Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7597686, 14 pages
https://doi.org/10.1155/2018/7597686

Review Article

Towards Distributed Data Management in Fog Computing

Vasileios Moysiadis,[1] Panagiotis Sarigiannidis,[1] and Ioannis Moscholios[2]

[1] *Department of Informatics and Telecommunications Engineering, University of Western Macedonia, 50100 Kozani, Greece*
[2] *Department of Informatics and Telecommunications, University of Peloponnese, 22100 Tripolis, Greece*

Correspondence should be addressed to Panagiotis Sarigiannidis; psarigiannidis@uowm.gr

Received 6 June 2018; Revised 2 August 2018; Accepted 7 August 2018; Published 2 September 2018

Academic Editor: Paul Honeine

In the emerging area of the Internet of Things (IoT), the exponential growth of the number of smart devices leads to a growing need for efficient data storage mechanisms. Cloud Computing was an efficient solution so far to store and manipulate such huge amount of data. However, in the next years it is expected that Cloud Computing will be unable to handle the huge amount of the IoT devices efficiently due to bandwidth limitations. An arising technology which promises to overwhelm many drawbacks in large-scale networks in IoT is Fog Computing. Fog Computing provides high-quality Cloud services in the physical proximity of mobile users. Computational power and storage capacity could be offered from the Fog, with low latency and high bandwidth. This survey discusses the main features of Fog Computing, introduces representative simulators and tools, highlights the benefits of Fog Computing in line with the applications of large-scale IoT networks, and identifies various aspects of issues we may encounter when designing and implementing social IoT systems in the context of the Fog Computing paradigm. The rationale behind this work lies in the data storage discussion which is performed by taking into account the importance of storage capabilities in modern Fog Computing systems. In addition, we provide a comprehensive comparison among previously developed distributed data storage systems which consist of a promising solution for data storage allocation in Fog Computing.

1. Introduction

During the last decade, an impressive development of integrated circuit and networking technologies has expanded the usage of computers from desktop computers and laptops to a wide range of devices specialized for use on specific purposes, such as environmental sensing, Global Positioning System (GPS) navigation, surveillance, home automation, and health monitoring. All of these devices consist of an ecosystem called the Internet of Things (IoT). The term IoT was firstly proposed from the Automatic Identification (Auto-ID) Labs in the Massachusetts Institute of Technology (MIT) in 1999 [1]. Four years later, the International Telecommunication Union (ITU) designates the standards for IoT [2]. By 2020, it is expected that 20.8 billion connected devices will exist [3], which requires automated mechanisms to overcome various aspects for their management. A significant percentage of IoT devices will be mobile, a fact that yields an extra complexity for their management.

The data management in the IoT domain becomes of paramount importance given the extremely large amount of data which is produced from a huge amount of interconnected devices [4]. In addition, a high percentage of the involved devices have limited resources, especially subject to their battery life, computation power, and storage capacity. To overcome these obstacles and manipulate data storage effectively, it is necessary to develop novel and efficient mechanisms. Cloud and Fog Computing seem to be an ideal solution to support data-demanding networks.

Storage resources of Fog nodes extend storage resources of the Cloud offering Storage as a Service (STaaS) with low latency to edge devices in geographical proximity. Furthermore, such a solution aims to enhance network bandwidth, mobility, security, and privacy. In particular,

(i) low latency is one of the main advantages of Fog Computing. Using Fog nodes the distance to the edge devices is minimized compared to legacy systems connecting Cloud and edge devices. As a result, latency is significantly lower, a fact that leverages the supported storage services in the Fog Computing paradigm;

(ii) Fog Computing is designed with the native support of mobile devices. Fog nodes can be dispersed across a wide geographical area, offering enhanced Cloud services to mobile devices;

(iii) security and privacy are two of the main pillars in the design of Fog Computing. When we are dealing with large-scale public networks, having a huge amount of private data, security and privacy are of paramount importance.

Providing storage services is one of the primary key features of Fog Computing [5–7]. However, implementing a distributed storage system, which will offer an abstraction storage layer to edge users, is still an open issue [8]. Cloud and Fog Computing seem to be an ideal solution to support data-demanding networks. Despite the great potential of Fog Computing, only a limited number of works is found in the literature related to data storage.

Three main service models [9, 10], for data storage and workload execution, are possible in Fog Computing, namely, the offloading, the aggregation, and the peer-to-peer model:

(i) In the offloading model, the data generated from edge devices are offloaded to the nearest Fog node and then at the Cloud (i.e., up-offloading) and in the reverse order from the Cloud to edge devices (i.e., down-offloading).

(ii) In the aggregation model, data streams generated by multiple edge devices are aggregated and possibly processed at the nearest Fog node before being uploaded to Cloud datacenter.

(iii) Finally, under the peer-to-peer (P2P) model, Fog nodes, which are at the proximity of edge devices, share their computing and storage capabilities and cooperate in order to offer an abstraction storage and computing layer to edge users.

This survey paper intends to familiarize readers with distributed data storage mechanisms in large-scale networks based on three tier levels, i.e., Cloud-fog-edge. We will focus on efficient data storage in Fog Computing which constitutes a Cloud to the edge continuum. Important capabilities and features of modern distributed data storage systems are discussed. The most compelling distributed storage systems are presented, analyzed, and discussed. In addition, the Social Internet of Things is discussed, and its requirements in data storage are analyzed.

The remaining of this paper is organized as follows: In Section 2, the basic concepts of Fog Computing are introduced. In Section 3, the known simulators and tools for Fog Computing are presented. In Section 4, the Social Internet of Things is discussed under the base of distributed storage. Section 5 is focused on the ongoing projects in Fog Computing. Section 6 presents the current state of the art in data storage associated with Fog Computing. In Section 7, the data storage requirements in large-scale networks are described and the benefits of utilizing Fog Computing are also discussed. Section 8 is focused on the most common techniques of distributed data storage and its related domains such as data dissemination and data replication. In Section 9, we provide a comprehensive analysis of the most important distributed storage systems. Finally, Section 10 concludes this survey paper while presenting a summary of future research trends.

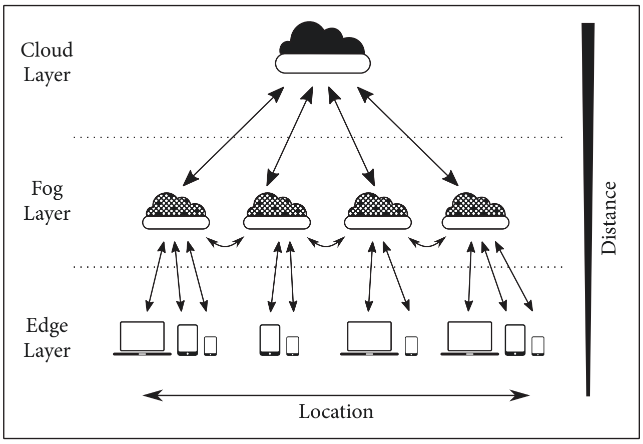


Figure 1: Fog Computing architecture.

2. Fog Computing

In large-scale IoT networks, central control combined with central data storage sounds necessary for supporting efficient and effective IoT infrastructures. To this end, Cloud Computing offers significant resources to IoT networks, such as computational power and storage capacity. Nevertheless, Cloud Computing comes with some drawbacks. One of the primary drawbacks of Cloud Computing is the latency induced in the intercommunication between an IoT device and the Cloud. A new layer is added in the IoT architecture to address this drawback, as illustrated in Figure 1, between the edge devices and the Cloud. This layer is called Fog Computing which introduces a new interesting research topic in the IoT domain. The term Fog Computing was first used by Bonomi et al. from Cisco [11], while its main feature is to extend Cloud Computing services at the edge of the network.

