective.*

Proof: Let us assume that the co-domain of f is V. In a sensor-cloud, the virtual sensors are created in a demand-based manner. Thus, the range V' of f is never a proper subset of co-domain, i.e., $V' \not\subset V$. The CSP cannot have a virtual sensor v that is created, but not assigned to any user-organization. Thus, $\nexists v \in V | f^{-1}(v) = App_i, App_i \in A$. From this, we infer, $V' = V$.

Let us assume, $f(App_i) = v_{App_i}$. We try to allocate v_{App_i} to another application App_j. The physical sensor nodes within v_{App_i} is $S_1 = f_{alloc}(App_i)$. So, we have to allocate S_1 to App_j. But $\forall s \in S_1, s.st = 1$. We have $f_{alloc}(App_j) \neq S_1$. Thus, the following inequalities hold.

$$f_{alloc}(App_i) \neq f_{alloc}(App_j)$$

or, $f_{vir}(f_{alloc}(App_i)) \neq f_{vir}(f_{alloc}(App_j))$ or, $v_{App_i} \neq v_{App_j}$

Thus, we infer, $v_{App_i} = v_{App_j} \Rightarrow App_i = App_j$. This completes the proof. ■

Proposition 2. *The mapping $f_{vir}(\cdot)$ of physical to virtual sensor for an application App_i is surjective (onto).*

Proof: We prove it by the method of contradiction. Let us assume that a particular running application, App_i, requires a single physical sensor node, and f_{vir} does not have a pre-image, i.e., $f_{vir}^{-1}(\cdot) = \emptyset$. As mentioned in Equation (2), $f_{vir}(f_{alloc}(App_i)) = v_{App_i}$. We have,

$$f_{vir}^{-1}(v_{App_i}) = f_{alloc}(App_i) \Rightarrow f_{alloc}(App_i) = \emptyset \Rightarrow S_1 = \emptyset \quad (5)$$

This means that no physical sensor node serves application App_i. Thus, App_i is not currently served by the sensor-cloud. This completes the proof. ■

Proposition 3. *The worst case asymptotic computational complexity of $f_{alloc}(\cdot)$ for an application App_i, involving t type of sensors, $t \in T$, is O(n(t)), where n(t) is the total number of physical sensors of type t.*

Proof: From Equation 1, we obtain t of App_i, $t \in T$. After that $f_{alloc}()$ computes and selects sensor nodes s, such that, $s.t = t, s \in \hat{S}, |\hat{S}| = n(T)$. Thus, all sensor nodes of type t are picked up. Followed by this, functions $g_1()$ and $g_2()$ are executed. Hence the worst case asymptotic computational complexity of $f_{alloc}(\cdot)$ is $O(n(t))$. This completes the proof. ■

Using Propositions 1 and 2, we analyze an example run-time scenario, shown in Table II, consisting of 100 sensor nodes and 3 running applications. The services of the physical sensor nodes for an application App_i, at a particular time instant t, constitute a virtual sensor $v_{i,t}$. We find that, $v_{1,t_0} = \{s_1, s_3, s_7\}$. Thus, $f_{vir}(s_1) = v_1$. Due to the surjective property of f_{vir}, $\nexists v_i | \exists s_j \in S f_{vir}(s_j) = v_i$. Also, it is evident that, at a particular time instant t, $\forall v_i, v_j \in V$, v_i and v_j are disjoint. Thus, $\nexists s_k \in S : (s_k \in v_i) \wedge (s_k \in v_j)$.

Table II: Illustration of a runtime scenario within sensor-cloud

Applications	Serving time	Resources	Virtual sensor
App_1	t_0	s_1, s_3, s_7	v_1
App_2	t_0	s_8, s_4, s_2, s_5	v_2
App_3	t_1	s_1, s_9, s_{10}	v_3
App_2	t_2	s_1, s_7, s_6	v_2
App_1	t_2	s_2, s_3, s_8	v_1
App_3	t_2	s_4, s_9, s_{10}	v_3
App_1	t_3	s_3, s_9, s_{10}	v_1

V. Performance Evaluation

In this Section, we evaluate and compare the performance of sensor-cloud against a traditional WSN.

