A Review on Fog Computing Systems

[*1]Farhoud Hosseinpour, [1&2]Yan Meng, [1]Tomi Westerlund, [1]Juha Plosila, [2]Ran Liu, and [1&3]Hannu Tenhunen

[1]*Department of Information Technology, University of Turku, Turku, Finland,*
[2] *School of Information Science and Technology, Fudan University, Shanghai, P. R. China, 200433*
[3]*Department of Industrial and Medical Electronics, KTH Royal Institute of Technology, Stockholm, Sweden*
{farhos; yanmen; tovewe; juplos; hatenhu}@utu.fi
rliu@fudan.edu.cn

Abstract

The current decade has witnessed a wide deployment of Internet of Things (IoT) technology in various application domains, and its pervasive role will continue to strengthen in the future. For dealing with a vast number of connected devices and the big data generated by them, an efficient computing platform is required. Fog computing has been proposed as a solution. It is a paradigm extending cloud computing and services to the edge of the network, thus reducing the latency of dynamic decision making and improving real-time performance in general. This paper provides a view on the current state-of-the-art research in the area of fog computing and internet of things (IoT) technology.

Keywords*: Fog computing, Internet of Things, Cloud computing*

1. Introduction

Internet of Things (IoT) is composed of small, resource constraint devices that are connected to the Internet. In essence, they are dedicated to perform specific functionalities without the need to perform complicated, resource-consuming tasks. This means that most of the data is transmitted to the cloud without pre-processing or analysis. This implies that the amount of data that is transmitted to the cloud is increasing even more rapidly than the number of IoT devices itself. Indeed, cloud computing is being recognized as a success factor for IoT, providing ubiquity, reliability, high-performance and scalability [1]. However, cloud computing based IoT fails in applications that require very low and predictable latency and which are geographically distributed, fast mobile and large-scale distributed control systems because of its geographically centralized nature and communication implications [2]. A promising technology to tackle the low-latency and geographical distribution required by IoT devices is fog computing.

The fog computing layer is an intermediate layer between the edge of the network and the cloud layer (Figure 1). The fog computing layer extends the computation paradigm geographically providing local computing and storage for local services. Fog computing does not outsource cloud computing. It aims to provide a computing and storage platform physically closer to the end nodes provisioning new breed of applications and services with an efficient interplay with the cloud layer [2]. The expected benefit is a better quality of service for applications that require low latency. Lower latency is obtained by performing data analysis already at the fog computing layer. Data analysis at the fog computing layer is lightweight, and therefore more advanced analyses and processing will be done at the cloud layer. Naturally, some applications do not require real-time computation, or they need high processing power, and therefore they are performed at the cloud layer. For example, in the case of a smart community [3], where homes in a neighborhood are connected to provide community services, low latency is expected for making urgent decisions, and thus computation is performed within the neighborhood, instead of a cloud layer which can be located on another continent, for example.

In this article, we will discuss different aspects of fog computing and how it relates to cloud computing and Internet of Things. The rest of the paper is organized as follows: Section 2 introduces the meaning, structure and features of the Internet of Things cloud computing and fog computing. Section 3 presents the application domains that fog computing is utilized. Section 4 studies the different challenges of integrating

[1] Corresponding Author: Email: farhos@utu.fi

the fog computing platform to IoT systems from different perspectives such as reliability, security, privacy and resource allocation. Finally, Section 5 concludes the paper.

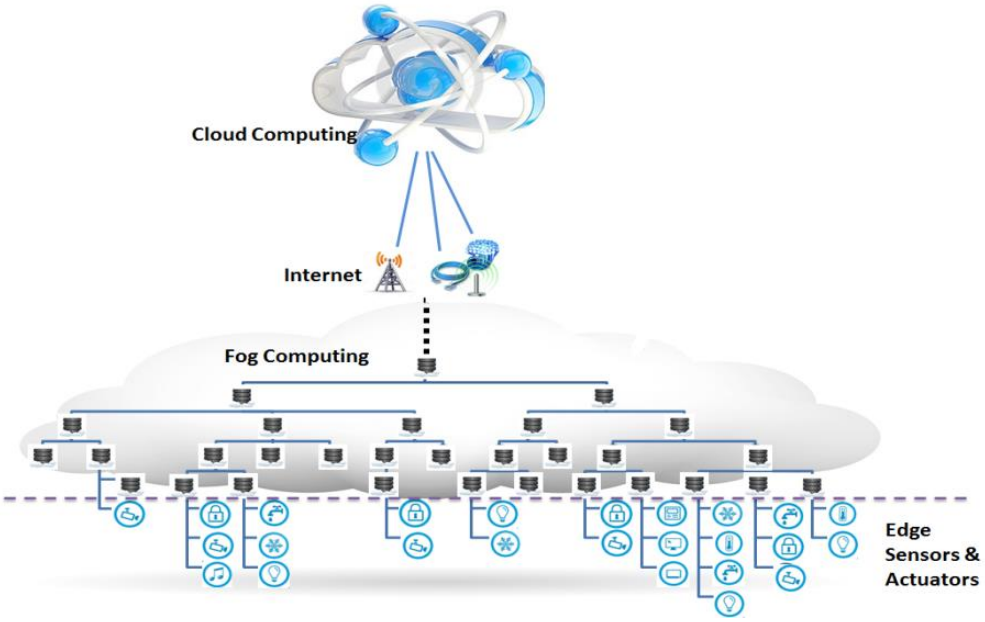

Figure 1: Hierarchical Fog Computing Architecture in IoT systems.

2. Internet of Things and Computing Solutions

Internet of Things is a concept that realizes the communication and control among a very large set of different devices[4]. Through connecting the devices, such as sensors, communication devices and data processing units, IoT allows distributed, autonomous decision making and intelligent data processing and analysis [5]. However, the enormous amount of connected devices produce a huge amount of data that is required to be transported to the cloud for analysis, which causes a demand to increase storage capacity [5]. Therefore, new concepts and structures are needed to promote the development of the IoT.

There are several ways to describe the structure of IoT. The most common way is to divide it into layers. H. Madsen et al. introduced a *H/M/P* model (Figure. 2(a)) that has three layers [6] : *H* layer refers to hardware layer which is composed of, for example, sensors, terminals, RFID tags and reader-writers, and embedded communication devices; the *M* layer is a middleware layer in which data analysis and storage are located; the *P* layer is a presentation layer in which the processed data is presented and interpreted. The data is gathered at the *H* layer, after which the *M* layer will process and transmit the data to the *P* layer that provides services directly to the end users.

