

# Optimal Data Center Scheduling for Quality of Service Management in Sensor-Cloud

Subarna Chatterjee, *Student Member, IEEE*, Sudip Misra<sup>ID</sup>, *Senior Member, IEEE*, and Samee U. Khan, *Senior Member, IEEE*

**Abstract**—The proposed work concentrates on the networking facets of sensor-cloud infrastructures—one of the first attempts of its kind. In a sensor-cloud, multiple sets of physical sensor nodes that are activated based on an application demand, in turn give rise to multiple distinct virtual sensors (VSs). The VSs are considered to span across multiple geographical regions; thereby, depositing the data (from each of the VS) to the closest cloud data center (DC). Quite obviously, multiple geospatial DCs get involve with an application. However, the principle of sensor-cloud is to store and conglomerate the data from various VSs, before they can be provisioned as Sensors-as-a-Service (Se-aaS). The assortment of data occurs within a single Virtual Machine (VM) (or in some cases multiple VMs) residing inside a particular DC. This work addresses the problem of scheduling a particular DC that congregates data from various VSs, and transmit the same to the end-user application. The work follows the general pairwise choice framework of the Optimal Decision Rule. The scheduling of the DC is performed under several network constraints, such as data migration cost, data delivery cost, and service delay of an application that ensures the preservation of the Quality-of-Service (QoS) and maintenance of the user satisfaction. The work quantifies the effective QoS of Se-aaS and determines an optimal decision rule for electing a particular DC. While arriving at a collective decision, the work incorporates the fallible decision making ability of a DC; thereby, excluding the loss of generality. Experimental results depict that the proposed algorithm for generating the optimal decision rule finds applicability in real-time cloud computing scenarios.

**Index Terms**—Sensor-cloud, cloud networks, virtualization, data centers

## 1 INTRODUCTION

RECENT research has acknowledged the sensor-cloud infrastructure as a potential substitute of traditional Wireless Sensor Networks (WSNs) [1], [2], [3]. The sensor-cloud is a new dimension of cloud computing [4], [5], [6], which is conceptualized as a sensor-management platform that functions as an interface between the physical and cyber world [7]. Such infrastructure virtualizes the physical resources (the sensor nodes) within the cloud platform and renders *Sensors-as-a-Service (Se-aaS)* to the end-users [8], [9]. Therefore, such a new technology allows the end-users to envision the sensor nodes as a service, rather than as a typical hardware.

Prior works on sensor-clouds have primarily addressed the ideology, dogma, and the challenges involved with this new aspect [2], [3], [10]. Very few of the works have focussed on some of the technical aspects of sensor-cloud [9], [11]. This work focuses on the networking aspects within sensor-cloud infrastructures—the first attempt of its kind. As mentioned previously, sensor-cloud thrives on the

virtualization of physical sensor nodes that are chosen as per the application demand and are grouped to form a virtual sensor (VS). The VSs serving a particular user application are stored, and processed within a single dedicated Virtual Machine (VM) within a physical server. Data from the VM is provisioned to the applications in the form on instantaneously obtainable service (Se-aaS).

The VSs may be formed across multiple overlapping regions. This work addresses the problem that arises due to routing and channelization of the data (of the VSs), originating from multiple regions, to geographically distributed sensor-cloud data centers (DCs). The goal of the work is to design and propose an algorithm for the dynamic scheduling of a cloud DC that would serve a particular user application with data from the respective VSs.

### 1.1 Motivation

As previously mentioned, research has not been initiated in the networking aspects of sensor-cloud infrastructure, which is a strong motivation of this work. In a sensor-cloud infrastructure, an end-user application requests for Se-aaS through a web interface in the form of high level requirement through Sensor Modeling Language (SensorML) equipped templates [7], [12]. The high level requirement is interpreted at the sensor-cloud end in terms of the allocation of the physical sensor nodes. For every request from a particular application, a set of physical sensors are allocated. Each of the set of allocated physical nodes, serving a particular application, form a VS. Intuitively, the set of VSs serving a particular application span across multiple regions. As the data from multiple VSs are channelized into different

- S. Chatterjee and S. Misra are with the School of Information Technology, Indian Institute of Technology, Kharagpur, India.  
E-mail: subarna@sit.iitkgp.ernet.in, smisra@sit.iitkgp.ernet.in..
- S.U. Khan is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58105.  
E-mail: samee.khan@ndsu.edu.

Manuscript received 9 Oct. 2014; revised 16 Aug. 2015; accepted 21 Sept. 2015. Date of publication 7 Oct. 2015; date of current version 6 Mar. 2019.

Recommended for acceptance by D. Kliazovich, B. Dorransoro, P. Bouwry, and A. Zomaya.

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/TCC.2015.2487973

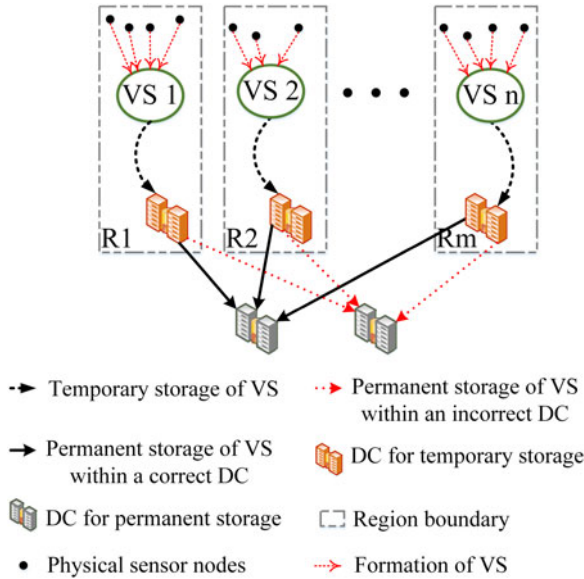


Fig. 1. Different storage types in DCs.

cloud DCs, multiple DCs that are spread geographically across the globe, may get involved in the process.

In sensor-cloud the VSs (serving a particular application) are stored inside a single VM, within a particular DC. Therefore, it becomes essential to migrate the data of different VSs (temporarily stored within geographically scattered DCs) to a single VM residing within a particular DC that would serve the application. Quite obviously, the urge for an application specific scheduling of a particular sensor-cloud DC is realized.

For example, as shown in Fig. 1, we observe that  $n$  number of VSs at different regions serve an application. Data from the physical sensor nodes constituting to a single VS are transmitted to a DC for temporary storage. In real-life scenarios, these DCs are the ones that are the have the maximum proximity with the physical sensor nodes. However, it is required to create a VM within a single DC to serve the application. Therefore, it is eventually required to transmit these VSs (from the DCs enabling temporary storage) to a single DC in which the VM, serving the application, resides. In such a scenario, a random selection of a DC, as also indicated by the figure, to serve an application might be unjust and inappropriate as it will incur huge networking overhead from the underlying sensor networks to the cloud platform thereby, reducing the Quality of Service (QoS). Herein, it is imperative to optimize the performance of the application by analyzing and selecting the DC that provisions with the maximum QoS.

## 1.2 Contributions

The main contribution of the work is the attempt to focus on the networking dimensions of sensor-cloud, that is totally unexplored to this date. The above-mentioned problem arises the need to select a single DC for serving a particular application. This work focuses on a dynamic scheduling of DCs, given a particular application, and a set of geographically scattered DCs. While scheduling a particular DC, the QoS of the application is also considered into account.

Initially, the proposed work quantifies the QoS to be offered to an application by sensor-cloud in terms of the migration cost within the DCs, the delivery cost to the

application from the scheduled DC, and the overall service delay of provisioning Se-aaS. The user satisfaction also accounts to the effective QoS. Finally, the process for the scheduling of the DC is performed, and the QoS is also simultaneously maintained.

This work addresses the problem by a collective decision making of various geographically distributed DCs. While arriving at a final solution, the work assumes the fallible decision making ability of the DCs; thereby, orienting the proposed problem to fit in well with the real-life scenarios. The proposed solution for scheduling a DC also takes into account the four different types of asymmetry arising due to two different states of nature (good or bad), and two different alternatives of the DCs while making a decision (yes or no), that is elaborately discussed in Section 5. This provides substantial credibility of the solution to be applicable de facto.

## 1.3 Organization of the Paper

Our work is organized as follows. Section 2 presents the work done so far on this aspect. Section 3 highlights the problem scenario. In Section 4, we present the formal definition of the problem. Section 5 discusses the system model. In Section 6, we provide the analytical results of the work. Section 7 discussed the experimental results on the performance of the system. Finally, Section 8 discusses and analyzes the complexity of the proposed solution. Section 9 concludes the work.

## 2 RELATED WORK

Prior to the advent of sensor-cloud infrastructures, some works suggested the integration of wireless sensor technology to cloud computing [3], [13], [14], [15]. Some of the works addressed the problem of dynamic gateway allocation while transmitting the sensed data from the networks to the cloud [16], [17], [18]. However, the concept for virtualization of physical sensor nodes was not realized.

The virtualization of sensor nodes was initially presented by Yuriyama and Kushida [7]. Subsequent research explorations were also performed in which some detailed aspects of virtualization and the sensor data management in cloud was proposed [1], [11]. Hassan et al. identified the principal benefits, and challenges involved in implementation of sensor-cloud infrastructure. However, all of the works mainly focussed on the dogma and ideology of sensor-clouds.

In addition to the theoretical works [19], [20], recently, few works have explored some of the technical aspect of sensor-clouds from an implementation point of view. Nguyen and Huh discussed the security aspects of sensor-cloud [21]. In [9], a scenario of multiple target tracking was explored to examine the implementation of sensor-cloud from an application perspective. However, to this date research has not been initiated in the networking aspects of sensor-clouds, and the challenges behind the system.