A. Performance Metrics

We define some performance metrics that have been taken into consideration for analysis.

1) Energy Consumption: The analysis for consumption of energy E is analyzed as per the equation,

$$E = E_{tr} + E_r + E_s + E_{proc} \quad (6)$$

where, E_{tr}, E_r, E_s, and E_{proc} are the energy expenses due to transmission, receiving, sensing, and computation, respectively. The unit of energy consumption for each of these components are assumed to be same for both WSN and sensor-cloud. However, the policies of communication vary and hence, the amount of energy expended varies.

2) Fault Tolerance: Fault-tolerance, $\mathcal{F}$ of a network is defined as the total number of non-faulty nodes present in the network at a particular time. Mathematically,

$$\mathcal{F}_t = F_{t-1} - P_f \times F_{t-1}, \quad \mathcal{F}_0 = N \quad (7)$$

where N and P_f are the total number of operative nodes initially present in the network and the probability of being faulty of a node, respectively. Also, a fixed fault tolerance rate for each sensor node in the network is assumed.

3) Lifetime of a Sensor Node: Lifetime of a sensor node $\mathcal{L}$ is computed as the number of sensing operations that can be performed by the node starting from the time of its deployment T till the time when its residual energy reaches below a threshold value E_{thresh}. Assuming every successful sensing operation requires τ amount of time, $\mathcal{L}$ is expressed as,

$$\mathcal{L} = T - \left(\frac{E_{act} - E_{thresh}}{E_s} \times \tau \right) \quad (8)$$

where E_{act} is the initial amount of available energy within a sensor node.

4) Evaluation of Cost-effectiveness: For evaluating cost-effectiveness, an analysis of flow of cash for every actor and a WSN user is studied. Lines of cumulative cost along the negative ordinate represents a cash outflow CO from the actor, whereas the one along the positive ordinate represents cash inflow CI to the actor. The costs due to deployment, maintenance, and rent are denoted by C_{deploy}, $C_{maintain}$, and C_{rent} respectively.

For a sensor-owner, the flow of cash is governed by the Equations 9 and 10, as follows:

$$CO_{sensor-owner} = n_1 \times (C_s + C_{deploy}) \quad (9)$$

$$CI_{sensor-owner} = n_1 \times C_{rent} \quad (10)$$

where n_1 is the number of sensors registered by the sensor-owner. C_s is the unit cost price of a sensor node. For a WSN user, we have,

$$CO_{wsn} = n_2 \times (C_s + C_{deploy} + C_{maintain}) + n_3 \times C_{deploy} \quad (11)$$

where n_2 and n_3 are the total number of sensor nodes in the WSN and the number of faulty nodes, respectively. The cash inflow for a WSN user is basically in terms of the service acquired from the sensed data.

From a sensor-cloud end-user point of view, the cash outflow is expressed as follows.

$$CO_{end-user} = n_4 \times C_{Se-aaS} \tag{12}$$

where n_4 is the total number of sensors nodes of which the user has obtained service in a particular month. C_{Se-aaS} is the cost incurred per unit usage of Se-aaS.

For a CSP, the monthly inflow and outflow of cash are also analyzed with the help of the following equations.

$$CO_{csp} = \eta_1 \times CI_{sensor-owner} + \frac{30n_5(C_{deploy} + C_{maintain})}{\Omega} \tag{13}$$

$$CI_{csp} = \eta_2 \times CO_{end-user} \tag{14}$$

where η_1, η_2, and Ω are respectively the total number of registered sensor-owners, total number of end-users, and the periodic time interval (in days) after which maintenance and deployment activities are performed by the CSP. n_5 is the number of faulty sensor nodes after Ω interval of time.

B. Simulation Setup

The simulation setup of this work is performed for a period of 5 years (60 simulation months). We have considered a cloud environment of 1000 sensors with 5 sensor-owners ($\eta_1 = 5$), 10 end-users ($\eta_2 = 10$) and a single CSP. The values for different costs are assigned as $C_s = 20, C_{deploy} = 10, C_{maintain} = 3, C_{rent} = 10, C_{Se-aaS} = 10$. The constants are also assigned specific values: $\tau = 1, \Omega = 5, \mathcal{L} = 200$.