As stated above, the *H/M/P* model divides IoT into three parts with different functionalities. However, the model, owing to its 3 layers, does not work well in for very large networks, for which a more fine-grained structure is better. M. Aazam et al. introduced in [7] a 5-layer model to describe an IoT architecture. The layers in the model are *perception, network, middleware, application and business* (Figure. 2(b)). The *perception* layer gathers data from the environment, after which the *network* layer moves the data from the middleware layer to the perception layer working as a bridge in between. The *middleware* layer provides service management and storage of data. The *application* layer, on the other hand, provides the global service management based on the information provided by the middleware layer. Data can be processed, analyzed, and finally presented in different forms depending on the application such as smart city, smart home, smart transportation, vehicle tracking, smart farming, and smart health [8]. The *business* layer contains the global service and business plans. This layer cares about the strategy and development of services – for both non-profit organizations and companies.

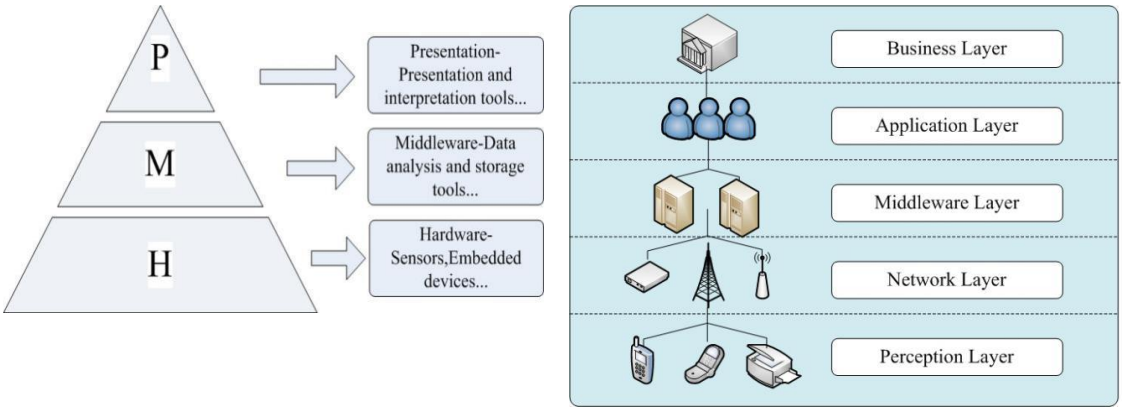

Fig. 1. a)The H/M/P model, b) The 5 layers model of IoT Structure

M. Armbrust et al. define that cloud computing includes services delivered over the internet as well as the hardware and system software in the data centers that provide those service [9]. H. Madsen et al. provide another more holistic definition: A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers[6]. Nevertheless of its definition, cloud computing has transformed the way in which computing resources are obtained, used and paid. Because all the computing hardware and software is hosted in a remote data center owned and operated by a service provider, users can just focus on their core business and pay for only the services accessed on a utility costing basis.

Current cloud computing cannot satisfy end users' requirements of mobility support, low time latency and local awareness [10]. Let us take latency as an example. For some real-time applications, users need to interact with data directly. If the data is located in the cloud, the interaction will be affected by latency. Latency is very hard to control in the Internet because of the scale of the network [11]. One viable solution to this problem is fog computing. Fog computing, as defined in [12], constitutes a highly virtualized platform that provides computing, storage and networking services between end devices and traditional cloud computing data centers, typically, but not exclusively, located at the edge of the network. Therefore, with fog computing, we are able to decrease latency and improve a user experience by performing computing nearby the source of data. That is, fog computing extends the computing and services to the edge of the system. By having an open application environment, the appearance of new devices, applications and services can be promoted to gather, analyze and exchange data [13].

Instead of the general idea of viewing fog computing as the evolution of the current cloud model, Vaquero et al. offer a much broader view to the fog. They consider the appearance of the fog as the result of the emerging trends in technology usage patterns and the advances on enabling technologies [14]. They do not try to provide a definition but more an analysis of the appearance of fog computing from the technical point of view. This offers a new angle to better understand the concept of the fog. H. Madsen et al. point out that the appearance of the fog is driven by the need for geographical distribution of devices; the need for communication among a large number of sensors; the need for real-time communication and the requirement of on-line analysis and processing with the data center [6].

An architectural view of the fog in comparison to the cloud is shown in Fig. 3. On the left side a cloud model is shown where all the gathered data is transported from the edge of the network upwards towards the cloud, whereas on the right side the fog model is shown where data can traverse not only vertical but also horizontal direction. The horizontal movement is the basis for fog computation because not all the data is transported to cloud anymore. The computation can be done at the lower level of the network in sensor nodes and smart devices such as smartphones and vehicles. Also, micro servers can be utilized to serve as a local cloud for smart buildings. For example, fog computing does not make cloud computing

futile, yet there is still a need for cloud-based services and application because some services cannot be localized.

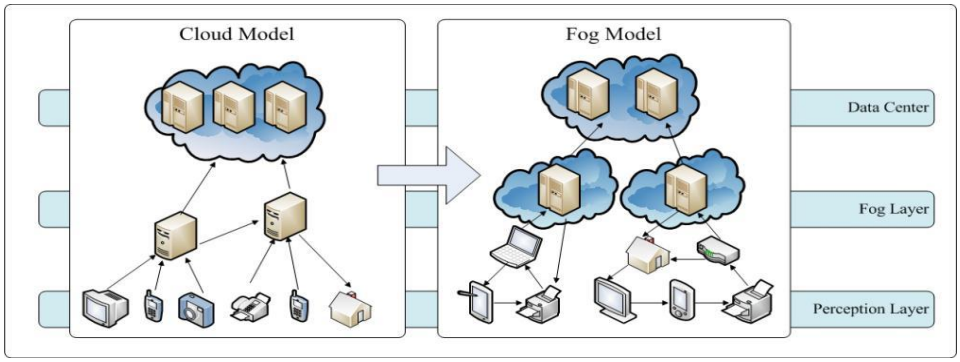

Fig. 3. Transformation from Cloud to Fog

Both cloud and fog can provide similar functionalities, but their strong points are different. In Table I below, some main differences between the fog and cloud are compared. The differences stem from the very nature of fog, which is, having intelligence at the edge of the network. In fog computing, large-scale computing can also be performed in the cloud [13]. From these differences we can draw the conclusion that fog computing is not just an extension of cloud computing, it rather complements cloud computing.