Quite a good number of works have explored, and addressed the technical issues of cloud networking [22], [23], [24], [25]. In [26], a Tabu Search Algorithm is proposed for optimal positioning of the cloud DCs. The work also focuses on efficient routing of information while taking the link capacities into account. The credibility of the work is also strengthened by the case study of a Web search engine. Mastroianni et al. addressed the problem of managing

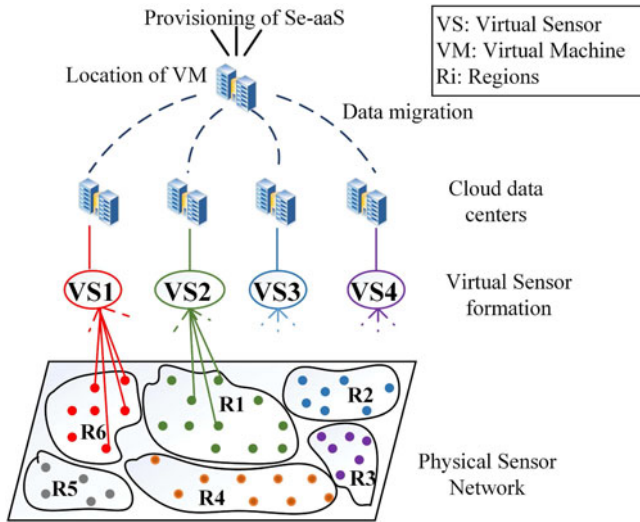


Fig. 2. Diagrammatic representation of the problem scenario.

power consumption of the DCs [27]. The work propounds *ecoCloud* for consolidation of VMs within the DCs. Another work [28] also addressed the problem of consolidations of VMs under QoS constraints. In [29], Bruneo examined the performance of the cloud DCs for Infrastructure-as-a-Service platforms of cloud computing. The work proposes an analytical model for evaluating several performance metrics, such as utilization, availability, waiting time, and responsiveness. Various other works have explored the issues of traffic management in cloud systems [30], [31]. For mobile cloud systems, research has found directions in cooperative resource management [32], energy-efficient offloading policies [33], and inter domain resource allocation [34]. However, for all of the above works, the data are transmitted from the DCs directly to the client. None of the work concerns intra data center scheduling that is essential primarily for sensor-cloud infrastructure as it involves collection of data from various geographical locations to a single DC. Additionally, the flow of request in the aforementioned cases is downward, from the client to the cloud; whereas, in our case, the flow of data is bottom-up, from the physical networks to cloud.

This work proposes scheduling of a single DC for serving a particular application. The scheduled DC collects information from several temporary DCs, which were used in intermediate storage and logging of data from the physical sensors directly to the cloud. The scheduler process is designed under constraints to ensure user satisfaction and maintenance of QoS, simultaneously.

### 3 PROBLEM DESCRIPTION

The problem scenario considers one or more end-user applications requesting for various types of sensor data from different regions in the form of Se-aaS. In case of Se-aaS, the end-users generally request for sensors through web templates. In return, the sensor-cloud service provider allocates physical sensor nodes and forms virtual sensors. Eventually, the sensed data is provisioned as a service to the end-users. It appears to the end-users that s/he is being served with dedicated physical sensor nodes as per requirement. The requests from the different applications are interpreted

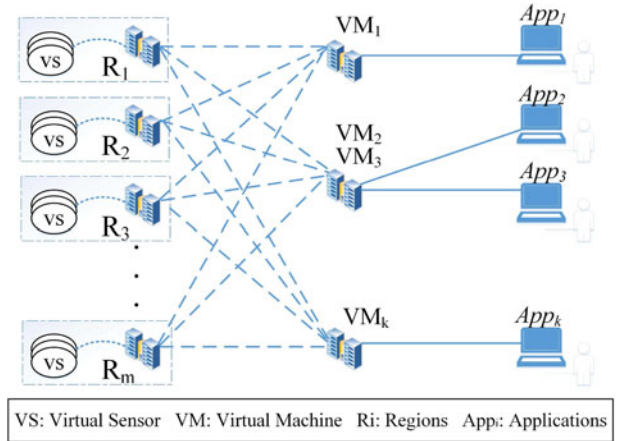


Fig. 3. Network model of the problem scenario.

and the physical sensors are allocated accordingly, as shown in Fig. 2. The figure clearly indicates the projection of the physical hosts within a DC and the projection of VMs within a physical host. As mentioned in Section 1, for every distinct request from the end-user, a distinct VS is formed. As the VSs are spread across multiple regions, the data from the VSs are temporarily stored into the closest sensor-cloud DC. A sensor-cloud DC is essentially a cloud DC that renders Se-aaS. However, the principle of a sensor-cloud is to serve a particular application with the data from multiple VSs residing within a single VM [9], [11]. Therefore, to provide Se-aaS, it is required to build a VM within a single DC.

The goal of this work is to perform a scheduling of the DC within which the VM will be allocated. By scheduling of a DC, we essentially mean the selection of a particular DC that will serve an end-user through its VM and VSs. The VSs, that are temporarily scattered within different DCs, are migrated to the particular VM, selected for serving a particular application. In the process of selection of the DC, the service delay of the end-user, as well as the migration cost of the DCs are optimized, thereby ensuring that the QoS is preserved.

### 4 FORMAL DEFINITION OF THE PROBLEM

We consider a set  $\mathcal{D}$  of  $\omega$  DCs within a sensor-cloud,  $\mathcal{D} = \mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_\omega$ , located at regions  $R_1, R_2, \dots, R_\omega$ , respectively, as shown in Fig. 3. Table 1 is a notation table that illustrates the significant functions and variables used in the proposed system. An application  $App_i$  creates  $m$  different types of requests for Se-aaS. The VSs formed are represented as,  $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_m$ , and are stored in temporary DCs,  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_m$ , ( $m \leq \omega$ ), respectively. Temporary DCs refer to those that temporarily store the data of the geographically scattered VSs. The data from these DCs are eventually transmitted to some other final DC where the VM of the particular user will be created. The objective of our work is to select a DC,  $\mathcal{D}_w^{App_i}$ , for  $App_i$  by maximizing QoS.

To ensure QoS, initially, we design a metric of QoS of a particular application. The QoS of Se-aaS is designed by considering several related factors, as described below.

**Definition 1.** The migration cost of sensor data from DC  $\mathcal{D}_i$  to DC  $\mathcal{D}_j$  for transmitting  $p$  packets is denoted by  $\mathcal{M}(\mathcal{D}_i, \mathcal{D}_j)$ , and is defined in terms of the latency  $\mathcal{L}$  involved in delivering a packet from  $\mathcal{D}_i$  to  $\mathcal{D}_j$ . The function  $\mathcal{M}(\cdot, \cdot)$  is expressed as:



TABLE 1  
Table of Notation

Parameters	Values
$\mathcal{D}$	Set of DCs
$R$	Set of regions
$\omega$	Number of DCs
$d_{i,j}$	Distance between $\mathcal{D}_i$ and $\mathcal{D}_j$
$\mathcal{V}_i$	VS stored in $\mathcal{D}_i$
$\mathcal{D}_w^{App_i}$	Winner DC for application $App_i$
$\mathcal{M}(\mathcal{D}_i, \mathcal{D}_j)$	Migration cost from $\mathcal{D}_i$ to $\mathcal{D}_j$
$\mathcal{L}(\cdot, \cdot)$	Latency involved in delivering a packet from $\mathcal{D}_i$ to $\mathcal{D}_j$
$(l_{1,j}, l_{2,j})$	Absolute location of $App_j$
$(\mathcal{D}_i.x, \mathcal{D}_i.y)$	Location coordinates of $\mathcal{D}_i$
$\eta_2$	Transmission rate from a DC to an end-user
$\delta(\cdot, \cdot)$	Delivery cost
$\mathcal{S}(\cdot)$	Service delay
$\mathcal{Q}(App_i)$	QoS offered to $App_i$
$\mathcal{Q}_{net}(App_i)$	Effective QoS offered to $App_i$
$\mathcal{U}(\cdot)$	User dissatisfaction-delay product
$\lambda$	Demand rate
$\mathcal{P}(\cdot, \cdot)$	Payoff for approving or disapproving DCs
$\mathcal{D}^{nom}$	Set of nominated DCs
$X_{ij}$	Decision outcome of $\mathcal{D}_i$ for $\mathcal{D}_j$
$W$	System events
$\Upsilon_i$	Decision making ability of $\mathcal{D}_i$
$\mathcal{J}_{\mathcal{D}_i}$	Decision profile of $\mathcal{D}_i$

$$\mathcal{M}(\mathcal{D}_i, \mathcal{D}_j) = \mathcal{L}, \quad (1)$$

$$\text{where, } \mathcal{L}(\mathcal{D}_i, \mathcal{D}_j, p) = p \times P \times d(i, j) / \eta_1. \quad (2)$$

The function  $\mathcal{L}(\mathcal{D}_i, \mathcal{D}_j, p)$  is the latency involved in migrating  $p$  packets, (each of size  $P$  byte), from  $\mathcal{D}_i$  to  $\mathcal{D}_j$ . The variable  $\eta_1$  is the migration latency of unit byte meter per second.

**Definition 2.** The delivery cost of  $p$  packets from  $\mathcal{D}_i$  to an application  $App_j$  at absolute location  $(l_{1,j}, l_{2,j})$  is denoted as  $\delta(\mathcal{D}_i, l_{1,j}, l_{2,j})$ , and is expressed as:

$$\delta(\mathcal{D}_i, l_{1,i}, l_{2,i}) = \left( \sqrt{(\mathcal{D}_i.x - l_{1,i})^2 + (\mathcal{D}_i.y - l_{2,i})^2} \right) / \eta_2, \quad (3)$$

where  $\eta_2$  is in meter per second through the link connecting a DC to an end-user. Therefore,  $\delta$  is finally expressed in second.

**Definition 3.** The service delay of  $App_i$ ,  $\mathcal{S}(App_i)$ , is the summation of its migration cost and the delivery cost. It is defined as:

$$\mathcal{S}(App_i) = \sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_s) + \delta(\mathcal{D}_s, l_{1,i}, l_{2,i}), \quad (4)$$

where the data of the VSs are migrated to  $\mathcal{D}_s$  from multiple DCs,  $\{\mathcal{D}_i\}, 1 \leq i \leq m$ .

Now, we define the QoS offered (in byte per second) for transmission of  $p$  packets to an application proportional to the costs due to migration, and data delivery. Therefore, the QoS offered to  $App_i$  by a sensor-cloud is given as:

$$\mathcal{Q}(App_i) = \frac{pP}{\left[ \sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_s) + \delta(\mathcal{D}_s, l_{1,i}, l_{2,i}) \right]}. \quad (5)$$

However, the above metric of QoS does not consider user satisfaction into account. Therefore,  $\mathcal{Q}$  is modified as  $\mathcal{Q}_{net}$ , as described below.

**Definition 4.** The user dissatisfaction-delay product,  $\mathcal{U}(\mathcal{V}_k)$ , of obtaining data from each  $\mathcal{V}_k$  is defined as the product of the mean service delay in provisioning data from  $\mathcal{V}_k$  and the corresponding demand rate  $\lambda_k$ . It is expressed as:

$$\mathcal{U}(\mathcal{V}_k) = \lambda_k \times \frac{\mathcal{S}(App_i)}{\sum_{i=1}^m \lambda_k}, \quad (6)$$

where  $\frac{\mathcal{S}(App_i)}{\sum_{i=1}^m \lambda_k}$  accounts for the mean service delay of  $App_i$  for  $\mathcal{V}_k$ .

