Assessment of the Suitability of Fog Computing in the Context of Internet of Things

Subhadeep Sarkar, *Student Member, IEEE*, Subarna Chatterjee, *Student Member, IEEE*, and
Sudip Misra, *Senior Member, IEEE*

Abstract—This work performs a rigorous, comparative analysis of the fog computing paradigm and the conventional cloud computing paradigm in the context of the Internet of Things (IoT), by mathematically formulating the parameters and characteristics of fog computing—one of the first attempts of its kind. With the rapid increase in the number of Internet-connected devices, the increased demand of real-time, low-latency services is proving to be challenging for the traditional cloud computing framework. Also, our irreplaceable dependency on cloud computing demands the cloud data centers (DCs) always to be up and running which exhausts huge amount of power and yield tons of carbon dioxide (CO_2) gas. In this work, we assess the applicability of the newly proposed *fog computing* paradigm to serve the demands of the latency-sensitive applications in the context of IoT. We model the fog computing paradigm by mathematically characterizing the fog computing network in terms of power consumption, service latency, CO_2 emission, and cost, and evaluating its performance for an environment with high number of Internet-connected devices demanding real-time service. A case study is performed with traffic generated from the 100 highest populated cities being served by eight geographically distributed DCs. Results show that as the number of applications demanding real-time service increases, the fog computing paradigm outperforms traditional cloud computing. For an environment with 50 percent applications requesting for instantaneous, real-time services, the overall service latency for fog computing is noted to decrease by 50.09 percent. However, it is mentionworthy that for an environment with less percentage of applications demanding for low-latency services, fog computing is observed to be an overhead compared to the traditional cloud computing. Therefore, the work shows that in the context of IoT, with high number of latency-sensitive applications fog computing outperforms cloud computing.

Index Terms—Fog computing, cloud computing, Internet of things (IoT), service latency, power consumption, carbon-dioxide emission

✦

1 INTRODUCTION

RECENT advancements in computer technologies have led to the conceptualization, development, and implementation of cloud computing systems. From its inception, cloud computing has gained widespread popularity due to its applicability in diverse, widespread domains. Cloud computing systems are generally based on data centric networks (DCNs), which are treated as the sole, monopolized hubs responsible for computation and storage. For contemporary cloud-based systems, all service requests and resource demands are analyzed and processed within the data centers (DCs). However, with the steep rise in the number of Internet-connected devices and in the light of the emerging technology of the Internet of things (IoT), the amount of data to be handled by the cloud DCs is paramount. In 2012, global commercialization of IoT-based application systems generated a revenue of $\$4.8$ trillon [1]. It is statistically estimated that by 2015, around 25 billion autonomous devices will be connected to the Internet. Cisco estimates that due to IoT, the global corporate profits will also increase approximately by 21 percent [2]. Also, the cloud DCs exhaust massive amount of energy leading to the emission of enormous amount of greenhouse gases (GHGs), especially carbon dioxide (CO_2). This takes a deep toll on the environment.

The technology of IoT is reliant on cloud computing. Data from the billions of Internet-connected devices are voluminous and demand to be processed within the cloud DCs. Most of these IoT infrastructures, such as smart vehicular traffic management systems, smart driving and car parking systems, and smart grids are observed to demand real-time, low-latency services from the service providers. Since conventional cloud computing involves processing, computation, and storage of the data only within DCs, the massive data traffic generated from the IoT devices is anticipated to experience a huge network bottleneck and, in turn, high service latency and poor quality of service (QoS). Moreover, in order to process and serve this high number of requests the DCs are required to be up and running around the clock which results in the consumption of enormous amount of energy and massive emission of CO_2.

In this work, we analyze the suitability of a recent computing paradigm—*fog computing* to serve the demands of the real-time, latency-sensitive applications in the context of IoT. Fog (From cOre to edGe) computing, a term coined by Cisco in 2012 [3], is a distributed computing paradigm, that empowers the network devices

- S. Sarkar is with the School of Information Technology and the School of Medical Science and Technology, Indian Institute of Technology, Kharagpur 721302, India. E-mail: subhadeep@smst.iitkgp.ernet.in.
- S. Chatterjee and S. Misra are with the School of Information Technology, Indian Institute of Technology, Kharagpur 721302, India.
  E-mail: {subarna, smisra}@sit.iitkgp.ernet.in.

at different hierarchical levels with various degrees of computational and storage capability. These devices are equipped with an 'intelligence' which allows them to examine whether an application request requires the intervention of the cloud computing tier or not. The idea is to serve the requests which demand real-time, low-latency services (e.g., live streaming, smart traffic monitoring, smart parking etc.) by the fog computing devices and the connected work stations and small-scale storage units. However, the requests which demand semi-permanent and permanent storage or require extensive analysis involving historical data-sets (e.g., social media data, photos, videos, medical history, data backups etc.), these devices only act as routers or gateways to redirect the requests to the core cloud computing framework. The focus of this work is to assess the suitability of fog computing in the context of real-time service requests, and compare its performance against the traditional cloud computing frameworks.

Judging by the different types of applications served by fog and cloud, it is clear that fog computing is not a replacement of cloud computing, rather these two technologies complement one another. The complementary functions of cloud and fog enable the end-users to experience a new breed of computing technology that serves the requirements of the real-time, low-latency IoT-applications running at the network edge, and also supports complex analysis and long-term storage of data at the core cloud computing framework. The work mathematically characterizes fog computing in terms of power consumption, service latency, and we investigate the eco-friendliness of the technology as well.

1.1 Motivation

The primary storage and computing centers of the cloud computing architecture are the geographically scattered DCs which communicate among themselves through the DCNs. These DCNs are huge consumers of energy and, in turn, generate and emit heavy high amount of CO_2 gas. Motivated by the design of the mathematical model for cloud networks by Zhang et al. [4], in this work, we develop the network model for the fog paradigm and assess its performance while supporting the IoT. With the increase in the number of the IoT devices demanding real-time services from the service providers, tradition cloud computing framework is expected to face the following challenges:

(i) The International Data Corporation (IDC) forecast says that the worldwide market for IoT-based technologies and solutions will grow from $1.9 trillion in 2013 to $7.1 trillion in 2020 [5]. With this increase in the number IoT devices, the DCNs encounters a heavy network traffic which affects the service latency by a great extent, and consequently, applications requesting for real-time services would experience a deterioration in the QoS.

(ii) The US Environmental Protection Agency (EPA) report [6] stated that in the year of 2006, the DCs of US consumed about 61 billion kilowatt-hours of power with a total financial expenditure worth $4.5 billion. It was also observed that in 2007, 30 million worldwide servers were accounted for 100 TWh of the world's energy consumption at a cost of $9 billion which is expected to rise upto 200 TWh in the next few years [6], [7]. Therefore, it is important to exempt the cloud DCs from being bombarded with service requests, and serve a part of those requests from the network edge. This would relax the load experienced by the DCs and also would serve the latency-sensitive application requests in a better way with increased QoS.

(iii) Moreover, with the cloud DCs made to run during around the clock irrespective of the traffic rate, the amount of GHGs emitted remains unreasonably high. Presently, the planet's annual electricity consumption Information-Communications-Technologies (ICT) ecosystem is about $1,500$ TWh – equal to the total electricity produced by Japan and Germany [8]. Controlling the traffic which requires to be directed to the core cloud computing module would thereby reduce the effective up-time of the DCs. This would help in keeping down the amount of GHGs generated by the DCs and help system to be more eco-friendly.

1.2 Contribution

We discuss the contributions of our work in this subsection. As mentioned earlier, fog computing is not a substitution of cloud computing; rather in this work, we analyze the suitability of fog computing combined with the traditional cloud computing in supporting the ever-increasing demands of the latency-hungry IoT-based applications. The primary contributions of this work are listed below.

(i) Initially, this work constructs the network model of fog computing—one of the first attempts of its kind in this direction. We define the different network devices and networking links within the fog computing architecture and explain the traffic exchange pattern for the same.

(ii) Based on this model, the work mathematically characterizes the performance metrics of fog computing in terms of the service latency, power consumption, CO_2 emission for different renewable and non-renewable energy resources, and the corresponding costs incurred.

(iii) The work also performs a fair and equitable comparative study for both cloud and fog computing systems. We analyze the suitability of the fog computing architecture to support the demands of IoT devices and while serving latency-sensitive applications.

1.3 Paper Organization

The rest of the paper is organized as follows. Section 2 describes the work done so far on this domain. In Section 3, the detailed architecture of the of the fog computing paradigm is presented. We discuss the details of the fog networking model in Section 4. The performance metrics are presented and modeled in Section 5. Section 6 presents experimental setup for the case study. In Section 7, the performance evaluation of fog computing paradigm is

performed and a comparative study of both fog and cloud models is presented. Finally, the work is concluded in Section 8.

2 RELATED WORK

In this section, we present and discuss the prior research works which were done in this domain.

2.1 Internet of Things

Recent research has spawned the concept of IoT [9], [10] that connects billions of things across the globe to the Internet and enables machine to machine (M2M) communication [11] among these devices. Thus, IoT framework is a dynamic and persuasive platform for data storage, computation, and management [12]. Contemporary devices or Internet based systems are gradually converging towards IoT [13]. It is estimated and reported by Cisco that, by 2020, around 50 billion devices will be connected to the Internet [3]. Thus, by 2020, it is estimated that a large number of applications will be required to be processed and served through the technology of IoT [14], [15]. Analyzing contemporary data trends of large volume, heavy heterogeneity, and high velocity ('Big' data), it is also anticipated that a vast majority of these applications are highly latency-sensitive and require real-time processing [16], [17], [18]. Therefore, to provision the resource management and heavy computational potential to the applications, IoT leans highly on cloud computing [19], [20], [21], [22]. Consequently, the performance of IoT is profoundly dependent on the ability of cloud platforms to serve billions of devices and their applications, in real-time [12], [23].