C. Results

We study and analyze the performance of WSN and sensor-cloud based on the metrics discussed and defined above.

Energy Consumption

We now analyze the performance of a single sensor node in terms of its battery life. Fig. 5 shows the cumulative energy expenses of a sensor node in terms of sensing, computation and transmission of packets. In a WSN, intra-network communication occurs by repetitive multi-hop communication followed by transmission of packets to a data center. However, in a sensor-cloud environment, energy expenses due to transmission are mainly attributed to reach the cloud platform via multi-hop communication. Communication among sensor nodes is very rare (or does not occur), and, hence, large amount of energy is conserved. Moreover, unlike WSN, a particular sensor node does not necessarily serve a user-organization, even if it is application-compatible. Periodic scheduling is followed by the CSPs among multiple application-compatible sensor nodes with a view to distribute load and conserve resources. The figure presents that sensor-cloud achieves 36.68% decrease in energy consumption, compared to that of a WSN.

Fault Tolerance

We examine the performance of sensor-cloud from a network point of view. Fig. 6 illustrates a comparative study of fault-tolerance in WSNs and sensor-cloud. Fault-tolerance is a major cause of concern in WSN. Assuming a specific fault-tolerance rate, a WSN reaches a dead state unless a redeployment scheme is considered atleast once during its lifetime. On the other hand, sensor-cloud involves multiple service providers who can render the best possible sensor nodes at any point of time to address fault-tolerance of resources. Once a user-organization's application demand is recognized, the cloud infrastructure allocates a CSP, which can best serve the user-organization in terms of energy level, accuracy, QoS, compatibility of sensor node specification, and location specific feasibility. Fig. 6 indicates the increase in network performance with the increase in the number of CSPs.

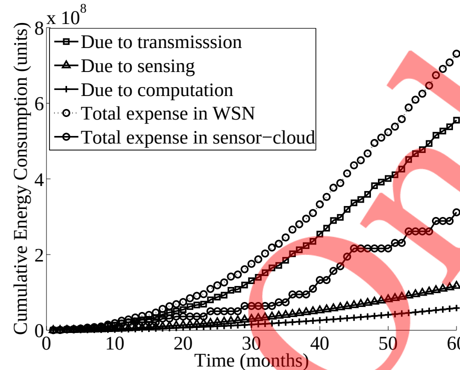


Figure 5: Comparative analysis for energy consumption

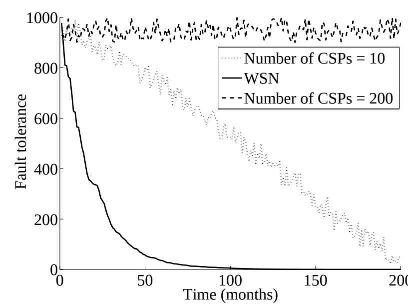


Figure 6: Comparative analysis of fault tolerance

Lifetime of a Sensor Node

As the energy consumption of a single sensor node is highly reduced in a sensor-cloud environment, it positively reflects the sensor node lifetime as well. Fig. 7 plots how the lifetime of a sensor node decreases over time for performing various operations within it. Results show that sensor-cloud increases the lifetime by 3.25%. From this, we can conclude that usage of sensor-cloud positively affects the network lifetime to a great extent, also.

Evaluation of Cost-effectiveness

This subsection puts forth a comparative study of various sensor-cloud actors and a WSN-user from a profit perspective.

Fig. 8 illustrates the perspective of a sensor-owner, who simply owns and deploys his/her sensor nodes within the