Table1: Comparison between cloud computing and fog computing

	Cloud	**Fog**
Latency	High	Low
Access	Fixed and wireless	Mainly wireless
Distribution	Centralised	Distributed
Location awareness	Limited	Supported
Support of mobility	Limited	Supported
Service access	Through core	At the edge
Price per device	$1500-3000	$50-200
Main data generator	User	Device
Computing Center	Provider's server	User's device

Fog computing can be used everywhere in our daily life. For example, in fields such as smart grids, smart traffic, smart hospitals and smart buildings, fog computing can make a big difference. Take the smart traffic as an example. In smart traffic, vehicles, roadside units and traffic lights can act as sensor nodes to gather data. Data analysis can be done at cloud or fog level depending on the analysis. Deeper analysis to recognize patterns can be done at cloud and traffic control at fog layer to regulate traffic, and thus prevent traffic jams based on the identified patterns. K. Hong et al. have simulated a traffic situation using the fog computing method [15]. The simulation result illustrates that compared with the cloud computing, the fog computing model using vehicles and roadside units as fog nodes significantly decrease latency.

2.1. Fog Computing Architecture

In a systematic view, a fog computing system is composed of distributed and heterogeneous resources that are deployed based on a hierarchal model. At the edge of the network, fog computing is extended and distributed from the network's gateways and routers to intelligent access points or smartphones that communicate directly with the edge devices. In this model, fog nodes deploy a virtualized and hierarchical topology and provide a distributed computing platform. Figure 4 illustrates a physical fog node that is composed of several virtual fog nodes based on the structure introduced by [16]. Each physical node is comprised of computing and storage components and has interfaces for communication with neighboring fog nodes at the same, one step higher, or one step lower level of the hierarchy. A virtual fog node is also composed of computing, storage and communication components and provides a multi-layered and hierarchical structure as well as collaborative distributed computing. The virtualized topology of fog computing supports multi-tenancy of various applications and processes and enables fog computing to

provide seamless computing services for different applications within each local fog node. Moreover, a hierarchical architecture that is composed of several physical and virtual fog nodes forms the fog computing platform. In this paradigm, the virtual fog nodes are defined as software agents that are consist of a virtual machine with the ability to run independently on different physical nodes. Figure 5 illustrates a hierarchical architecture of physical fog computing nodes at different levels [16].

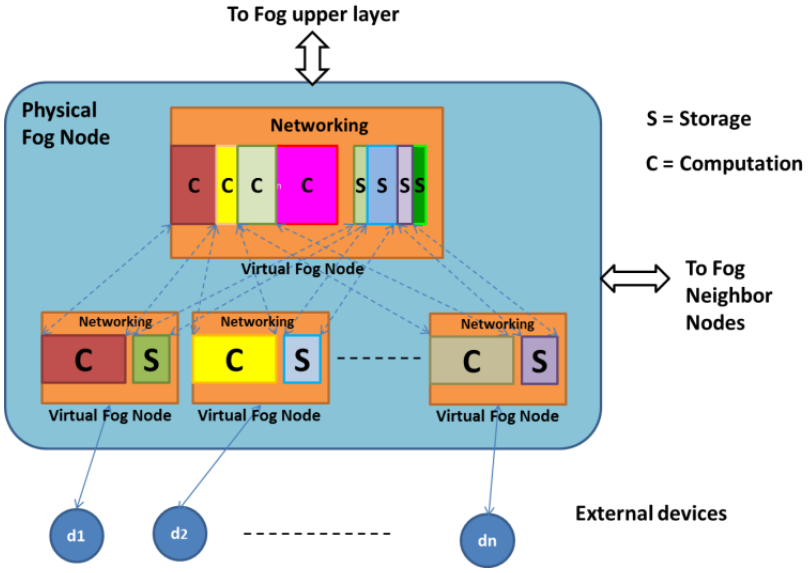

Figure 4. A Physical Fog node containing hierarchical Virtual Fog node.

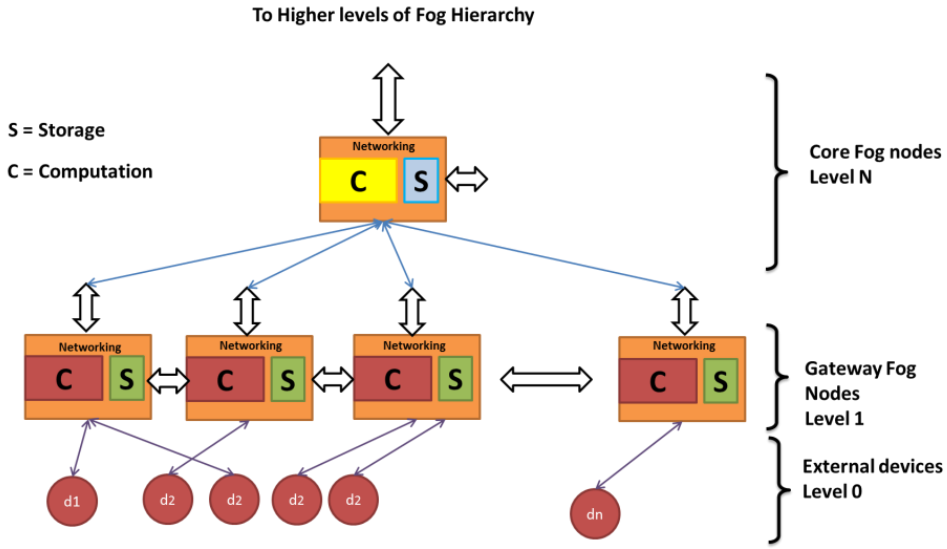

Figure 5. Hierarchical architecture of fog computing.

3. Fog Computing Application Domains

Fog computing is a new technology, but it has been already utilized in different domains and applications. In this section, we briefly review some recent studies that demonstrate or represent these key application domains. Since fog computing indeed is a relatively new technology at the time of writing this paper, most of the mentioned projects are not yet mature, i.e., they are at a very primary stage.

3.1. Fog Computing in e-Health

Due to delay-sensitive requirements of electronic healthcare, fog computing has been widely applied to this domain. Several studies investigated the utilizing of fog computing to increase the reliability of the e-health systems by coping with the latency issues in this application domain. Azam et al. [17] present a fog

computing based architecture for emergency alert services. They developed a smartphone-based service for emergency calls that utilizes a fog computing platform for a swift notification of related emergency departments. They aimed to overcome the complexity and delay in emergency notifications by offloading the collected information from mobile devices to fog and pre-processing them in fog computing while synchronizing the data with cloud computing for further analysis.