2.2 Cloud Computing

Over the last few years, a good number of works [24], [25], [26], [27], [28], [29], [30] on cloud computing illustrate the detailed underlying process behind the provisioning of cloud services. The process of complete virtualization of cloud services involves several cloud DCs, dispersed across multiple geographical locations. Cloud systems are, therefore, DCN-centric and for every user-request, service provisioning involves one or more DCNs. In [31], Xiao et al. addressed the problem of design and optimal positioning of DCs to improve the QoS in terms of service latency and cost efficiency. However, the work is strongly affected by the efficiency of the DCNs. In another work, Chen et al. [32] focused on the problem of latency for video streaming services. The work suggests the usage of a single DC under a single CSP. However, the situation might be hypothetical as in real-life scenarios of IoT, a single DC under a single global CSP may hinder the overall service efficiency due to lack of proper management and shortage of cloud storage. Tziritas et al. [33] addressed process migration to improve the performance of cloud systems and demonstrated experimental results with $1,000$ processes. However, IoT concerns billions of processes and in such a scenario, process migration within DCs might be of overhead degrading the performance. Similarly, job scheduling techniques to improve QoS were also compared by Chandio et al. [34] using $22,385$ jobs. However, compared to IoT systems,

the count is too low less to be considered. Other scheduling techniques that focus on real-time workload scheduling [35] or energy-efficient scheduling [36] have also worked with low scale scenarios comprising of a maximum of 128 Virtual Machine instances and $1,000$ cloud servers, respectively.

For each of the above works, the DCs form the hub of computing and the DCNs are invoked everytime an application makes a service request. Therefore, with the increase in the number of IoT consumers and with every request being required to be processed within the DCs, it is likely that the cloud DCNs will encounter a serious difficulty in serving the IoT applications real-time. Additionally, with the increase in the number of latency-sensitive applications, the efficiency of service provisioning will also reduce to a significant extent. Moreover, considering the contemporary state of our environment, it is observed and reported [37], [38], [39], [40] that as we are more and more advancing towards technology, we are driving our nature to an alarming state. Therefore, it is imperative to simultaneously maintain the eco-friendliness of our surroundings.

2.3 Fog Computing

The contemporary trends in data volume, velocity, and variety and the limitations of cloud computing make it easy to speculate the need to propose new techniques of data management and administration. In this context, Cisco proposed the revolutionary concept of *fog* computing [3], [41]. Fog computing is defined as a distributed computing infrastructure that is able to handle billions of Internet-connected devices. The underlying principle of the technology is *edge computing* in which the services are hosted within the edge devices inclusive of the gateways, routers, and access points. Bonomi et al. [41] explicates the different architectural components of the fog computing paradigm and illustrated few of its real-life applications. The authors also highlighted the importance of fog-cloud interplay and the role of fog computing in the context of IoT. In [42], the authors defined the characteristics of the paradigm in terms of latency, location awareness, geographical distribution, mobility, heterogeneity, and the predominant access to wireless devices. However, these works mostly center around the theory and dogma associated with fog computing. Hong et al. [43] designed a programming model to support large-scale IoT applications through mobile fog computing. The model supports the service provisioning to geographically scattered, latency-sensitive applications.

Recent researches, however, have revealed some of the important aspects of fog computing. The importance and applicability of fog computing were assessed by Yannuzzi et al. [44] and Preden et al. [45] at a superficial level. In [46], the authors considered various computing paradigms inclusive of cloud computing, and investigated the feasibility of building up a reliable and fault-tolerant fog computing platform. Do et al. [47] and Aazam and Huh [48], [49] have inspected the different intricacies of resource allocation in a fog computing framework. Of late, few researchers have also explored the security and privacy aspects of fog computing [50], [51], [52]. However, most of the works on fog computing have primarily
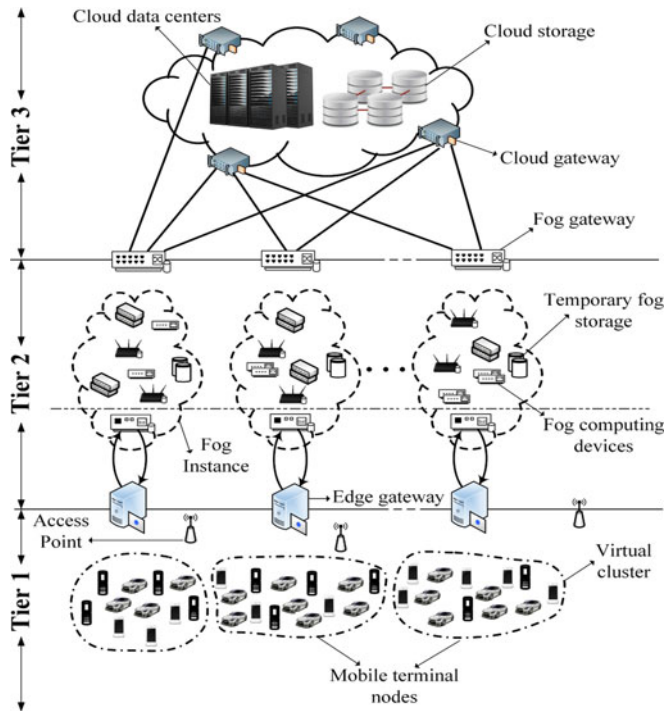
Fig. 1. Fog computing architecture.

focused on the principles, basic notions, and the doctrines of it. Not many works have contributed in the technical aspect of the paradigm in terms of an implementation perspective.

2.4 Focus of the Paper

This work studies the suitability and applicability of fog computing as a potential platform to support IoT. The novelty of the paper is to model the paradigm of fog computing and perform a comparative study in terms of power consumption, cost and latency with respect to cloud systems. The work further investigates the eco-friendly aspects of the paradigm to judge its appropriateness to serve the world of Internet connected devices.

3 FOG COMPUTING ARCHITECTURE

In this section, we present the fog computing architecture and its minute details. It is imperative to mention that fog computing is a non-trivial extension of cloud computing and extends the services of cloud to the network edge.

3.1 Assumptions

Fog paradigm is still in its early stage of research and is yet to shape up. We, therefore, draw few simple, yet realistic assumptions with justification for the same.

- Every terminal node (TN) is aware of its absolute geo-spatial location and is able to share the information through technologies such as GPS, GIS, or GNSS so that location-based services are provided to the TNs in real-time based on region-specific data analysis.
- The devices of the fog computing tier of are "intelligent" in terms of their computational and storage ability [41], [43]. Apart from their typical functionalities (routing or packet forwarding), these devices are capable of dynamic decision-making on whether an application request can be served within the fog tier or it needs to be forwarded to the cloud computing core for processing.

- The networking devices of the fog computing layer, as shown in Fig. 2, are self-adjusting to dynamic load sharing in terms of the network, computational, and storage load among themselves.
- Every fog computing devices can support the mobility of the TNs. As the Internet-connected devices are highly mobile in nature, intra-fog tier communication and information transfer is necessary to provided continued service in real-time to the TNs without any disruptions.

3.2 System Outline

This subsection illustrates the distinct tiers of a generic fog computing architecture. As depicted in Fig. 1, it is essentially a three tier architecture. The tiers are discussed below.

(a) *Tier 1*. This is the bottom-most tier of the architecture. The tier comprises of several TNs. The TNs are majorly smart, wireless sensor nodes that sense heterogeneous physical parameters and transmit the same to the immediate upper tier.

(b) *Tier 2*. The tier 2 or the middle layer is also known as the fog computing layer. The primary components of this tier are intelligent intermediate devices (such as routers, gateways, switches, and access points) that possess the ability of data storage, computation, routing, and packet forwarding.

(c) *Tier 3*. The uppermost tier is commonly known as the cloud computing tier. Several high-end servers and DCs comprise this tier.

3.3 Architecture Details

The principle of fog computing architecture is based on edge computing. As already discussed, the bottom-most tier comprising of smart Internet connected TNs are one of the fundamental components of IoT. The TNs are assumed to form location-based logical clusters, which are termed as virtual clusters (VCs). The VCs together form an edge virtual private network (EVPN) that transmits data to multiple fog instances (FIs). An FI is conceptualized specific to a geographic location. The mobility of a TN makes the mapping of a TN to an FI flexible and non-static. While the data are transmitted upwards (towards the fog tier) they are processed within the intermediate fog devices.

The fog devices can stretch from different networking components, such as routers, switches, gateways, and access points to high-end proxy servers and computing machines. As proposed by Bonomi et al., [41] the fog computing architecture can be classified into two sub-parts, viz., (a) the fog abstraction layer and (b) the fog orchestration layer. While the former manages the fog resources, enables virtualization, and preserves tenant-privacy, the latter beholds the exclusive fog properties. The fog orchestration layer comprises of a small software agent—foglet which monitors the state of the devices, a distributed database to
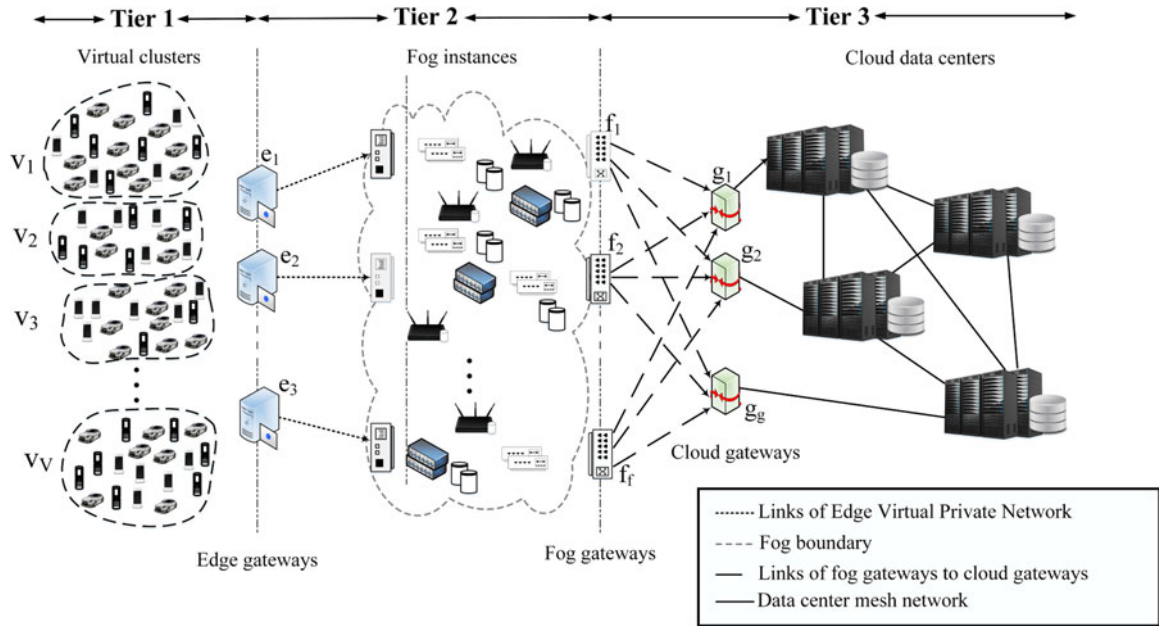
Fig. 2. Networking links and components of fog computing.