Consequently,  $p$  packets are transmitted in an overall time of  $\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_s) + \delta(\mathcal{D}_s, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \mathcal{U}(\mathcal{V}_k)$ .

**Definition 5.** The effective QoS of application  $App_i$  served by  $\mathcal{V}_k$  is mathematically expressed as:

$$\begin{aligned} \mathcal{Q}_{net}(App_i) &= \frac{pP}{\mathcal{S}(App_i) + \mathcal{U}(\mathcal{V}_k)} \\ &= \frac{pP}{\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_s) + \delta(\mathcal{D}_s, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \left( \lambda_k \times \frac{\mathcal{S}(App_i)}{\sum_{i=1}^m \lambda_k} \right)}. \end{aligned} \quad (7)$$

Therefore, the final objective function of our work is stated as:

$$f : \mathcal{D}^m \rightarrow \mathcal{D}, \mathcal{D} = \{\mathcal{D}_i\}, 1 \leq i \leq m \leq d, \quad (8)$$

which maps a set of DCs to a single  $\mathcal{D}_w$ . The function  $f$  maximizes the effective QoS,  $\mathcal{Q}_{net}$ , of an application. Therefore, for a particular application  $App_i$ ,  $f$  belongs to the solution set of the maximization function:

$$\arg \max \left( \mathcal{Q}_{net}(App_i) \right), \quad (9)$$

where  $\mathcal{Q}_{net}(App_i)$  is obtained from Definition 5. Having obtained the formal objective of our work in Equation (8) and Equation (42), we propose our approach towards achieving the solution in Section 5.

It is to be noted here that, the problem definition of the proposed work considers all applications of sensor-cloud, i.e., it targets all sensor-based applications (e.g., environmental monitoring, surveillance applications, multimedia applications, and so on). As different applications possess different demands (especially in terms of sensor hardware and configurations), sensor-cloud manages the interoperability and compatibility issues which we have already discussed in our previous works [8], [35], [36], [37].

## 5 SYSTEM MODEL

Motivated by the *Optimal Decision Rule*, as illustrated in [38], we consider the “general pairwise choice framework” implemented over the cloud network.

### 5.1 Optimal Decision Rule

The optimal group decision-making considers the fallibility in human nature within a fixed-sized committee. Such rules

are applicable for selection of investment projects in economic organizations, for design of reliable systems, and for decision making in other political or legal applications. The general pairwise choice framework of Optimal Decision Rule takes into account the four possible types of asymmetry. For example, in an  $n$  member committee of an organization, a decision making is required for accepting/rejecting a good/bad project. Hence, the decision of the committee has four potential alternatives. The Optimal Decision Rule focuses to combine the individual opinions (assuming the fallibility of humans) and generate the rule that maximizes the payoff of the organization in terms of the profit obtained. In this work, the Optimal Decision Rule is analogously used. The rationale of the analogy is illustrated below.

## 5.2 Proposed Model

Each DC of the set  $\mathcal{D}_{App_i} \subseteq \mathcal{D}$  ( $\mathcal{D}_{App_i} = \{\mathcal{D}_i\}, 1 \leq i \leq m$ ), involved in temporary storage and processing of  $m$  VSs participate in a decision rule to elect a particular DC from a nominated set of DCs,  $\mathcal{D}^{nom}$ , to serve  $App_i$ . As per the *Optimal Decision Rule*, we assume two possible states of nature for every nominated DC—good (1) and bad (−1). The payoff associated with the approval of a good DC and the disapproval of a bad DC is denoted by  $\mathcal{P}(1 : 1)$  and  $\mathcal{P}(-1 : -1)$ , respectively. As the model assumes four types of asymmetry, while making a pairwise choice, we also consider the payoffs associated with the approval of a bad DC and the disapproval of a good DC, expressed as,  $\mathcal{P}(1 : -1)$  and  $\mathcal{P}(-1 : 1)$ , respectively.

## 5.3 Assumptions of the Model

- Every DC,  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ , is heterogeneous in terms of its decision making ability.
- Each DC,  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ , possesses imperfect decision making ability.
- In the context of pairwise choice, four types of asymmetry are possible.
- The decision of  $\mathcal{D}_{App_i}$  and the individual DCs have a binary interpretation.
- The likelihood of a nominated DC being good is a time variant, *a priori* probability ( $\alpha_i$ ),  $\alpha_o = 1/2$ .
- The load capacity of a DC is known to the other DCs of set  $\mathcal{D}$ .

To form  $\mathcal{D}^{nom}$ , we initially compute the mean distance of each data centers from the location of the user application  $App_i$  ( $L = \langle l_{i,1}, l_{i,2} \rangle$ ), denoted by  $\xi_{avg}^{App_i}$ . Therefore, we have,

$$\xi_{avg}^{App_i} = \frac{\sum_{j=1}^{\omega} d(\mathcal{D}_j, L)}{\omega}. \quad (10)$$

Consequently,  $\forall \mathcal{D}_j \in \mathcal{D}, \mathcal{D}_j \in \mathcal{D}^{nom}$ , if and only if:

$$d(\mathcal{D}_j, L) \leq \xi_{avg}^{App_i}. \quad (11)$$

Now,  $X_{ij} = \{1, -1\}$  at a particular time,  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ ,  $\mathcal{D}_j \in \mathcal{D}^{nom}$ .  $X_{ij}$  is modeled as:

$$X_{ij}(t) = \begin{cases} 1, & \text{if } (d(i,j) \leq d_{avg}^i) \wedge (\mathcal{C}(j) \leq \mathcal{C}_{avg}^{nom}) \\ -1, & \text{otherwise} \end{cases}, \quad (12)$$

where  $d_{avg}^i$  is the mean distance of  $\mathcal{D}_i$  from the other DCs of the system and is expressed as:

$$d_{avg}^i = \frac{\sum_{k=1}^{\omega} d(i, k)}{\omega - 1}, \forall \mathcal{D}_k \in \forall \mathcal{D}, i \neq k, \quad (13)$$

and  $\mathcal{C}(\mathcal{D}_j)$  computes the current load of DC  $\mathcal{D}_j$  given as:

$$\mathcal{C}(\mathcal{D}_j) = |\mathcal{A}_j|, \quad (14)$$

where  $\mathcal{A}_j$  is the current set of applications served by  $\mathcal{D}_j$ . Therefore,  $\forall \mathcal{D}_j \in \mathcal{D}^{nom}$ , we have:

$$\mathcal{C}_{avg}^{nom} = \frac{\sum_{j=1}^{|\mathcal{D}^{nom}|} \mathcal{C}(\mathcal{D}_j)}{|\mathcal{D}^{nom}|}. \quad (15)$$

**Definition 6.** The events of our model are:

- $W_1^i = \{1, -1\}$ : Approving or disapproving a DC by  $\mathcal{D}_i$ ;
- $W_2^j = \{1, -1\}$ : A DC  $\mathcal{D}_j$  appearing as a “good” or “bad”.

**Definition 7.** The correct decision making ability of a DC  $\mathcal{D}_i$ , is denoted by  $\Upsilon_i$ , and is expressed as:

$$\Upsilon_i^+ = P(W_1^i = 1 | W_1^j = 1) = P(1 : 1), \quad (16)$$

$$\Upsilon_i^- = P(W_1^i = -1 | W_1^j = -1) = P(-1 : -1). \quad (17)$$

**Definition 8.** The incorrect decision making ability of a DC  $\mathcal{D}_i$ , is denoted by  $\Upsilon'_i$ , and is expressed as:

$$\Upsilon_i'^+ = P(W_1^i = 1 | W_1^j = -1) = P(1 : -1), \quad (18)$$

$$\Upsilon_i'^- = P(W_1^i = -1 | W_1^j = 1) = P(-1 : 1). \quad (19)$$

However, a bias exists, that is expressed as:

$$\Upsilon_i^+ > \Upsilon_i'^+, \Upsilon_i^- > \Upsilon_i'^-. \quad (20)$$

The probability of approving a good DC is greater than that of a bad DC and the probability of disapproving a bad DC is greater than that of a good DC. Having defined the decision making abilities of every DC, we formally define a “good” and a “bad” DC. The “goodness” or “badness” is studied only for the elements of  $\mathcal{D}^{nom}$ . The criteria for “goodness” of a DC,  $\mathcal{D}_j \in \mathcal{D}^{nom}$ , for a particular application  $App_k$ , holds true if the total number of positive decisions for  $\mathcal{D}_j$  exceeds a pre-negotiated threshold ( $X_{th}$ ). That is to say that, if

$$\sum_{i=1}^m (X_{ij} + 1)! - 1, \text{ such that } \forall \mathcal{D}_i \in \mathcal{D}_{App_k}, \quad (21)$$

where  $(X_{ij} + 1)! - 1$  returns 1 or 0 for  $X_{ij} = 1$  or  $-1$ , respectively.

**Definition 9.** The metric of “goodness” of  $\mathcal{D}_j$ ,  $\mathcal{G}(\mathcal{D}_j)$ , is defined as the ratio of the difference of the current and the maximum load to the maximum load of  $\mathcal{D}_j$  that it supports. Therefore, we have:

$$\mathcal{G}(\mathcal{D}_j) = \frac{\mathcal{C}^{max}(\mathcal{D}_j) - \mathcal{C}(\mathcal{D}_j)}{\mathcal{C}^{max}(\mathcal{D}_j)}, 0 \leq \mathcal{G}(\mathcal{D}_j) \leq 1. \quad (22)$$

The metric of “badness” of  $\mathcal{D}_j$ ,  $\bar{\mathcal{G}}(\mathcal{D}_j)$ , is obtained as the complement of the “goodness” of  $\mathcal{D}_j$ . Therefore, we have:

$$\bar{\mathcal{G}}(\mathcal{D}_j) = 1 - \mathcal{G}(\mathcal{D}_j) = \frac{\mathcal{C}(\mathcal{D}_j)}{\mathcal{C}^{max}(\mathcal{D}_j)}, 0 \leq \bar{\mathcal{G}}(\mathcal{D}_j) \leq 1. \quad (23)$$

Consequently, every DC possesses a measure of “goodness”, and “badness”. Intuitively, a DC that is exactly half loaded has identical metrics for “goodness” and “badness”. The proportion of good DCs is:

$$\alpha = \frac{|\mathcal{D}_s^{nom}|}{|\mathcal{D}^{nom}|} \text{ such that, } \mathcal{D}_s^{nom} = \{\mathcal{D}_u\}, \mathcal{G}(\mathcal{D}_u) > \bar{\mathcal{G}}(\mathcal{D}_u). \quad (24)$$