Smart decision-making is a crucial aspect of pervasive or electronic healthcare systems (e-health) such as health monitoring systems. Health monitoring involves the development of sensors in order to monitor a chronic condition of a particular body organ to predict and prevent critical health conditions such as stroke or heart attacks. Such applications are very delay sensitive and therefore fog computing is a promising technology to enhance their quality of service (QoS). Shi et al. [18] discussed the advantages of using fog computing and services in in e-healthcare systems. They debate that cloud computing at the center of the network is not a suitable solution for e-health systems because 1) cloud computing being far away from the sensors is a problem for delay-sensitive applications, 2) the big data produced in these systems increases the burden on the cloud computing resulting unacceptable QoS. They discussed the key features and functionalities that fog computing will provide for e-health applications such as 1) being adjacent to physical locations that eliminates the bottlenecks causing the delay for communication with the cloud computing, instead communication the data within the LAN, 2) enabling dynamic and real-time analysing of collected data, 3) processing and storage with smart but not powerful devices, therefore makes it suitable for pre-processing of data while deep and long-term analysis is carried out in the cloud 4) providing interoperability by supporting various communication protocols and 5) distributed computing rather than centralized way by decomposing large computation tasks into smaller ones and assigning each to one device.

Cao et al. [19] utilized a fog computing in the development of real-time fall detection system to mitigate stroke in patients. The fog computing which is composed of users' smart devices such as mobile phone or smart watches is used to carry out the detection analysis in a distributed fashion.

Gia et al. [20] utilized fog computing in an IoT-enabled healthcare system for monitoring Electrocardiogram (ECG) signals in order to enhance the bandwidth utilization and real-time response. They present a hardware setup for a smart gateway as fog computing node. The smart gateway in their work offers more functionality for the fog computing services compared to the traditional gateways. A smart gateway is composed of an embedded router and one or several sink nodes. The embedded router is composed of several hardware components operated with an operating system that provides a range of functionalities. While, the sink nodes are tiny resource constrained nodes, which are connected through USB connections to the main router component. However, they have not clearly defined the role of sink nodes in their implementations. Assuming that the purpose of having the sink nodes is to provide extendibility to the main structure, the drawbacks of their work include: first, the number of sink nodes is limited to the number of available USB connections, and, second, using the physical link (instead of wireless ad-hoc structure for example) between the sink nodes and the main router makes a dynamic scalability of the system impossible. Third, indeed, a fog computing platform is a distributed computing system over a geographical area at the edge of the network. Yet, the smart gateway presented in their work lacks such a distributed structure and offers only a single smart gateway component at the edge. This structure has the issue of the traditional gateways of being a single point of failure. If the smart gateway does not function properly, the whole IoT system and services will be affected accordingly.

3.2. Fog Computing in Food Chain

Food safety and quality are very important issues in the food industry. In an advanced food chain (from the farm to the customers) maintaining the quality of food by monitoring the quality parameters is often done by utilizing the IoT infrastructure. For example, it is crucial in a supply chain to ensure that the food has been stored at proper temperature all the time. Traceability of such parameters requires continuous monitoring through specific sensors that will generate a large amount of data to be processed. Chen et al. [21] proposed utilizing a fog computing system using an Artificial Neural Network (ANN) for evaluating the temperature data in a food chain management system. They argued that fog computing compared to cloud computing has advantages of mobility support, location awareness and low latency that are essential for traceability of foods. They propose to set up a fog computing platform using smart vehicles, ubiquitous mobile devices to perform real-time analysis for temperature monitoring. However, there is no real implementation or simulation of the fog computing platform presented in this work.

3.3. Fog Computing in Energy Management

The next generation of power systems involves a variety of smart devices and technologies such as smart meters, smart appliances, renewable energy resources, and energy efficiency resources. Such devices and technologies are key components of a smart electrical grid. It is evident that powerful computing resources are needed to handle the data generated by growing number of sensors and smart devices in smart grids. Cloud computing traditionally used in such systems suffers from scalability and adaptability issues. To cope with such problems, Faruque et al. [22] presented a fog computing based energy management-as-service for residential building for managing their energy consumption. Their proposed system addresses the interoperability, scalability, and adaptability issues by utilizing a fog computing system. Their fog computing system is composed of low-power and low-cost devices which are based on open source architecture. They developed a two-level hierarchical fog computing platform consisting of sensor nodes with processing capabilities which act as access points and base stations. The access points are connected to end devices with no computing or storage capacity and at a higher level, base stations are connected to multiple access points. Data is collected from end devices and aggregated in access points and base stations hierarchically. Base stations send the collected data to the main energy management platform reside in the local fog computing system. They deployed a Devices Profile for Web Services (WS4D) for communication of devices that supports SOAP-based protocols providing interoperability, scalability, and interactivity in their system. However, the performance evaluation of fog computing is not presented in this work.

3.4. Fog Computing to Support Mobile Applications

Mobile devices such as tablets and smartphones are increasingly getting more attention as replacements for traditional PC or laptop computers. Software applications supported by such smart mobile devices are developing and offer more services and functionalities, yet, on the other hand, they require more computational and storage resources. Limited bandwidth and high latency make the utilization of cloud computing solutions infeasible for smart mobile applications. Also, energy consumption for communication with a cloud through high-performance communication protocols such as TCP is a major factor for limited energy mobile devices. So, a fog computing platform in the proximity of these devices is a preferable chose to provide a fast, reliable and energy efficient computing platform for offloading their computation tasks. Hassan et al. [23] explored an edge intelligence by utilizing nearby resources to speed up the mobile computations. They argued that a proper computation offloading decision is critical for the performance of an application. Therefore, they aimed to predict the performance of applications in different computational settings. They investigated the parameters that impact the offloading performance. They used a Multilayer Perception (MLP) model for predicting the offloading performance of different parts of the application in different commutating environment (fog or cloud or mobile device itself). This is done by training the MLP by monitoring the different computing environment and their response time.

4. Challenges in Integrating Fog Computing in IoT Systems:

4.1. Reliability

Just like in the case of cloud computing, the reliability of fog computing not only is crucial but also difficult to analyze because of the complexity and the huge size of the network. Because of its unique characteristics, the way and the model for testing the reliability of the former networks cannot be used here. When considering the reliability of the fog computing network, the reliability of single sensors, end-user nodes, application interfaces, software, and a network supporting data processing and information exchange should be considered [24]. Moreover, the failure detection methods used nowadays are not mature enough, the failure itself and the fault leading to the failure are both hard to recognize. All of these make the reliability of fog computing an important but difficult task. Y.-S.Dai et al. have raised models and algorithms to analysis the reliability of the cloud and grid computing, which can be the guidance to the reliability of fog computing [25], [26].

The appearance of failures will affect the reliability of the whole system, Y.-S.Dai et al. analysed the failures in the cloud computing which are also very common in fog computing [26]. Failures can be divided into *request stage failures* and *execution stage failures* according to the time it happens. In a request stage, the failures include overflow and timeout, While, in the execution stage, the failures

include: data and computing resource missing, software and hardware failure, database and connection failure.