account for scalability and fault tolerance, and a service orchestration module which is responsible for policy-based routing of application requests. Within the FIs, the data are processed and analyzed to decide whether it needs to be transmitted to the cloud DCs. Application requests which require storage or historical data based analytics are redirected to the cloud, else, the data are processed within the fog units. The fog devices possess limited semi-permanent storage that allow temporary data storage and serve the latency-sensitive applications in real-time.

The cloud computing tier is commonly responsible for permanent storage of huge, voluminous data chunks within its powerful DCs. The DCs are equipped with massive computational ability. However, unlike conventional cloud architecture, the core cloud DCs are not bombarded for every single query. Fog computing enables the cloud tier to be accessed and utilized in an efficient and controlled manner.

4 NETWORKING MODEL

Motivated by the works of Zhang et al. [4], we develop the network model of the fog computing paradigm. Analogous to the above work, the proposed work designs the metric of fog network model. Any cloud based analytics are excerpted from the above cited work, especially while performing a comparative study of cloud versus fog in Section 7. However, all fog based evaluations are based on the metrics designed in this work.

In this section, we define the functional components and the data transfer links associated with the fog computing paradigm. The set of N geo-spatially distributed mobile TNs is denoted by $\mathcal{N}$, where $|\mathcal{N}| = N$. At time instant t, we consider these N TNs to be grouped under different VCs. The set of all VCs is denoted by $\mathcal{V}$, such that $\mathcal{V} = \{v_i | i \in [1, V], i \in \mathbb{I}\}$ and $|\mathcal{V}| = V$, where V is the total number of VCs present in the system and v_i denotes the ith VC. The total number of TNs mapped to v_i is denoted by n_i,

$\forall i = 1(1)V$. As total coverage of the TNs by the VCs is assumed for any instant of time, we have:

$$\sum_{i=1}^{V} |v_i| = \sum_{i=1}^{V} n_i = N. \tag{1}$$

The data generated by all the constituent TNs within a VC are transported through the edge gateways towards the fog computing tier. The set of all edge gateways present at the lowest tier in the computing architecture is denoted by $\mathcal{E}$, with $|\mathcal{E}| = E$. The data transport links between the VCs and the edge gateways are considered to be of high bandwidth. However, the data transmission bandwidth between the EVPNs and the FIs are considered to be restricted. The set of all fog instances present in the system at any given time is given by $\mathcal{F}$, with $|\mathcal{F}| = F$. All data and queries generated from the applications instances running with the TNs in tier 1 are forwarded through the (e, f) link, i.e., through the link between the EVPN e and the FI f, $\forall e \in \mathcal{E}$ and $\forall f \in \mathcal{F}$.

Let $P_r^{v_i}(t)$ and $P_s^{v_i}(t)$ be the total amount of data, in bytes, that are generated from the VC v_i in time-slot t which demand to be served and stored, respectively. Note that, based on the type of request a stream of bytes are forwarded to the cloud computing tier. If a request demands real-time services, it is processed and served from within the FI without the intervention of the cloud computing core. However, the requests which require intervention of the cloud computing layer for analysis based on historical data-sets and for long-term (semi-permanent or permanent) storage, are redirected to the upper tier after each time-slot. Let $Q_r^{v_i}(t)$ and $Q_s^{v_i}(t)$ be the total number of bytes generated from the VS v_i which is redirected to the cloud computing layer for computation and storage purposes, respectively, in the tth time-slot. Clearly, $P_r^{v_i}(t) \leq Q_r^{v_i}(t)$ and $P_s^{v_i}(t) \leq Q_s^{v_i}(t), \forall i = 1(1)V$.

The fog gateways, located between the fog computing tier and the cloud computing tier, are represented by the

set $\mathcal{G}$, where $|\mathcal{G}| = G$. Also, $\forall f \in \mathcal{F}$ and $\forall g \in \mathcal{G}$, the data communication link (f, g), between the FI f and the fog gateway g is also bandwidth constrained. Finally, we discuss the data communication and aggregation involving the cloud computing framework. The byte-stream transmitted by the fog gateways reaches a cloud DC through a channel of limited bandwidth. The set of all DCs at the cloud-end is represented by $\mathcal{K}$, where $|\mathcal{K}| = K$. $\forall g \in \mathcal{G}$ and $\forall c \in \mathcal{K}$, (g, c) the communication link between the cloud gateway g and the DC c is denoted by (g, c) which is the route through which data reach the cloud computing tier from the fog computing tier. However, we assume that at the cloud-end not all DCs are responsible data aggregation at every instant of time. We consider that at any given time instant, data aggregation and subsequent analysis take place only in one DC. Data migration between the cloud DCs takes place through high bandwidth links.

5 MODELING THE PERFORMANCE METRICS

We define the fog tier routing variable (FTRV) as $X^{fog}_{v_i,e,f}(t)$, $\forall v_i \in \mathcal{V}, \forall e \in \mathcal{E}, \forall f \in \mathcal{F}$, which indicates the uplink route through which the data generated from the VC v_i in time-slot t reaches the FI f for real-time processing and temporal storage. However, for application instances which require to be referred to the cloud framework for aggregation, historical analysis, and long-term storage, the cloud tier routing variable (CTRV) is defined as $X^{cld}_{f,g,c,d}(t)$, $\forall f \in \mathcal{F}, \forall g \in \mathcal{G}, \forall c, d \in \mathcal{D}$, which denotes the route through which data from the FI f reaches its destination DC in time-slot t. For the FTRV, the route $v_i \rightarrow e \rightarrow f$ represents the path along which the data originated from v_i moves to the FI f through the intermediate edge gateway e. Similarly, for the CTRV, $f \rightarrow g \rightarrow c \rightarrow d$ indicates that the data-stream is redirected from the FI f to the cloud DC c through the intermediate fog gateway g, and from there it is once again transferred to the DC d for aggregation and further processing.

For any given route, its corresponding FTRV or CTRV value is set as equal to the proportion of the total data (in bytes) generated in time-slot t which traverse through the route. Clearly, if a given route for data transmission is valid (at least some proportion of the total data generated traverses via the route), the corresponding value of the FTRV or the CTRV is set as non-zero, i.e., $X^{fog}_{v_i,e,f}(t) > \lambda$ or $X^{cld}_{f,g,c,d}(t) > \lambda$, accordingly, and is set to zero, otherwise. λ denotes the proportion of the data which traversed through the concerned route, and $\lambda \in (0, 1]$. Clearly, $\sum_{v_i \in \mathcal{V}, e \in \mathcal{E}, f \in \mathcal{F}} X^{fog}_{v_i,e,f}(t) = 1$ implies that every byte of data originated from v_i reaches f without any loss, whereas, $\sum_{v_i \in f \in \mathcal{F}, g \in \mathcal{G}, c, d \in \mathcal{D}} X^{cld}_{f,g,c,d}(t) = 1$ indicates that the amount of data redirected towards the DC d by the FI f has reached its destination without any information loss. At any given time instant t, the set of all feasible FTRVs ($\mathcal{X}^{fog}$) is expressed as

$$\mathcal{X}^{fog} = \Big\{ X^{fog}_{v_i,e,f}(t) \mid X^{fog}_{v_i,e,f}(t) = [0, 1] \text{ and} \atop \sum_{v_i \in \mathcal{V}, e \in \mathcal{E}, f \in \mathcal{F}} X^{fog}_{v_i,e,f}(t) = 1, \forall v_i \in \mathcal{V}, \forall e \in \mathcal{E}, \forall f \in \mathcal{F} \Big\}. \quad (2)$$

Similarly, at time t, the set of feasible CTRVs ($\mathcal{X}^{cld}$) is given by

$$\mathcal{X}^{cld} = \Big\{ X^{cld}_{f,g,c,d}(t) \mid X^{cld}_{f,g,c,d}(t) = [0, 1] \text{ and} \atop \sum_{v_i \in f \in \mathcal{F}, g \in \mathcal{G}, c, d \in \mathcal{D}} X^{cld}_{f,g,c,d}(t) = 1, \atop \forall f \in \mathcal{F}, \forall g \in \mathcal{G}, \forall c, d \in \mathcal{D} \Big\}. \quad (3)$$

5.1 Power Consumption

The total power consumption is divided into three broad categories for application requests which are served by the fog computing tier without inference of the cloud framework, and into four broad categories for requests which are required to be served by the cloud computing tier. Following is the list of factors which are responsible for power consumption during big data handling using the fog computing framework.