Fog Computing is characterized from intense virtualization [12]. Each Fog node can be composed of one or more devices, thus creating a virtual node to support the coverage area. These devices can be routers, switches, gateways, or specialized deployment of local servers. A comprehensive survey of what a Fog node could be can be found in [13].

Edge devices are connected with the Fog Computing layer, which in turn is connected to the centralized Cloud Computing layer. This type of connection forms a hierarchical computing platform architecture, where on the higher layer a plethora of computing power and storage capacity exist. On the contrary, limited computing power and storage capacity per device exist at the lower level.

The layer of Fog Computing can be organized in many different domains. Also, it can be distributed in different geographical areas in order to cover a wide area on large-scale IoT networks. For example, this architecture can be used to cover the network of a smart city, where one or more Cloud data centers are used for centralized control. Fog nodes are desirable to be dispersed in a city in various places, such as subway stations, bus stations, cell phone towers, public areas, public services buildings, department stores, and coffee shops or even at users home.

Edge devices, like smart phones, sensors, or actuators, can be connected through gateways with Fog nodes and gain Cloud-like services with low latency. This consists of the primary benefit of Fog Computing since Fog nodes are close enough to provide services within a few milliseconds. Instead, Cloud data centers need hundreds of milliseconds to provide the same services as a result of the large geographical distance. Fog Computing is able to remedy this phenomenon by applying data distribution, mobility support extension, and heterogeneity.

Data communication among nodes in the IoT domain can occur between edge-edge, edge-fog, fog-fog, and fog-Cloud nodes. Figure 2 illustrates a specific geographical area where Fog nodes communicate with each other via Wireless Sensor Networks (WSNs), Local Area Networks (LANs), or even 5th Generation (5G) networks. However, if a Fog node from area A wants to exchange data with a Fog node in area B, data should be transferred through an Internet connection which is a time-consuming and high-cost procedure.

3. Fog Computing Simulators and Tools

To evaluate Fog Computing architecture in real testbeds is costly and in most cases not practical since it requires many hundreds of Fog nodes and more than thousands of IoT devices. To overcome these limitations, a (small) number of simulators and tools exist which can be used to simulate such environments and measure various characteristics like latency, network congestion, and energy consumption. In this section, we analyze the state of the art in the domains of simulators and tools destined for Fog Computing.

3.1. iFogSim. iFogSim [14] is one of the most promising solutions for Fog Computing simulation and experimentation. It allows the design of different Fog architecture scenarios and the assessment of various aspects such as bandwidth and energy consumption. In iFogSim, any infrastructure can be simulated by adding Fog nodes and edge devices with various features including CPU, RAM, and storage capabilities. It supports different classes corresponding to Fog devices, sensors, tuples, actuators, and applications. A Graphical User Interface (GUI) is used to create network topologies as a supplementary of the programmatically ability to the creation of the whole infrastructure in JavaScript Object Notation (JSON) format via Java APIs.

By using iFogSim, we can measure performance metrics and simulate edge devices, sensors, network links, and Cloud datacenters. Moreover, iFogSim integrates simulated services for power monitoring and resource management in two separate levels, i.e., the application placement and the application scheduling. Two application module placement strategies are packaged to support multiple deployment scenarios, namely, (a) Cloud-only placement, where all applications modules run in data centers and (b) edge-ward placement, where application modules run on Fog nodes close to edge devices.

In [15] the authors presented an extension of iFogSim, which is able to optimize data placement in Fog nodes and IoT devices. The extension enables the management of data placement according to specific objectives such as minimization of service latency, network congestion, and energy consumption.

It is worth mentioning that iFogSim is based and implemented over CloudSim [16]. Since Fog concept has many similarities to Cloud one, CloudSim can also be used as a standalone application to implement many features of Fog Computing, although it has been designed as a Cloud Computing simulator.

3.2. EmuFog. EmuFog [17] is an emulator designed to support large-scale scalable topologies, by allowing the evaluation of various Fog Computing environments. Emulation of real applications is another supported objective, which allows developers to implement and evaluate their behavior as well as the induced workload the network topology. Finally, EmuFog supports extensibility, while allowing developers to override any component and adapt it according to their needs.

The implementation process in the emulated environment of EmuFog consists of four stages. First, a network topology is either generated or loaded from a file, supporting thus real-world topology datasets. Next, the network topology is converted in an undirected graph, where nodes represent network devices (e.g., routers) and links correspond to the connections between them. Latency and throughput values can be configured in a custom way. On the next stage, the edge devices are determined and the Fog nodes are placed according to a placement policy. Users are able to define the computational capabilities of Fog nodes as well as the number of clients expected to be served by each node. Finally, Fog nodes are emulated from the network emulated environment, while the applications in any individual Fog node are running under Docker containers.

3.3. FogTorchΠ. In [18] the FogTorchΠ prototype was introduced, which extends FogTorch [19]. It is an open source prototype developed in Java, able to evaluate Fog Computing infrastructures deployments, which fulfill various requirements in software, hardware, and QoS on latency and bandwidth. FogTorchΠ uses Monte Carlo simulations to implement variations in communications links, which are used as inputs. The final output consists of the aggregated results in terms of QoS-assurance and Fog resource consumption, by indicating, thus, the percentage of consumed RAM and storage.

4. Social Internet of Things

Social relations among humans are nowadays a part of their daily activity, such as exchanging knowledge or products, establishing friendships, and collaborating and working together.

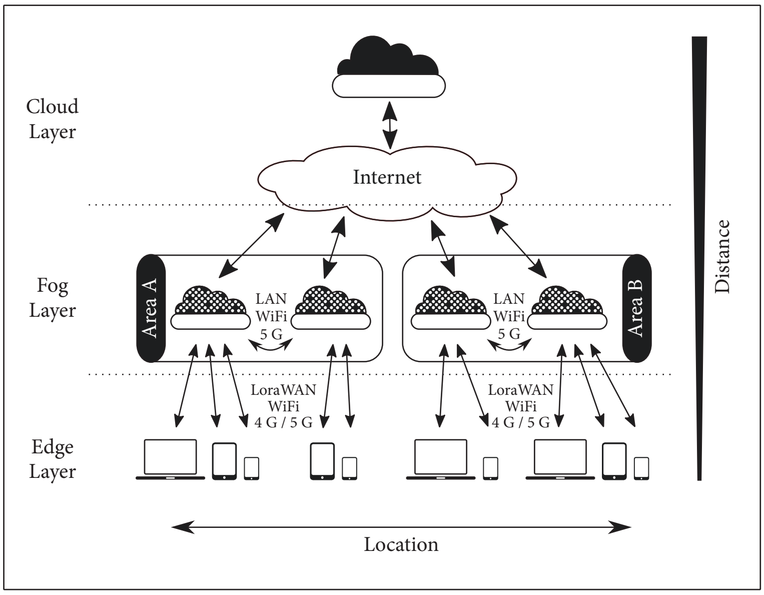


Figure 2: Communication between different domains of Fog Computing.

In analogy with real life, many works during the last years deal with the term of social IoT (SIoT), where smart objects can interact with each other to accomplish specific works or to share resources. SIoT is defined thoroughly in [20], where the social relationships between smart objects are divided into five main categories:

(i) Parental Objects Relationship (POR), which refers to homogeneous devices produced by the same manufacturer and probably belongs to the same production batch

(ii) Colocation Objects Relationship (C-LOR), which is established among devices, which are always in the same place

(iii) Cowork Objects Relationship (C-WOR), which refers to objects, where the objects cooperate on a common IoT application

(iv) Ownership Object Relationship (OOR), which is established among heterogeneous objects of the same user

(v) Social Object Relationship (SOR), which refers to objects that come into contact continuously or sporadically, as a result of their owners relationship.