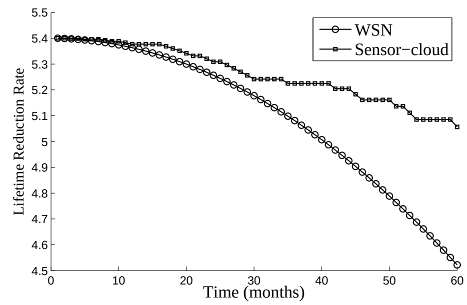


Figure 7: Comparative analysis for sensor node lifetime

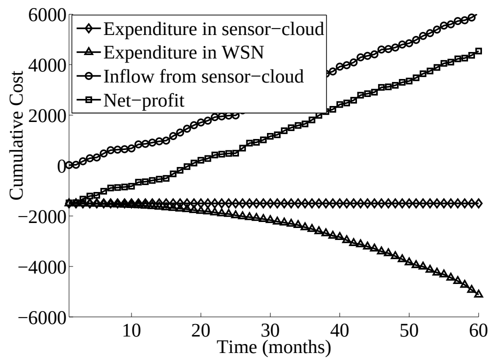


Figure 8: Analysis of cost-effectiveness for sensor-owner

sensor-cloud environment. In a WSN, the sensor-owner is eventually the WSN user. It is the responsibility of a WSN user to buy, deploy, maintain and redeploy sensor nodes, as and when needed. The cumulative cash outflow of a WSN user and a sensor-owner are indicated over time. The cash outflow of the sensor-owner occurs only once during the network lifetime, due to ownership and deployment of sensor nodes. The inflow of the sensor-owner is measured by the monthly rental fee that it obtains from the CSP. Finally, the overall profit of the sensor-owner is also denoted in the figure. Fig. 8 depicts that a single sensor-owner can reduce 33.83% of cash-outflow in sensor-cloud environment, compared to a WSN.

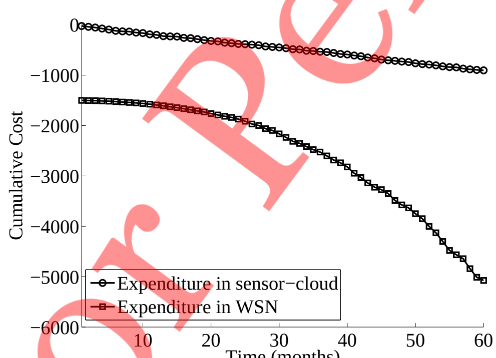


Figure 9: Analysis of cost-effectiveness for end-user

The perspective and profit analysis for an end-user is different. Fig. 9 illustrates a comparison with respect to the cost incurred by an end-user. End-user of a WSN is responsible for several jobs involving maintenance and overhead. However, in sensor-cloud, an end-user perceives a sensor as an instantaneous service (just like electricity, water), rather than as a hardware. Thus, s/he is liable to pay for only those units of Se-aaS that s/he has actually consumed. The profit of an end-user cannot be measured in terms of monetary units as it is relevant in terms of countable units of Se-aaS. The figure shows an average of 14.72% decrease in the expenditure of an end-user-organization.

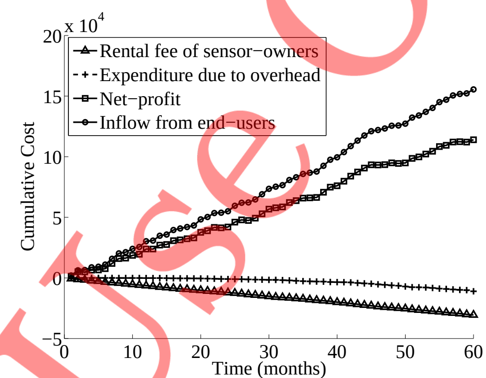


Figure 10: Analysis of cost-effectiveness for CSP

In Fig. 10, we depict the profit perspective of a CSP within sensor-cloud. As seen in Fig. 2, the CSP has to pay a monthly rental-fee to each sensor owner, from whose resources s/he renders services to the end-users. Fig. 10 illustrates the cumulative cash outflow for multiple sensor-owners. Some amount of cash outflow occurs due to the periodic maintenance and redeployment of the physical sensor nodes. The principal source of cash inflow is the end-users, who use the on-demand service and pay to the CSP accordingly. The net profit of the CSP is also indicated over time.

It is worthy to mention that a sensor-cloud can perform, only when the required resource type is actually available. Therefore, some sensor nodes have to be deployed by some sensor-owner. If a sensor-type is quite uncommon, it involves high overhead and maintenance cost compared to that of usage. Thus, if the number of end-user-organizations demanding for a particular resource type T_i is typically low, the performance of sensor-cloud reduces almost similar to that of WSN. Fig. 11 reflects a scenario where end-user-organizations demand a specific resource type. As the number of such users reduces, the profit of CSP reduces, eventually turns into loss. In such cases it is better to deploy a customized sensor network on behalf of the end-user-organizations.