5.1.1 Data Forwarding

While forwarding of data packets, the overall power consumed due to reception of the byte-stream, initial processing required for routing, and subsequent transmission of the same, is categorized as *power consumption due to data forwarding*. For data packets which require real-time, low-latency services, and are processed at the fog computing tier, and the corresponding power consumption due to data forwarding $(\Psi^{fog}_{df}(t))^{[1]}$ at time t is computed as

$$\Psi^{fog}_{df}(t) = (\gamma_{eg} + \gamma_{fi}) \left[\sum_{i=1}^{V} \left\{ P^{v_i}_r(t) - Q^{v_i}_r(t) + P^{v_i}_s(t) \right. \right.$$
$$\left. \left. - Q^{v_i}_s(t) \right\} \sum_{v_i,e,f} X^{fog}_{v_i,e,f}(t) \right]$$
$$= (\gamma_{eg} + \gamma_{fi}) \left[\sum_{v_i,e,f} \left\{ P^{v_i}_r(t) - Q^{v_i}_r(t) + P^{v_i}_s(t) \right. \right.$$
$$\left. \left. - Q^{v_i}_s(t) \right\} X^{fog}_{v_i,e,f}(t) \right], \quad (4)$$

where γ_{eg} and γ_{fi} represent the amount of energy required per second (power) to forward unit byte of data by the edge gateways and the fog instances, respectively.

Similarly, for data packets which demand to be processed at the core cloud computing module for complex analysis and long-term storage, the corresponding power consumption required for forwarding of the data packets $(\Psi^{cld}_{df}(t))^{[2]}$ at time t is expressed as

$$\Psi^{cld}_{df}(t) = (\gamma_{eg} + \gamma_{fi}) \left[\sum_{i=1}^{V} \left\{ Q^{v_i}_r(t) + Q^{v_i}_s(t) \right\} \sum_{v_i,e,f} X^{fog}_{v_i,e,f}(t) \right]$$
$$+ \gamma_{cl} \left[\sum_{i=1}^{V} \left\{ Q^{v_i}_r(t) + Q^{v_i}_s(t) \right\} \sum_{f,g,c,d} X^{cld}_{f,g,c,d}(t) \right]$$
$$= \sum_{i=1}^{V} \left\{ Q^{v_i}_r(t) + Q^{v_i}_s(t) \right\} \left[(\gamma_{eg} + \gamma_{fi}) \sum_{v_i,e,f} X^{fog}_{v_i,e,f}(t) \right.$$
$$\left. + \gamma_{cl} \sum_{f,g,c,d} X^{cld}_{f,g,c,d}(t) \right], \quad (5)$$

---

1. The notation $\sum_{v_i,e,f}$, as used in the paper essentially means the same as the notation $\sum_{v_i \in \mathcal{V}, e \in \mathcal{E}, f \in \mathcal{F}}$.

2. The notation $\sum_{f,g,c,d}$ is used as an alternative for the notation $\sum_{v_i \in f \in \mathcal{F}, g \in \mathcal{G}, c, d \in \mathcal{D}}$ in the manuscript.

where γ_{cl} is the power required to forward unit byte of data by a cloud gateway.

5.1.2 Computation

The power consumption due to computation also occur at both fog tier and cloud tier. At the fog computing layer, these computations are mostly real-time, low-latency services which are required to meet the continuous and ever-increasing demand of the TNs, otherwise stated as the IoE devices. Computation and analysis at the fog tier involves the temporarily stored data-sets within the fog computing devices. Let τ denote the time-to-live for every data packet after which it is removed from the temporary fog storage. Therefore, at time t, the computational power consumption at the fog layer ($\Psi_{cp}^{fog}(t)$) depends on the data stored within the FIs from time $(t - \tau)$ till the present time-slot. ($\Psi_{cp}^{fog}(t)$) is mathematically expressed as

$$\Psi_{cp}^{fog}(t) = \beta^{fog} \sum_{j=t-\tau}^{t} \phi_j^{fog} \sum_{i=1}^{V} \{P_s^{v_i}(j) - Q_s^{v_i}(j)\}, \qquad (6)$$

where β^{fog} is the average per byte computational power consumption (obtained as the ratio of the power exhausted for the processing of an instruction to the number of bytes in the instruction). ϕ_j^{fog} is the weight-factor associated with the data-set which is required for analysis, and the magnitude of ϕ_j^{fog} decreases with the increase in the *age* of the data. The magnitude of ϕ_j^{fog} lies within $(0,1]$, with $phi_j^{fog} = 1$ for $j = t$. The expression $\sum_{i=1}^{V}\{P_s^{v_i}(j) - Q_s^{v_i}(j)\}$ indicates the cumulative amount of data which is stored at time t within the temporary fog storage units, for analysis and computation.

For the cloud computing tier, computations and subsequent analysis are highly extensive in terms of the volume of data involved and the complexity associated. At time instant t, the power consumption at the cloud computing framework due to computation and analysis, $\Psi_{cp}^{cld}(t)$, is dependent on the cumulative amount of data which stored within the DCs, starting from the very beginning, i.e., $t = 0$. ($\Psi_{cp}^{cld}(t)$) is computed as

$$\Psi_{cp}^{cld}(t) = \beta^{cld} \sum_{j=0}^{t} \phi_j^{cld} \sum_{i=1}^{V} Q_s^{v_i}(j), \qquad (7)$$

where β^{cld} id the mean computational power required to process unit byte at the cloud-end, and $\sum_{i=1}^{V} Q_s^{v_i}(j)$ is the total amount of data aggregated within a DC for processing and computation, with $\phi_j^{cld} \in [0,1]$. Clearly, for $j = t$, $\phi_j^{cld} = 1$, whereas, as $j \to 0$, $\phi_j^{cld} \to 0$.

5.1.3 Storage

The power consumption due to storage, similar to the computational power consumption, depends on the number of bytes of data which is stored in the database for processing and analysis. However, as the storage power consumption is independent of the *age* of the data, unlike the computational power, this term is void of any such weight-

factors. The mathematical expression for the storage power consumption at time t for the fog tier ($\Psi_{st}^{fog}(t)$) is given by:

$$\Psi_{st}^{fog}(t) = \alpha^{fog} \sum_{j=0}^{t} \sum_{i=1}^{V} \{P_s^{v_i}(j) - Q_s^{v_i}(j)\}. \qquad (8)$$

Similarly, for the cloud computing tier, the storage power consumption ($\Psi_{st}^{cld}(t)$) is computed as:

$$\Psi_{st}^{cld}(t) = \alpha^{cld} \sum_{j=0}^{t} \sum_{i=1}^{V} Q_s^{v_i}(j), \qquad (9)$$

where α^{fog} and α^{cld} represent the per byte per unit time energy consumption to storage data within the databases in the fog tier and the cloud DC, respectively.

5.1.4 Data Migration

The power consumption due to data migration is only associated with the cloud computing layer. In other words, applications which require real-time services data migration is irrelevant as they are served by the fog computing tier without the intervention of the cloud computing framework. However, for applications which demand complex and historical analysis of data, migration of data from different geo-spatially distributed DCs towards the aggregator DC is required. As the aggregator DC may vary in every time-slot, at any given time-slot t, it is important to migrate all data to the aggregator DC for subsequent processing. At time t, the overall migration cost ($\Psi_{mg}^{cld}(t)$) within the cloud computing framework is given by

$$\Psi_{mg}^{cld}(t) = \begin{cases} \sum_{c \in \mathcal{D}} \sum_{d \in \mathcal{D}} \eta_{cd} \sum_{j=0}^{t-1} \phi_j^{cld} \sum_{i=1}^{V} Q_s^{v_i}(j), & \text{if } \mathcal{A}_t \neq \mathcal{A}_{t-1} \\ 0, & \text{otherwise,} \end{cases}$$
$$(10)$$

where η_{cd} is the per byte power consumption for cost migration of data from DC c to DC d, $c, d \in \mathcal{D}$. $\mathcal{A}_t$ denotes the aggregator DC at time-slot t. Therefore, $\mathcal{A}_t = \mathcal{A}_{t-1}$ implies that the aggregator DC in time t is same as it was in the previous time-slot, indicating no additional data migration is required.

Therefore, at time t, the overall power consumption for applications which are served by the fog computing tier, $\Psi^{fog}(t)$ is computed as

$$\Psi^{fog}(t) = \Psi_{df}^{fog}(t) + \Psi_{cp}^{fog}(t) + \Psi_{st}^{fog}(t). \qquad (11)$$

Whereas, for application seeking intervention of the core cloud computing, the cumulative power consumption at time t is calculated as

$$\Psi^{cld}(t) = \Psi_{df}^{cld}(t) + \Psi_{cp}^{cld}(t) + \Psi_{st}^{cld}(t) + \Psi_{mg}^{cld}(t). \qquad (12)$$

5.2 Service Latency

Service latency for a request sent by an application instance running within a TN is basically its response time, and is computed as the sum of the transmission latency and the processing latency for the request. As mentioned earlier, the communication among the different TNs within a EVPN incur insignificant latency as sufficient bandwidth-support

is provided within every EVPN for the purpose. Similarly, inter-DC communication and data migration, which take place along the (c, d) link are also considered to invoke negligible latency due to the high-bandwidth, $\mathbb{W}(c, d)$, dedicated available within the cloud-core, $\forall c, d \in \mathcal{D}$ [4]. However, the bottleneck of communication is formed by the links (e, f) and (f, g), $\forall e \in \mathcal{E}, \forall f \in \mathcal{F}, \forall g \in \mathcal{G}$. The bandwidth between the edge gateways and the FIs, $\mathbb{W}(e, f)$ is limited, and at any given time instant, as the active number of TNs in tier 1 increases, the service latency also increases. The communication links between fog gateways and the cloud gateways are bandwidth constrained ($\mathbb{W}(f, g)$) as well. In this Section we discuss the service latency for the applications, and mathematically formulate the expression for the same. The service latency is divided into two subheads as mentioned below.