M. A. Mahmood et al. divided the reliability protocols into two types: packet reliability and event reliability [27]. Based on their analysis, the packet reliability will ensure that the nodes can transport all of the data packet and the event reliability make sure that at least one of the packets with the event that the sensor detected is delivered. They point out that by combining these two reliabilities and the reliability requirement of the sensor nodes and the whole network, it is possible to design a reliable fog computing system.

4.2. Privacy

British Internet users' personal information and sensitive data on cloud storage are under surveillance by Government Communications Headquarters (GCHQ). National Security Agency (NSA) and GCHQ have developed a technology that is able to record and filter through very large volumes of the information traffic. This is only one example about the personal data leakage. From this, we can find that with the communication among devices increases exponentially, the privacy issue faces more and more challenges.

Privacy-preserving techniques are common topics which have been investigated in different situations, such as in cloud computing network, in a smart grid system, in an online social network, in wireless communication network. Because of the structure of fog computing, when users enjoy the service, the data related to them will be transferred to the computing center to be processed, which is out of the users' control. These data might be location related or time-related, which may be related to the users' privacy.

In [10], the author points out that the privacy issue about fog computing means hiding the details which may leak the personal information and allow the correct summary data for the accurate charging. Even though fog devices cannot decrypt and change the information from a smart meter, the device can transmit the information to other gateways. They also point out that the security and privacy work should notice the following issues:1) Whether the important information is only accessible to privileged users; 2) Whether the devices and the infrastructure can meet the security requirements; 3) Whether the network provider can store and process the data under the specific jurisdictions; 4)Whether the data of one user can be totally isolated from another users data; 5)Whether the network provider can recover the data if some disasters happen; 6)Whether there is the mature regulation to support the user to investigate the network provider if it is needed; 7)Whether the users' data can be stored for a long time even the network provider has been changed.

Because both fog layer and cloud centers have computing and storage units, privacy-preserving algorithms and models can be run between the fog layer and cloud center. In [28], R. Lu et al. proposed an Efficient and Privacy-Preserving Aggregation scheme, named EPPA, for smart grid communications. This scheme uses a homomorphic Paillier cryptosystem technique to allow privacy-preserving aggregation at the local gateways without decryption. Furthermore, techniques such as batch verification can be used in EPPA to reduce authentication cost. Their work demonstrates that with less computation and communication overhead, EPPA can be used to preserve users' privacy effectively.

4.3. Security

The distributed nature of fog computing causes more complex security issues. Dealing with heterogeneous computing nodes, implementing a robust security mechanism in a fog computing system is very challenging. In a fog computing, most of the devices use a temporary power supply like batteries. On the other hand, most of the devices are not powerful devices and have limited computing and storage capabilities. Therefore, developing low power security mechanisms with acceptable level of security is very challenging. Even though more data processing will be done locally in the fog computing network, users' data will still be stored remotely. Users have concerns about where their data is and who will use it and for what reason. There exist protocols in the cloud systems for the service providers and users to regulate delivery of a service, data positioning and reporting, security rights and responsibility [29]. In fog computing, how to formulate the similar protocols is still an open question.

Compared to cloud computing, fog computing structure is more flexible, and thus service providers can deploy more functions and services that make the effective governance and regulation even more important. Because service providers have different services and different roles, it may produce a security gap in data availability and integrity [30]. The security gap may lead to the blind spot for the governance. Without an efficient system or organization for governance, users' security cannot be guaranteed.

A) Naming Issues:

As a fog computing network will connect billions of smart devices, an efficient naming and identity management system is of great importance to efficiently manage and organize a large number of devices. Presently, different applications often use their own standards, and are therefore not compatible. For example, EPC Global and Ubiquitous ID are two entirely different ways of identifying objects. Devices using these two modes to be identified cannot communicate with each other smoothly. A probable solution is using IPv6 [31] due to its large address space. In this case, an efficient mechanism of IPv4-IPv6 coexistence is necessary to support the interoperability in fog computing. Depending on the application, devices use different communication protocols, such as Wireless HART, ZigBee, or 6LowPAN. Therefore, the interoperability of such protocols is a key issue. Therefore, we need devices capable of converting a protocol to another one. In order to handle such a problem, P. P. Hung et al. proposed that the mapping of standardized protocols would be done in a gateway [7]. However, fog computing provides a more flexible solution for conversion, as it can provide conversions as services when needed. From the security point of view, poor encryption technologies and network protocols raise the risks that must be taken into account in designing the conversion services.

B) Data Security:

In the shared data environment used in fog computing, some data can be migrated from one region to another making the integrity of the data more challenging. Having several users, shared computing resources such as CPU caches and disk partitions can be a target for an attack [30]. Existing data protection mechanisms such as encryption and decryption, and standard access controls, have been demonstrated to fail. S. J. Stolfo et al. proposed a new security mechanism to protect the information security from insider attack, which includes two parts: user behavior profiling and decoys. User behavior profiling is based on the assumption that regular users are familiar with the structure and the content of the system, they have an idea about what specific file they will get, so their search behavior will be aimed and finite. The decoy files are stored in the system, and unauthorized users cannot tell them apart from the correct files according to the name or location. If the system notices the unusual access, the decoy files would be returned to the user in a reasonable and legitimate way.

The storage location of sensitive data is a key data privacy and security issue. There is not a perfect model or algorithm to describe and solve such issues at the moment. When data and information have been stolen or attacked, it is impossible to recover, which will be a great loss for both a service provider and a user. Compared with a cloud system, more data will be processed at the edge of the network that makes it easier to steal or attack data and information. The lack of effective authorization, authentication and audit are making the situation even worse. Dsouza et al. [32] proposed a policy-driven security management framework which using an attribute based authentication scheme provides in which all users and devices are authenticated based on a set of attributes that they represent. They utilized eXtensible Access Control Markup Language (XACML) to specify operational, security, and network policy specification in their framework. Such security policies are built based on a modular structure where each security policy module could be plugged and played in real-time based on the corresponding applications and devices. They evaluated the feasibility and practicality of their framework for a Smart Transportation System (STS).

C) Potential Attacks

A local computing center increases the risk of the man-in-the-middle attacks through compromised or fake fog computing nodes. If the secure socket layer cannot cover all the security gaps, an attacker can access the data exchange between two parties, which is dangerous to the data communication. In highly dynamic fog computing with limited power computing resource, Denial of Service (DoS) attacks have a higher risk and are easy to run [33].

Intrusion detection systems are widely used in cloud computing to detect attacks and malicious activities such as port scanning and spamming [34]. However, in fog computing, due to its special characteristics, the intrusion detection faces more challenges. Anomaly detection techniques due to high power consumption and computational demands are not a feasible option for fog computing systems. Lee et al. [35] propose a hybrid intrusion detection system utilizing both signature and anomaly based engines. In this case, the anomaly detection is done in the cloud, and a signature of detected anomalies is generated and stored in local fog computing servers for signature based intrusion detection. Wang et al. [36] investigated the new issues in fog computing security forensics.