Now, the estimated probability of approving any DC irrespective of its being “good” or “bad”, by  $\mathcal{D}_i$  is its decision making ability that is influenced, and affected by the present workload of the DC [39], [40]. Here, we estimate  $P(W_i^1 = 1)$  on the basis of learning its ability for the last  $h$  instants. Assuming  $\mathcal{D}_i$  voted for  $h$  distinct  $\mathcal{D}_j$ s for the last  $h$  time instants, we have:

$$\hat{P}(W_i^1 = 1)(t) = \begin{cases} \frac{1}{h} \sum_{j=1}^h (X_{ij} + 1)! - 1, & \text{if } t \leq h \\ \frac{1}{t} \sum_{j=1}^t (X_{ij} + 1)! - 1, & \text{if } t > h \end{cases}, \quad (25)$$

where  $P(W_i^1 = 1)$  at time  $t$  is estimated as the mean decision making ability of  $\mathcal{D}_i$  over the last  $h$  instants if  $t \leq h$ . For  $t > h$ , the mean is computed till the current time instant. Moreover, we have:

$$\hat{P}(W_i^1 = -1)(t) = \begin{cases} \frac{1}{h} \sum_{j=1}^h (1 - X_{ij})! - 1, & \text{if } t > h \\ \frac{1}{t} \sum_{j=1}^t (1 - X_{ij})! - 1, & \text{if } t \leq h \end{cases}, \quad (26)$$

where  $(1 - X_{ij})! - 1$  returns 0 or 1 for  $X_{ij} = 1$  or  $-1$ , respectively. Having obtained  $\alpha$ ,  $\hat{P}(W_i^1 = 1)$ ,  $\hat{P}(W_i^1 = -1)$ , we obtain  $\Upsilon_i^+$  and  $\Upsilon_i^-$  using Bayesian classification [41], [42].

$$\begin{aligned} \Upsilon_i^+ &= P\left(\frac{W_1^i = 1}{W_2 = 1}\right) \\ &= \frac{P(W_1^i = 1)P\left(\frac{W_2=1}{W_1^i=1}\right)}{P(W_1^i = 1)P\left(\frac{W_2=1}{W_1^i=1}\right) + P(W_1^i = -1)P\left(\frac{W_2=1}{W_1^i=-1}\right)} \end{aligned} \quad (27)$$

$$\begin{aligned} \Upsilon_i^- &= P\left(\frac{W_1^i = -1}{W_2 = -1}\right) \\ &= \frac{P(W_1^i = -1)P\left(\frac{W_2=-1}{W_1^i=-1}\right)}{P(W_1^i = 1)P\left(\frac{W_2=-1}{W_1^i=1}\right) + P(W_1^i = -1)P\left(\frac{W_2=-1}{W_1^i=-1}\right)}. \end{aligned} \quad (28)$$

The expressions of  $\Upsilon_i^{'+}$ , and  $\Upsilon_i'^-$  can be evaluated simply using Equation (27) and Equation (28), respectively. Therefore, we have:

$$\Upsilon_i^{'+} = 1 - \Upsilon_i^-, \quad (29)$$

$$\Upsilon_i'^- = 1 - \Upsilon_i^+. \quad (30)$$

**Definition 10.** The decision profile of  $\mathcal{D}_{App_i}$  for a particular  $\mathcal{D}_j$ , is defined as:

$$\mathcal{J}_{\mathcal{D}_j} = \{X_{ij}\}, \forall \mathcal{D}_i \in \mathcal{D}_{App_i}, \mathcal{J}_{\mathcal{D}_j} \in \mathcal{J}, \mathcal{J} = \{1, -1\}^m, \quad (31)$$

where  $\mathcal{J}$  is the set of all of the possible decision profiles.

The outcome of the aggregation rule is  $\mathcal{O} : \mathcal{J}_{\mathcal{D}_j} \rightarrow \{1, -1\}$ . Therefore, the set of decision profiles can be partitioned into  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+}$  and  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-}$ , where  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} = \{\mathcal{J}_{\mathcal{D}_k} \mid \mathcal{O}(\mathcal{J}_{\mathcal{D}_k}) = 1\}$ , and  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-} = \{\mathcal{J}_{\mathcal{D}_k} \mid \mathcal{O}(\mathcal{J}_{\mathcal{D}_k}) = -1\}$ . Moreover, we also have  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} \cup \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-} = \mathcal{J}$  and  $\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} \cap \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-} = \emptyset$ .

A particular decision profile  $\mathcal{J}_{\mathcal{D}_j}$  for  $\mathcal{D}_j$  can be partitioned into  $A(\mathcal{J}_{\mathcal{D}_j})$  and  $R(\mathcal{J}_{\mathcal{D}_j})$ , where  $X_{ij} = 1, \forall \mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})$  and  $X_{ij} = -1, \forall \mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})$ . If  $h(1 : 1)$ , and  $h(-1 : -1)$  be the respective probabilities of approving or disapproving  $\mathcal{D}_j$ , then under decision rule  $\mathcal{O}$ , we have:

$$h(1 : 1) = \prod_{\mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})} \Upsilon_i^+ \prod_{\mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})} (1 - \Upsilon_i^+), \quad (32)$$

$$h(-1 : -1) = \prod_{\mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})} \Upsilon_i^- \prod_{\mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})} (1 - \Upsilon_i^-). \quad (33)$$

Now, given a  $\mathcal{D}_j$  is “good” or “bad”,  $\mathcal{D}_{App_i}$  approves or disapproves it for a particular decision profile  $\mathcal{J}_{\mathcal{D}_j}$ , under decision rule  $\mathcal{O}$  with probability  $P_{\mathcal{D}}(\mathcal{O} : 1)$  and  $P_{\mathcal{D}}(\mathcal{O} : -1)$ , respectively. Therefore we have:

$$P_{\mathcal{D}}(\mathcal{O} : 1) = P(\mathcal{J}_{\mathcal{D}_j} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} : 1), \quad (34)$$

$$P_{\mathcal{D}}(\mathcal{O} : -1) = P(\mathcal{J}_{\mathcal{D}_j} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-} : -1). \quad (35)$$

However, for Type I and Type II errors in decision making, by the DCs themselves, we also have:

$$P_{\mathcal{D}}^e(\mathcal{O} : 1) = 1 - P_{\mathcal{D}}(\mathcal{O} : 1), \quad (36)$$

$$P_{\mathcal{D}}^e(\mathcal{O} : -1) = 1 - P_{\mathcal{D}}(\mathcal{O} : -1). \quad (37)$$

From Equation (32) and Equation (33), we obtain:

$$P_{\mathcal{D}}(\mathcal{O} : 1) = \sum_{\mathcal{J}_{\mathcal{D}_k} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+}} h(1 : 1), \quad (38)$$

$$P_{\mathcal{D}}(\mathcal{O} : -1) = \sum_{\mathcal{J}_{\mathcal{D}_k} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-}} h(-1 : -1). \quad (39)$$

The goal of our problem is to maximize the expected payoff in terms of the QoS,  $\mathcal{Q}_{net}(App_i)$ , for the set of all possible aggregation rules  $F$ . We model the payoff associated with the approval or disapproval of a  $\mathcal{D}_j$  proportional to the QoS of the provisioned service to  $App_i$ . Therefore, we have:

$$\mathcal{P}(\zeta : \zeta) \propto \mathcal{Q}_{net}(App_i),$$

$$\Rightarrow \mathcal{P}(\zeta : \zeta) = \text{sign}(\mathcal{D}_j) \left\{ \left[ \sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_j) + \delta(\mathcal{D}_j, l_{1,i}, l_{2,i}) \right] - \sum_{i=1}^m \left( \lambda_k \times \frac{\mathcal{S}(\text{App}_i)}{\sum_{i=1}^m \lambda_k} \right) \right\}, \quad (40)$$

where  $\zeta = \{1, -1\}$  and  $\text{sign}(\mathcal{D}_j)$  is defined as:

$$\text{sign}(\mathcal{D}_j) = \begin{cases} 1, & \text{if } \mathcal{G}(\mathcal{D}_j) > \bar{\mathcal{G}}(\mathcal{D}_j) \\ -1, & \text{if } \mathcal{G}(\mathcal{D}_j) \leq \bar{\mathcal{G}}(\mathcal{D}_j) \end{cases}. \quad (41)$$

Consequently, based on the ‘‘goodness’’ or ‘‘badness’’ of a DC, a positive or negative payoff is computed. Now, following the *Optimal Decision Rule*, our goal is to maximize our expected payoff. Therefore, the goal is modified as:

$$\arg \max_{f \in F} \mathcal{E}(\text{App}_i), \quad (42)$$

where  $\mathcal{E}(\text{App}_i)$  is expressed as:

$$\begin{aligned} \mathcal{E}(\text{App}_i) &= \alpha [\mathcal{P}(1:1)P_{\mathcal{D}}(\mathcal{O}:1) + \mathcal{P}(-1:1)(1 - P_{\mathcal{D}}(\mathcal{O}:1))] \\ &\quad + (1 - \alpha) [\mathcal{P}(-1:-1)P_{\mathcal{D}}(\mathcal{O}:-1) \\ &\quad + \mathcal{P}(1:-1)(1 - P_{\mathcal{D}}(\mathcal{O}:-1))] \end{aligned} \quad (43)$$

Now, the net effective payoff of for approval of a good DC is  $\mathcal{P}(1) = \mathcal{P}(1:1) - \mathcal{P}(-1:1)$ . Similarly, the net payoff for disapproving a bad DC is  $\mathcal{P}(-1) = \mathcal{P}(-1:-1) - \mathcal{P}(1:-1)$ . Therefore, simplifying Equation (43), we obtain:

$$\mathcal{E}(\text{App}_i) = \alpha P_{\mathcal{D}}(\mathcal{O}:1)\mathcal{P}(1) + (1 - \alpha)P_{\mathcal{D}}(\mathcal{O}:-1)\mathcal{P}(-1) + \alpha\mathcal{P}(-1:1) + (1 - \alpha)\mathcal{P}(1:-1). \quad (44)$$

Based on the above, the net goal function can be rewritten as:

$$\begin{aligned} &\arg \max_{f \in F} \mathcal{E}(\text{App}_i) \\ &= \alpha P_{\mathcal{D}}(\mathcal{O}:1)\mathcal{P}(1) + (1 - \alpha)P_{\mathcal{D}}(\mathcal{O}:-1)\mathcal{P}(-1) \\ &= \alpha\mathcal{P}(1) \sum_{\mathcal{J}_{\mathcal{D}_k} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+}} h(1:1) + (1 - \alpha)\mathcal{P}(-1) \sum_{\mathcal{J}_{\mathcal{D}_k} \in \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-}} h(-1:-1). \end{aligned} \quad (45)$$