5.2.1 Transmission Latency

Let δ_{ef} and δ_{fg} are the delays in unit byte data transmission from a EVPN to the corresponding FI, and from an fog gateway to the cloud gateway, respectively. For applications which are served by the cloud computing tier, the transmission latency at time t, ($\delta_{tr}^{fog}(t)$), is computed as

$$\delta_{tr}^{fog}(t) = \delta_{ef} \sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t) - Q_r^{v_i}(t) - Q_s^{v_i}(t)\}. \quad (13)$$

Similarly, for requests which are handled at the cloud computing tier, the corresponding transmission latency ($\delta_{tr}^{cld}(t)$) is expressed as

$$\delta_{tr}^{cld}(t) = (\delta_{ef} + \delta_{fg}) \sum_{i=1}^{V} \{Q_r^{v_i}(t) + Q_s^{v_i}(t)\}. \quad (14)$$

Therefore, in presence of the fog computing tier, the mean transmission latency at time t (Δ_{tr}^{fog}) is computed as

$$\Delta_{tr}^{fog}(t) = \frac{\delta_{ef} \sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\} + \delta_{fg} \sum_{i=1}^{V} \{Q_r^{v_i}(t) + Q_s^{v_i}(t)\}}{\sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\}}. \quad (15)$$

On the contrary, in the traditional cloud computing framework, the corresponding mean transmission latency ($\Delta_{tr}^{cld}(t)$) is given as

$$\Delta_{tr}^{cld}(t) = \frac{\delta_{eg} \sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\}}{\sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\}}, \quad (16)$$

where δ_{eg} is the latency associated with unit byte data transmission from a TN to the cloud DC. Clearly, from the triangle inequality, we have $\delta_{eg} \geq \delta_{ef} + \delta_{fg}$.

5.2.2 Processing Latency

Processing latency for an application instance request within the fog computing tier is defined as the time required to serve the request after analyzing the data accumulated during the previous τ time-slots within the fog computing devices. Mathematically, at time t, the processing latency, $\delta_{pr}^{fog}(t)$, within the fog computing tier is expressed as

$$\delta_{pr}^{fog}(t) = \left(P_r^{v_i}(t) - Q_r^{v_i}(t)\right)\zeta^{fog} \sum_{j=t-\tau}^{t} \phi_j^{fog} \sum_{i=1}^{V} \{P_s^{v_i}(j) - Q_s^{v_i}(j)\}. \quad (17)$$

Similarly, for applications which are referred to the cloud computing layer for processing purpose, the processing latency is defined in terms of the time required to analyze the cumulative migrated data-sets from the cloud DCs within the aggregator DC. The processing latency ($\delta_{pr}^{cld}(t)$) within the cloud computing core is, therefore, computed as

$$\delta_{pr}^{cld}(t) = Q_r^{v_i}(t)\zeta^{cld} \sum_{j=t-\tau}^{t} \phi_j^{cld} \sum_{i=1}^{V} Q_s^{v_i}(j), \quad (18)$$

where ζ^{fog} and ζ^{cld} represent the per byte processing latency at the fog computing and cloud computing tiers, respectively. The mean processing delay at time t for a fog computing environment is

$$\Delta_{pr}^{cld}(t) = \left[\left(P_r^{v_i}(t) - Q_r^{v_i}(t)\right)\zeta^{fog} \sum_{j=t-\tau}^{t} \phi_j^{fog} \sum_{i=1}^{V} \{P_s^{v_i}(j) - Q_s^{v_i}(j)\}\right.$$
$$\left. + Q_r^{v_i}(t)\zeta^{cld} \sum_{j=t-\tau}^{t} \phi_j^{cld} \sum_{i=1}^{V} Q_s^{v_i}(j)\right] \Big/ \sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\}. \quad (19)$$

However, the average transmission latency, $\Delta_{pr}^{cld}(t)$, at time t for a traditional cloud computing framework is given as

$$\Delta_{pr}^{cld}(t) = \frac{P_r^{v_i}(t)\zeta^{cld} \sum_{j=t-\tau}^{t} \phi_j^{cld} \sum_{i=1}^{V} P_s^{v_i}(j)}{\sum_{i=1}^{V} \{P_r^{v_i}(t) + P_s^{v_i}(t)\}}. \quad (20)$$

Finally, to compute the mean service latency, we simply add the mean transmission latency and the mean processing latency for the corresponding computing framework.

6 CASE STUDY: SIMULATION SETUP

This Section illustrates the network setup, power consumption, CO_2 emission rate, and the cost variables corresponding to the fog computing framework.

6.1 Network Topology

The essential nodes of the fog computing network includes the sets of the TNs $\mathcal{N}$, the FIs $\mathcal{F}$, and the cloud DCs $\mathcal{D}$. We consider a global deployment of these essential nodes. For this purpose, the work considers 100 highest populated cities across the globe [53], the respective population of people using Internet services [54], and the corresponding geographic location of the cities [55], as shown in Fig. 3. The matrix $L_c[1..100][1..100]$ stores the relative Euclidean distance between any pair of cities. The TNs within a particular city are logically grouped to form a VC.

6.2 Network Traffic

The data traffic generated from the cities (or VCs) is proportional to the population of Internet users of the corresponding city. Data from the EVPNs are transmitted to the fog computing tier in form of packets. These packet sizes
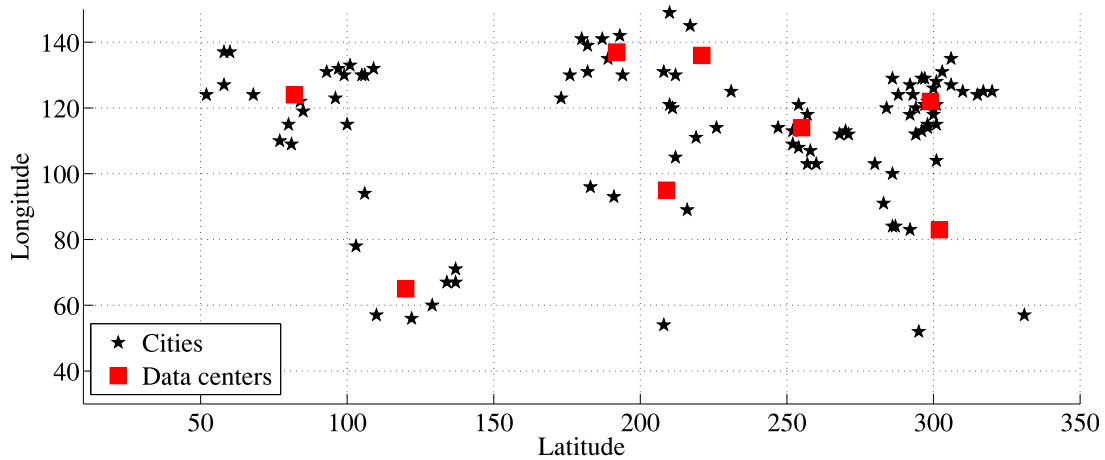
Fig. 3. Global deployment scenario.

typically vary between a minimum of 34 bytes (header + FCS only) to a maximum of 65,550 bytes. The instruction size is taken as 64 bits. The packet arrival is considered to follow a Poisson distribution with the mean packet arrival rate being 1 packet per node per second. Communication among the TNs within a EVPN and that among the DCs within the cloud computing core are assumed to take place through bandwidth unconstrained channels, as mentioned earlier. However, the capacity of the (e, f) links, $\forall e \in \mathcal{E}, \forall f \in \mathcal{F}$ is considered to be 1 Gbps. On the other hand, the (f, g), $\forall f \in \mathcal{F}, \forall g \in \mathcal{G}$ link capacity is taken as 10 Gbps.

6.3 Terminal Nodes

The total number of TNs in the system is treated as a variable, within the range [10,000, 100,000], to assess the system performance against varied network conditions. The number of TNs in each of the 100 cities is taken as proportional to the population of Internet users of the city. The TNs transmit their data through access points distributed within each city. All TNs within a city are considered to send their data to a single FI.

6.4 Data Centers

The number of DCs present worldwide is considered to be 8. Based on the clustering of the city-population, the location of the DCs are determined. The pair-wise Euclidean distances between the DCs are stored in the matrix $L_d[1..8][1..8]$. Every DC is assumed to accommodate varied number of IT components within the discrete set {16,000, 32,000, 64,000, 128,000}, based on the network traffic is processes.

6.5 Power Consumption

The power consumed by the different network, computing, and storage elements are used to compute the overall power consumption by the system under the discussed scenario. The power consumed by each 1 and 10 Gbps router is taken as 20 and 40 W, respectively [56]. Each three-layer network switch and storage component consumes 350 and 600 W, respectively. The fog computing devices, within an FI, consumes a total of 3.7 W collectively. Power consumption by

the cloud DCs are taken to be proportional to the IT elements within those, and taken from the range {9.7, 19.4, 38.7, 77.4} MW.

6.6 CO_2 Emission

The sources of the energy supplied to the cloud DCs determines the amount the CO_2 gas produced. Table 1 depicts the CO_2 emission rate for the different energy sources taken for analysis [57]:

6.7 Cost

For analysis of the cost both in terms of the operational cost and the CO_2 emission penalty, we define the cost of the different component elements. Each 1 and 10 Gbps router port costs \$50/year. The Cost for each server is taken as \$4,000/year [56]. The uploading cost for each byte of data is taken as \$ 12, and the storage cost per GB of data is taken uniformly from within the range $0.45-0.55$ \$/hour. The electricity cost is taken as uniformly distributed between \$30/MHWh and \$70/MWh [56]. The penalty for CO_2 emission, however, was taken in higher degree. The penalty was decided to be \$1,000/ton CO_2 emitted.

7 CASE STUDY: PERFORMANCE EVALUATION

In details of the results obtained in terms of the performance metrics are presented and analyzed in this Section. The simulation setup is mentioned in details in Section 6. Additionally, we compare the performance of the fog computing paradigm wit that of the traditional cloud computing architecture, and present a thorough study against the same.