4.4. Resource management

In this section, we review and analyze ongoing research works in resource management in fog computing. Efficient management of resources is a key research problem in fog computing because of the inherent heterogeneity and delay-sensitivity requirements of the fog. Moreover, different constraints, such as limited energy and limited wireless coverage, make the research more challenging. Taking the scalability and reliability into account, fast and appropriate distributed resource allocation provides enhanced QoS for IoT applications as well as efficient and fair consumption of the resources in fog [37].

A) Optimization Techniques

A precise resource allocation in fog computing requires the optimization of the involved parameters. Several researchers have focused on resource allocation in fog computing with considering different parameters in their optimization problems. Deng et al. [38] investigated the trade-off between power consumption and delay in a cloud-fog computing system. They formulated the workload allocation problem in a cloud-fog scenario by modeling the power consumption and delay functions in cloud and fog a well as communication delay function for dispatch. Then, they decomposed the primal problem into three sub problems of power consumption-delay trade-off for fog computing (leading to a convex problem with linear constraints), power consumption-delay trade-off for cloud computing (leading to a mixed integer nonlinear programming (MINLP) problem), and communication delay minimization for dispatch (leading to an assignment problem). The numerical results reported in their work show that allocating the workload to the fog computing system while the power consumption is increased in this case decreases the delay. This reveals that cloud computing is more energy efficient than fog computing while fog computing due to the proximity to the users can improve the performance of cloud computing by reducing communication latencies.

Because of heterogeneity of applications, devices, and users in an IoT system, the utility of the system is significantly diverse based on geographical distribution of such entities. "Utility, or usefulness, is the (perceived) ability of something to satisfy needs or wants. Utility is an important concept in economics and optimization theory because it represents satisfaction experienced by the consumer of goods [39]." To optimize the utility of the fog computing system, one important way is to control the fraction of the traffic. Do et al. [39] propose the optimization of resource allocation in fog computing by maximizing the utility with joint consideration of carbon footprint in cloud-fog data centers. Inspired by some related studies for management of traffic for data centers in a wide area [40], they utilized proximal algorithms [41] which are faster with relatively moderate accuracy compared to traditional gradient methods [42]. Proximal algorithms are robust in the sense that they do not strongly rely on the strict convexity of the constraint functions. They considered a widely used affine function for utility in [40] and a cost function for carbon footprint [43] from literature. They formulate a joint optimization problem to maximize utility and minimize carbon footprint in fog computing as a general convex optimization.

In an alternative approach to optimizations techniques, Abedin et al. [44] utilized a refined version of Irving's matching algorithm [45] by considering quota for each node to be able to serve more than one pair. Their approach provides a stable pair-matching scheme between one-to-many fog nodes to utilize their shared resources. They defined a utility preference list for each fog node based on a pricing factor. They formulate a utility function based on the data transmission and reception power consumption and the price of consumed energy. After defining preference list for each fog node using the proposed Irving's matching algorithm, requesting fog nodes pair with the best matching nodes among their preferred nodes in a way that no individual requesting node prefers a particular node from a set of matching nodes over their currently matched node (stable matching). The quota-based algorithm in a large number of node set enhances the overall utility of the entire fog network.

Communication costs between fog computing nodes are very important parameters to be optimized. Finding an optimal path to communicate the processing requests is a key task in building a robust resource allocation scheme in fog computing. To this end, Jingtao et at. [46] proposed a Steiner tree based caching scheme to produce an optimal Steiner tree in a fog computing cluster in order to minimize the cost total cost of the communication path in a way that total cost of cashing resources is minimized.

Allocation of resources in a fog computing system for IoT applications requires the deployment of Virtual Machines (VM) corresponding to the requesting application in the selected fog computing node. It is evident that deploying the VM in all fog nodes is a costly and not efficient approach. Also, VMs usually require certain resources to meet the requirements of the application. Therefore, the deployment of

VM imposes another challenge in resource allocation. To cope with this problem, Gu et al. [47] propose a method called FC-MCPS for medical devices which utilize fog computing for offloading their computation. They studied a cost-efficient resource management in this scenario by avoiding the development of VM of the applications in all fog nodes and extra communication costs to send the computing request to the nodes that already contains the VM for the corresponding application. They formulated cost efficiency as minimization problem in the form of mixed-integer nonlinear programming (MINLP) by jointly considering constraints such as VM deployment, communication and computation Base Station (BS) association, task distribution and subcarrier allocation. They linearized the problem as mixed-integer linear programming (MILP) to cope with the complexity of MINLP. Their main objective was to minimize the overall cost of deploying VM of requesting applications in a given fog computing infrastructure. They found out that an appropriate set of fog nodes to host the VMs for each application is a key factor for minimizing the cost of fog computing resource allocation.

B) Smart Data Approach

Looking from another perspective, Hosseinpour et al. [48], [49] present a new framework for managing big data in IoT based applications. They propose a concept called Smart Data, an active and intelligent data structure that utilizes a fog computing system. A Smart Data element is a cell of encapsulated data consisting of a payload, metadata, and a virtual machine. Such a data cell is initially very simple and lightweight, but it evolves (grows) when traveling through the hierarchical fog computing system towards the cloud, merging with other cells (or vice-versa, if the data moves towards the actuators). The virtual machine part in Smart Data acts as a platform which enables and manages the execution of the rules specified in the metadata part. In order to avoid the overhead of integrating the whole application code to each Smart Data, at early stage, Smart Data includes only basic application code that provides it some basic functionality such as communication and lightweight encryption. New modules of the application code integrate to the base code whenever there is a need for new functionality. Each module contains a set of program codes that provides a certain service or accomplishes a certain operation on the data. For example, there could be an aggregation module, an encryption module, and compression modules. A code repository, located in the core of the network, transmits the code modules upon requests coming from Smart Data cells. Although this technology is in a very early stage of development, it is a promising approach to cope with issues of deploying the application code and VM of IoT services which were addressed in [47].

C) Resource Allocation

A proper federation of resources between cloud and fog computing nodes is essential to facilitate interplay between a cloud layer and a fog layer. In [50], a Mobile-IoT-Federation-as-a-Service (MIFaaS) paradigm is presented. In this work, fog computing is utilized for the federation of resources to support the interplay between mobile cloud nodes at the edge of the network and the cloud computing at the center of the network. When a user's local mobile could is not able to handle the computing requests, the edge fog nodes allocates the overloaded tasks to available computing resources by virtualizing the computational, communication and storage profiles of the tasks through a decomposer module in fog nodes. By exploiting an opportunistic computing cooperation between mobile clouds and cloud computing, fog computing triggers the federation of tasks and resources between mobile cloud and cloud computing. The proposed federation algorithm utilizes a utility function that measures the ability of any combination of IoT cloud providers (ICP) to maximize the efficiency of the system. Based on presented results, the ICP federation can achieve a higher percentage of usefully executed tasks. Considering the mobile cloud nodes as local fog computing cluster, the proposed algorithm can be utilized to federate the task allocation between cloud and fog layer. However, considering more parameters based on the application requirements, such as delay, cost and energy consumption, could complement this work.