VI. Application Specific Case studies

In this Section, we discuss an application specific study of workflow for both sensor-cloud and WSNs. We depict a general workflow within sensor-cloud in Fig. 12 following the prototype of sensor-cloud infrastructure [13]. From Fig. 12, it is evident that the end-user organization requests for *Se-aaS* to the sensor-cloud service provider. The requests are encoded in the form of XML templates which are decoded by the SensorML interpreter. Based on the requirements of the end-user,

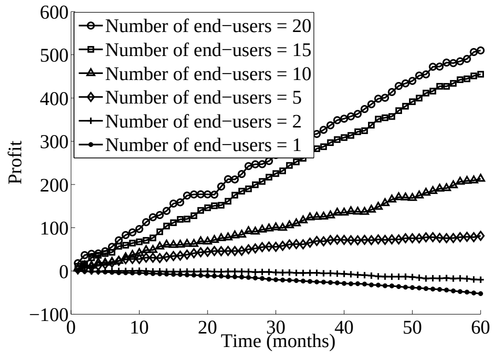


Figure 11: Profit Analysis of CSP in a Sensor-cloud

the Resource Manager allocates or deallocates physical sensor nodes. The allocation of the physical resources conforms to the definition and the application-specific compatibility of the sensor nodes. The Virtual Sensor Manager, and the Virtual Sensor Controller manages the entire processing logic behind virtualization, on-demand provisioning of resources and maintenance of abstraction.

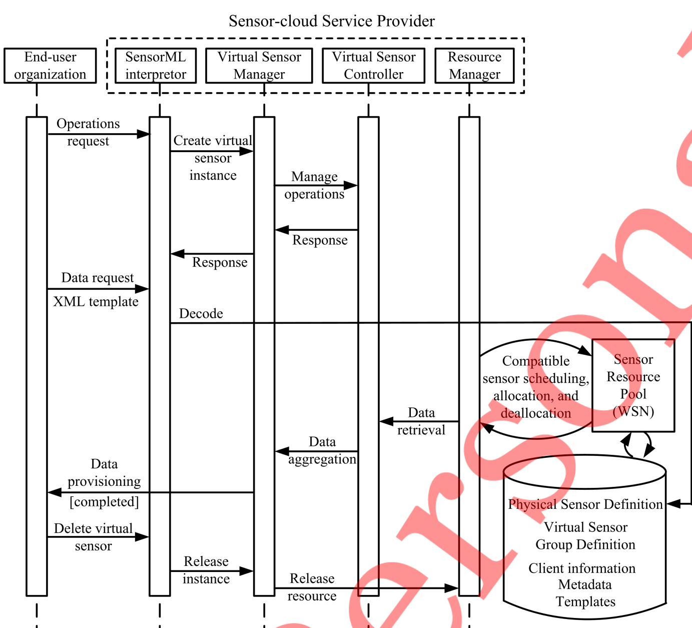


Figure 12: Workflow in Sensor-cloud

Target Tracking Application

We consider a WSN-based target tracking application in which a WSN-owner refuses to share the sensed information with an external body, even in exchange of money. Consequently, any organization that wishes to detect intrusion within a particular zone, has to deploy its own WSN. This leads to a long-term investment due to costly network set-up and maintenance overheads. However, in a sensor-cloud environment, the same organization can use the same tracking application and still get the service without actually owning the WSN. As indicated in Fig. 12, the CSP allocates the physical sensors in an on-demand manner (corresponding to the zone of interest, and the sensor type). On behalf of the organization, a virtual sensor is instantiated. The virtual sensor is kept alive till the organization terminates its data-request thread. The Virtual Sensor Manager ensures the real-time processing and management of physical sensors. Thus, the organization obtains *Se-aaS* effortlessly without bearing the overhead and responsibilities that are generally involved with a typical hardware.