TABLE 1
CO_2 Emission Rates

Type of energy source	Energy source	CO_2 emission rate (in g/kWh)
Non-renewable	Coal	960
	Diesel	778
	Natural gas	443
Renewable	Geothermal	38
	Hydroelectric	10
	Wind	9

(a) Tranamission latency vs No. of TNs (b) Processing latency vs No. of TNs (c) Service latency vs No. of TNs
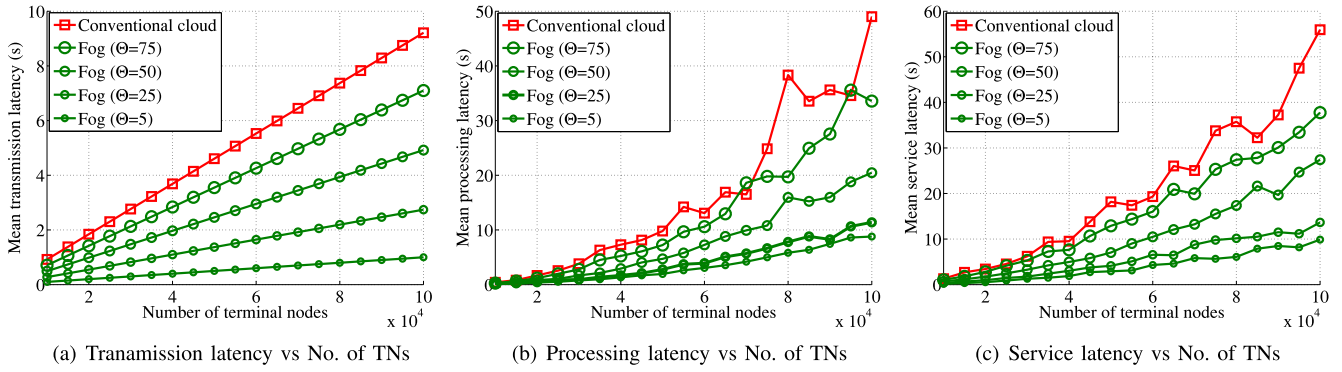
Fig. 4. Analysis of service latency.

7.1 Service Latency

First, we analyze the variation of the service latency (transmission latency + processing latency) with the number of TNs present in tier 1. We define the ratio of the total bytes transmitted to the fog computing tier to the number of bytes referred to the cloud computing core as the cloud transmission ratio, mathematically represented by the variable Θ. With the change in the magnitude of Θ, within the range $[0.05, 0.75]$, we plot the transmission latency and propagation latency for both fog computing and the conventional cloud computing architectures, and observe the change in the corresponding services latencues.

In Figs. 4a and 4b, the mean transmission latency and mean processing latency are plotted, separately, against variable number of TNs, for different magnitudes of Θ. It is observed that with the decrease in the magnitude of Θ, as more number of application requests demand real-time and latency-sensitive services, the mean transmission latency and the mean processing latency are observed to diminish. Also, with the increase in the number of TNs, the tranmission and processing latencies increase.

Consequently, the overall service latency for both these computing paradigm follow the same pattern. In summary, for a very low percentage of applications which demand real-time services, i.e., for a very high magnitude of Θ ($\Theta \simeq 100$), the service latency in fog computing becomes almost same with that in a cloud computing environment.

7.2 Power Consumption

The power consumption for due to the individual effects of data forwarding, computation, and data storage, and due to

their collective effect is analyzed in this subsection. As shown in Fig. 5a, although with the increase in the number of TNs in the lowest tier, the mean power consumption due to data forwarding increases linearly, the impact of the change in the power consumption for different magnitudes of Θ is observed to be very low. However, compared to a conventional cloud computing framework, this mean power consumption was significantly less.

In Figs. 5b and 5c, the variation in the mean power consumption due to computation and storage, respectively, are shown against the change in the number of TNs. Similar inferences are drawn from the two figures, as the power consumption in both these cases are noticed to decrease by a significant margin as the magnitude of Θ is decreased.

In presence of the fog computing tier, the mean cumulative power consumption, as shown in Fig. 6, is always less than that in a conventional cloud computing. We observe that compared to the power consumption due to data forwarding, the amount of power consumed due to computation and storage are considerably high. For $\Theta = 75$, the overall mean power consumption is calculated to be 42.2 percent less in fog computing.

7.3 CO_2 Emission

The impact of power consumption on the environment is depicts through the amount of CO_2 gas emission in the process. As mentioned in Section 6, we divide our analysis based on the type of the energy source—non-renewable and renewable. For analysis of CO_2 emission in both cloud and fog systems, the predominant factor that we have assumed is the power consumption of the DCs in each

(a) Power consumption due to data forwarding (b) Power consumption due to computation (c) Power consumption due to data storage
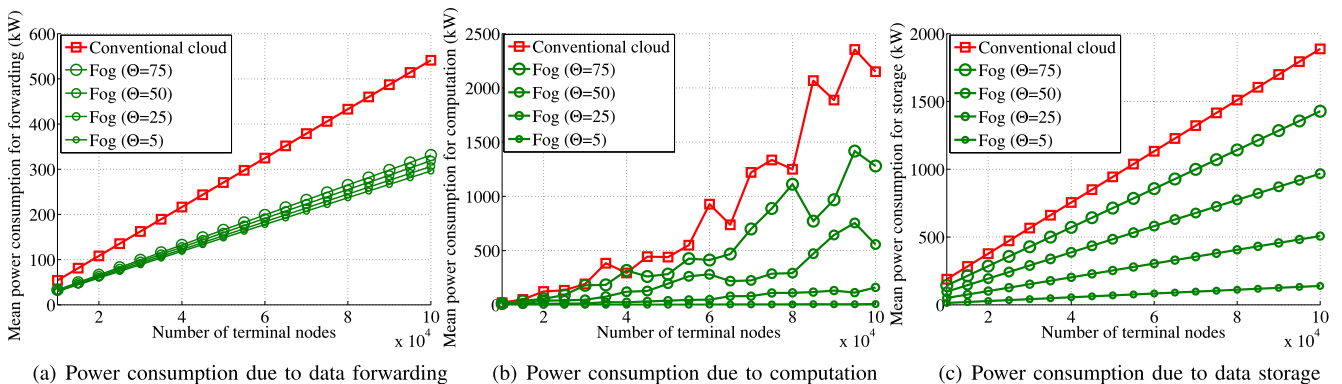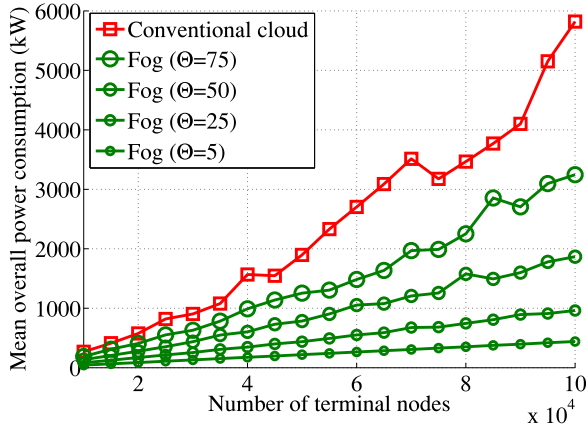
Fig. 5. Analysis of power consumption.

Fig. 6. Overall power consumption versus No. of TNs.

case. The experiment considers distinct multiplicative factors for various non-renewable and renewable energy sources (Table 1).

In Fig. 7, variation in the mean CO_2 emission for each of the individual factors, viz., data forwarding, computation, and data storage is plotted against variable number of TNs. As indicated in Fig. 7a, the average CO_2 emission for packet forwarding is notably high in cloud computing paradigm compared to fog computing paradigm when coal, diesel, and natural gas are considered. In Fig. 7a, it is observed that, for the same non-renewable energy source type, the mean CO_2 emission stands greener compared to conventional cloud systems while taking com-

putation into account. Lastly, even for the purpose of storage, we observe a significant reduction in CO_2 emission in Fig. 7c.

We now examine the case of green computation for renewable energy sources, as shown in Fig. 8. For the purpose of study, we have assumed three types of renewable energy sources – geothermal, hydroelectric, and offshore wind. Similar to Fig. 7, it is observant that fog computing platform are distinctly greener in terms of CO_2 emission compared to cloud platforms, considering packet forwarding (Fig. 8a), computation (Fig. 8b), and storage (Fig. 8c) into account.

A combined analysis of Figs. 7 and 8 is presented in Figs. 9a and 9b, respectively. For the sake of data aggregation we have assumed $\Theta = 75$. For non-renewable energy sources, we find that the over CO_2 emission is decreased by 59.26 percent for coal, 57.58 percent for diesel, and 55.56 percent for natural gas through fog computing, compared to cloud computing. On the other hand, for renewable energy sources, we observe that fog computing achieves a reduction of 56.94 percent for geothermal energy, 55.79 percent for hydroelectric energy, and 54.95 percent for offshore wind energy.

7.4 Cost

In this section, we analyze the cost incurred in both cloud and fog computing environments for various operations.