Due to the limitations of most IoT devices on computation capabilities, storage resources, and energy consumption, IoT devices are not able to handle social relationships on their own. Only few works so far have proposed a suitable infrastructure to support SIoT and extend their capabilities.

Fog Computing can provide the required platform to support IoT devices at the edge, extending their capabilities to interact with each other under a social model. Virtualization, as one of the main features in Fog Computing, can be used to create virtual objects as representatives of the real SIoT devices. Thus, many limitations are addressed since computational capabilities, energy consumption, and storage resources are implemented in the Fog layer and if needed in the Cloud layer. In addition, management of Social Virtual Objects, which reside in the Fog layer, requires enhanced storage mechanisms supporting replication to ensure data integrity and migration methods from one Fog node to another in case the SIoT devices are moving.

Ivan Farris et al. [21] proposed a first approach of Cloud technologies at the edge for the implementation of a SIoT platform. They argue that the main features of this platform should be (a) low latency (b) scalability, (c) autonomy, (d) flexibility, and (e) mobility management. They call Social Virtual Objects (SVOs) the counterparts of the physical objects which are able to run in Cloud-like environments (e.g., Fog environment).

In [22], the authors presented a platform called Lysis, which is suitable to support Social Virtual Objects (SVOs). The platform entails that autonomous social agents exist which are capable of establishing social relationships. Although the authors do not directly propose Fog Computing as the infrastructure of Lysis, they claim that such a solution might help in reducing latency when an SVO follows the physical device.

The authors in [23] proposed an application platform for Fog-based Cyber-Physical Social Computing and Networking (CPSCN) systems. In particular, they developed an independent coordination model to support a reusable and scalable application model.

Trust and privacy are also of paramount importance in SIoT. Trust ensures faith between the social objects, whereas privacy secures the sensitive personal information. In this context, the authors in [24] proposed a novel architecture for the maintenance of trust and preservation of privacy rules in SIoT based on edge-crowd integration. For the implementation of the proposed solution, mini-edge servers are used as crowdsources.

A distributed attack detection system based on Fog Computing was proposed in [25]. The authors proved that such an approach is better in cyberattack detection than the centralized algorithms, because many parameters are exchanged among the participating nodes.

Social Internet of Vehicles (SIoV) [26] as an extension of SIoT applies the same concepts in the vehicular domain. Vehicles may interact with each other to share information about traffic, weather, and parking slots or even share media and news. Moreover, socializing in SIoV is not exclusive for vehicles, as drivers or passengers might participate in the social network. An extensible and scalable SIoV architecture based on Fog Computing is presented in [27].

5. Ongoing Research Projects in Fog Computing

Nowadays, a lot of work has been done in research projects towards investigating the Fog Computing architecture potential in line with exploring the benefits of supporting large-scale IoT networks. Big data, security, and privacy are some of the most important requirements in these projects. In this section, we provide the ongoing research projects on Fog Computing and discuss them in terms of architecture, security, privacy, and data storage.

5.1. DITAS. The European project DITAS [28] (data-intensive applications improvement by moving daTA and computation in mixed Cloud/fog environments) is focused on providing an abstraction layer for data storage by hiding the complex infrastructure based on different platforms, storage systems, and network capabilities. In more detail, it proposes a framework based on novel strategies for data and computation movements to decide when, where, and how to save data. Thus, the data could be saved on the Cloud, on the Fog nodes, or on edge devices taking into account security, privacy, reliability, sustainability, and performance metrics.

The proposed framework is composed of the Software Development Kit (SDK) and the Execution Environment (EE). The DITAS SDK is designed to allow developers to specify resources on Cloud or edge devices by using Virtual Data Containers (VDCs) and defining constraints and preferences. The abstraction layer described above is based on VDCs and allows developers to focus on data instead of where and how data are stored. The DITAS EE is able to manage the distributed architecture and maintain the computation and data movement as well as all other involved resources of the infrastructure.

5.2. PrEstoCloud. The PrEstoCloud [29] (Proactive Cloud Resources Management at the Edge for Efficient Real-Time Big Data Processing) is a European H2020 project, aiming at providing a configurable Fog Computing architecture in order to support Big Data streams at the edge. The proposed solution is based on five distinct conceptual layers: meta-management, control, Cloud infrastructure, Cloud/edge communication, and devices/layers. Logistics, mobile journalism, and security surveillance are the three pilots on which PrEstoCloud will be tested and demonstrated to prove its strength.

5.3. mF2C. The European project mF2C [30] (Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem) constitutes an open, secure, and decentralized management framework. The m2FC project will try to set the bases of a distributed system architecture based on Fog Computing with integrated novelty programming models, data storage techniques, privacy, and security, as well as innovative service creation, brokerage solutions, SLA policies, and resource orchestration methods.

The designed framework is tested on three real-world use cases. The first use case provides emergency situation management in a smart city. The second one is focused on the unreliability of the connection to the Cloud and is tested in a Smart Boat where the mF2C platform is deployed between onboard devices and the Cloud. The third use case is taking place in an airport where a Smart Fog Hub (SFH) service is deployed to interact with all objects within the scope of coverage and offer value-added services on marketing, prediction of path, or behavior of the consumers and take real-time decisions.

5.4. REDESIGN. The REDESIGN (distRibutED, sElf-adaptable, and Scalable Wireless foG Networks) is a European project which will start in 2019. REDESIGN aims to design distributed and scalable Wireless Fog Networks (WFNs) with ground and mobile (through drones) Fog nodes. The designed WFNs will be integrated into cellular networks and each node will be able to continuously sense the network topology and autonomously configure network parameters by using deep learning algorithms in order to guarantee the required QoS. In particular, Fog nodes will be self-adaptable and guarantee requirements for high energy-efficiency, high data-rate, and high-reliability applications for IoT, such as smarter power grids, disaster management, and healthcare. In conclusion, the REDESIGN project aims to propose a novel distributed, software-based, and self-adaptable, core-centric cellular network to support new Fog Computing applications.

5.5. FOGHORN. The FOGHORN (FOG-aided wireless networks for communication, caching and computing: theoretical and algorithmic foundations) is a European project which aims at developing the theoretical and algorithmic bases of fog-aided wireless networks. FOGHORN focuses on fundamental theoretical insights and algorithmic principles for problems such as optimal communication management and efficient caching and computing resources of the Fog Computing architecture. The theoretical framework is based on network information theory, signal processing, and distributed computing to develop and analyze the algorithmic solutions.

5.6. RECAP. The European project RECAP [31] (Reliable Capacity Provisioning and Enhanced Remediation for Distributed Cloud Applications) aims to develop the next generation of Cloud/Fog Computing architecture with elastic services in the proximity of end users, according to the user needs. It will be built on advanced machine learning, optimization, and simulation techniques. It will provide advanced state-of-the-art features such as automation of the creation of applications, while collecting and synchronizing data in multiple nodes allocated in different geographical areas. In addition, it will advance the automation process for detecting and correcting failures at the network and the infrastructure, while maintaining the QoS.

The RECAP project encloses four use cases: an infrastructure and network management use case to demonstrate how automated service characteristics can ensure the desired QoS and a second use case for complex Big Data analytics engine to evaluate if complex applications and virtual data centers can be modeled and automatically improved in Cloud environments to improve performance and reduce costs. A Fog Computing and large-scale IoT infrastructure will be also implemented as a use case to demonstrate the capabilities of resource allocation automated techniques at the edge of the network in order to reduce latency. Lastly, a use case for Network Function Virtualization, QoS management, and remediation will demonstrate how these techniques can mitigate different types of failures thanks to the automation of orchestration and scheduling and rescheduling of virtualized network functions.