**Theorem 1.** The optimal decision rule  $\hat{f}$  of our problem is denoted as:

$$\hat{f} = \sigma(\varphi + \beta + \rho + \Psi),$$

where

$$\varphi = \ln \frac{\alpha}{1 - \alpha}, \beta = \ln \frac{\mathcal{P}(1)}{\mathcal{P}(-1)},$$

$$\rho = \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} (1 + X_{ij})! - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^+} (1 - X_{ij})! \right],$$

$$\Psi_i = \ln \frac{\Upsilon_i^+ (1 - \Upsilon_i^+)}{\Upsilon_i^- (1 - \Upsilon_i^-)}, \Psi = \sum_{i=1}^m \Psi_i,$$

$$\sigma(\chi) = \begin{cases} +1, & \chi \geq 0 \\ -1, & \text{otherwise} \end{cases}.$$

**Proof.** For any decision profile  $\mathcal{J}_{\mathcal{D}_j}$ ,  $\hat{f}(\mathcal{J}_{\mathcal{D}_j}) = 1$  if and only if,

$$\alpha\mathcal{P}(1)h(1:1) > (1 - \alpha)\mathcal{P}(-1)h(-1:-1). \quad (46)$$

Now, according to the *Optimal Decision Rule* [38] the sufficient condition for the optimality of  $\hat{f}$  is satisfied by the partition of  $\mathcal{J}$ , abiding by the condition:

$$\begin{aligned} \mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} &= \left\{ \mathcal{J}_{\mathcal{D}_j} : \hat{f}(\mathcal{J}_{\mathcal{D}_j}) = 1 \right\} \\ &= \left\{ \mathcal{J}_{\mathcal{D}_j}, \alpha\mathcal{P}(1)h(1:1) > (1 - \alpha)\mathcal{P}(-1)h(-1:-1) \right\} \\ &= \left\{ \mathcal{J}_{\mathcal{D}_j}, \frac{\alpha}{1 - \alpha} \frac{\mathcal{P}(1)}{\mathcal{P}(-1)} \prod_{\mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})} \Upsilon_i^+ \prod_{\mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})} (1 - \Upsilon_i^+) \right. \\ &\quad \left. > \prod_{\mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})} \Upsilon_i^- \prod_{\mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})} (1 - \Upsilon_i^-) \right\} \\ &= \left\{ \mathcal{J}_{\mathcal{D}_j}, \ln \frac{\alpha}{1 - \alpha} + \ln \frac{\mathcal{P}(1)}{\mathcal{P}(-1)} + \sum_{\mathcal{D}_i \in A(\mathcal{J}_{\mathcal{D}_j})} \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} \right. \\ &\quad \left. > \sum_{\mathcal{D}_i \in R(\mathcal{J}_{\mathcal{D}_j})} \ln \frac{\Upsilon_i^-}{(1 - \Upsilon_i^+)} \right\}. \end{aligned}$$

The optimality condition can be further simplified as:

$$\begin{aligned} &= \ln \frac{\alpha}{1 - \alpha} + \ln \frac{\mathcal{P}(1)}{\mathcal{P}(-1)} + \sum_{i=1}^m \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} ((X_{ij} + 1)! + 1) \\ &\quad - \sum_{i=1}^m \ln \frac{\Upsilon_i^-}{(1 - \Upsilon_i^+)} ((1 - X_{ij})! - 1) > 0 \\ &= \ln \frac{\alpha}{1 - \alpha} + \ln \frac{\mathcal{P}(1)}{\mathcal{P}(-1)} + \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} (1 + X_{ij})! \right. \\ &\quad \left. - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^+} (1 - X_{ij})! \right] + \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^+} \right] > 0 \\ &= \ln \frac{\alpha}{1 - \alpha} + \ln \frac{\mathcal{P}(1)}{\mathcal{P}(-1)} + \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^-} (1 + X_{ij})! \right. \\ &\quad \left. - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^+} (1 - X_{ij})! \right] + \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+ (1 - \Upsilon_i^+)}{\Upsilon_i^- (1 - \Upsilon_i^-)} \right] > 0. \end{aligned}$$

Finally, we obtain:

$$\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^+} = \{ \mathcal{J}_{\mathcal{D}_j}, \varphi + \beta + \rho + \Psi > 0 \}. \quad (47)$$

From the above, it directly follows that:

$$\mathcal{J}_{\mathcal{D}_j}^{\mathcal{O}^-} = \{ \mathcal{J}_{\mathcal{D}_j}, \varphi + \beta + \rho + \Psi \leq 0 \}. \quad (48)$$

Therefore, we infer,  $\hat{f} = \sigma(\varphi + \beta + \rho + \Psi)$ .  $\square$

Clearly, only  $\rho$  includes the choices of the individual DCs and assigns weights to the individual preferences.  $\varphi$ ,  $\beta$ , and  $\Psi$  are the bias elements [38].

## 6 ANALYTICAL RESULTS

**Proposition 1.** *The mean weight,  $M(1:-1)$ , assigned to the decision of an individual DC, for the two states of nature, is positive.*

**Proof.** From Theorem 1, we obtain:

$$\rho = \sum_{i=1}^m \left[ \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^+} (1 + X_{ij})! - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^-} (1 - X_{ij})! \right], \quad (49)$$

where  $X_{ij} = \{1, -1\}$ . Therefore,  $1 \leq (1 + X_{ij})!$ ,  $(1 - X_{ij})! \leq 2$ . For a particular  $\mathcal{D}_i$ , if  $X_{ij} = 1$ , then Equation (49) can be rewritten as:

$$\rho_i = \ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^+} (1 + X_{ij}) - \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^-}. \quad (50)$$

Therefore, the weight of  $X_{ij}$  is  $\ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^+}$ . Similarly, for  $X_{ij} = -1$ , weight of  $X_{ij}$  is  $\ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^-}$ . Therefore,

$$M(1 : -1) = \frac{\ln \frac{\Upsilon_i^+}{1 - \Upsilon_i^+} + \ln \frac{\Upsilon_i^-}{1 - \Upsilon_i^-}}{2} = \frac{1}{2} \ln \frac{\Upsilon_i^+ \Upsilon_i^-}{(1 - \Upsilon_i^+)(1 - \Upsilon_i^-)}.$$

From Equation (20), it directly follows,

$$\Upsilon_i^+ \Upsilon_i^- > (1 - \Upsilon_i^+)(1 - \Upsilon_i^-) \text{ i.e., } M(1 : -1) > 0. \quad (51)$$

□

**Proposition 2.** *The proposed optimal decision rule satisfies the Potential Pareto criterion.*

**Proof.** As per the *Potential Pareto criterion* (PPC) [43], [44], the winner compensates the losers and still remains better off. We assume  $\mathcal{D}_w$  is selected for  $App_i$  through decision rule  $\mathcal{O}$ . Therefore, we have,  $\mathcal{O}(\mathcal{J}_{\mathcal{D}_w}) = 1$  and  $\mathcal{E}_{max}(App_i) = \mathcal{E}_{\mathcal{D}_w}(App_i)$ :

$$\begin{aligned} \mathcal{E}_{max}(App_i) &= \alpha \left[ \mathcal{P}(1 : 1) P_{\mathcal{D}_w}(\mathcal{O} : 1) + \mathcal{P}(-1 : 1)(1 - P_{\mathcal{D}_w}(\mathcal{O} : 1)) \right] \\ &+ (1 - \alpha) \left[ \mathcal{P}(-1 : -1) P_{\mathcal{D}_w}(\mathcal{O} : -1) + \mathcal{P}(1 : -1)(1 - P_{\mathcal{D}_w}(\mathcal{O} : -1)) \right], \end{aligned}$$

which gives us:

$$\mathcal{E}_{\mathcal{D}_w}(App_i) > \mathcal{E}_{\mathcal{D}_j}(App_i), \forall \mathcal{D}_j \in \mathcal{D}^{nom}, \mathcal{D}_j \neq \mathcal{D}_w. \quad (52)$$

Therefore, the compensation of any  $\mathcal{D}_j$ , is expressed as  $\mathcal{E}_{\mathcal{D}_j}(App_i)$  and the net benefit of the winner DC is  $\mathcal{E}_{\mathcal{D}_w}(App_i) - \mathcal{E}_{\mathcal{D}_j}(App_i)$ . However, the winner DC must have a positive benefit. Using Equation (52), we have:

$$\begin{aligned} &\alpha P_{\mathcal{D}_w}(\mathcal{O} : 1) \mathcal{P}(1) + (1 - \alpha) P_{\mathcal{D}_w}(\mathcal{O} : -1) \mathcal{P}(-1) \\ &> \alpha P_{\mathcal{D}_j}(\mathcal{O} : 1) \mathcal{P}(1) + (1 - \alpha) P_{\mathcal{D}_j}(\mathcal{O} : -1) \mathcal{P}(-1), \end{aligned}$$

$$\begin{aligned} &\Rightarrow \alpha P_{\mathcal{D}_w}(\mathcal{O} : 1) \mathcal{P}(1) + (1 - \alpha) P_{\mathcal{D}_w}(\mathcal{O} : -1) \mathcal{P}(-1) \\ &+ \alpha \mathcal{P}(-1 : 1) + (1 - \alpha) \mathcal{P}(1 : -1) > \alpha P_{\mathcal{D}_j}(\mathcal{O} : 1) \mathcal{P}(1) \\ &+ (1 - \alpha) P_{\mathcal{D}_j}(\mathcal{O} : -1) \mathcal{P}(-1) + \alpha \mathcal{P}(-1 : 1) + (1 - \alpha) \mathcal{P}(1 : -1), \end{aligned}$$

$$\Rightarrow \mathcal{E}_{\mathcal{D}_w}(App_i) = \mathcal{E}_{\mathcal{D}_j}(App_i) + c'.$$

Therefore,  $\mathcal{D}_w$  compensates other DCs and also incurs a positive benefit  $c'$ ; thereby, satisfying the PPC. □

**Proposition 3.** *The proposed decision rule guarantees the unanimity criterion.*

**Proof.** The unanimity criterion [45] is one of the most rational properties of a decision rule. For two nominated DCs  $\mathcal{D}_{j_1}, \mathcal{D}_{j_2} \in \mathcal{D}^{nom}$ . For a particular DC,  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ ,  $\mathcal{D}_{j_1}$  is preferred to  $\mathcal{D}_{j_2}$ , if either:

- $X_{i1} = 1$ , and  $X_{i2} = -1$  (when  $\mathcal{D}_i$  rejects  $\mathcal{D}_{j_2}$  clearly),
- $X_{i1} = X_{i2} = 1$  (when  $\mathcal{D}_{j_2}$  is not rejected, however,  $\mathcal{D}_{j_1}$  is preferred).