Fig. 10 indicates the standard cost incurred. Based on the power consumption due to computation, we evaluate the cost
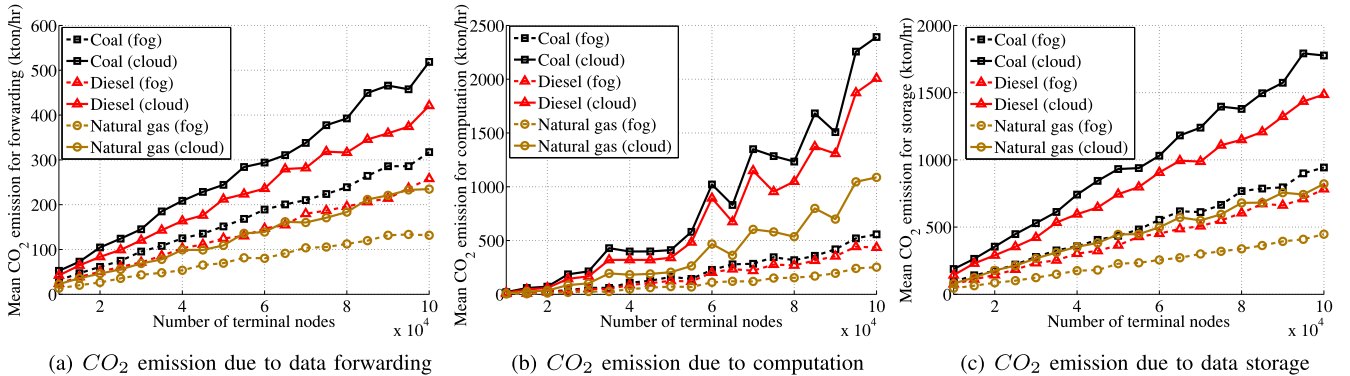

(a) CO_2 emission due to data forwarding

(b) CO_2 emission due to computation

(c) CO_2 emission due to data storage

Fig. 7. Analysis of CO_2 emission for non-renewable energy sources.

(a) CO_2 emission due to data forwarding

(b) CO_2 emission due to computation

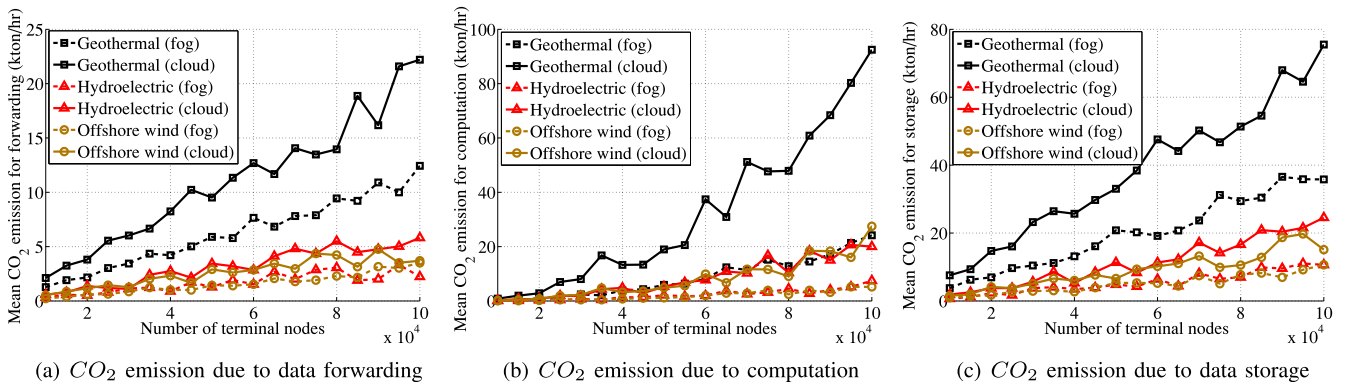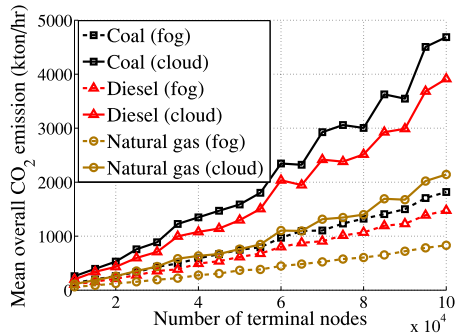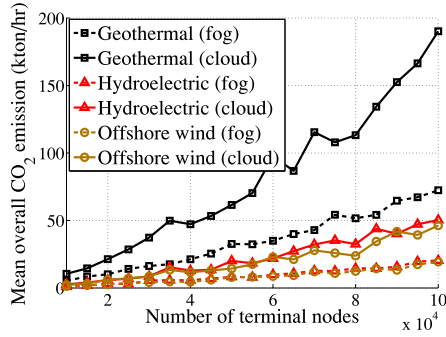(c) CO_2 emission due to data storage

Fig. 8. Analysis of CO_2 emission for renewable energy sources.

(a) Total CO_2 emission for renewable sources

(b) Total CO_2 emission for non-renewable sources

Fig. 9. Comparison of total CO_2 emission for different sources.

(a) CO_2 emission penalty for non-renewable energy sources

(b) CO_2 emission penalty for renewable energy sources

Fig. 11. Analysis of cost for non-renewable energy sources.

for computation. Similarly, we also evaluate the cost for routing and storage. Both the costs are observed to be considerably higher in cloud environments than fog environments.

In Fig. 11, we evaluate the penalty to be paid by the TNs for consumption of renewable and non-renewable energy sources. From Fig. 11a we observe that the penalty is considerably high for coal in cloud environments, whereas, in fog environments, it is remarkably low. Similarly, for diesel and natural gas, the penalty are much lower in fog environments than cloud platforms.

The analysis of penalty for consumption of renewable energy sources is shown in Fig. 11b. The penalty due to consumption of geothermal energy is much higher in cloud systems. The other types viz. hydroelectric and offshore wind energy penalty are also reduced in fog platforms, compared to cloud platforms.

Thus, the effects of Figs. 10 and 11 are combined in Fig. 12 to obtain the total incurred cost. It is observed that
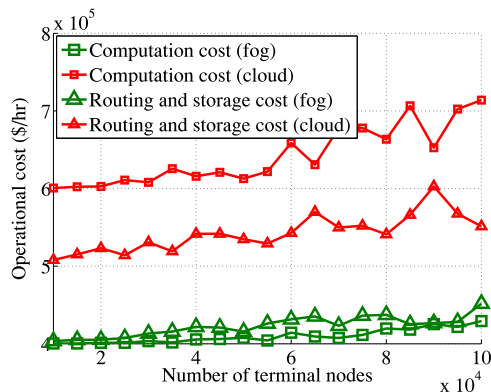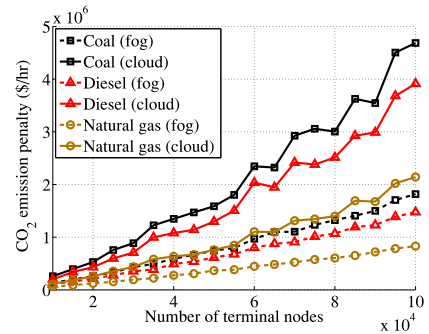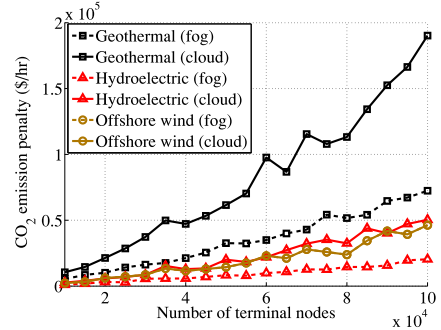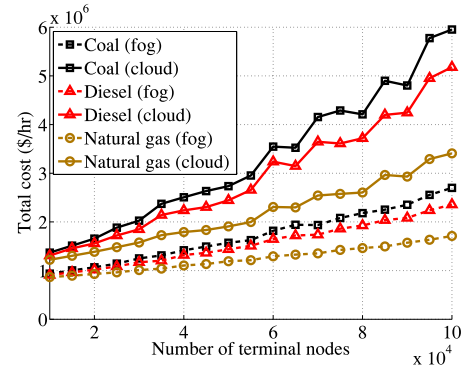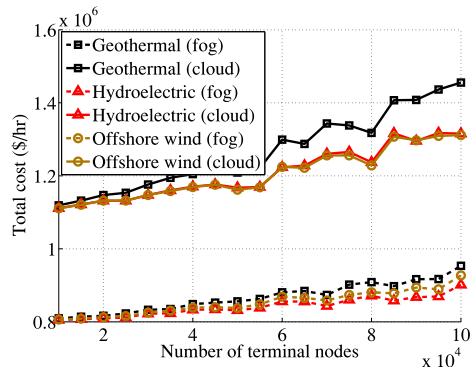
both for non-renewable (Fig. 12a) and renewable (Fig. 12b) energy sources the fog based systems exhibit a cheaper nature compared to cloud based systems.

(a) Total cost for non-renewable energy sources

(b) Total cost for renewable energy sources

Fig. 12. Analysis of cost for renewable energy sources.

Fig. 10. Operational cost per year.

8 CONCLUSION

The work focuses on analyzing the suitability of fog computing within the framework of IoT. The goal of this paper is to develop a mathematical model of fog computing and assess its applicability in the context of IoT where it is pivotal to meet the demands of the latency-sensitive applications running at the network-edge. The work further performs a comparative performance evaluation of cloud computing with that of fog computing for an environment with high number of Internet-connected devices demanding real-time services. Results clearly depict the enhanced performance of fog computing both in terms of the provisioned QoS and eco-friendliness under such situations. We eventually justify fog paradigm as an improved, eco-friendly computing platform that can support IoT better compared to the existing cloud computing paradigm.

In the future, we plan to extend this work by proposing a working fog computing prototype to support real-time implementation. The results of real network traffic and CO_2 emission rate of the DCs can be utilized to strengthen the model and support the future 'green' IoT technologies.

REFERENCES

[1] L. Piras. (2014, Mar.). A brief history of the internet of things [infographic]. [Online]. Available: http://www.psfk.com/2014/03/internet-of-things-infographic.html

[2] (2011). Internet of things market forecast. Cisco. [Online]. Available: http://postscapes.com/internet-of-things-market-size

[3] (2014, Jan.). Cisco delivers vision of fog computing to accelerate value from billions of connected devices. Press release. Cisco. [Online]. Available: http://newsroom.cisco.com/release/1334100/Cisco-Delivers-Vision-of-Fog- Computing-to-Accelerate-Value-from-Billions-of-Connected-Devices-utm-medium-rss

[4] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. M. Lau, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2710–2721, Dec. 2013.