A highly dynamic fog computing system that consists of mobile nodes with fluctuating connectivity behavior requires complex techniques for a predictable resource allocation and provisioning. Therefore estimating the required resources for a request beforehand increases the efficiency of the cloud or fog computing resources and provides an appropriate resource management. Azam et al. [51] mentioned that "prediction and pre-allocation of the resources for each service in IoT also depends on the users' behavior and the probability of using those resources in future." They formulated a resource allocation technique for cloud service customers (CSC) through fog micro data centers by considering relinquish probability of utilizing the requested services. To this end, they have proposed a service oriented resource management

scheme for different traits of the customers. They formulated required resources (storage, memory, and bandwidth) based on the basic price of requested service, service-oriented relinquish probability of a customer to give up on a service and also the history of overall relinquish probabilities. In this model, fog computing is used for establishing a fair resource allocation for CSC from a cloud computing or fog computing resources in a way that resource wastage is minimized if a customer gave up on a service. They complemented their work in [52] by incorporating a more accurate and fair pricing model in order to provide a more realistic resource allocation in the case that consumers discontinue the service (especially with mobile devices). After terminating a service, computing calculates the consumed resources and services and the remaining service value of agreed total preliminary service.

Because of heterogeneity of the devices and applications in IoT systems and also the dynamic nature of the processing requests based on application's requirement, enabling scalable, flexible and real-time strategies for resource allocation is very challenging. Oueis et al. [53] propose a cluster-based resource allocation scheme for fog computing in which a cluster of fog computing resources is logically built depending on the profile of computation offloading request from an IoT device or a fog node. This computation cluster is established based on a backhaul topology [54]. The size of the computing cluster also depends on the communication link quality between fog nodes as well as the computation capability of each node. Their proposed method is a multi-user case, where the computing clusters have adaptive size and load distribution based on all users' requests jointly. So, they formulate the clustering problem as a joint optimization problem with the objective of minimizing power consumption and optimizing computational rate and computational load for each mobile user and considering the latency requirements of each request. The optimization problem in their work is a non-convex problem because of non-convexity of the delay function in that scenario. Thus, they cast the primary problem into another convex problem that could be easily optimized. Even though their approach is non-optimal, their experimental results show that the Quality of Expertise (QoE) for each requesting node is increased when using the clustering method for resource allocation in fog computing.

5. Conclusion

Fog computing is a promising computing model and concept that provides a hierarchical and distributed computing platform at the edge of the network for IoT applications. Because of its unique characteristics, a fog computing system provides better means of computing for delay-sensitive and geographically distributed IoT applications. A fog computing system does not replace a cloud computing system, but it complements cloud computing by a new breed of services. This paper reviewed key aspects of the fog computing technology for IoT applications. We analyzed and discussed the architectural models and the interplay of cloud and fog computing. We presented some application areas in which fog computing systems are used for enhancing services. Fog computing, however, faces many challenges in integrating fully with the current IoT domain. We categorized, analyzed and discussed these challenges in this paper.

Fog computing research is at a primary stage. Researchers have different perceptions and understanding about fog computing systems, which lead to various implementations and deployments of this technology. This issue makes the evaluation and comparison of these studies more difficult. Evidently, there is a need for a standard fog computing architecture that can be used as a reference for utilizing fog computing in different application domains. Moreover, in addition to latency aspects, essential parameters such as throughput, energy efficiency, scalability, and quality of service need more emphasis in future to answer the challenges posed by the growing demand of new IoT applications. The aforementioned challenges together with design, specification, and development of a fog computing platform for IoT applications will be in front and center in our research.

Acknowledgement

This work was supported by Turku University Foundation and EIT Digital.

6. References

[1] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," *2014 IEEE World Forum Internet Things*, pp. 375–376, Mar. 2014.

[2] 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*, *Studies in Computational Intelligence*, vol. 546, N. Bessis and C. Dobre, Eds. Cham: Springer International Publishing,

2014, pp. 169–186.
[3] V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, "Smart Human Security Framework Using Internet of Things, Cloud and Fog Computing," in *Advances in Intelligent Systems and Computing*, vol. 321, R. Buyya and S. M. Thampi, Eds. Cham: Springer International Publishing, 2015, pp. 251–263.
[4] Y. J. Guo-Zhen TAN, Hao Wang, "IoT-based Distributed Situation Awareness for Traffic Emergent Events," *IJACT Int. J. Adv. Comput. Technol.*, vol. 5, no. 7, pp. 1050–1059, 2013.
[5] L. Tan, "Future Internet: The Internet of Things," in *International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, pp. 376–380.
[6] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable Fog computing," *2013 20th Int. Conf. Syst. Signals Image Process.*, pp. 43–46, Jul. 2013.
[7] M. Aazam and E. N. Huh, "Fog computing and smart gateway based communication for cloud of things," *Proc. - 2014 Int. Conf. Futur. Internet Things Cloud, FiCloud 2014*, pp. 464–470, 2014.
[8] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet : The Internet of Things Architecture , Possible Applications and Key Challenges," in *10th International Conference on Frontiers of Information Technology Future*, 2012.
[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of Cloud Computing," *Communications of the ACM*, pp. 50–58, 2010.
[10] I. Stojmenovic and S. Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," in *Federated Conference on Computer Science and Information Systems*, 2014, vol. 2, pp. 1–8.
[11] N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14 – 23, 2009.
[12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things Characterization of Fog Computing," in *MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–15.
[13] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," *2014 Australas. Telecommun. Networks Appl. Conf. ATNAC 2014*, pp. 117–122, 2015.
[14] L. M. Vaquero and L. Rodero-merino, "Finding your Way in the Fog : Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.
[15] K. Hong, D. Lillethun, B. Ottenwälder, and B. Koldehofe, "Mobile Fog : A Programming Model for Large – Scale Applications on the Internet of Things," in *second ACM SIGCOMM workshop on Mobile cloud computing*, 2013, pp. 15–20.
[16] M. Nemirovsky, "Fog Computing," in *Cloud Assisted Services in Europe (CLASS) Conference, Bled 2012*, 2012.
[17] M. Aazam and E. N. Huh, "E-HAMC: Leveraging Fog computing for emergency alert service," *2015 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2015*, pp. 518–523, 2015.
[18] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," *2015 2nd Int. Symp. Futur. Inf. Commun. Technol. Ubiquitous Healthc.*, pp. 1–5, 2015.
[19] D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," *2015 IEEE Int. Conf. Networking, Archit. Storage*, pp. 2–11, 2015.
[20] T. N. Gia, M. J. A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog Computing in Healthcare Internet-of-Things : A Case Study on ECG Feature Extraction," *IEEE Int. Conf. Comput. Inf. Technol.*, p. 8, 2015.
[21] R. Chen, "Fog Computing-based Intelligent Inference Performance Evaluation System Integrated Internet of Thing in Food Cold Chain," in *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 879–886.
[22] M. Al Faruque and K. Vatanparvar, "Energy Management-as-a-Service Over Fog Computing Platform," *IEEE Internet Things J.*, vol. PP, no. 99, pp. 1–1, 2015.
[23] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help Your Mobile Applications with Fog Computing," in *Sensing, Communication, and Networking - Workshops (SECON Workshops)*, 2015.
[24] C. Dabrowski, "Reliability in grid computing systems," *Concurr. Comput. Pract. Exp.*, 2009.
[25] Y. Dai, Y. Pan, S. Member, and X. Zou, "A Hierarchical Modeling and Analysis for Grid Service Reliability," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 681–691, 2007.
[26] Y. S. Dai and M. Xie, "Reliability Analysis of Grid Computing Systems," in *Pacific Rim International Symposium on Dependable Computing (PRDC'02)*, 2002.
[27] M. A. Mahmood and W. K. G. Seah, "Event Reliability in Wireless Sensor Networks," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2011, pp. 377–382.
[28] R. Lu, X. Liang, S. Member, and X. Li, "EPPA : An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1632, 2012.
[29] S. Ramgovind, E. Mm, and E. Smith, "The Management of Security in Cloud Computing," in *Information Security for South Africa*, 2010.
[30] A. E. Youssef and M. Alageel, "A framework for secure cloud computing," *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 4, pp. 487–500, 2012.