**Case a:** If for every  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ ,  $\mathcal{D}_{j_1} \succ_{\mathcal{D}_i} \mathcal{D}_{j_2}$  hold true, then it readily follows:

$$\mathcal{Q}_{net}^{\mathcal{D}_{j_1}}(App_i) > \mathcal{Q}_{net}^{\mathcal{D}_{j_2}}(App_i), \forall \mathcal{D}_i \in \mathcal{D}_{App_i} \quad (53)$$

$$\begin{aligned} &\Rightarrow \frac{pP}{\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_{j_1}) + \delta(\mathcal{D}_{j_1}, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \left( \lambda_k \times \frac{S(App_i)}{\sum_{i=1}^m \lambda_k} \right)} \\ &> \frac{pP}{\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_{j_2}) + \delta(\mathcal{D}_{j_2}, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \left( \lambda_k \times \frac{S(App_i)}{\sum_{i=1}^m \lambda_k} \right)}. \end{aligned}$$

Consequently,  $\mathcal{E}_{\mathcal{D}_{j_1}}(App_i) > \mathcal{E}_{\mathcal{D}_{j_2}}(App_i)$ ; thereby, inferring  $\mathcal{O}(\mathcal{J}_{\mathcal{D}_{j_1}}) = 1$ .

**Case b:** In this case, for every  $\mathcal{D}_i \in \mathcal{D}_{App_i}$ , we have  $\mathcal{D}_{j_1} \succeq_{\mathcal{D}_i} \mathcal{D}_{j_2}$ . This implies that:

$$\begin{aligned} &\frac{pP}{\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_{j_1}) + \delta(\mathcal{D}_{j_1}, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \left( \lambda_k \times \frac{S(App_i)}{\sum_{i=1}^m \lambda_k} \right)} \\ &\geq \frac{pP}{\sum_{i=1}^m \mathcal{M}(\mathcal{D}_i, \mathcal{D}_{j_2}) + \delta(\mathcal{D}_{j_2}, l_{1,i}, l_{2,i}) + \sum_{i=1}^m \left( \lambda_k \times \frac{S(App_i)}{\sum_{i=1}^m \lambda_k} \right)}, \\ &\Rightarrow \mathcal{P}_{\mathcal{D}_{j_1}}(1) \geq \mathcal{P}_{\mathcal{D}_{j_2}}(1), \text{ and } \mathcal{P}_{\mathcal{D}_{j_1}}(-1) \leq \mathcal{P}_{\mathcal{D}_{j_2}}(-1) \quad (54) \end{aligned}$$

$$\Rightarrow \mathcal{E}_{\mathcal{D}_{j_1}}(App_i) > \mathcal{E}_{\mathcal{D}_{j_2}}(App_i) \Rightarrow \mathcal{O}(\mathcal{J}_{\mathcal{D}_{j_1}}) = 1. \quad (55)$$

□

## 7 PERFORMANCE EVALUATION

This Section presents and analyzes the performance of the proposed system of scheduling a DC for serving a particular application  $App_i$ . The details of the testbed information are provided in Table 2.



TABLE 2  
Testbed Information

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

## 7.1 Single Application Scenario

The experiments were initially performed for a single application that demands for 10 distinct VSs. The underlying sensor network was simulated over a uniform, random deployment of 100 physical sensor nodes over a region of 500 m × 500 m. The experimental setup is illustrated in Table 3. The experiments were performed to select 10 temporary DCs forming  $\mathcal{D}_{App_i}$ . Based on the proximity of the DCs with the application center  $l_1 = 28$  and  $l_2 = 196$ ,  $\mathcal{D}^{nom}$  was built. The parameters of the component DCs of  $\mathcal{D}^{nom}$  are illustrated in Table 4. For the purpose of selection and analysis of the optimal decision rule. We considered the set of decision rules  $F = \{f_i(\cdot)\}$ , such that,  $\forall f_i(\cdot) \in F$ :

$$f_i(\mathcal{J}_{\mathcal{D}_i}) = 1, f_i(\mathcal{J}_{\mathcal{D}_j}) = -1, \forall \mathcal{D}_i, \mathcal{D}_j \in \mathcal{D}^{nom}, \mathcal{D}_i \neq \mathcal{D}_j. \quad (56)$$

The performance of the rules in  $F$  are studied and analyzed, as shown in Fig. 4.

As per Equation (56), every decision rule in  $F$  schedules a unique DC from the set  $\mathcal{D}^{nom}$ . Following ever  $f_i \in F$ , the cumulative migration cost of all the DCs of  $\mathcal{D}_{App_i}$  to the DCs of  $\mathcal{D}^{nom}$  are analyzed in Fig. 4a. It is observed that  $\mathcal{D}_5$  has the lowest migration cost followed by  $\mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_1$ , and  $\mathcal{D}_2$ . Fig. 4b depicts the cumulative delivery cost of the DCs, in which  $\mathcal{D}_5$  performs poorly. On the other hand,  $\mathcal{D}_2$  bears a low cost for delivering the packets to the application center. Fig. 4c highlights the closeness of  $\mathcal{D}_3, \mathcal{D}_2$ , and  $\mathcal{D}_1$  in terms of the overall service delay. However, in Fig. 4d, the basic QoS is evaluated in which the outcome of the decision rule  $f_3(\cdot)$ , i.e.,  $\mathcal{D}_3$  obtains the maximum QoS. The user-dissatisfaction delay product is evaluated and analyzed in Fig. 4e. It is observed that  $\mathcal{D}_3$  provides the minimum dissatisfaction to  $App_i$ . The effective QoS is shown in Fig. 4f, in which  $f_3(\cdot)$  emerges as the optimum decision rule.

Now, we analyze the correctness of the decision rule  $f_3(\cdot)$  in terms of the heterogeneous, and fallible decision making ability of  $\mathcal{D}_{App_i}$  and the “goodness” or “badness” of the

TABLE 3  
Experimental Setup for Single Application

Parameters	Values
Sensor deployment area	500 m × 500 m
Deployment	Uniform, random
Number of nodes	100
Number of VSs	10
Number of temporary DCs ( $ \mathcal{D}_{App_i} $ )	10
Number of nominated DCs ( $ \mathcal{D}^{nom} $ )	5
Size of each packet	2 B
Transmission latency	100 m/s
Distribution of demand rate	Poisson

TABLE 4  
Parameters of  $\mathcal{D}^{nom}$

	$\mathcal{D}_1^{nom}$	$\mathcal{D}_2^{nom}$	$\mathcal{D}_3^{nom}$	$\mathcal{D}_4^{nom}$	$\mathcal{D}_5^{nom}$
Abscissa	38	224	48	273	453
Ordinate	296	464	188	56	317
Migration latency (Bps)	330.93	456.58	236.24	86.54	398.73
Delivery latency (m/s)	248.19	305.37	142.86	142.86	215.61

components of  $\mathcal{D}^{nom}$ . As shown in Fig. 5, the decision making abilities of the DCs are studied by learning the behavior of each DC for last  $k$  instants of time. Here, we assume that  $k = 50$ . In Fig. 5a, it is found that the mean probability of a correct decision is 0.6, 615; whereas, that of a wrong decision is 0.3, 385. Moreover, the mean probability of rejecting a “good” DC is 0.684; whereas, that of a “bad” DC is 0.639. The probabilities of the Type I and Type II error are depicted in Fig. 5b, the mean probability being 0.316, and 0.361, respectively. With this fallible decision making ability of  $\mathcal{D}_{App_i}$ ,  $f_3(\cdot)$  eventually schedules the optimal DC.

Fig. 6 illustrates the capacity, “goodness” and “badness” of every  $\mathcal{D}_j \in \mathcal{D}^{nom}$ . For the sake of simplicity, Equation (22) and Equation (23) are simplified to obtain the capacity  $\mathcal{C}$ , “goodness” and “badness” of a DC,  $\mathcal{C}^{max}(\mathcal{D}_j)$ , and  $\mathcal{C}(\mathcal{D}_j)$  are the total number of applications that can be supported by, and that are running withing  $\mathcal{D}_j$ , respectively. Therefore:

$$\mathcal{G}(\mathcal{D}_j) = \mathcal{C}^{max}(\mathcal{D}_j) - \mathcal{C}(\mathcal{D}_j), \bar{\mathcal{G}}(\mathcal{D}_j) = \mathcal{C}^{max}(\mathcal{D}_j) - \mathcal{G}(\mathcal{D}_j). \quad (57)$$

Now, the outcome of  $f_3(\cdot)$  is clearer, as  $\mathcal{D}_3$  exhibits the maximum “goodness”, and minimum “badness”, although  $\mathcal{D}_2$  has a higher capacity to support applications.

## 7.2 Multiple Application Scenario

Having studied the performance evaluation of the proposed system for a single application, we now examine and discuss the system performance for a multiple applications. The experimental setup is depicted in Table 5. The performance metrics that have been considered are:

- Mean turnaround time;
- Mean throughput;
- Channel utilization; and
- Mean data center utilization.

Based on the above metrics, the system performance is evaluated by varying the number of applications and the number of DCs, as shown in Fig. 7.

### 7.2.1 Mean Turnaround Time

For the purpose of examination of the system performance in terms of the mean turnaround time  $\tau_{mean}$  for every application, the metric is defined as the mean of the turnaround time of all the applications running in the system. The turnaround time,  $\tau$ , of a particular application is computed as the service delay incurred to serve a particular demand of the application.  $\tau$  is expressed as:

$$\tau(App_i) = \sum_{j=1}^{g(i)} \lambda_j \times \frac{\mathcal{S}_{tot}}{\sum_{p=1}^n \sum_{k=1}^{g(p)} \lambda_k}, \quad (58)$$