5.7. CHARIOT. The project CHARIOT [32] (Cognitive Heterogeneous Architecture for Industrial IoT) is a European project aiming to provide a unified approach towards Privacy, Security, and Safety (PSS) for IoT Systems based on Fog Computing. The project includes novel applications such as a privacy and security protection method based on Public Key Infrastructure (PKI) technologies. In addition, Blockchain technology will be responsible for authorizing any IoT device entering the system as well as recording and approving any change occurred.

Three Living Labs will be implemented to demonstrate and evaluate the proposed solutions regarding safety and privacy. The Living Labs will take place at the Italian railways, at the IBM Ireland Campus and at the International Airport of Athens.

5.8. SOFIE. The European project SOFIE [33] (Secure Open Federation for Internet Everywhere) is envisioned on creating a secure and open IoT federation architecture and framework. Through the project, decentralized solutions with virtualization and unlimited scalability will be implemented along with Blockchain connections and interledger technologies, to enable actuation, auditability, smart contracts, and management of identities and encryption keys. Hence, end-to-end security, key management, authorization, accountability, and auditability will be implemented. SOFIE will be based on existing open standards, such as FIWARE, W3C Web of Things (WoT), and oneM2M. Three use cases will demonstrate the developed approach in three different sectors: food chain, gaming, and energy market.

5.9. 5G-CORAL. The 5G-CORAL [34] (5G Convergent Virtualized Radio Access Network Living at the Edge) is a European project aiming to leverage the Edge and Fog Computing in the Radio Access Network (RAN). The proposed solution is composed of two major building blocks, namely, (a) the Edge and Fog Computing System (EFS), including the Edge and Fog computing infrastructure, which offers a shared hosting environment suitable for virtualized functions, services, and applications, and (b) the Orchestration and Control System (OCS) which is responsible for managing and controlling the EFS. Three large-scale testbeds will be deployed in Taiwan to evaluate the implemented system in domains such as augmented reality, car safety, and IoT gateway.

6. Data Storage Research Efforts

Fog Computing constitutes an emerging technology which will undoubtedly contribute to the design of the future Internet. Companies like Cisco, Dell, Intel, and Microsoft are participating in an open fog consortium to define the appropriate standards. In addition, many research papers were published in the last years, but only few of them are related to the area of data storage. This section is devoted to presenting the state of the art in data storage of the Fog Computing research field.

In [35], the authors proposed a hybrid data dissemination framework to utilize the fog-Cloud bandwidth and guarantee the download performance. In detail, they analyze the Fog Computing network in two planes. The first plane is attached to the Cloud, which composes a control plane to process content update queries and organize data flows. The second plane is related to the Fog network using geometrically distributed nodes. The proposed framework tries to deliver data among Fog nodes with delay tolerant techniques using mobile devices which belong to humans or vehicles.

A data dissemination method using instantly decodable binary codes is proposed in [36]. The method is applied to a Fog Radio Access Network (F-RAN), while trying to reduce the communication time required to spread all files between devices.

In [37], the authors proposed a Fog-supported peer-to-peer (P2P) architecture for wireless content delivery. The main concept is to enhance the speed of data searching operations performed at serving Fog nodes. In addition, they proposed an adaptive probabilistic search algorithm, which is distributed over the available nodes. The proposed algorithm implements file searching with low latency over Fog-supported P2P overlay networks.

In [38], the authors discussed and evaluated three object store solutions, i.e., the Rados [39], the Cassandra [40], and the InterPlanetary File System (IPFS) [41], in terms of Fog Computing suitability. The evaluation was based on measuring the access time for data object delivering and on the produced amount of network traffic.

The authors in [42] proposed an object store service based on a revised version of IPFS [41] to meet the requirements of the Fog Computing. Their approach leverages a scale-out Network Attached Storage system (NAS) as a storage backend of IPFS nodes. The evaluated results show improvement in access times as well as in the produced amount of network traffic.

In [43], the authors proposed the FogStore system that provides novel design goals for existing distributed data systems. In detail, a Fog-aware replica placement algorithm was described in addition to a context-sensitive differential consistency.

The authors in [44] proposed a model for enabling data movements in Fog Computing environment by transferring data at the right time in the right place according to the needs of several customers. Data storage systems may belong to data providers as well as to data consumers.

In [45], a processing optimization mechanism was proposed aiming to synchronize storage and computation in Fog environment. The suggested mechanism intends to minimize the processing cost of the network in computation and storage level, while improving the overall performance over investment.

In [46], the authors introduced a computation offloading and storage expansion of mobile devices in Fog Computing stratum. According to this research effort, Fog Computing can leverage the intelligence and storage capabilities of resource-constrained mobile devices.

Caching problem in F-RAN was the main topic discussed in [47–50]. All these research endeavors aim to reduce latency and download delay to end users.

The authors in [51] focused on data availability on unreliable P2P networks by using Fog nodes as the peers trying to provide a reliable data storage framework. They provide a mathematical analysis on formulating different storage and network use conditions, while they developed a practical P2P system with Raspberry Pi devices to evaluate their approach.

In [52], the authors discussed two coding concepts that use the available computing resources of Fog nodes. They also considered the possibility of applying them in large-scale systems in order to reduce the bandwidth consumption and latency of delivery of files.

7. Distributed Data Storage Requirements

When considering distributed data storage architectures and schemes, it is clear that a lot of research efforts can be found in the literatures that are related to Cloud Computing. Of course, Cloud Computing is synergic with Fog Computing; thus, a variety of Cloud-related data storage schemes can be adopted or can be slightly altered to meet the requirements of Fog Computing. However, the major drawback to this end is the high latency the Cloud-related services present when dealing with edge devices. Fog Computing comes to surpass this obstacle and offers Cloud-like services with low latency. To manipulate the enormous amount of data produced or consumed from edge devices, which probably yields heavy network traffic, there is a need for improved algorithms able to support data storage distribution and related mechanisms.

The data storage distribution problem is defined as the distribution of managed data in a fairly manner to all nodes of the network. Storage capacity is an essential parameter of this distribution process but many other parameters have to be also taken into account for ensuring efficient and reliable data storage. For example, we have to take into consideration many aspects of the whole system, such as fault tolerance, heterogeneity, scalability, bandwidth consumption, low latency, security, and energy consumption. In the following subsections, these critical parameters are discussed in detail.

7.1. Fault Tolerance. Fault tolerance means that there should be no data loss in case of node failure due to occasionally restarts or shutdown, battery drainage, or system failure. To achieve that, many replication mechanisms have been proposed, which duplicate data from one node to another to ensure data integrity.

In Fog Computing, Fog nodes may also be sensitive to data loss. A simple solution would be to replicate all data in the Cloud, but although Cloud is characterized by sufficient storage, there is a limit in the storage capacity. Hence, replication mechanisms should also be used between Fog nodes as well as between Fog and Cloud nodes when necessary. Replication mechanisms can take advantage of two or more layers of Fog nodes.

7.2. Heterogeneity. Data produced in IoT could be derived from different sources with different characteristics. For example, in a smart city thousands of sensors exist which can measure the temperature and humidity of the atmosphere or can observe the traffic on highways. Additional applications are health monitoring of patients or old people, smart meters for power, gas or water, and, finally, more bandwidth consuming applications, like streaming and web browsing that can coexist in a smart city environment. The complexity of the generated data, combined with the heterogeneity of the involved devices, entails innovative mechanisms for their management in order to store the corresponding data efficiently at the most suitable position.

Fog nodes may be from different vendors with different characteristics like process speed, network bandwidth, and memory or storage capacity. Thus, heterogeneity in the Fog layer is a fact which should be taken into consideration in designing of any mechanism. Conversely, Fog layer offers virtualized services at edge devices in such way that they will not realize differences from one Fog node to another. For example, when an edge device requests a data unit from its nearest Fog node, it will not care where data is stored and how it will be transferred. In addition, in case a device moves from one Fog node to another, the Fog layer has the responsibility of data delivery.