[31] E. Engin, “Impact of IPv4-IPv6 Coexistence in Cloud Virtualization Environment,” *Ann. Telecommun. - Ann. des télécommunications*, no. October, 2013.
[32] C. Dsouza, G. J. Ahn, and M. Taguinod, “Policy-driven security management for fog computing: Preliminary framework and a case study,” *Proc. 2014 IEEE 15th Int. Conf. Inf. Reuse Integr. IEEE IRI 2014*, pp. 16–23, 2014.
[33] G. P. Pandeeswari Nagarajan, “Detection of Denial of Service Attack in Cloud using Fuzzy Time Series Analysis and EM Algorithm,” *IJACT Int. J. Adv. Comput. Technol.*, vol. 7, no. 5, pp. 25–36, 2015.
[34] P. Kasinathan, C. Pastrone, M. a. Spirito, and M. Vinkovits, “Denial-of-Service detection in 6LoWPAN based Internet of Things,” *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, pp. 600–607, 2013.
[35] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, “On security and privacy issues of fog computing supported Internet of Things environment,” *2015 6th Int. Conf. Netw. Futur.*, pp. 1–3, 2015.
[36] Y. Wang, T. Uehara, and R. Sasaki, “Fog Computing: Issues and Challenges in Security and Forensics,” *2015 IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, pp. 53–59, 2015.
[37] M. H. Rashid Alakbarov, Fahrad Pashayev, “A Model of Computational Resources Distribution Among Data Center Users,” *IJACT Int. J. Adv. Comput. Technol.*, vol. 7, no. 2, pp. 01–06, 2015.
[38] R. Deng, R. Lu, C. Lai, and T. H. Luan, “Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing,” *IEEE Int. Conf. Commun.*, vol. 2015-Septe, pp. 3909–3914, 2015.
[39] 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,” *Int. Conf. Inf. Netw.*, vol. 2015-Janua, pp. 324–329, 2015.
[40] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, “To Coordinate Or Not To Coordinate ? Wide-Area Traffic Management for Data Centers,” 2012.
[41] N. Parikh and S. Boyd, “Proximal Algorithms,” *Found. TrendsR ? Optim.*, vol. 1, no. 3, pp. 123–231, 2013.
[42] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient Methods The subgradient method,” vol. 1, pp. 1–21, 2003.
[43] H. Xu, C. Feng, B. Li, and S. Member, “Temperature Aware Workload Management in Geo-Distributed Data Centers,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1743–1753, 2015.
[44] S. F. Abedin, G. R. Alam, N. H. Tran, and C. S. Hong, “A Fog based System Model for Cooperative IoT Node Pairing using Matching Theory,” in *Network Operations and Management Symposium (APNOMS)*, 2015, pp. 309–314.
[45] R. W. Irving, “An Efficient Algorithm for the ‘ Stable Roommates ’ Problem,” *J. Algorithms*, vol. 595, pp. 577–595, 1985.
[46] C. Engineering and T. Beijing, “Steiner Tree B ased Optimal Resource Caching Scheme in Fog Computing,” *China Commun.*, no. August, pp. 161–168, 2015.
[47] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, “Cost-Efficient Resource Management in Fog Computing Supported Medical CPS,” *IEEE Trans. Emerg. Top. Comput.*, vol. 6750, no. c, pp. 1–1, 2015.
[48] F. Hosseinpour, J. Plosila, and H. Tenhunen, “Smart Data: Reshaping Data Structure in IoT for Tackling the Five Vs of Big Data using Fog Computing,” *TUCS Tech. Reports*, no. 1159, 2016.
[49] F. Hosseinpour, P. Juha, and H. Tenhunen, “An Approach for Smart Management of Big Data in the Fog Computing Context,” in *IEEE CloudCom 2016 Conference*, 2016.
[50] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, “Federated Edge-assisted Mobile Clouds for Service Provisioning in Heterogeneous IoT Environments,” in *Internet of Things (WF-IoT)*, 2015, pp. 1–6.
[51] M. Aazam and E. N. Huh, “Dynamic resource provisioning through Fog micro datacenter,” *2015 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2015*, pp. 105–110, 2015.
[52] M. Aazam and E. N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, vol. 2015-April, pp. 687–694, 2015.
[53] J. Oueis, E. C. Strinati, and S. Barbarossa, “Small cell clustering for efficient distributed cloud computing,” *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, vol. 2015-June, pp. 1474–1479, 2015.
[54] J. Oueis, E. Calvanese-strinati, and S. Barbarossa, “On the Impact of Backhaul Network on Distributed Cloud Computing,” in *Wireless Communications and Networking Conference Workshops (WCNCW)*, 2014, pp. 12–17.