Weather Services

We consider the application scenario of obtaining sensor-enabled weather services. If an end-user A (organization/person) is interested to obtain the weather services (such as rainfall, temperature, and humidity), s/he deploys his/her own WSN and extracts the sensed information, as and when required. A second end-user B may obtain environmental information from A in a rental manner, however, it will be extremely difficult to collect information at a global level, as A's network spans over a limited area. Additionally, WSNs can be extremely inconvenient for A, if A is concerned about environmental information for a very short span of time. In such cases the cost of deployment, redeployment, and maintenance of the WSN will be an overhead.

In a similar situation in a sensor-cloud environment, as A requests the CSP for weather information, a virtual sensor is instantiated for A. The CSP allocates the appropriate sensor nodes (rainfall sensor or temperature sensor) and the sensed information is collected within the cloud from where it is delivered to A. Thus, A extracts information very easily from the CSP by rental payment. A remains free from the network management overhead and other responsibilities. Also, the rental cost is incurred on a pay-per-use basis.

VII. Conclusion

The proposed work presents a theoretical model of virtualization for sensor-cloud environment. The process of mapping an application to its physical resources and the procedure for virtualization of the resources are also discussed. Finally, we show a comparative evaluation of performance between sensor-cloud and WSN. Results show that sensor-cloud accomplishes better performance compared to WSN in most of the cases. However, in some exceptional situations, sensor-cloud was found not to perform reasonably better than traditional WSNs. Thus, we justify that a paradigm shift for applications from the existing WSN-based technology to a sensor-cloud platform will be beneficial in terms of performance, usability and profit.

As sensor-cloud is an emerging concept, there is substantial scope for research. Future work will include details of design issues, and standardization of communication protocols. Schemes for optimization of sharing and coherence of resources can also be proposed. Additionally, each type of application can be analyzed for understanding the distinctness of its behavior within a sensor-cloud environment.

References

[1] G. Han, J. Jiang, L. Shu, J. Niu, and H.-C. Chao, "Management and applications of trust in wireless sensor networks: A survey," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 602 – 617, May 2014.

[2] W. Guo, N. Xiong, H. chieh Chao, S. Hussain, and G. Chen, "Design and analysis of self-adapted task scheduling strategies in wireless sensor networks," *Sensors*, vol. 11, no. 7, pp. 6533–6554, 2011.

[3] S. Misra and S. Singh, "Localized policy-based target tracking using wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 8, no. 3, July 2012.

[4] S. Chatterjee and S. Misra, "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2014.

[5] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, "Design for the next generation of wireless sensor networks in battlefield based on zigbee," in *Defense Science Research Conference and Expo (DSR)*, August 2011, pp. 1–4.

[6] H. Yang, Y. Qin, G. Feng, and H. Ci, "Online monitoring of geological CO_2 storage and leakage based on wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 556 – 562, February 2013.

[7] S. Jimenez-Fernandez, P. de Toledo, and F. del Pozo, "Usability and interoperability in wireless sensor networks for patient telemonitoring in chronic disease management," *IEEE Transactions on Bio-Medical Engineering*, vol. 60, no. 12, pp. 3331 – 3339, November 2013.

[8] C. Bachmann, M. Ashouei, V. Pop, and M. Vidojkovic, "Low-power wireless sensor nodes for ubiquitous long-term biomedical signal monitoring," *IEEE Comm. Magazine*, vol. 50, no. 1, pp. 20 – 27, Jan 2012.

[9] S. Misra and S. Chatterjee, "Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare: Data aggregation and channelization," *Information Sciences*, vol. 284, pp. 95 – 117, 2014.

[10] Y. Yuan, J. Zhao, and C. Qiu, "Estimating crowd density in an RF-based dynamic environment," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3837 – 3845, Oct 2013.

[11] X. Wang, M. Chen, T. Kwon, and H.-C. Chao, "Multiple mobile agents itinerary planning in wireless sensor networks: Survey and evaluation," *IET Communications*, vol. 5, no. 12, p. 7691776, 2011.

[12] M. M. Hassan, B. Song, and E.-N. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *International Conference on Ubiquitous Information Management and Communication*, 2009.

[13] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure physical sensor management with virtualized sensors on cloud computing," in *International Conference on Network-Based Information Systems*, 2010.

[14] 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," *International Journal of Distributed Sensor Networks*, vol. 2013, November 2012.