7.3. Scalability. Modern networks are characterized by intense scalability since it is quite likely to have nodes able to move in any direction or nodes that can be added or removed at any time, changing thus the network topology. To cope with these phenomena, a data storage mechanism should have dynamic knowledge of the network and should be using tools to prevent data loss in case of node subtraction, system shutdown, or moving outside of the network area.

Scalability exists at both Fog and edge layers. Edge devices are commonly moving, and they can log in or log out from the network at any time. Although Fog Computing has not been completely defined, many research works describe Fog nodes as moving nodes which are free to be connected or not in the network at any time [53].

7.4. Bandwidth Consumption. Bandwidth is one of the valuable resources in modern networks. In addition, due to the expected exponentially increasing of traffic, the limited network bandwidth is likely to be an obstacle in the Fog layer. Hence, proactive data distribution strategies should be adapted to reduce data exchange among Fog nodes as well as among Fog and Cloud nodes.

7.5. Low Latency. Fog Computing differentiates with Cloud Computing towards solving the low latency obstacle [54]. However, it is crucial to keep data near the place we need it in order to achieve low latency on data access as well as to decrease network traffic. For example, in conventional smart city environments, the measurements of the smart meters should be forwarded at a node close to the headquarters of the corresponding company. Hence, when the company requests the measurements, the query will be delivered with as low latency as possible.

7.6. Energy Consumption. Energy consumption is not a critical parameter for Fog nodes since most of the nodes are expected to have a permanent power supply. Nevertheless, as the number of Fog nodes exponentially increases compared to the number of the existing Cloud servers, the development of new distribution algorithms should highly consider the decrease of the carbon footprint. To cope with this, low weight data dissemination schemes should be used to decrease network traffic. In addition, keeping the corresponding data where we need it, except achieving low latency, decreases the overall energy consumption of the whole network. For example, an energy-efficient virtualized Fog Computing architecture was proposed in [10, 55], for dynamic resource management of real-time streaming applications.

7.7. Security and Privacy. Security is one of the most important parameters in the design of new systems, especially when they are related to Fog and Cloud Computing. In addition, privacy is another critical parameter when dealing with data storage. A comprehensive survey on security and privacy issues in Fog Computing can be found in [56].

Fog nodes may be owned by different vendors as well as users who intend to share their storage resources under a leasing plan. Since Fog nodes have local proximity, sensitive and private user information may be exposed such as when users are at their homes, which placed they visit, and what kind of services they use. Strong encryption schemes have to be applied in order to store all these data safely. Moreover, isolation is also applied in Fog Computing for ensuring data privacy.

Many different cases of attacks may occur in the case of Fog Computing such as Man in the Middle attack (MitM), Denial of Service (DoS), and Sybil attack. A Fog node which belongs to an individual user may be malicious even when it declares itself legitimate. Authentication protocols are needed among the various Fog nodes for recognizing a nodes identity. Associating an incoming request with a set of identifying credentials in Fog Computing environment, many security threats can be prevented. Authentication mechanisms seem to be necessary since Fog nodes are densely deployed at a large scale, while they are characterized by enhanced automation with minimal human intervention. Autoupdate or even remote update should be used to enhance the resilience of Fog nodes. Finally, intrusion detection systems would be an excellent solution to prevent cyberattacks in Fog Computing.

Eventually, the Fog layer will provide a trusted distributed platform over the edge devices. It will be able to manage and update security credentials, scanning for malware, and distribute timely software patches.

8. Distributed Data Storage Mechanisms

In Fog Computing, both control and data storage are performed centrally. However, storing everything in the Cloud is not realistic because of (a) the high latency [57], (b) the high traffic demand between edge and Cloud, and (c) the high storage cost. In addition, the mobility of the Fog or edge devices is an important parameter under consideration. By distributing stored data at the Fog layer depending on various parameters, such as geographical location of producers and consumers, the aforementioned drawbacks could be addressed. This solution lies in the central control of Cloud Computing in combination with the distributed data storage of Fog nodes [58]. In particular, load balancing techniques can be applied in changing network topology over time based on the data usage needs from consumers in order to achieve time dependency.

Data storage mechanisms consist of various research areas such as data distribution, data dissemination, and data replication, which will be further discussed in this section.

8.1. Data Distribution. In large-scale networks where there is a huge amount of cooperating devices, data distribution strategies are able to eliminate data storage restrictions by spreading data fairly to all nodes. In Fog Computing, data should be distributed to Fog nodes. To the best of our knowledge, there are a few related works in Fog Computing to this end [59–61].

Big companies like Amazon and Google implemented their own solutions to support their storage needs in their highly demanding environments in complex infrastructures. In addition, P2P storage systems have been proposed as a distributed storage solution for distributed computing in many research efforts over the past years. Meanwhile, P2P storage techniques have been used in special systems like Content Delivery Systems (CDN) [59] and Cloud Computing. Thus, P2P storage techniques appear to be an appropriate storage mechanism to manipulate enormous information in Fog Computing paradigm. Such systems can transfer data between different Fog nodes of the same vendor to provide STaaS to edge devices. In addition, different vendors can share their storage resources under a business model to decrease the cost of installation and maintenance of extra Fog nodes and in parallel to increase their capabilities in cases of high demand.

P2P systems have been evolved from unstructured P2P networks, where node connections were arbitrarily established, to structured P2P systems where node connections follow a predefined form. Many systems in the latter approach use routing mechanisms to ensure a maximum number of search steps, which is desirable in Fog Computing in order to reduce latency.

8.2. Data Dissemination. Data dissemination is a topic well investigated in WSNs [62–65]. Furthermore, in low bandwidth networks, dissemination algorithms are applied in determining the shortest path. In addition, dissemination algorithms can be adapted in deciding different routes of transmission based on current load of network connections.

In Fog Computing, dissemination mechanisms can be used among different Fog nodes to reduce the overall traffic by deciding the best method of a transmission. Data dissemination mainly takes place among Fog nodes. However, it is possible for data dissemination to even involve interconnection among (a) Cloud and Fog nodes and (b) Fog nodes and edge devices. In the last case, edge devices can extend the Fog layer by taking the role of communication paths to deliver a package [35].

8.3. Data Replication. Replication mechanisms are significant to ensure data integrity; therefore, they are highly recommended in infrastructures where failure is not only common but pervasive [60]. Data loss may occur for various reasons such as system failures or battery drainage. System failures are very likely to happen in Fog nodes. Thus, replication mechanisms are also crucial at the Fog layer. Replication can be applied over entire virtual machines as well as over data record. In the first case, replicas are only used to ensure data integrity. Instead, in the second case, replicas can be kept in another location where it is more probably to be queried from another edge device. In addition, data consistency should be also taken into account to ensure that all replicas are updated before they will be delivered to an edge device.

Replication strategies are used to improve data integrity and to increase data availability with low latency and retrieval time [61]. However, high amounts of data redundancy reduce the overall storage capacity of the system and increase maintenance costs.

9. Distributed Storage Systems Compilation and Discussion

The most important distributed storage systems are evaluated and discussed in this section in terms of features related to Fog Computing. The evaluation involves the most recent and compelling systems such as the Granary [66], the Comet [67], the ElaStraS [68], the PAST [69], the OceanStore [70], the IPFS [41], the Dynamo [71], the Facebook Cassandra [40], and the Google File System (GFS) [72].

A distributed storage system has to meet certain requirements for being utilized in Fog Computing environment. Such systems should be able to achieve high performance with low latency in heterogeneous systems featured by hundreds or thousands Fog nodes. In this context, a node failure is deemed as a fact rather than a rare incident. Thus, consistency and integrity should be supported by replication mechanisms.