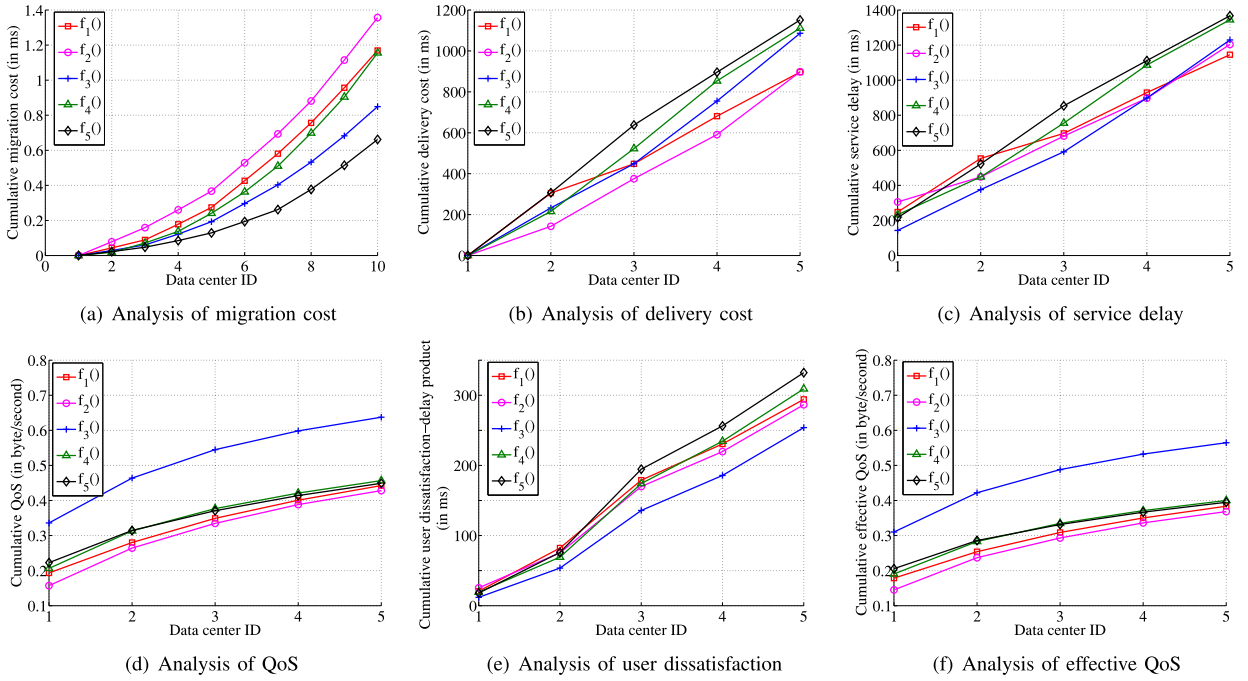


Fig. 4. Performance evaluation of decision rules of set  $F$ .

$$\tau_{mean}(App_i) = \sum_{q=1}^{|\mathcal{D}^{nom}|} \tau_q, \quad (59)$$

where  $S_{tot}$  is the total service delay incurred in processing  $n$  applications (per DC), each requesting for  $g(p)$  distinct services, and  $\lambda$  being the demand rate. As indicated in Fig. 7a, we find that when the number of DCs are low, the value of  $\tau_{mean}$  increases with the increase in the number of applications. With the increase in the number of nominated DCs, the mean turnaround time falls reasonably, as is the case of 12–15 DCs.

### 7.2.2 Mean Throughput

The mean throughput,  $\mathcal{T}_{mean}$ , of the system is defined as the average outflow of information in bits per unit time from every nominated DC. If the  $p_i$  packets are generated from a  $\mathcal{D}_i \in \mathcal{D}^{nom}$  per  $\Delta t$  unit of time, then the  $\mathcal{T}_{mean}$  is expressed as:

$$\mathcal{T}_{mean} = \left( \sum_{q=1}^{|\mathcal{D}^{nom}|} \frac{P \times p_q}{\Delta t} \right) / |\mathcal{D}^{nom}|, \quad (60)$$

where  $P$  being the size of each packet. Now Fig. 7b clearly shows the variation of the mean throughput with the

variation of the number of applications being served and the total number of nominated DCs. It is observed that the mean throughput increases significantly with both the increase in the count of the applications and the DCs.

### 7.2.3 Channel Utilization

For the purpose of evaluating the channel utilization  $\chi_B$ , the metric is defined as the ratio of the amount of channel utilized in unit time to the total bandwidth available  $\chi_B^{max}$ , expressed as percentage. Mathematically,  $\chi$  is expressed as:

$$\chi_B = \frac{\sum_{q=1}^{|\mathcal{D}^{nom}|} P \times p_q}{\chi_B^{max}} \times 100\%. \quad (61)$$

Fig. 7c indicates that percentage of channel utilization is initially low for lower number of nominated DCs serving the applications. As the count of DCs increase with the utilization percentage increases reasonably with the increase in the number of currently running applications.

### 7.2.4 Mean Data Center Utilization

The mean data center utilization is defined as the ratio of the number of DCs required to handle the current set of

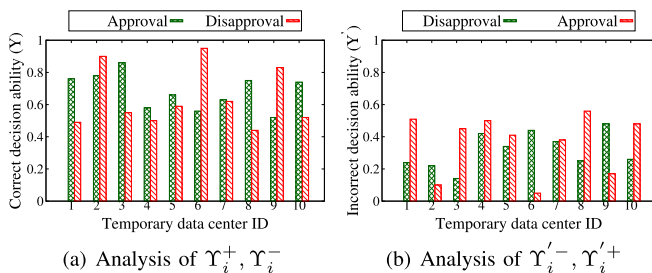


Fig. 5. Analysis of the decision making abilities of  $\mathcal{D}_{App_i}$ .

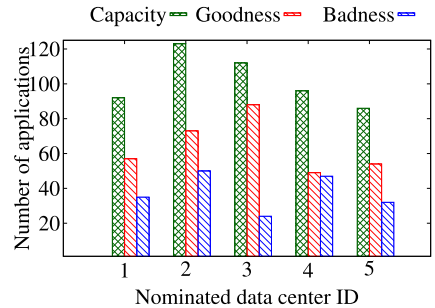


Fig. 6. Analysis of the “goodness” and “badness” of the nominated data centers.

TABLE 5  
Experimental Setup for Multiple Applications

Parameters	Values
Sensor deployment area	500 m × 500 m
Number of applications	{10, 20, ..., 100}
Number of temporary data centers ( $ \mathcal{D}_{App_i} $ )	50
Number of nominated data centers ( $ \mathcal{D}^{nom} $ )	{3, 6, ..., 15}
Intra data center bandwidth	Very high, > 1 MBps
Channel bandwidth	Moderate, $\leq 1$ MBps
Size of each packet	2 B
Transmission latency	100 m/s
Distribution of demand rate	Poisson
Maximum number of demand	1 – 10

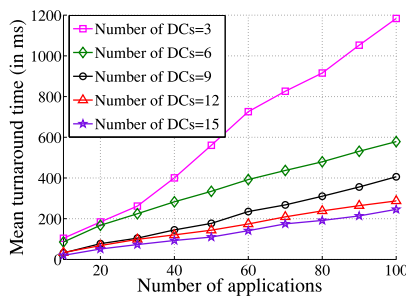
running applications to the total number of DCs that are available in the system, expressed as percentage. If every  $\mathcal{D}_j \in \mathcal{D}^{nom}$  has a capacity to handle  $n_i$  number of applications, and we assume  $n = \{10, 20, \dots, 100\}$  to be the total number of applications then  $\chi_D$  is expressed as:

$$\chi_D = \frac{\left[ \left( \sum_{p=1}^n \sum_{k=1}^{g(p)} \lambda_k \right) \bmod (|\mathcal{D}^{nom}|) + 1 \right]}{|\mathcal{D}^{nom}|} \times 100\%. \quad (62)$$

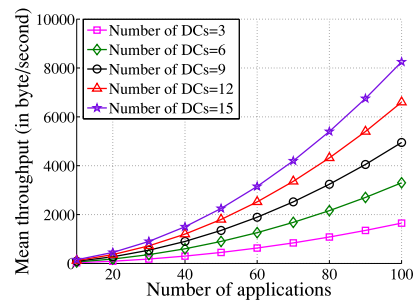
From Fig. 7d, it is observed that the utilization percentage is negligible with three-six nominated DCs, and is almost independent to the number of applications being served. As the count of DCs increase from 9 onwards, the utilization percentage rises significantly with the increase in the application count.

## 8 COMPLEXITY ANALYSIS

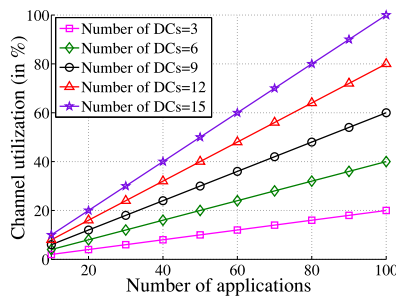
In this Section, we discuss, and analyze the asymptotic computational complexity of the proposed work to justify its applicability for real-time processing. The metric for



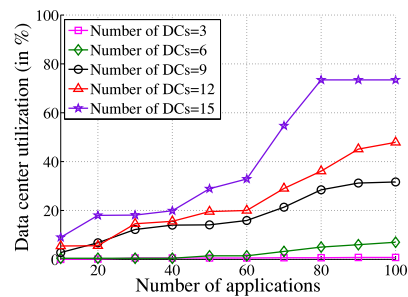
(a) Study of mean turnaround time



(b) Study of mean throughput

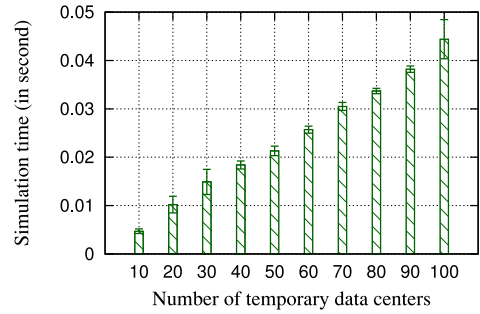


(c) Study of channel utilization

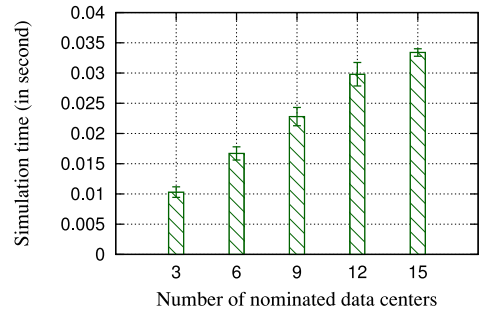


(d) Study of data center utilization

Fig. 7. Performance evaluation for multiple applications.



(a) Plot of number of temporary DCs versus simulation time



(b) Plot of number of nominated DCs versus simulation time

Fig. 8. Complexity analysis by varying system components.

computational complexity has been measured in terms of the simulation time by varying the parameters of simulation, as shown in Fig. 8. Initially, the setup for performing the experiment assumes a fixed number of applications ( $n = 100$ ). The variation of the simulation time are recorded by varying the number of DCs in different iterations.

As shown in Fig. 8a, the number of temporary DCs are varied from 10 to 100 with a step of 10, and the number of nominated DCs are kept fixed,  $\mathcal{D}^{nom} = 9$ . For a particular

number of temporary DC, values are recorded for 50 iterations and are plotted with 95 percent confidence. It was observed that the simulation time averages at 0.024 within interval [0.022, 0.026] with 95 percent confidence.