[15] A. Marchiori and Q. Han, "A two-stage bootloader to support multi-application deployment and switching in wireless sensor networks," in *Intl. Conf. on Computational Science and Engineering*, 2009, pp. 71–78.

[16] C.-F. Lai, H. Wang, H.-C. Chao, and G. Nan, "A network and device aware qos approach for cloud-based mobile streaming," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 747–757, June 2013.

[17] C.-F. Lai, Y.-X. Lai, H.-C. Chao, and J. Wan, "Cloud-assisted real-time transrating for http live streaming," *IEEE Wireless Communications Magazine*, vol. 20, no. 3, pp. 62–70, June 2013.

[18] S. Misra, S. Bera, A. Mondal, R. Tirkey, H.-C. Chao, and S. Chattopadhyay, "Optimal gateway selection in sensor-cloud framework for health monitoring," *IET Wireless Sensor Systems*, vol. 3, no. 4, December 2013.

[19] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 3, pp. 114–131, 2003.

[20] M. Eggert, R. Haubling, M. Henze, L. Hermerschmidt, R. Hummen, D. Kerpen, A. N. Perez, B. Rumpe, D. Thiben, and K. Wehrle, "Sensorcloud: Towards the interdisciplinary development of a trustworthy platform for globally interconnected sensors and actuators," RWTH Aachen University, Germany, Tech. Rep., 2013.

[21] L. D. Kumar, S. Grace, A. Krishnan, V. M. Manikandan, R. Chinraj, and M. R. Sumalatha, "Data filtering in wireless sensor networks using neural networks for storage in cloud," in *International Conference on Recent Trends In Information Technology (ICRTIT)*, 2012.

[22] S. Olariu, A. Wada, and L. Wilson, "Wireless sensor networks: leveraging the virtual infrastructure," *IEEE Netw*, vol. 18, pp. 51–56, Jul 2004.

[23] T. Ojha, M. Khatua, and S. Misra, "Tic-Tac-Toe-Arch: a self-organising virtual architecture for Underwater Sensor Networks," *IET Wireless Sensor Systems*, vol. 3, no. 4, pp. 307–316, April 2013.

[24] H. Dong, Q. Hao, T. Zhang, and B. Zhang, "Formal discussion on relationship between virtualization and cloud computing," in 11^{th} *International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2010.

[25] Open Geospatial Consortium. http://www.opengeospatial.org/.

[26] U. o. A. i. H. Mike Botts, Ed., *Sensor Model Language (SensorML) for In-situ and Remote Sensors*, Open Geospatial Consortium Inc., 2004.

[27] A. P. Jayasumana, Q. Han, and T. H. Illangasekare, "Virtual sensor networks - a resource efficient approach for concurrent applications," in 4^{th} *Intl. Conf. on Info. Tech.*, 2007, pp. 111 – 115.

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 Ph.D. 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. Dr. Misra 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. Dr. Misra is the author of over 180 scholarly research papers (including 90 journal papers). He has won eight research paper awards in different conferences. Dr. Misra has edited 6 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.

Subarna Chatterjee is presently working as a Junior Research Fellow and pursuing her Ph.D. from the School of Information Technology, Indian Institute of Technology, Kharagpur, India. She received her B.Tech. 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.

Prof. Mohammad S. Obaidat is an internationally well-known academic, researcher, and scientist. He received his Ph.D. and M. S. degrees in Computer Engineering with a minor in Computer Science from The Ohio State University, Columbus, Ohio, USA. Dr. Obaidat is currently a full Professor of Computer Science at Monmouth University, NJ, USA. Among his previous positions are Chair of the Department of Computer Science and Director of the Graduate Program at Monmouth University and a faculty member at the City University of New York. He has received extensive research funding. He has authored or co-authored five books and over three hundred (300) refereed scholarly journal and conference articles. Dr. Obaidat is editor of many scholarly journals, including being the Chief Editor of the International Journal of Communication Systems published by John Wiley. In 2002, he was the scientific advisor for the World Bank/UN Workshop on Fostering Digital Inclusion. Recently, Dr. Obaidat was awarded the distinguished Nokia Research Fellowship and the Distinguished Fulbright Scholarship. Prof. Obaidat is a Fellow of the Society for Modeling and Simulation International SCS, and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).