Moreover, when dealing with large-scale networks, which are scattered in large geographical areas, where a number of mobile devices is supported, load balancing algorithms are required for improving the availability of data with low latency at specific areas in a specific time having high demand of specific data. Load balancing algorithms could be implemented either through adaptive replication mechanisms or through intelligent caching techniques.

Data partitioning and replication with the use of erasure codes [52, 73] require less redundancy than simple replication of whole files and offer the same level of reliability.

In order to use Cloud storage, as a supporting infrastructure and not as a primary one, either decentralized implementation or semicentralized is also desirable. In the first case, the Cloud can play an equal role with Fog nodes, for example, as an equivalent peer in case of a P2P network. In the second case, the Cloud can take the central role of the orchestration of the Fog nodes and can share its storage resources when necessary.

Ensuring security and privacy preserving are two of the most critical parameters in Fog Computing because large-scale networks become potentially vulnerable to any (external) attacker. Thus, encryption and security mechanisms in addition to intrusion detection systems are mandatory.

Extra features of a distributed storage system are also significant for the quality of service, such as operation of a performance monitoring manager, error detection mechanism, automatic recovery system, Application Programming Interfaces (APIs) for third-party implementations, file versioning, and filesystem.

Table 1 illustrates the main characteristics of the evaluated distributed storage systems such as the type of replication mechanism, the level of scalability, the existence of security or/and privacy mechanisms, and the support of load balancing methods based on locality demand of data to reduce latency. In addition, the analysis involves other important features such as the license type, the element type of data storage unit, the ability to maintain data versioning, data partitioning on data elements, and file system support to the edge user.

Table 2 provides the pros and cons for each one of the compared distributed storage systems. We briefly compare them regarding performance, security, privacy, and extra features.

In [66], the authors proposed Granary, a wide-area data storage system based on P2P technology. The infrastructure of Granary is based on dedicated nodes at a global scale and can provide online storage services to end users. The system supports file upload, download, and modification. An important feature of Granary lies in its file sharing among end users. Granary stores meta-data file properties in a distributed hash table storage layer which is also distributed on wide-area systems. A replication mechanism is used to support consistency and to enhance file sharing with low bandwidth consumption.

Table 1: Comparison of distributed data storage.

	License	Replication	Load Balancing	Security Privacy	Scalability	Element Type	Data Versioning	Data Partitioning	File System
Granary [66]	Open	Adaptive	Replication	-	Medium	File	-	-	Yes
Comet [67]	Open	Adaptive	Replication	Yes	High	Object	-	-	-
ElaStraS [68]	Open	Adaptive	Caching	-	High	Table	Yes	Yes	-
PAST [69]	Open	Random	Caching	Yes	High	File	-	-	-
Ocean Store [70]	Open	Adaptive	Replication	Yes	High	Object	Yes	Yes	Yes
IPFS [41]	Open	Adaptive	-	Yes	High	Object	Yes	-	Yes
Dynamo [71]	Internal at Amazon	Fixed	-	-	High	Object	Yes	Yes	-
Cassandra [40]	Internal at Facebook	Adaptive	-	-	High	Table	-	Yes	-
GFS [72]	Internal at Google	Adaptive	-	-	High	File	Yes	Yes	Yes

In [67], the authors proposed Comet, a key-value storage system applicable to large-scale distributed storage systems. It encloses the Active Storage Object (ASO) technique which involves a key, an associated value and optionally a set of simple handlers. The implementation of ASO offers extra flexibility as it allows multiple applications with diverse requirements to share a typical storage system of a distributed infrastructure.

ElasTras was introduced in [68], where the scalability and the elasticity of data storing are addressed in Cloud Computing. ElasTras is suitable for applications which are designed for transactions to a single database partition.

A large-scale, Internet-based storage utility, named PAST, was proposed in [69]. It is self-organized and can provide strong assurances, efficient storage access, scalability, and load balancing. Moreover, PAST provides a simple, lean storage abstraction for insistent, immutable files and targets to global, archival storage.

OceanStore [70] is a utility designed to provide continuous access to persistent information. It is based on an infrastructure with untrusted servers, so it uses cryptographic techniques to protect the privacy of data and applied data redundancy for reliability. In addition, OceanStore supports a replication strategy to improve performance when access request overwhelms a specific replica.

The authors in [41] proposed the InterPlanetary File System (IPFS) which constitutes a P2P distributed file system. It attempts to provide a single file system for all connected nodes in a Fog environment. It is designed for applications such as encrypted data sharing system and root file system of a Virtual Machine.

Dynamo is a key-value storage system which was presented in [71]. It is used by Amazon to support storage services with demands of high availability. The underlying infrastructure consists of tens of thousands of data centers distributed worldwide. In such enormous number of nodes, failures are a common issue. Thus, scalability, which is the main feature of Dynamo, sounds as a significant feature. To achieve high availability, Dynamo sacrifices consistency in specific failure scenarios.

In [40], the authors presented the Cassandra, which aims to run on top of an infrastructure with hundreds of nodes. Cassandra has been developed by Facebook to support their complex infrastructure and their demanding storage needs. It provides scalability, high performance with low latency, and wide applicability.

In [72], the authors presented the Google File System (GFS) which has been designed and implemented by Google to support its storage needs. GFS is a distributed file system deployed across thousands of disks over a thousand of machines and accessed by hundreds of clients. A GFS cluster has a centralized control through a single master server and multiple chunkservers to store chunk data. Data is divided into chunk blocks with a size of 64MB and distributed across chunk servers. Replication placement policy tries to maximize data availability and reliability and to maximize network bandwidth utilization.

10. Conclusions

Although distributed data storage has been analyzed and used in Cloud Computing and Internet-based infrastructure, it is still an open issue when we are dealing with Fog Computing. Depending on the Fog Computing environment, different features are indicated. For instance, in case of static Fog nodes, algorithms should support optimal storage capacity determination at each location depending on data demand of edge devices with geographical proximity. On the other hand, in case of mobile Fog nodes, algorithms that move Fog nodes on demand when extra storage space are indicated.

To this end, the desired requirements and features of modern distributed storage systems are presented and discussed. Moreover, the most important and compelling storage systems in Fog Computing environments are introduced, evaluated, and discussed.

Table 2: Pros and cons of the distributed data storage techniques.

	Pros	Cons
Granary [66]	Evaluated on real environment.	Evaluated on few storage nodes.
	File sharing between edge users.	Scalability limitations due to routing.
	Load balancing based on replication mechanisms.	Does not support security and privacy.
Comet [67]	Increased flexibility.	It is not an autonomous distributed storage system.
	Monitoring mechanism for third party implementations.	Scalability limitations due to routing.
	Does not support load balancing.	
	Supports security and privacy.	
ElaStraS [68]	Light-weight implementation.	Does not support load balancing.
	Integrates a monitoring manager.	Does not support security and privacy.
PAST [69]	Storage utilization can approach 100%.	Not evaluated in real environment.
	Modular design provides extra flexibility.	Smartcards reduce performance in write operations.
	Smartcards offers extra flexibility and security.	Suitable as an archive storage system.
	Load balancing - caching mechanisms (popularity/locality).	
	Supports security and privacy.	
	Open source under BSD licence.	
Ocean Store [70]	Provides observation and optimization techniques.	Not evaluated in real environment.
	Supports load balancing based on a replication mechanism.	Not an active project during the last years.
	Supports security and privacy.	
	Can support commercial business model.	
	Open source under BSD licence.	
IPFS [41]	Can be used as a filesystem in a Virtual Machine.	Not evaluated in real environment.
	Supports file sharing between edge users.	Does not support load balancing.
		Does not support security and privacy.
Dynamo [71]	Evaluated and used internally at Amazon in a real environment.	Sacrifices consistency under certain failures scenarios.
	Achieves high availability and guaranteed performance.	Does not support security and privacy.
	Optimization based on application needs.	Does not support load balancing
	System monitoring, error detection and configuration management.	
Cassandra [40]	Evaluated and used internally at Facebook in a real environment.	Does not support security and privacy.
	High update throughput and low latency.	Does not support load balancing.
	High performance and wide applicability.	
	Integrated distributed performance monitor tool.	
	Open Source under Apache License.	
GFS [72]	Evaluated and used internally at Google in a real environment.	Garbage collection mechanism could be unstable.
	High aggregate performance.	Does not support security and privacy.
	Constant monitoring, error detection and automatic recovery mechanism.	Does not support load balancing

The next few years Fog Computing is expected to set the bases of the future Internet, while many future applications such as autonomous driving, vehicular computing, smart energy grid, and smart cities will be based on this infrastructure.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Sarma, D. Brock, and K. Ashton, "The networked physical world, auto-id center," *Auto-ID Labs White Paper*, 2000.