Fig. 8b indicates the variation in the computational complexity as the number of nominated DCs vary. For this experiment, the number of temporary DCs is kept constant,  $D_{App_i} = 50$ , and the count of the nominated DCs are varied from 3 to 15, with a step size of 3. It is observed that the simulation time centers at 0.0226 within the interval [0.0214, 0.0238], with 95 percent confidence.

The mean variance of complexity with the increase in DCs (for Figs. 8a, and 8b) is found to be  $8.9 \times 10^{-6}$  second, and  $5 \times 10^{-6}$  second, respectively. Therefore, the increase of the computational complexity of the proposed work with the increased number of DCs is negligible.

## 9 CONCLUSIONS

The proposed work focused on the networking of a sensor-cloud infrastructure. In a sensor-cloud scenario, data from various physical sensors are grouped to form a VS. An application requests for multiple such VSs spanning across multiple geographical regions. Therefore, data from the VSs are collected by several geo-spatial DCs. However, sensor-cloud involves collection of these VSs to a single VM within a single DC for further analysis and collective processing. This necessitates scheduling of a single DC to serve each application. For the purpose of scheduling, the work propounds a QoS based optimal decision rule.

Future scope of research will focus on extending the problem for dynamic shifting of VMs among DCs for serving applications; thereby, ensuring localized load sharing and balancing among the DCs. Revisiting the problem for a mobile sensor-cloud scenario also induces research interest.

## ACKNOWLEDGMENTS

This work was partially supported by a fellowship sponsored by the Tata Consultancy Services (TCS), India.

## REFERENCES

- [1] M. Yuriyama, T. Kushida, and M. Itakura, "A new model of accelerating service innovation with Sensor-cloud infrastructure," in *Proc. Annu. SRII Global Conf.*, Mar. 2011, pp. 308–314.
- [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. Sens. Netw.*, vol. 2013, p. 18, Nov. 2012.
- [3] K.-L. Tan, "What's next?: Sensor + cloud!?" in *Proc. 7th Int. Workshop Data Manage. Sens. Netw.*, 2010, p. 1.
- [4] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 142–157, Jul. 2013.
- [5] A. Dastjerdi and R. Buyya, "Compatibility-aware cloud service composition under fuzzy preferences of users," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 1–13, Jan. 2014.
- [6] I. Moreno, P. Garraghan, P. Townsend, and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 208–221, Apr. 2014.
- [7] 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.*, 2010, pp. 1–8.
- [8] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor-cloud: A paradigm shift from wireless sensor network," *IEEE Syst. J.*, 2014, Doi: 10.1109/JSYST.2014.2362617.
- [9] 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, 2014.
- [10] M. M. Hassan, B. Song, and E.-N. Huh, "A framework of sensor-cloud integration opportunities and challenges," in *Proc. 3rd Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2009, pp. 618–626.
- [11] 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.
- [12] Open Geospatial Consortium [Online]. Available: <http://www.opengeospatial.org/>
- [13] 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 Inf.*, vol. 19, no. 2, pp. 541–548, 2015.
- [14] S. Misra, A. Singh, S. Chatterjee, and M. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military Tri-services operations and decision making," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–9, 2014, Doi: 10.1109/JSYST.2014.2316013.
- [15] M. Mahmoud and X. Shen, "A Cloud-based scheme for protecting Source-location privacy against Hotspot-locating attack in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1805–1818, Oct. 2012.
- [16] S. Misra and S. Chatterjee, "Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare: Data aggregation and channelization," *Inf. Sci.* [Online], 284, pp. 95–117. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514005489>
- [17] 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 Sens. Syst.*, vol. 4, no. 2, pp. 61–68, Jun. 2014.
- [18] S. Misra, S. Das, M. Khatua, and M. Obaidat, "QoS-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 181–193, Apr. 2014.
- [19] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," *IEEE Sens. J.*, vol. 13, no. 10, pp. 3733–3741, Oct. 2013.
- [20] S. Mori, Y.-C. Wang, T. Umedu, A. Hiromori, H. Yamaguchi, and T. Higashino, "Design and architecture of cloud-based mobile phone sensing middleware," in *Proc. 2nd Symp. Netw. Cloud Comput. Appl.*, Dec. 2012, pp. 102–109.
- [21] M. V. Nguyen and E.-N. Huh, "An efficient key management for secure multicast in sensor-cloud," in *Proc. 1st ACIS/JNU Int. Conf. Comput. Netw. Syst. Ind. Eng.*, May 2011, pp. 3–9.
- [22] K. Nagothu, B. Kelley, M. Jamshidi, and A. Rajae, "Persistent Net-AMI for microgrid infrastructure using cognitive radio on cloud data centers," *IEEE Syst. J.*, vol. 6, no. 1, pp. 4–15, Mar. 2012.
- [23] X. Xu, J. Wu, G. Yang, and R. Wang, "Low-power task scheduling algorithm for large-scale cloud data centers," *J. Syst. Eng. Electron.*, vol. 24, no. 5, pp. 870–878, Oct. 2013.
- [24] H. Wu, A. Tantawi, Y. Diao, and W. Wang, "Adaptive memory load management in cloud data centers," *IBM J. Res. Develop.*, vol. 55, no. 6, pp. 5:1–5:10, Nov. 2011.
- [25] Z. Zheng, J. Wang, J. Ren, W. Hou, and J. Wang, "Least maintenance batch scheduling in cloud data center networks," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 901–904, Jun. 2014.
- [26] F. Larumbe and B. Sanso, "A tabu search algorithm for the location of data centers and software components in green cloud computing networks," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 22–35, Jan. 2013.
- [27] C. Mastroianni, M. Meo, and G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 215–228, Jul. 2013.
- [28] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1366–1379, Jul. 2013.
- [29] D. Bruneo, "A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, pp. 560–569, Mar. 2014.
- [30] Y. Hua, X. Liu, and H. Jiang, "Antelope: A Semantic-aware data cube scheme for cloud data center networks," *IEEE Trans. Comput.*, vol. 63, no. 9, pp. 2146–2159, Sep. 2014.
- [31] C. Assi, S. Ayoubi, S. Sebbah, and K. Shaban, "Towards scalable traffic management in cloud data centers," *IEEE Trans. Commun.*, vol. 62, no. 3, pp. 1033–1045, Mar. 2014.



- [32] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Select. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.
- [33] W. Zhang, Y. Wen, and H.-H. Chen, "Toward transcoding as a service: Energy-efficient offloading policy for green mobile cloud," *IEEE Netw.*, vol. 28, no. 6, pp. 67–73, Nov. 2014.
- [34] H. Liang, L. Cai, D. Huang, X. Shen, and D. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2222–2232, Jun. 2012.
- [35] S. Chatterjee and S. Misra, "Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud," in *IEEE Int. Conf. Adv. Netw. Telecommun. Syst.*, Dec. 2014, pp. 1–6.
- [36] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous Service-oriented architecture of Sensor-cloud infrastructure," *IEEE Trans. Serv. Comput.*, p. 1, 2015, Doi: 10.1109/TSC.2015.2453958.
- [37] S. Chatterjee and S. Misra, "Optimal composition of a virtual sensor for efficient virtualization within Sensor-cloud," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 448–453.
- [38] R. C. Ben-Yashar and S. I. Nitzan. (1997). The optimal decision rule for fixed-size committees in dichotomous choice situations: The general result. *Int. Econ. Rev.* [Online]. 38(1), pp. 175–186. Available: <http://www.jstor.org/stable/2527413>
- [39] (2012). Cultivating a crystal ball for data center availability and performance. Emerson Network Power [Online]. Available: [Online]. Available: <http://www.emersonnetworkpower.com/documentation/en-us/solutions/cio-topics/documents/cultivating-a-crystal-ball-for-dat-a-center-playbook.pdf>
- [40] G. Leopold. (2014, Apr. Survey finds disconnect between big data and Decision-making [Online]. Available: <http://www.datanami.com/2014/04/02/survey-finds-disconnect-between-big-data-and-decision-making/>
- [41] E. Baralis, L. Cagliero, and P. Garza, "EnBay: A novel pattern-based Bayesian classifier," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2780–2795, Dec. 2013.
- [42] C. Ordonez and S. K. Pitchaimalai, "Bayesian classifiers programmed in SQL," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 139–144, Jan. 2010.
- [43] B. P. Brownstein, "Pareto optimality, external benefits and public goods: A subjectivist approach," *J. Libertarian Stud.*, vol. 4, no. 1, pp. 93–106, 1980.
- [44] Z. He, G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.
- [45] C. Borgers, *Mathematics of Social Choice: Voting, Compensation, and Division*. Philadelphia, PA, USA: SIAM, 2010.



**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.



**Sudip Misra** received the PhD degree in computer science from Carleton University, in Ottawa, Canada, and the master's and bachelor's degrees respectively from the University of New Brunswick, Fredericton, Canada, and the Indian Institute of Technology, Kharagpur, India. He 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, Yale University, Nortel Networks (Canada) and the Government of Ontario (Canada). He has several years of experience working in the academia, government, and the private sectors in research, teaching, consulting, project management, architecture, software design and product engineering roles. His current research interests include algorithm design for emerging communication networks. He is the author of over 180 scholarly research papers (including 90 journal papers). He has won eight research paper awards in different conferences. He was awarded the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at IEEE GLOBECOM 2012, Anaheim, CA. He also received 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 Generals 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 Governments prestigious NSERC Post Doctoral Fellowship and the Humboldt Research Fellowship in Germany. He is the editor-in-chief of the *International Journal of Communication Networks and Distributed Systems (IJCNDS)*, Inder-science, United Kingdom. He has also been serving as the associate editor of the *Telecommunication Systems Journal (Springer)*, *Security and Communication Networks Journal (Wiley)*, *International Journal of Communication Systems (Wiley)*, and the *EURASIP Journal of Wireless Communications and Networking*. He is also an editor/editorial board member/editorial review board member of the *IET Communications Journal*, *IET Wireless Sensor Systems*, and *Computers and Electrical Engineering Journal (Elsevier)*. He has edited six 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 was invited to chair several international conference/workshop programs and sessions. He served in the program committees of several international conferences. He was also invited to deliver keynote/invited lectures in over 20 international conferences in, Canada, Europe, Asia, and Africa. He is a senior member of the IEEE.



**Samee U. Khan** received the BS degree in 1999 from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, and the PhD degree in 2007 from the University of Texas, Arlington, TX. He is currently an associate professor of electrical and computer engineering at the North Dakota State University, Fargo, ND. His research interests include optimization, robustness, and security of: cloud, grid, cluster and big data computing, social networks, wired and wireless networks, power systems, smart grids, and optical networks. His work has appeared in over 250 publications. He is a fellow of the Institution of Engineering and Technology (IET, formerly IEE), and a fellow of the British Computer Society (BCS). He is a senior member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).