[5] D. Lund, C. MacGillivray, V. Turner, and M. Morales, "Worldwide and regional internet of things (IoT) 2014-2020 forecast: A virtuous circle of proven value and demand," Int. Data Corp., Framingham, MA, USA, Tech. Rep. 248451, 2014.

[6] D. Bouley. (2010, Apr.). Estimating a data center's electrical carbon footprint. [Online]. Available: https://www.insight.com/content/dam/insight/en-US/pdfs/apc/apc-estimati ng-data-centers-carbon-footprint.pdf

[7] "Report to congress on server and data center energy efficiency public law 109-431," U.S. Environmental Agency, Energy Star Program, Tech. Rep. OCLC no. 163585191, Jun. 20, 2008.

[8] M. P. Mills. (2013, Aug.). The cloud begins with coal. Digital Power Group, Washington, DC, USA, Tech. Rep. [Online]. Available: http://www.tech-pundit.com/wp-content/uploads/2013/07/Cloud_Begins_With_Coal.pdf

[9] D. He and S. Zeadally, "An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography," *IEEE Internet of Things Journal*, vol. 2, no. 1, pp. 72–83, Feb. 2015.

[10] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, "The internet of bio-nano things," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 32–40, Mar. 2015.

[11] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 103–112, Apr. 2015.

[12] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between internet of things and social networks: Review and research challenges," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014.

[13] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[14] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[15] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2014.

[16] J. Wei, "How wearables intersect with the cloud and the internet of things : Considerations for the developers of wearables," *IEEE Consumer Electron. Mag.*, vol. 3, no. 3, pp. 53–56, Jul. 2014.

[17] L. Wang and R. Ranjan, "Processing distributed internet of things data in clouds," *IEEE Cloud Comput.*, vol. 2, no. 1, pp. 76–80, Jan. 2015.

[18] X. Zheng, P. Martin, K. Brohman, and L. D. Xu, "Cloudqual: A quality model for cloud services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1527–1536, May 2014.

[19] H. Yue, L. Guo, R. Li, H. Asaeda, and Y. Fang, "Dataclouds: Enabling community-based data-centric services over the internet of things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 472–482, Oct. 2014.

[20] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1253–1266, May 2014.

[21] X. Zheng, P. Martin, K. Brohman, and L. D. Xu, "Cloud service negotiation in internet of things environment: A mixed approach," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1506–1515, May 2014.

[22] S. U. Khan, P. Bouvry, and T. Engel, "Energy-efficient high-performance parallel and distributed computing," *J. Supercomput.*, vol. 60, pp. 163–164, 2012.

[23] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, May 2014.

[24] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 4, pp. 373–392, Dec. 2012.

[25] H. Liang, L. X. 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.

[26] T. H. Noor, Q. Z. Sheng, A. H. H. Ngu, and S. Dustdar, "Analysis of web-scale cloud services," *IEEE Internet Comput.*, vol. 18, no. 4, pp. 55–61, Jul. 2014.

[27] A. V. 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.

[28] S. Chatterjee and S. Misra, "Optimal composition of a virtual sensor for efficient virtualization within sensor-cloud," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 448–453.

[29] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Trans. Serv. Comput.*, 2015, DOI: 10.1109/TSC.2015.2453958.

[30] S. Das, M. Khatua, S. Misra, and M. Obaidat, "Quality-assured secured load sharing in mobile cloud networking environment," *IEEE Trans. Cloud Comput.*, 2015, DOI: 10.1109/TCC.2015.2457416.

[31] J. Xiao, H. Wen, B. Wu, X. Jiang, P.-H. Ho, and L. Zhang, "Joint design on DCN placement and survivable cloud service provision over all-optical mesh networks," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 235–245, Jan. 2014.

[32] W. Chen, J. Cao, and Y. Wan, "QoS-aware virtual machine scheduling for video streaming services in multi-cloud," *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 308–317, Jun. 2013.

[33] N. Tziritas, S. U. Khan, C. Z. Xu, T. Loukopoulos, and S. Lalis, "On minimizing the resource consumption of cloud applications using process migrations," *J. Parallel Distrib. Comput.*, vol. 73, pp. 1690–1704, 2013.

[34] A. A. Chandio, K. Bilal, N. Tziritas, Z. Yu, Q. Jiang, S. U. Khan, and C. Z. Xu, "A comparative study on resource allocation and energy efficient job scheduling strategies in large-scale parallel computing systems," *Cluster Comput.*, vol. 17, pp. 1349–1367, 2014.

[35] F. Zhang, J. Cao, W. Tan, S. U. Khan, K. Li, and A. Y. Zomaya, "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud," *IEEE Trans. Emerging Topics Comput.*, vol. 2, no. 3, pp. 338–351, Sep. 2014.

[36] D. Kliazovich, S. T. Arzo, F. Granelli, P. Bouvry, and S. U. Khan, "e-STAB: Energy-efficient scheduling for cloud computing applications with traffic load balancing," in *Proc. IEEE Int. Conf. IEEE Cyber, Phys. Social Comput. Green Comput. Commun.*, Aug. 2013, pp. 7–13.

[37] L. Xu, C. Li, L. Li, Y. Liu, Z. Yang, and Y. Liu, "A virtual data center deployment model based on the green cloud computing," in *Proc. IEEE/ACIS 13th Int. Conf. Comput. Inf. Sci.*, Jun. 2014, pp. 235–240.

[38] N. N. Naik, K. Kanagala, and J. P. Veigas, "Achieving green computing by effective utilization of cloud resources using a cloud OS," in *Proc. IEEE Int. Conf. Emerging Trends Comput., Commun. Nanotechnol.*, 2013, pp. 687–690.

[39] T. T. Huu and C.-K. Tham, "An auction-based resource allocation model for green cloud computing," in *Proc. IEEE Int. Conf. Cloud Eng.*, 2013, pp. 269–278.

[40] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U. S. Khan, A. Abbas, N. Jalil, and S. U. Khan, "A taxonomy and survey on green data center networks," *Future Gener. Comput. Syst.*, vol. 36, no. 0, pp. 189–208, 2014.

[41] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, " Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments* (ser. Studies in Computational Intelligence), vol. 546, N. Bessis and C. Dobre, Eds. New York, NY, USA: Springer, 2014, pp. 169–186.

[42] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. First Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[43] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proc. 2nd ACM SIG-COMM Workshop Mobile Cloud Comput.*, 2013, pp. 15–20.

[44] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing," in *Proc. IEEE 19th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw.*, Dec. 2014, pp. 325–329.

[45] J. Preden, J. Kaugerand, E. Suurjaak, S. Astapov, L. Motus, and R. Pahtma, "Data to decision: Pushing situational information needs to the edge of the network," in *Proc. IEEE Int. Inter-Disciplinary Conf. Cognitive Methods Situation Awareness Decision Support*, Mar. 2015, pp. 158–164.

[46] H. Madsen, G. Albeanu, B. Burtschy, and F. L. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," in *Proc. 20th Int. Conf. Syst., Signals Image Process.*, Jul. 2013, pp. 43–46.

[47] C. T. Do, N. H. Tran, C. Pham, M. G. R. Alam, J. H. Son, and C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," in *Proc. Int. Conf. Inf. Netw.*, Jan. 2015, pp. 324–329.

[48] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2015, pp. 687–694.

[49] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2015, pp. 105–110.

[50] C. Dsouza, G. J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integration*, Aug. 2014, pp. 16–23.

[51] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proc. IEEE Symp. Security Privacy Workshops*, May 2012, pp. 125–128.

[52] S. Kulkarni, S. Saha, and R. Hockenbury, "Preserving privacy in sensor-fog networks," in *Proc. 9th Int. Conf. Internet Technol. Secured Trans.*, Dec. 2014, pp. 96–99.

[53] T. Brinkoff. (2013, Jan.). City population. [Online]. Available: http://www.citypopulation.de/

[54] (2013, Jan.). Internet world stats. [Online]. Available: http://www.internetworldstats.com/

[55] Google map. Google. [Online]. Available: https://www.google.com/maps

[56] 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.–Jun. 2013.

[57] B. K. Sovacool, "Valuing the greenhouse gas emissions from nuclear power: A critical survey," *Energy Policy*, vol. 36, no. 8, pp. 2950–2963, Aug. 2008.

Subhadeep Sarkar (S'13) received the BTech degree in computer science and engineering from the West Bengal University of Technology, India, in 2012. He is currently working toward the PhD degree at the School of Medical Science and Technology, Indian Institute of Technology, Kharagpur, India. He is also a senior research fellow at the School of Information Technology, Indian Institute of Technology Kharagpur, in a project funded by the Department of Electronics and Information Technology, Government of India. His current research interests include networking and communication aspects of wireless body area networks, cloud computing, fog computing, and internet of things. He is a student member of the IEEE.

Subarna Chatterjee (S'13) received the BTech degree in computer science and engineering from the West Bengal University of Technology, India, in 2012. She is a TCS Research Scholar working toward the PhD degree at the School of Information Technology, Indian Institute of Technology, Kharagpur, India. She is also a recipient of the prestigious Google Anita Borg Memorial Scholarship 2015. Her current research interests include networking and communication aspects of cloud computing in wireless sensor networks. She is a student member of the IEEE.

Sudip Misra (SM'13) received the PhD degree in computer science from Carleton University, Ottawa, Canada. He is currently an associate professor at the School of Information Technology, Indian Institute of Technology Kharagpur, India. Prior to this, he was associated with Cornell University, Yale University, Nortel Networks (Canada), and the Government of Ontario (Canada). He has received eight research paper awards in different conferences. He received the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at IEEE GLOBECOM 2012. He also received the Canadian Government's prestigious NSERC Post Doctoral Fellowship and the Humboldt Research Fellowship in Germany. 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.