[2] I. Peña-López et al., *Itu Internet Report 2005: The Internet of Things*, 2005.

[3] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the Internet of Things in a Quantum World," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 116–120, 2017.

[4] Q. Yongrui, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: a survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137–153, 2016.

[5] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, 2017.

[6] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proceedings of the 3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 73–78, 2015.

[7] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges," *IEEE Communications Surveys & Tutorials*, 2017.

[8] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and Software Architecture for Fog Computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 44–53, 2017.

[9] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, "Fog of Everything: Energy-Efficient Networked Computing Architectures, Research Challenges, and a Case Study," *IEEE Access*, vol. 5, pp. 9882–9910, 2017.

[10] P. G. Vinueza Naranjo, E. Baccarelli, and M. Scarpiniti, "Design and energy-efficient resource management of virtualized networked Fog architectures for the real-time support of IoT applications," *The Journal of Supercomputing*, vol. 74, no. 6, pp. 2470–2507, 2018.

[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–16, August 2012.

[12] M. Aazam and E.-N. Huh, "Fog Computing: The Cloud-IoT/IoE Middleware Paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, 2016.

[13] E. M. Tordera, X. Masip-Bruin, J. Garca-Almiñana et al., "What is a fog node a tutorial on current concepts towards a common definition," 2016, https://arxiv.org/abs/1611.09193.

[14] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[15] M. I. Naas, J. Boukhobza, P. Raipin Parvedy, and L. Lemarchand, "An Extension to iFogSim to Enable the Design of Data Placement Strategies," in *Proceedings of the 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp. 1–8, May 2018.

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[17] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran, "EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures," in *Proceedings of the 2017 IEEE Fog World Congress (FWC)*, pp. 1–6, Santa Clara, CA, USA, October 2017.

[18] A. Brogi, S. Forti, and A. Ibrahim, "How to best deploy your fog applications, probably," in *Proceedings of the 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017*, pp. 105–114, 2017.

[19] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1–8, 2017.

[20] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (SIoT)—when social networks meet the internet of things: concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[21] I. Farris, R. Girau, L. Militano et al., "Social Virtual Objects in the Edge Cloud," *IEEE Cloud Computing*, vol. 2, no. 6, pp. 20–28, 2015.

[22] R. Girau, S. Martis, and L. Atzori, "Lysis: A platform for iot distributed applications over socially connected objects," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 40–51, 2017.

[23] N. K. Giang, R. Lea, and V. C. Leung, "Exogenous Coordination for Building Fog-Based Cyber Physical Social Computing and Networking Systems," *IEEE Access*, vol. 6, pp. 31740–31749, 2018.

[24] V. Sharma, I. You, D. N. K. Jayakody, and M. Atiquzzaman, "Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social Internet of Things," *Future Generation Computer Systems*, 2017.

[25] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[26] M. Nitti, R. Girau, A. Floris, and L. Atzori, "On adding the social dimension to the Internet of Vehicles: Friendship and middleware," in *Proceedings of the 2014 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2014*, pp. 134–138, May 2014.

[27] T. A. Butt, R. Iqbal, S. C. Shah, and T. Umar, "Social internet of vehicles: Architecture and enabling technologies," *Computers Electrical Engineering*, vol. 69, pp. 68–84, 2018.

[28] "Project DITAS (Data-intensive applications Improvement by moving daTA and computation in mixed cloud/fog environmentS)," 2018, https://www.ditas-project.eu.

[29] "Project PrEstoCloud (Proactive Cloud Resources Management at the Edge for Efficient Real-Time Big Data Processing)," 2018, https://prestocloud-project.eu.

[30] "Project mF2C (Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem)," 2018, https://www.mf2c-project.eu.

[31] "Project RECAP (Reliable Capacity Provisioning and Enhanced Remediation for Distributed Cloud Applications)," 2018, https://recap-project.eu.

[32] "Project CHARIOT (Cognitive Heterogeneous Architecture for Industrial IoT)," 2018, https://www.chariotproject.eu.

[33] "Project SOFIE (Secure Open Federation for Internet Everywhere)," 2018, https://www.sofie-iot.eu.

[34] "Project 5G-CORAL (5G Convergent Virtualised Radio Access Network Living at the Edge)," 2018, http://5g-coral.eu.

[35] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-Based Data Dissemination Model in Fog Computing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, 2017.

[36] A. Douik and S. Sorour, "Data dissemination using instantly decodable binary codes in fog-radio access networks," in *Proceedings of the 13th IEEE International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, pp. 604–609, June 2017.

[37] M. Shojafar, Z. Pooranian, P. G. V. Naranjo, and E. Baccarelli, "FLAPS: bandwidth and delay-efficient distributed data searching in Fog-supported P2P content delivery networks," *The Journal of Supercomputing*, pp. 1–22, 2017.

[38] B. Confais, A. Lebre, and B. Parrein, "Performance analysis of object store systems in a fog and edge computing infrastructure," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIII*, pp. 40–79, Springer, 2017.

[39] S. A. Weil, A. W. Leung, S. A. Brandt, and C. Maltzahn, "RADOS: A scalable, reliable storage service for petabyte-scale storage clusters," in *Proceedings of the 2nd International Petascale Data Storage Workshop, PDSW '07, held in Conjunction with Supercomputing '07*, pp. 35–44, November 2007.

[40] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

[41] J. Benet, "Ipfs-content addressed, versioned, p2p file system," 2014, https://arxiv.org/abs/1407.3561.

[42] B. Confais, A. Lebre, and B. Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS," in *Proceedings of the 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017*, pp. 41–50, IEEE, Madrid, Spain, 2017.

[43] R. Mayer, H. Gupta, E. Saurez, and U. Ramachandran, "FogStore: Toward a distributed data store for Fog computing," in *Proceedings of the 2017 IEEE Fog World Congress (FWC)*, pp. 1–6, Santa Clara, CA, October 2017.

[44] P. Plebani, M. Salnitri, and M. Vitali, "Fog computing and data as a service: A goal-based modeling approach to enable effective data movements," in *Proceedings of the International Conference on Advanced Information Systems Engineering*, pp. 203–219, Springer, 2018.

[45] Z. Song, Y. Duan, S. Wan et al., "Processing Optimization of Typed Resources with Synchronized Storage and Computation Adaptation in Fog Computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3794175, 13 pages, 2018.

[46] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *Proceedings of the 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops (SECON Workshops)*, pp. 1–6, Seattle, WA, USA, June 2015.

[47] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in Fog-RANs: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7039–7051, 2017.

[48] A. Roushdy, A. S. Motahari, M. Nafie, and D. Gunduz, "Cache-aided fog radio access networks with partial connectivity," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.

[49] J. S. P. Roig, F. Tosato, and D. Gündüz, "Storage-latency trade-off in cache-aided fog radio access networks," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, 2018.

[50] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning based edge caching for fog-ran," 2018, https://arxiv.org/abs/1801.06449.

[51] F. H. Fitzek et al., "On network coded distributed storage: How to repair in a fog of unreliable peers," in *Proceedings of the International Symposium on Wireless Communication Systems (ISWCS)*, pp. 188–193, 2016.

[52] S. Li, M. A. Maddah-Ali, and A. Salman Avestimehr, "Coding for Distributed Fog Computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 34–40, 2017.

[53] S. Alonso-Monsalve, F. Garcia-Carballeira, and A. Calderon, "Fog computing through public-resource computing and storage," in *Proceedings of the 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017*, pp. 81–87, May 2017.

[54] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Networks*, vol. 5, no. 2, pp. 23–29, 2016.

[55] N. Cordeschi, M. Shojafar, D. Amendola, and E. Baccarelli, "Energy-efficient adaptive networked datacenters for the QoS support of real-time applications," *The Journal of Supercomputing*, vol. 71, no. 2, pp. 448–478, 2015.

[56] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 685–695, Springer, 2015.

[57] W. Ramirez, X. Masip-Bruin, E. Marin-Tordera et al., "Evaluating the benefits of combined and continuous Fog-to-Cloud architectures," *Computer Communications*, vol. 113, pp. 43–52, 2017.

[58] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "An architecture for the Internet of Things with decentralized data and centralized control," in *Proceedings of the 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–8, Marrakech, Morocco, November 2015.

[59] M. E. Dick, E. Pacitti, and B. Kemme, "A Highly Robust P2P-CDN under Large-Scale and Dynamic Participation," in *Proceedings of the 2009 First International Conference on Advances in P2P Systems (AP2PS)*, pp. 180–185, Sliema, Malta, October 2009.

[60] R. Bhagwan, D. Moore, S. Savage, and G. M. Voelker, *Replication Strategies for Highly Available Peer-To-Peer Storage Systems*, Department of Computer Science and Engineering, University of California, San Diego, CA, USA, 2002.

[61] L. Pamies-Juarez, M. Sanchez-Artigas, P. García-López, R. Mondéjar, and R. Chaabouni, "On the interplay between data redundancy and retrieval times in P2P storage systems," *Computer Networks*, vol. 59, pp. 1–16, 2014.

[62] R. S. Carbajo and C. Mc Goldrick, "Decentralised Peer-to-Peer data dissemination in Wireless Sensor Networks," *Pervasive and Mobile Computing*, vol. 40, pp. 242–266, 2017.

[63] E. B. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 31–37, 2008.

[64] M. Chaqfeh, A. Lakas, and I. Jawhar, “A survey on data dissemination in vehicular ad hoc networks,” *Vehicular Communications*, vol. 1, no. 4, pp. 214–225, 2014.

[65] F. Kangling et al., “Overview of data dissemination strategy in wireless sensor networks,” in *Proceedings of the 2010 International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT)*, pp. 260–263, Shenzhen, China, April 2010.

[66] H. Yu, F. Zhang, and Y. Wu, “Granary: A sharing oriented distributed storage system,” *Future Generation Computer Systems*, vol. 38, pp. 47–60, 2014.

[67] R. Geambasu, A. A. Levy, T. Kohno, A. Krishnamurthy, and H. M. Levy, “Comet: An active distributed key-value store,” in *OSDI*, pp. 323–336, 2010.

[68] S. Das, A. El Abbadi, and D. Agrawal, “Elastras: An elastic transactional data store in the cloud,” *HotCloud*, vol. 9, pp. 131–142, 209.

[69] P. Druschel and A. Rowstron, “PAST: a large-scale, persistent peer-to-peer storage utility,” in *Proceedings of the 8th Workshop on Hot Topic in Operating Systems*, pp. 75–80, 2001.

[70] J. Kubiatowicz, D. Bindel, Y. Chen et al., “OceanStore: An architecture for global-scale persistent storage,” *ACM SIGOPS Operating Systems Review*, vol. 34, no. 5, pp. 190–201, 2000.

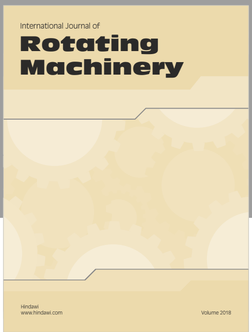
[71] G. DeCandia, D. Hastorun, M. Jampani et al., “Dynamo: amazon’s highly available key-value store,” *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 205–220, 2007.

[72] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The google file system,” *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 29–43, 2003.

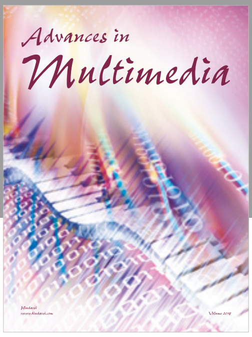
[73] H. Chen, H. Zhang, M. Dong et al., “Efficient and Available In-Memory KV-Store with Hybrid Erasure Coding and Replication,” *ACM Transactions on Storage (TOS)*, vol. 13, no. 3, pp. 1–30, 2017.

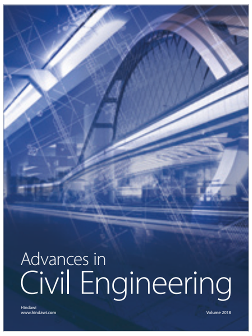
Journal of
Engineering

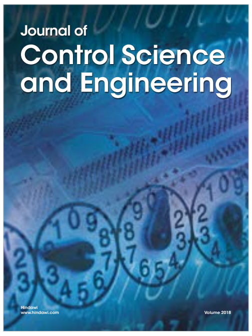
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

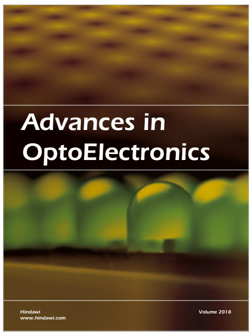

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

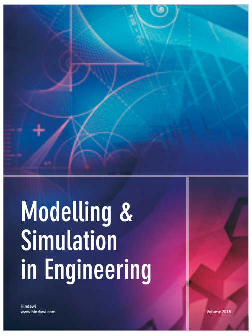
Journal of
Robotics

Journal of
Electrical and Computer Engineering

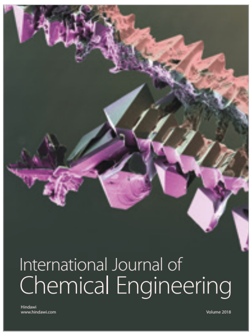

Advances in
OptoElectronics

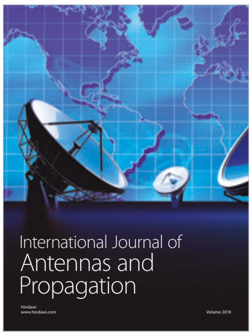
VLSI Design

International Journal of
Navigation and Observation

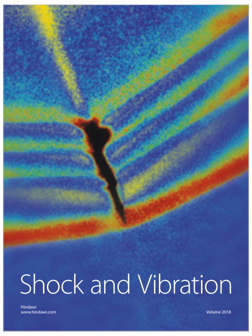

Modelling & Simulation in Engineering

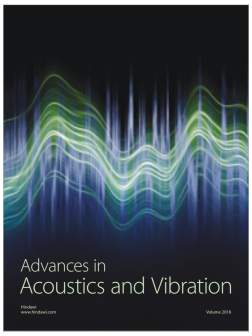
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration