

Article

Modeling and Simulation Tools for Fog Computing—A Comprehensive Survey from a Cost Perspective

Spiridoula V. Margariti ^{1,*}, Vassilios V. Dimakopoulos ²  and Georgios Tsoumanis ¹ 

¹ Department of Informatics and Telecommunications, University of Ioannina, GR-47100 Arta, Greece; gtsoum@uoi.gr

² Department of Computer Science and Engineering, University of Ioannina, GR-45110 Ioannina, Greece; dimako@cse.uoi.gr

* Correspondence: smargar@uoi.gr

Received: 17 April 2020; Accepted: 13 May 2020; Published: 16 May 2020



Abstract: Fog computing is an emerging and evolving technology, which bridges the cloud with the network edges, allowing computing to work in a decentralized manner. As such, it introduces a number of complex issues to the research community and the industry alike. Both of them have to deal with many open challenges including architecture standardization, resource management and placement, service management, Quality of Service (QoS), communication, participation, to name a few. In this work, we provide a comprehensive literature review along two axes—*modeling* with an emphasis in the proposed fog computing architectures and *simulation* which investigates the simulation tools which can be used to develop and evaluate novel fog-related ideas.

Keywords: fog computing; modeling; simulation tools; cost

1. Introduction

Fog computing is a new architecture which bridges the cloud and the Internet-of-Things (IoT) world, acting as a layer between them in order to provide services directly to the network edge. As per Reference [1], fog computing is as “*a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment*”. It provides a promising approach for processing the huge data volume produced and consumed by applications, machines and their users. Complex processing and distributed computing is placed on clouds and devices around us, respectively [2]. Fog is an alternative technology to the cloud aiming to meet user requirements for SCALE [3]—Security, Cognition, Agility, Latency, Efficiency.

Nowadays, IoT devices are used for storing, processing, collecting and exchanging data that are generated in huge volumes from emerging applications. IoT interacts with applications that are time-sensitive and have demands for location-aware, high-speed data processing and low energy consumption [4], while having to handle the large scale of produced data. However, in most cases IoT devices do not have the capabilities to perform all these tasks, so instead they rely on cloud computing facilities [5].

The choice to move data and computation from the edges to the cloud can become inefficient and costly while posing security issues related to privacy. Moreover, the centralized nature of cloud, the location dependency in terms of distance, and the cost of services provided by the cloud are important shortcomings of the cloud-IoT integration [5]. Fog computing comes to

overcome these limitations. It works complementary to cloud computing and improves on it in terms of latency reduction, bandwidth savings, mobility support, geographically distributed and decentralized deployment, heterogeneity, interoperability, data security and privacy protection, and energy consumption [6,7]. These factors affect the cost and the performance of communication, computation and storage. In order to reduce traffic load and latency the fog architecture tries to process data or to provide services closer to end devices.

Traditionally IoT applications which exploit cloud computing facilities need to deal with heterogeneity, high latency, and large scale issues. In contrast, fog computing establishes a decentralized framework and shifts parts of applications, management, and data analytics to the network edge. However, moving computation from cloud centers closer to network edges requires decisions on where to place data and applications within the cloud-to-end-point continuum and how to address the issues of heterogeneity, geographic distribution, low latency, low power, low bandwidth, security and privacy in cost-effective manner.

Researchers and practitioners in the fog computing area have to deal with the system complexity arising from the large number of participating objects and their interactions, different technologies and disparate applications. In order to overcome these issues, they typically employ models and simulation tools to approximate the actual fog system. A model is a representation of a real or a planned system or part of it, in an ideal form. It is used to gain understanding of the system under study, to interpret the various phenomena and the interactions among the components, to make predictions about the behavior of the whole system or some of its subsystems, and finally to act on them. Simulation frameworks provide solutions in cases where mathematical modeling techniques are difficult or impossible to apply due to the scale, complexity and heterogeneity of a fog computing system [8]. Simulation is a way to mimic the operation of real systems, with the freedom to modify the inputs, and to model a number of characteristics, analyze existing systems or support the design of new systems and helps to identify and balance the cost [9]. It should be, thus, apparent that modeling and simulation are appropriate tools to study and analyze how fog computing and its applications are deployed and act in the intermediate layer between the cloud and the network edges.

There are several surveys on fog computing and related issues. They cover fog computing platforms [10]; they analyze challenges, architectures and open issues [2,4,6,11,12]; they summarize emerging computing paradigms to drive the shift from the centralized cloud computing to distributed edge computing [13]; they focus on resource management [14–16], or applications [17]. Other relevant surveys examine various aspect of modeling and simulation in the fog environment. Svorobej et al. [8] investigate simulation challenges in fog and edge computing environments. Markus and Kertesz [18] consider fog models and the quality of simulators from a software viewpoint. Ashouri et al. [19] assess the quality of simulators. Abreu et al. [20] provide a conceptual review of cloud/fog simulation tools. We note the relation of fog computing to other technologies such as edge computing and mobile edge computing. Edge computing also extends the cloud but in a limited manner, focusing on the IoT devices end, while mobile edge computing provides cloud computing services close to mobile users. While facing similar issues to fog computing, these technologies are distinct [4,6,13]. As such, they are out of the scope of this work. We focus on models and on the most recent fog computing simulators. In contrast to previous works, for the first time we consider the cost parameter and its significance in fog computing deployments.

In this work, we present a comprehensive survey of modeling techniques and simulation tools for fog computing and investigate how to reduce the system cost in terms of network, application and management/configuration requirements. More specifically, our contributions are:

- We identify the major research aspects in the area and how modeling and simulation helps towards their solution.
- We provide an extensive overview of state-of-the-art fog computing simulation tools, pinpointing their distinct characteristics and the way they address several critical issues such as latency, energy consumption, and so forth.

- We underline the limitations of existing tools and highlight open issues and research trends
- We highlight various quantities that can be used as metrics or system parameters when considering the cost dimension

The rest of the paper is organized as follows. Section 2 serves as an overview of the basic notions and concepts of fog computing. In Section 3 we identify fog computing challenges for the study of which, modeling and simulation can play an important role. Sections 4 and 5 focus on modeling techniques and simulation tools, respectively, for fog computing systems. Finally, Section 6 concludes this work.

2. Basic Concepts

Since 2009, Satyanarayanan et al. [21] have supported that cloud computing suffers from large scale and increasing latency, and have pointed out the need for new technologies. Among the newer developments, fog computing is a groundbreaking technology that bolsters and extends the cloud. The term “fog computing” was initially introduced by Cisco, in 2012, as a “cloud-to-thing continuum” [1]. It is defined in different ways, which encompass the cooperation of a huge number of distributed devices and the placement of storage, computational and networking services at the edges, the network or the cloud [2]. Fog computing is not a standalone paradigm, it is located between the cloud and the network edges, supporting virtualization and the cooperation of end devices. It arose as a result of the efforts of academia and industry in order to provide low latency, high scalability while reducing costs and energy. It is considered as a promising technology, capable of handling extraordinary amounts of data that are generated by IoT devices. The features of fog computing include heterogeneity support, geographical distribution, location awareness, ultra-low latency, support of real time and large-scale applications, and virtualization [4].

Fog Computing Architecture

There is no standard architecture for fog computing, and research works often rely on different architectures. Figure 1 shows a high-level overview of the fog computing structure. In the most common scenario, it consists of three key layers: cloud, fog and end devices:

- *End devices layer:* This ground layer represents the end devices such as sensors, actuators or things, mobile smartphones and tablets, smart meters, aircrafts and smart vehicles, desktops PCs, laptop computers accompanied with applications [2,22,23]. End devices may be considered as human-operated resources [23] and they provide a range of computing capabilities. All these elements form a communication network, and their data are transmitted to the cloud through the fog layer.
- *Fog layer:* It is the layer above the end devices layer. According to Reference [2], any device which is able to process, store and connect to the network can be considered as a fog processing devices. In this sense, some devices can be considered as both IoT and fog devices, smartphones begin a characteristic example. The fog layer is a collection of processing devices, gateways, and networked devices (routers and switches) that are deployed between network edges and clouds. Fog resources are interconnected and used to deliver a number of services to users such as computing, storage and network services. Fog devices are shaping a distributed system that offers services for a certain set of end devices in a specific location and handle data which are transmitted by such devices [24].
- *Cloud layer:* Physical data center nodes are placed in the cloud layer. Each node has CPU(s), main memory and network bandwidth and is used to satisfy user requests for resources. Control strategies allow the management and scheduling of cloud resources according to their load demands [22]. Clouds are connected to Wide Area Networks (WANs) and provide economic benefits, elastic services, data-intensive analysis for end-users [13], quality of the offered services and high degree of fault tolerance [25]. However, clouds suffer from high latency and bounded

capacity due to current WAN connections. Moreover, as cloud computing works in a centralized manner, it is incapable of supporting context-aware computing for IoT applications [13].

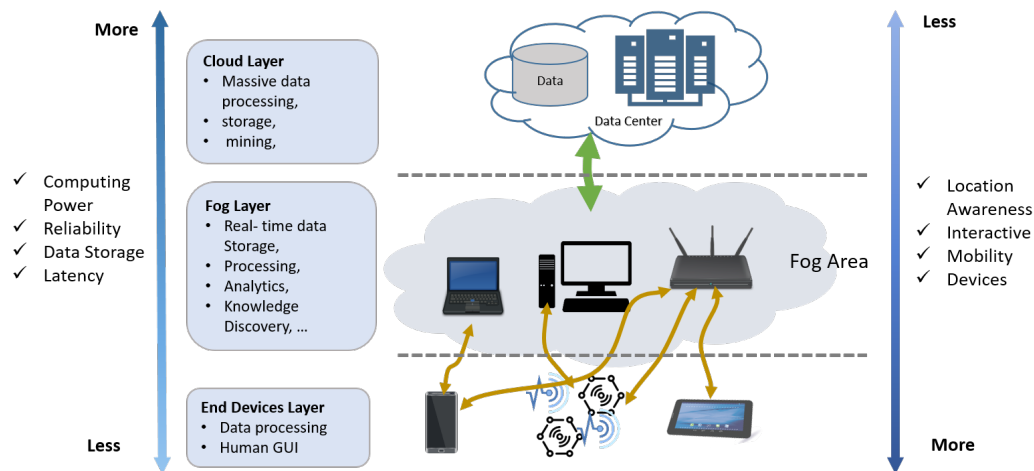


Figure 1. Fog computing technical architecture.

3. Modeling and Simulation Challenges

Fog computing emerges as a modern and competitive environment in a number of different areas: smart cities, intelligent transportation systems, smart healthcare, public safety, smart grid, industry 4.0, smart home and smart building are only some of the domains where applications have been developed and supported by a variety of fog computing technologies [10]. Often, these technologies and applications have a significant impact on daily peoples lives. However, not all of them have reached a satisfactory level of maturity and require more consideration. Existing approaches try to answer emerging questions, to solve a problem or make a decision in a specific field [17] mainly using modeling and simulation. Today, there are several simulation tools for fog computing environments with different mechanisms and characteristics. Before we investigate how they fulfill their purpose, we discuss the current challenges in fog computing that lead to new specialized and novel techniques. These innovations can be evaluated, tested and have their cost estimated by employing proper simulation tools.

It is a non-trivial task to design fog computing simulators meeting the needs for generality, scalability, efficiency, mobility support and low-latency. One of the hardest problems in this process is to determine the constituent parts of the simulator. Every decision is related to the actual cost of implementation and operation of the fog computing scheme and its components—infrastructure, computing, and supported applications. Next, we identify several deployment challenges.

3.1. Fog Computing Infrastructure

The fog infrastructure consists of a set of devices with no computational capabilities (e.g., sensors) and a set of resources offering compute power, storage capacity, and so forth (fog nodes, cloud servers) that are interconnected, forming a complex system [15]. There is a large diversity in the characteristics of all these devices. They may concern, among other things, the CPU power, RAM size, disk capacity, bandwidth, latency, and so forth. From a functional point of view, the infrastructure could be organized in layers, and is generally modeled as a connected graph where the vertices denote the set of devices/resources, and the edges indicate the links. Simulators should at least support the most common resource types and their characteristics: storage, compute and communication.

3.1.1. Storage

One of the main functions of fog computing is the provision of storage resources which are closer to the end devices than cloud storage. The storage requirements come from a large number

of end devices, which may be distributed in a vast geographical area, used by different users while running their applications. Many types of storage can be used in fog nodes in order to provide the reliability and data integrity required by the fog system [26]. The user needs, the application requirements or restrictions and the cost determine the type and capacity of storage, while a key issue is the placement of storage resources. In 5G wireless networks, proactive caching and distributed storage based on the integration of each user's personal storage space [6] have been proposed as storage technologies which can expand the service capabilities. Sharma et al. [27] propose a fog architecture that supports blockchain storage; blockchain technology introduces an efficient, scalable and secure way to manage resources.

According to Reference [16], the fog nodes may have limited storage capacity and, consequently, the cloud data centers can be used for compute and storage functions. In general, both of them could work in a complementary manner. In this scenario, a number of issues arise: where to place data (since data locality is an important factor for fast and efficient computing), how to deal with the huge amount of data transfer requests, how to manage storage resources or how to cope with the heterogeneity coming from the various devices and applications. Suitable modeling and simulation are necessary in order to test any conceived solutions, observing the degree to which their behavior satisfies the actual needs.

3.1.2. Computation

Any device with computing capabilities could be used for compute. Generally speaking, any device in the fog layer which has computational power and any server on the cloud can act as a compute device. Based on them, fog computing systems provide new computational opportunities and more efficient services.

Compute devices provide services for storing, processing and analyzing data [28]. They could act as servers, coordinators or controllers depending on the given application and its requirements. They may also discover, monitor and manage other devices and their requests. In a real implementation compute devices need to be purchased, connected and configured, while in a simulation environment, they need to have specific features, such as management and mobility capabilities. It is desirable to deploy a fog system with various types of computing devices and predefined QoS levels, which enhances the performance metrics in a cost-effective manner. Here two different kinds of cost must be considered—the purchase cost and the operating cost. Notice that since compute can be placed on both fog and cloud nodes, the operating costs may also include latency, network congestion, and energy consumption, which are related to possible data transfers among compute nodes [29].

3.1.3. Communication

As mentioned above, a fog computing system cooperates with the cloud layer and the end-devices layer. It works in a distributed manner and connects geographically distributed devices—mainly end-devices—with fog nodes and clouds. It usually exploits existing access and metro network infrastructure [4] in order to provide data transfer facilities. The communication channels can be wired or wireless, supported by different networking technologies (e.g., WLAN for wireless and part of IP networks for wired). Wired connections can be found between fog nodes and cloud data centers or among fog nodes, while wireless connections are typically used to connect end devices and fog nodes [6]. It is common for fog devices to maintain three different connection types [30]:

- For communication between fog and mobile users
- For communication between cloud and fog
- For communication between fogs

For each connection type, the links may be heterogeneous, may have different capacities and may utilize different technologies. In the first connection type, basic fog elements such as fog servers provide a local infrastructure where communication services take place between fog and end-users. The fog

layer is connected with end-devices using technologies such as WiFi, Bluetooth and NFC [14]. Here, a key feature is the prediction of the specific location of mobile users, which can then be used to handle localized issues and provide specialized services. Regarding the second connection type, fog and cloud communications are mostly based on IP core networks. In some cases, they exploit software-defined networking (SDN) technologies [6,27], disassociating the traffic routing task from the data forwarding process. Finally, the communication between fog nodes could take place in a centralized or distributed manner. The centralized approach utilizes SDN facilities, while distributed solutions take place through the traditional routing process using wired or wireless transmission. More recent approaches propose the use of Network Function Virtualization (NFV) for communication [6]. NFV utilizes virtualization technology to provide networking fully independent of hardware.

Throughput, average round trip time, average response time and transmission delay are among the issues that concern researchers [12]. Fog systems have to undergo additional investigation and evaluation to extract cost and performance trends. Modeling and simulation are flexible, scalable and low-cost tools to evaluate the performance of these systems, for every communication type.

3.2. Fog Computing Key Technologies

Fog nodes and end-devices make use of a wide variety of distributed computing technologies. These technologies are used to serve users' processing requests in an autonomous and independent manner, while virtualization technology is capable of providing scalable processing environments. Computing takes place on edge devices, intermediate nodes or in cloud data centers depending on the current needs and the the availability of the resources. This process gives rise to the challenging problem of resource management. Ghobaei-Arani et al. [16] classify resource management approaches in the following six categories:

- Application placement
- Resource scheduling
- Task offloading
- Load balancing
- Resource allocation
- Resource provisioning

The research community working on these issues and considering open perspectives and future research directions, set different evaluation factors and typically utilize simulators as evaluation tools. They try to improve resource management mechanisms and focus on minimizing key aspects of cost (e.g., execution cost, system cost, authentication cost) which are discussed in Section 3.4.

3.3. Fog Computing Applications

A broad range of applications, with low latency requirements, time sensitivity, and demands for accessibility or efficiency from mobile users fit well to the fog computing paradigm [6,7]. These applications come from several domains—health care, smart environments, augmented reality, Intelligent Transportation Systems (ITS), public safety, Smart Grid, Industry 4.0 [6,10,25,31]. The need for models and simulators capable of performing actual experiments is quite clear in this area. Generally speaking, simulation tools should provide facilities for [25]—physical or virtual resources, network infrastructure, control mechanisms and data management in order to support the simulation of different types of applications. The decoupling of the applications from the hardware simplifies the development of mobile crowdsensing applications [32]. Syed et al. [7] raise the multi-tenancy issue, that is the ability to support the execution multiple applications in fog computing environments, by sharing the available resources [7].

3.4. Cost Considerations

Fog computing is a complex environment, where services are provided to users, and users execute their applications. The acquisition of services or resources depends on the participating nodes and their roles as providers or users and always involves the notion of cost. Various aspects of fog computing costs have been considered and cost-aware schemes have been proposed for resource allocation [33], simulations tools [34,35], or modeling tools [36,37]. Battula et al. [33] propose a cost model for fog computing environments which concerns both devices and applications. It is calculated per device or application and summarizes different types of cost and various cost parameters. It is expressed as a function of:

- *Communication cost*: this cost is proportional to the number and size of messages received.
- *Processing cost*: this cost is related to user requirements for local or virtual execution. In the first case, it depends on the total number of executed tasks. In the second case, it depends on the number of required virtual resources and their time usage (CPU hours).
- *Cloud-network cost*: this is included if there are interactions with clouds. It concerns cloud integration pricing and data roundtrip times.
- *Migration cost*: it is the cost for migrating tasks from a fog node to another and is proportional to the total execution time of migrated tasks. This cost is paid by the fog node who was in charge of executing the tasks before the migration.
- *Storage cost*: it is calculated as a function of the storage size, the storage duration and the encryption cost.
- *Power cost*: this accounts for the battery costs of all sensors that are utilized during an application execution.
- *Software cost*: it is calculated according to the per-month pricing policy for each used commercial software product.
- *Sensors cost*: this expresses the cost per sensor request, summed over all served requests.
- *Operational cost*: this sums up the fog device, the sensor and the network operational cost.

The above cost model general enough, has many advantages and can fit in other fog computing settings with some modifications (for example, power cost may be also attributed to fog and/or cloud devices). Ideally, it could be embedded in simulations tools. Examples of cost metrics listed in Table 1. As Brogi et al. [38] state, the inclusion of cost models in simulators may give a significant boost to the deployment of eligible fog computing applications.

Table 1. The notion of cost in fog computing systems.

Cost Type	Works	Examples
Communication cost	[27,29,37,39]	No. of message, message size, throughput, transmission cost, network cost.
Processing cost	[25,27,29,32,35,37,40–45]	Computing cost, system response time CPU-memory-bandwidth utilization, failures per-connection, job loss probability, response time processing -service delay resource churn.
Cloud-network cost	[22,25,32,34,35,37,39,44,45]	Bandwidth, latency, propagation and transmission delays, round-trip-times routing costs.
Migration cost	-	-
Storage cost	[45]	Storage cost.
Power cost	[25,35,36,41,46]	Energy consumption
Software cost	[36,47,48]	Price cost, billing cost.
Sensors cost	-	-
Operational cost	[39,45,46,49–51]	Maintenance costs, operating expenses operational cost, reconfiguration, acquisition, power delivery, evacuating heat.

4. Modeling Fog Computing Systems

In this section, we present and compare typical proposed models for fog computing systems. The role of a model is two-fold; first to understand the real system under study and then to extract an applicable simulation model [52]. In the existing literature, several fog models have been used in lieu of real systems in order to evaluate or understand how those systems work. Models involve components that represent the different parts or states of the real system and the relationships between them which define the behavior of the fog system.

For fog computing systems the key components include the infrastructure (i.e., the devices and the network), the platform (encompassing resources, services and their management) and applications (which execute in the fog and have certain requirements) according to Reference [2]. Mahmud et al. [11] identify three essential dimensions involved in fog computing—system structural, services and security, while Svorobej et al. [8] distinguish four key factors relative to fog and edge computing modeling and simulation—infrastructure, resource management, mobility and scalability. Among the proposed approaches, only a few [33,38] focus on cost modeling. Both efforts seek to quantify the cost of developing IoT applications and the resource-allocation paradigm in fog computing environments. A fog model could be a static representation of the distributed system, or a dynamic one; it may include facilities for data-processing, resource management, networking, and storage services [53]. However, it is important for the research and industrial community to evaluate all those models in terms of costs.

Several types of models have been proposed to study fog system behavior or to evaluate its operation. We can classify them into two broad categories—mathematical and conceptual models. A mathematical model uses mathematical expressions, symbols and language to describe a system, while a conceptual model represents the system as a composition of components.

We can further distinguish mathematical models in:

- *Analytical models:* An analytical model uses mathematical symbols and expressions in order to describe the system evolution. Such models exploit probability theory, algebraic methods and other mathematical tools and techniques to solve a defined problem, sometimes resulting in closed-form solutions. Commonly they have relatively few parameters so as to keep the problem tractable and this can reduce their accuracy. In most cases they lead to rather simplistic representations, not well suited for representing highly complex systems [54].

Although not strictly analytical, numeric computer-based simulations can be considered in this category. Numeric simulation usually involve dividing a system into many small subsystems, model each subsystem using analytical models and test the overall system using computers. These models are relatively simple but can be useful for reproducing the behavior of complex systems.

Analytical models have been used to model the system load [54], the IoT service delay [44], and other performance metrics in fog computing. Numeric computer-based simulations have been used to test resource allocation or caching schemes [55,56].

- *Petri Nets models:* Petri Nets use a graphical representation of the system under study, in combination with a mathematical formalism to model, analyze, and verify system components and their activities/interactions, and have enjoyed wide applicability [57]. In [48] Petri Nets used for the decomposition of distributed edge systems into several components. Cheng et al. [58] utilize Petri Nets to discover the learning process in IoT edges. Merlino et al [32] adopt Petri Nets to model and evaluate mobile crowdsensing services. Ni et al. [47] investigate the resource allocation problem in fog computing and propose a solution based on priced timed Petri Nets. The user autonomously chooses the part of resources that satisfy her needs, out of all the available preallocated resources. The scheme considers monetary and time costs.
- *Markov Chains and other models:* A Markov chain is a random process characterized by the memoryless property: future system states depend only on the present state and not on events that

occurred in the past [45]. Markov chains allow identifying stimulating characteristics of complex systems, compacting potentially large mathematical models into a more concise representation, which can be quite helpful when studying the dynamics of these systems. There are a few works which adopt Markov chain models for fog computing systems. Haneefa et al. [59] model the fog architecture using Markov chains and they compare the performance of hierarchical and flat architecture variants in terms of computational power and completion time of a task. In Reference [39], Markov decision processes, an extension to Markov chains, were used to optimize the operational cost in the computation migration problem. Markov decision processes are also used in Reference [45] to optimize offloading in mobile fog computing, that is, the transfer of computational tasks from mobile devices to fog servers.

Sarkar et al. [22] construct a theoretical model using a mathematical formulation to parametrize individual components of fog computing systems. The authors consider how to reduce the cost in terms of service latency and energy consumption.

Regarding the second broad category, a conceptual model is an abstraction of a real system and it considers the systems as a collection of components which interact in various ways. Each component represents a class of devices or services in the real system, and is characterized by specific behaviors. Iorga et al. [31] propose a conceptual model to introduce an architectural structure for constructing, enhancing or expanding fog computing. A fog system is considered as a layered model consisting of fog nodes (physical or virtual) which interact and communicate with cloud and end-devices. This model aims to support the deployment of applications and services. Osanaiye et al. [60] design a conceptual migration framework. They use it for migration of virtual machines in order to minimize the downtime and the migration time.

Table 2 summarizes the literature with respect to the modeling approach used. For every work, we include information about the actual problem studied by the authors, the targeted cost metrics and the simulation tool which was used to evaluate the models (if any).

Table 2. Comparison of modeling tools (N/A: not applicable).

Work	Problem Studied	Cost Metrics	Simulation Tool
Analytical Models			
[61]	Sources of latency in health care applications	Communication latency, computation latency, and network latency	iFogSim [35]
[36]	Reduction of energy consumption and billing cost in audio/visual recognition and retrieval environment	Billing cost, energy consumption	N/A
[37]	Optimal distribution of processing load among the fog and the cloud	Transmission cost, processing cost per unit time, response time	Java Modeling Tool
[4,44]	Quality of service (QoS) in IoT networks	Service delay, propagation and transmission delays, processing delay	N/A
[42]	Allocation of fog computing resources under SLA and QoS constraints	CPU utilization, system response time, system loss rate, system throughput, number of messages	Java Modeling Tools
[40]	Modeling a typical healthcare monitoring system	Computing cost, response time	Java Modeling Tools
[41]	Data management in mobile crowdsensing environments	energy consumption	N/A
[49]	Management system design for NFV customers and service providers	Capital expenditures (CAPEX) and operating expenses (OPEX)	N/A

Table 2. Cont.

Work	Problem Studied	Cost Metrics	Simulation Tool
Petri Nets Models			
[47]	Resource allocation	Price and time cost	N/A
[48]	System decomposition into distributed components	Price and time cost	N/A
[58]	System for learning process models in IoT edges	Time for discovering and delivering the process model	N/A
[32]	Deployment Mobile Crowd Sensing application into a sensing Cloud.	Number of participating entities (contributors), rate of contributor arrival/departure	Emulation (Genymotion)
Markov Chain Models			
[46]	Energy consumption in wireless sensor networks	Maintenance costs, energy consumption	Ansys Software
[39]	Services migration from the edge to the cloud	Operational cost, transmission cost, routing costs, reconfiguration cost	N/A
[43]	System performance optimization via optimal management of energy storage system.	Monetary cost, job losses	N/A
[45]	Offloading process optimization	Processing and storage costs	Cloudsim
Other Models			
[22]	Fog computing performance for IoT applications.	latency, energy consumption	Custom

5. Simulation Tools

Simulators shed light on existing or prototype systems by imitating their operation. Simulators are used to study the system behavior and understand the factors that affect system performance as it evolves over time. Before applying new technologies to real systems it is salutary to test them using simulation tools. Fog computing trends as the latest extension of cloud computing. Although there is a wide range of simulation tools for cloud computing, they cannot be used as-is for studies in the field of fog computing; they have thus been adapted to meet the new needs. At the same time, novel simulation tools have been proposed and developed, specifically for the fog. In this section, we survey all such tools that are used in the fog computing area.

The Edge-Fog cloud simulator [23] is implemented in Python. It is composed of two layers: (i) the outer layer which includes the edge devices and (ii) the inner layer which consist of fog devices. All those devices are connected and form a distributed system. The authors also implement the Least Processing Cost First (LPCF) algorithm, which is used to assign tasks to available nodes. The assignment aims to reduce the processing time and network costs.

Gupta et al. [35] extend the CloudSim simulator [62] by adding new functionalities and developing a simulation toolkit for fog computing, called iFogSim. iFogSim is one of the most popular tools that can be used to model and simulate IoT and fog environments. It is written in Java and employs the JSON file format to represent physical topologies. iFogSim supports the simulation of entities and services. The communication is based on message passing. It is an event-based simulator that can be used to simulate applications in IoT and fog environments. It provides an environment capable of evaluating resource management policies based on different metrics, such as energy consumption, operational costs, and network congestion. The architecture of iFogSim provides physical, logical and management components. Physical components included fog devices, actuators, and sensors. Logical components represent processing modules and their interaction as a directed graph. User can draw physical elements, define their characteristics and build their topology using a user-friendly GUI. Alternatively, the user can define topologies programmatically using Java APIs. A management component assists in application placement, scheduling and monitoring. Despite its popularity, iFogSim has a few shortcomings: (i) the Java-based implementation introduces version compatibility issues while not

supporting basic network parameters [63], (ii) it focuses primarily on resource management, ignoring other important fog computing facets (e.g., mobility, infrastructure, cost) (iii) it supports only tree-like topologies (iv) it does not consider QoS requirements [64].

Brogi et al. [64] propose a model for fog computing which consists of QoS profiles, fog infrastructures, IoT applications and eligible deployments. This model is supported by a simulation tool named FogTorch. The simulator is written in Java and allows developers to customize the fog infrastructure (in terms of CPU cores, RAM and storage per node, to define QoS policies related to latency and bandwidth, and to specify the application requirements. FogTorch does not provide a cost model. FogTorchII [65] is an extension of FogTorch, capable of determining deployments of IoT applications over fog computing systems. FogTorchII inherits parameters related to hardware, software, QoS from FogTorch and provides facilities to define QoS for communication links. To simulate the behaviors of communication links, it exploits Monte-Carlo methods which are used to generate sampling probability distributions. FogTorchII provides metrics to estimate and compare resource consumption and QoS accuracy.

FogNetSim++ [63] is a simulation framework capable of handling fog networks and devices. It is based on OMNeT++ [66], an open source component-based C++ simulation library and framework, widely used in academia. FogNetSim++ is an event-driven simulator and its main objective is to provide a static or dynamic environment that supports sensors, fog nodes, distributed data centers, and a broker node. The role of the latter is to manage the other devices and their requests. FogNetSim++ supports the execution of resource scheduling algorithms while it also provides an energy model and different pricing models. The pricing model applies to network, storage and compute components for network, storage, compute and other tasks. The simulator provides users with detailed configuration options. It also provides modules for the creation of the network environment, and the users may incorporate their own extensions. The authors claim that FogNetSim++ is scalable in terms of average execution time, CPU and memory usage.

Tychalas and Karatza proposed Parallel Discrete Event Simulation (PDES) for fog computing, implemented in C [67]. In a predefined cloud/fog system, authors consider how to reduce costs for each incoming task by combining of all available computational resources and bringing all involved computing resources closer to users. The tool focuses on task scheduling algorithms.

The work in Reference [68] proposes Yet Another Fog Simulator (YAFS) that enables Cloud/fog simulations. Python was used for the development of YAFS is Python while is also supports the JSON format for input files that describe customized scenarios. The YAFS architecture consists of the following components: application, selection, placement, population, topology, and core. During simulation the selection, placement, and population processes have the responsibility of resource management and their interactions are dynamic. Topologies use graph-based representations and are imported from JSON files which can be generated from other tools (such as CAIDA and BRIT). Applications remain the same as in iFogSim, and consist of a set of modules, which provide services and exchange messages. Following a distributed data flow model, authors use a directed acyclic graph to represent them, where nodes are modules that act and edges are the exchanged data among modules.

FogDirMine [69] is a Python-based simulation tool that aims to model the behavior of the CISCO FogDirector, which is a tool for managing IoT applications on fog systems. Although FogDirMine refers to the availability of resources and QoS, there is no reference to costs. FogDirSim [70] is deployed to support also the CISCO FogDirector. It is built in Python and composed of independent REST services. The objective of FogDirSim is to compare different management policies for applications and infrastructure, considering energy consumption, uptime, and resource usage.

FogBus [71] is a framework for building fog environments. It is developed using RESTful technologies and a combination of scripting and programming languages. FogBus supports a wide range of infrastructure devices, application execution and interactions among nodes. It provides authentication and encryption facilities implementing blockchain. The FogBus framework includes

hardware components (fog nodes, IoT devices, fog gateway nodes, cloud data centers) software components (grouped in broker, repository, and computing services) and network components (topology, security, scalability, and fault tolerance). This simulator combines a large number of elements and services and evaluates their performance in terms of energy, latency, network and CPU usage.

ModFogSim [5] is an extension of iFogSim which requires significant effort to model user mobility and migration strategies. ModFogSim adds mobility and migration features as well as wireless connections to iFogSim. These additional features are implemented with the help of a Coordinate class (which maps devices to a Cartesian plane), an AppDevice class (acting as an access point, handling connections and disconnections), a number of Mobile classes (for devices, sensors and actuators) a number of Migration classes to support migration in the simulation. The authors used a lab-based testbed to evaluate their simulator. They conclude that the results demonstrate the usability of their tool. In terms of cost, it inherits the iFogSim metrics, improvements on the latency metric.

FogWorkflowSim [72] presents a different approach to fog computing simulation. It is written in Java and provides a user-friendly GUI to evaluate resource and task management strategies. The simulation process is done in simple steps: (a) setup of the fog computing environment, where the user defines and parametrizes devices for each of the three layers, (b) setting a workflow application, where the user chooses from available workflows or creates her own, (c) configuration, which defines the computational strategies, the scheduling algorithms, and the performance metrics and (d) results, where simulation outcomes can be visualizes and exported. FogWorkflowSim evaluates system performance with respect to three different quantities: time, energy and processing cost.

Finally, there is the class of custom simulator tools built for specific uses. They are based on simulators that target other areas, with some simplifications and modifications. Abbas et al. [73] developed a real client-side environment and collected benchmark values. Then they exploited an OPNET-based network simulator [74] to evaluate their proposed Fog Security Service mechanism considering the overall processing time as the performance metrics. The authors of Reference [75] use a PeerSim simulator in order to evaluate their proposed fog-caching peer-to-peer architecture in terms of success rate and latency. Hong et al. [76] consider communication costs for their Mobile Fog programming model (a communication API that allows applications to use fog resources) and use OMNeT++ [66] to conduct simulations.

Table 3 summarizes the available simulation tools. The first column refers to the year of the latest release of simulator. The third column refers to the technologies used for implementing the simulators. In the metrics column we list the metrics reported by each simulator and in the following we highlight the the main targets/objectives of each simulator. The citations column is an indication of the number of works that use each simulator; it is obtained from the citations the corresponding original article has received according to Google Scholar as of April 2020. Based on this data, iFogSim seems to be the most popular tool among researchers, who use it or extent it. The majority of the simulators are implemented to support a specific facet or problem of fog computing. For example, Edge-Fog, iFogSim, FogWorkflowSim, MyiFogSim focus on the resource management problem. FogNetSim, on the other hand, is a general purpose simulator and can be used to simulate large scale fog networks. All of them provide metrics or statistics, while only a few of them (Edge-Fog, FogTorchPi, iFoSim and FogNetSim) provide cost models. The lack of cost models is an issue that should receive considerable attention in future works.

Table 3. Simulator tools for fog computing.

Year	Simulator	Implementation Technologies	Metrics	Objective	Citations
2016	Edge-Fog [23]	Python		Distribute task processing on the participating cloud resources	73
2017	FogTorch [64]	Java	QoS, reliability of links and nodes, power consumption, security, monetary costs	Find eligible deployments of an application over a fog infrastructure	173
2017	FogTorchII [65]	Extension of FogTorch	Resource utilization and QoS accuracy	Same as FogTorch and many QoS profile according to a probability distribution	51
2017	iFogSim [35]	Extension of CloudSim, Java, JSON	Energy consumption, network congestion, and operational costs	Performance of resource management policies	509
2017	MyiFogSim [34]	Extension of iFogSim	Latency	Resource allocation	38
2018	FogDirSim [70]	RESTful API, Python	Performance in terms of uptime, energy consumption, resource usage	Compare application management and infrastructure management policies	4
2018	FogNetSim++ [63]	based on OMNeT++	Energy module, scheduling algorithms, pricing model	General simulation of fog environments	25
2019	FogWorkflowSim [72]	Java	Performance (time, energy and processing cost)	Evaluate resource and task management strategies	
2019	OPNET [73]	Visual studio, based on OPNET	Processing time	Cope with the massive amount of confidential and security-sensitive data	3
2019	YAFS [68]	Python, JSON	Energy models, network utilization, response time, network delay	Analyze the design and deployment of applications	7
2019	FogDirMime [69]	Python		Support FogDirector [77]	9
2019	Fogbus [71]		Latency, energy, network and CPU usage		27
2020	MobFogSim [5]	Extends iFogSim	Same as iFogSim	Evaluate application behaviour and performance	0

5.1. Discussion

In Table 4 we try to draw a technical comparison of the simulators, based on their capabilities. As can be seen, the existing simulators provide only partial coverage of the requirements for simulation in fog computing. Almost all of them support different devices or other equipment to build the fog system (column “Infrastructure”), but they differ in terms of the attributes they support, or in terms of their connections. For example, FogNetSim++ provides a flexible network model and YAFS supports a multitude of network topologies, but Edge-Fog provides only tree topologies. Also, most of them include a resource management module to support application execution. Some are distinguished for a unique feature. For example, FogWorkflowSim is an easy to use simulator, as it provides a web API for online execution. FogBus introduces a tool to analyze Sleep Apnea. Although it is not practical

to build a specific simulation tools for each specific application area, we are lacking tools that enable generic application simulation.

Next we underline the main limitations of fog computing simulation tools. These limitations also represent challenging issues for possible future work.

- *Efficiency*: An indicative measure for simulation efficiency is lacking; this could be the execution time, the CPU utilization, the memory consumption, and so forth, but there are only few and limited works which study this issues. An experimental comparison [20] shows that iFogSim has some scalability issues and has higher memory demands than YAFS.
- *Cost modeling*: As mentioned previously the presence of a cost model is quite limited in simulators. The cost is a way to quantify how the various parameters affect the operation of a fog computing system and it can be an indispensable tool for operators, developers or users in this area.
- *Network infrastructure/configuration*: In most cases, network models are limited and not suited for experiments aimed at evaluating the network entities and network connections. Specifically, they do not usually provide the possibility to define the physical topology, the link characteristics, or the capabilities of networked devices.
- *Security consideration*: Security and privacy protection are among the key technologies of fog computing systems [6]. A wide range of issues including identity management, resource access control, encryption, decryption, authentication and authorization and malicious attacks need to be dealt with. Simulation tools could be particularly helpful, but they currently lack such functionality. The only exception is FogBus which provides an authentication and encryption mechanism.
- *Applications integration*: A key question for researchers and developers is related to the needs and requirements of an application to be executed in a fog computing environment. As already mentioned, current tools lack the functionality to simulate most aspects of an application and its activities.
- *Extendibility*: Fog computing is a new paradigm and researchers strive to propose solutions (architectures, applications, models) that have to deal with heterogeneity, mobility, security, scalability, and other related issues. The ability to use a simulator that would allow them to easily integrate a new idea rather than build it from scratch would be very useful. Unfortunately, only a few of the existing simulators (namely FogNetSim++, iFogSim and YAFS) provide some hooks for such functionality.

We have also identified some non-technical albeit equally important limitations of contemporary fog simulation tools:

- *Documentation quality*: In general, there do not exist sufficient documentation materials and detailed tutorials. Typically, there only exist some basic documents related to installation and running the simulator; even this information is missing in a good percentage of the available simulations tools. IFogSim maintains a social media presence with instructional videos, but it comes mainly from the user community, not from the creators themselves.
- *Learning curve*: There is not a clear indication about the learning curve of each tool. For example YAFS seems to have a smaller learning curve than iFogSim [20]. While the learning curve also depends on the documentation quality/quantity, a systematic comparison among all tools is lacking.
- *Maintenance*: Although the simulators have been developed relatively recently, some of them seem to be already inactive, as they have not been updated for a quite a few months.

Table 4. Comparison of simulation tools for fog computing according to their capabilities.

Simulator	What It does	What It does Not	Infrastructure	Communication Network	Resource Management	Cost Model
Edge-Fog [23]	Generates a network of resources, supports task allocation, configuration parameters	Supports various model (network, energy), network configuration	Edge, fog and data store layer	No	Distributed task processing	Yes (partial)
FogTorch [64]	QoS-aware deployment of IoT applications	Most features lacking	Cloud, fog nodes and things			No
FogTorchII [65]	Extends FogTorch. Estimates monthly cost of QoS-aware deployment of IoT applications	Energy consumption, security constraints, mobility	Similar to FogTorch	Latency, bandwidth (limited)	Resource consumption (RAM and storage)	Yes
iFogSim [35]	Comparison of Resource management techniques, import topologies, calculates cost	Mobility, failure modeling	Sensors, actuators, fog devices and data centers	Network links, delay, network usage (limited), topology (tree)	Resource and power consumption, allocation policies	Yes (partial)
MyiFogSim [34]	Extends iFogSim to support virtual machine migration policies for mobile users	Failure modeling, network configuration	Sensors, actuators, devices, and data centers	Similar to iFogSim	Similar to iFogSim	Similar to iFogSim
FogDirSim [70]	Manage the entire life-cycle of IoT applications for CISCO FogDirector	Most features lacking				No
FogNetSim++ [63]	Provides built-in modules (sensors, mobile devices, fog nodes, broker), supports communication protocols, applications	VM migration, other cost types, real-network properties (packet loss, congestion, or channel collision)	Mobile devices, fog nodes, broker nodes, sensors and base stations	Network links, delay, handovers, bandwidth, topology (graph)	Resource consumption (RAM, CPU), scheduling policy	Yes
FogWorkflow-Sim [72]	Modeling and simulation of workflow scheduling, web API	Most features lacking	Cloud, fog nodes, end devices	N/A	Offloading and scheduling strategies	No
YAFS [68]	Resource allocation analysis, billing management, network design, and so on.	Network configuration, energy consumption, cost	Cloud, fog, sensors and actuators	Bandwidth and link propagation, topology (graph)	Allocation and orchestration algorithm	No
FogDirMime [69]	Fog application management with CISCO FogDirector	Focuses on the simulation of a specific application	Cloud and fog nodes			No
Fogbus [71]	Focuses on application simulation, security features, supports multi-applications	Network configuration, energy consumption	IoT devices, fog nodes, and cloud data centers	Limited	Sleep Apnea analysis, security considerations	No
MobFogSim [5]	Extends iFogSim adding mobility and migration facilities	Network configuration, importing topologies	Mobile sensors, actuators, broker, and data centers	Similar to iFogSim	Similar to iFogSim	Similar to iFogSim

We summarize the above non-technical characteristics in Table 5, where we also note the availability of a Graphical User Interface (GUI). All simulation tools are free, open-source, and available through open repositories. In the column “Actively developed” we consider the simulator as active if there exist commits during the last 18 months.

Table 5. Other information on the simulation tools for fog computing.

Simulator	GUI	Documen- Tation	Actively Developed	Licence	Available at
Edge-Fog [23]	No	No	No	GNU	https://github.com/nitinder-mohan/EdgeFogSimulator
FogTorch [64]	No	No	No	MIT	https://github.com/di-unipi-socc/FogTorch
FogTorchII [65]	No	No	Yes	MIT	https://github.com/di-unipi-socc/FogTorchPI
iFogSim	Yes	Yes	No	N/A	https://github.com/Cloudslab/iFogSim
MyiFogSim [35]	Same as iFogSim	No	No	N/A	https://github.com/marciocomp/myifogsim
FogDirSim [70]	Yes	Yes	Yes	MIT	https://github.com/di-unipi-socc/FogDirSim
FogNetSim++ [63]	Yes	Little	Yes	GNU	https://github.com/rtqayyum/fognetsimpp
FogWorkFlowSim [72]	Yes plus web API	Little	Yes	N/A	https://github.com/CCIS-AHU/ , http://47.74.84.61/FogWorkFlowSim
YAFS [68]	Yes	Yes	Yes	MIT	https://github.com/acsicuib/YAFS
FogDirMime [69]	No	No	Yes	Apache 2.0	https://github.com/di-unipi-socc/FogDirMime
Fogbus	Yes	Yes	Yes	GNU	https://github.com/Cloudslab/FogBus
MobFogSim [71]	Same as iFogSim	Yes	Yes	GNU	https://github.com/diogomg/MobFogSim

6. Conclusions and Future Directions

Fog computing is an evolving technology, and researchers strive to find solutions for emerging challenges related to its architecture and the management of its infrastructure, communication and resources. Their pursuit should be supported by models and simulation tools. In this work, we discuss modeling and simulation efforts and present the most recent approaches in the literature. Moreover, we consider cost as one more dimension that has to be considered, not only by the research community but the industry, too. While several cost metrics were detailed, it is a fact that only a few simulation tools incorporate them into their feature lists and functionalities. The cost issue is still in its infancy and needs further consideration. It is our hope that this work can help towards this end.

As discussed in Section 5.1, except the cost parameter, current simulators have also other limitations. Fog computing is an active research area with real-world implementations. It thus imperative to make improvements to existing tools, to add new features or even create new simulators that will meet the needs for infrastructure, network, and resource management modeling. The latter is important if we take into account that some of the simulators are adapted from previous cloud simulation frameworks.

As a final open issue and possible research direction, we consider the systematic comparison of available simulation tools. First, this could entail an evaluation of the potential and the applicability

of each tool for specific problems (such as resource availability, energy consumption, security and privacy, mobility, network access and scalability). Secondly, it could focus on the efficiency of each simulator, based on certain metrics (e.g., simulation time, memory consumption) by having all of them simulate similar systems. This is certainly a challenge, since no tool is able to simulate all possible system configurations; they rather aim at certain problems and provide specific and limited solutions. Finally, such a study could evaluate the accuracy of the tools by comparing the simulation results with actual measurements from a real world implementation.

Author Contributions: Conceptualization, S.V.M. and V.V.D.; methodology, S.V.M. and V.V.D.; investigation, S.V.M., G.T. and V.V.D.; resources, S.V.M., G.T. and V.V.D.; writing—original draft preparation, S.V.M. and V.V.D.; writing—review and editing, V.V.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vaquero, L.M.; Rodero-Merino, L. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 27–32. [[CrossRef](#)]
2. Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Gao, L.; Xiang, Y.; Ranjan, R. Fog Computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access* **2018**, *6*, 47980–48009. [[CrossRef](#)]
3. OpenFog Consortium Architecture Working Group. OpenFog reference architecture for fog computing. *OPFRA001* **2017**, *20817*, 162.
4. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [[CrossRef](#)]
5. Puliafito, C.; Gonçalves, D.M.; Lopes, M.M.; Martins, L.L.; Madeira, E.; Mingozzi, E.; Rana, O.; Bittencourt, L.F. MobFogSim: Simulation of mobility and migration for fog computing. *Simul. Model. Pract. Theory* **2020**, *101*, 102062. [[CrossRef](#)]
6. Hu, P.; Dhelim, S.; Ning, H.; Qiu, T. Survey on fog computing: architecture, key technologies, applications and open issues. *J. Netw. Comput. Appl.* **2017**, *98*, 27–42. [[CrossRef](#)]
7. Syed, M.H.; Fernandez, E.B.; Ilyas, M. A pattern for fog computing. In Proceedings of the 10th Travelling Conference on Pattern Languages of Programs, Leerdam, The Netherlands, 7–10 April 2016; pp. 1–10.
8. Svorobej, S.; Takako Endo, P.; Bendeche, M.; Filelis-Papadopoulos, C.; Giannoutakis, K.M.; Gravvanis, G.A.; Tzovaras, D.; Byrne, J.; Lynn, T. Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges. *Future Internet* **2019**, *11*, 55. [[CrossRef](#)]
9. Banks, J. *Discrete Event System Simulation*; Pearson Education: Delhi, India, 2005.
10. Puliafito, C.; Mingozzi, E.; Longo, F.; Puliafito, A.; Rana, O. Fog computing for the internet of things: A Survey. *ACM Trans. Internet Technol.* **2019**, *19*, 1–41. [[CrossRef](#)]
11. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog Computing: A Taxonomy, Survey and Future Directions. In *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*; Di Martino, B., Li, K.C., Yang, L.T., Esposito, A., Eds.; Springer: Singapore, 2018; pp. 103–130.
12. Mouradian, C.; Naboulsi, D.; Yangui, S.; Glitho, R.H.; Morrow, M.J.; Polakos, P.A. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 416–464. [[CrossRef](#)]
13. Ren, J.; Zhang, D.; He, S.; Zhang, Y.; Li, T. A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet. *ACM Comput. Surv.* **2019**, *52*, 1–36. [[CrossRef](#)]
14. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [[CrossRef](#)]
15. Ait Salaht, F.; Desprez, F.; Lebre, A. *An Overview of Service Placement Problem in Fog and Edge Computing*; Research Report RR-9295; Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP: Lyon, France, 2019.

16. Ghobaei-Arani, M.; Souri, A.; Rahmanian, A.A. Resource management approaches in fog computing: A comprehensive review. *J. Grid Comput.* **2019**, *18*, 1–42. [\[CrossRef\]](#)
17. Javadzadeh, G.; Rahmani, A.M. Fog computing applications in smart cities: A systematic survey. *Wirel. Netw.* **2020**, *26*, 1433–1457. [\[CrossRef\]](#)
18. Markus, A.; Kertesz, A. A survey and taxonomy of simulation environments modelling fog computing. *Simul. Model. Pract. Theory* **2020**, *101*, 102042. [\[CrossRef\]](#)
19. Ashouri, M.; Lorig, F.; Davidsson, P.; Spalazzese, R. Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics. *Future Internet* **2019**, *11*, 235. [\[CrossRef\]](#)
20. Abreu, D.P.; Velasquez, K.; Curado, M.; Monteiro, E. A comparative analysis of simulators for the Cloud to Fog continuum. *Simul. Model. Pract. Theory* **2019**, *101*, 102029. [\[CrossRef\]](#)
21. Satyanarayanan, M.; Bahl, P.; Caceres, R.; Davies, N. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Comput.* **2009**, *8*, 14–23. [\[CrossRef\]](#)
22. Sarkar, S.; Misra, S. Theoretical modelling of fog computing: A green computing paradigm to support IoT applications. *IET Netw.* **2016**, *5*, 23–29. [\[CrossRef\]](#)
23. Mohan, N.; Kangasharju, J. Edge-Fog cloud: A distributed cloud for Internet of Things computations. In Proceedings of the 2016 Cloudification of the Internet of Things (CIoT), Paris, France, 23–25 November 2016; pp. 1–6.
24. Muthanna, A.; A Ateya, A.; Khakimov, A.; Gudkova, I.; Abuarqoub, A.; Samouylov, K.; Koucheryavy, A. Secure and reliable IoT networks using fog computing with software-defined networking and blockchain. *J. Sens. Actuator Netw.* **2019**, *8*, 15. [\[CrossRef\]](#)
25. Baccarelli, E.; Naranjo, P.G.V.; Scarpiniti, M.; Shojafar, M.; Abawajy, J.H. Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study. *IEEE Access* **2017**, *5*, 9882–9910. [\[CrossRef\]](#)
26. Ai, Y.; Peng, M.; Zhang, K. Edge computing technologies for Internet of Things: A primer. *Digit. Commun. Netw.* **2018**, *4*, 77–86. [\[CrossRef\]](#)
27. Sharma, P.K.; Chen, M.Y.; Park, J.H. A software defined fog node based distributed blockchain cloud architecture for IoT. *IEEE Access* **2017**, *6*, 115–124. [\[CrossRef\]](#)
28. Faruque, M.A.A.; Vatanparvar, K. Energy Management-as-a-Service Over Fog Computing Platform. *IEEE Internet Things J.* **2016**, *3*, 161–169. [\[CrossRef\]](#)
29. Kertész, A.; Pflanzner, T.; Gyimóthy, T. A mobile IoT device simulator for IoT-Fog-Cloud systems. *J. Grid Comput.* **2019**, *17*, 529–551. [\[CrossRef\]](#)
30. Luan, T.H.; Gao, L.; Li, Z.; Xiang, Y.; Wei, G.; Sun, L. Fog computing: Focusing on mobile users at the edge. *arXiv* **2015**, arXiv:1502.01815.
31. Iorga, M.; Feldman, L.; Barton, R.; Martin, M.J.; Goren, N.S.; Mahmoudi, C. Fog computing conceptual model. In *Technical Report NIST Special Publication 500-325*; National Institute of Standards and Technology: North Charleston, MD, USA, 2018.
32. Merlino, G.; Arkoulis, S.; Distefano, S.; Papagianni, C.; Puliafito, A.; Papavassiliou, S. Mobile crowdsensing as a service: A platform for applications on top of sensing clouds. *Future Gener. Comput. Syst.* **2016**, *56*, 623–639. [\[CrossRef\]](#)
33. Battula, S.K.; Garg, S.; Naha, R.K.; Thulasiraman, P.; Thulasiram, R. A micro-level compensation-based cost model for resource allocation in a fog environment. *Sensors* **2019**, *19*, 2954. [\[CrossRef\]](#)
34. Lopes, M.M.; Higashino, W.A.; Capretz, M.A.; Bittencourt, L.F. Myifogsim: A simulator for virtual machine migration in fog computing. In Proceedings of the 10th International Conference on Utility and Cloud Computing, Austin, TX, USA, 5–8 December 2017; pp. 47–52.
35. Gupta, H.; Vahid Dastjerdi, A.; Ghosh, S.K.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw. Pract. Exp.* **2017**, *47*, 1275–1296. [\[CrossRef\]](#)
36. Renna, F.; Doyle, J.; Giotsas, V.; Andreopoulos, Y. Query processing for the internet-of-things: Coupling of device energy consumption and cloud infrastructure billing. In Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 83–94.

37. Tadakamalla, U.; Menascé, D. FogQN: An analytic model for fog/cloud computing. In Proceedings of the 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, 17–20 December 2018; pp. 307–313.
38. Brogi, A.; Forti, S.; Ibrahim, A. Deploying fog applications: How much does it cost, by the way? In *Proceedings of the 8th International Conference on Cloud Computing and Services Science—Volume 1: CLOSER; INSTICC, SciTePress: Funchal, Portugal, 2018*; pp. 68–77.
39. Urgaonkar, R.; Wang, S.; He, T.; Zafer, M.; Chan, K.; Leung, K.K. Dynamic service migration and workload scheduling in edge-clouds. *Perform. Eval.* **2015**, *91*, 205–228. [[CrossRef](#)]
40. El Kafhali, S.; Salah, K. Performance modelling and analysis of Internet of Things enabled healthcare monitoring systems. *IET Netw.* **2018**, *8*, 48–58. [[CrossRef](#)]
41. Skorin-Kapov, L.; Pripuzić, K.; Marjanović, M.; Antonić, A.; Žarko, I.P. Energy efficient and quality-driven continuous sensor management for mobile IoT applications. In Proceedings of the 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Miami, FL, USA, 22–25 October 2014; pp. 397–406.
42. El Kafhali, S.; Salah, K. Efficient and dynamic scaling of fog nodes for IoT devices. *J. Supercomput.* **2017**, *73*, 5261–5284. [[CrossRef](#)]
43. Conti, S.; Faraci, G.; Nicolosi, R.; Rizzo, S.A.; Schembra, G. Battery management in a green fog-computing node: A reinforcement-learning approach. *IEEE Access* **2017**, *5*, 21126–21138. [[CrossRef](#)]
44. Yousefpour, A.; Ishigaki, G.; Jue, J.P. Fog computing: Towards minimizing delay in the internet of things. In Proceedings of the 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, USA, 25–30 June 2017; pp. 17–24.
45. Rahbari, D.; Nickray, M. Task offloading in mobile fog computing by classification and regression tree. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 104–122. [[CrossRef](#)]
46. Gomes, T.; Fernandes, D.; Ekpanyapong, M.; Cabral, J. An IoT-based system for collision detection on guardrails. In Proceedings of the 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, 14–17 March 2016; pp. 1926–1931.
47. Ni, L.; Zhang, J.; Jiang, C.; Yan, C.; Yu, K. Resource allocation strategy in fog computing based on priced timed petri nets. *IEEE Internet Things J.* **2017**, *4*, 1216–1228. [[CrossRef](#)]
48. Wiśniewski, R.; Karatkevich, A.; Wojnakowski, M. Decomposition of distributed edge systems based on the Petri nets and linear algebra technique. *J. Syst. Archit.* **2019**, *96*, 20–31.
49. Faraci, G.; Schembra, G. An analytical model to design and manage a green SDN/NFV CPE node. *IEEE Trans. Netw. Serv. Manag.* **2015**, *12*, 435–450. [[CrossRef](#)]
50. Greenberg, A.; Hamilton, J.; Maltz, D.A.; Patel, P. The cost of a cloud: Research problems in data center networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *39*. [[CrossRef](#)]
51. Yu, L.; Jiang, T.; Zou, Y. Fog-assisted operational cost reduction for cloud data centers. *IEEE Access* **2017**, *5*, 13578–13586. [[CrossRef](#)]
52. Robinson, S. A tutorial on conceptual modeling for simulation. In Proceedings of the 2015 Winter Simulation Conference (WSC), Huntington Beach, CA, USA, 6–9 December 2015; pp. 1820–1834.
53. Dastjerdi, A.V.; Buyya, R. Fog computing: Helping the Internet of Things realize its potential. *Computer* **2016**, *49*, 112–116. [[CrossRef](#)]
54. Dede, J.; Förster, A.; Hernández-Orallo, E.; Herrera-Tapia, J.; Kuladinithi, K.; Kuppusamy, V.; Manzoni, P.; bin Muslim, A.; Udugama, A.; Vatandas, Z. Simulating opportunistic networks: Survey and future directions. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 1547–1573. [[CrossRef](#)]
55. Lin, F.; Zhou, Y.; Pau, G.; Collotta, M. Optimization-oriented resource allocation management for vehicular fog computing. *IEEE Access* **2018**, *6*, 69294–69303. [[CrossRef](#)]
56. Su, J.; Lin, F.; Zhou, X.; Lu, X. Steiner tree based optimal resource caching scheme in fog computing. *China Commun.* **2015**, *12*, 161–168. [[CrossRef](#)]
57. Bobbio, A. System modelling with Petri nets. In *Systems Reliability Assessment*; Springer: Singapore, 1990; pp. 103–143.
58. Cheng, L.; Liu, C.; Liu, Q.; Duan, Y.; Murphy, J. Learning process models in IoT Edge. In Proceedings of the 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 8–13 July 2019; Volume 2642, pp. 147–150.

59. Haneefa, N.K.; Pramod, S.; Pal, S.; Manivasakan, R. *A Markov Chain Based Framework for Analysis of Hierarchical Fog Computing Networks*; Kim, K.J., Kim, H.Y., Eds.; Information Science and Applications; Springer: Singapore, 2020; pp. 41–52.
60. Osanaiye, O.; Chen, S.; Yan, Z.; Lu, R.; Choo, K.K.R.; Dlodlo, M. From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access* **2017**, *5*, 8284–8300. [[CrossRef](#)]
61. Shukla, S.; Hassan, M.F.; Khan, M.K.; Jung, L.T.; Awang, A. An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment. *PLoS ONE* **2019**, *14*, e0224934. [[CrossRef](#)] [[PubMed](#)]
62. Calheiros, R.N.; Ranjan, R.; Beloglazov, A.; De Rose, C.A.; Buyya, R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exp.* **2011**, *41*, 23–50. [[CrossRef](#)]
63. Qayyum, T.; Malik, A.W.; Khattak, M.A.K.; Khalid, O.; Khan, S.U. FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access* **2018**, *6*, 63570–63583. [[CrossRef](#)]
64. Brogi, A.; Forti, S. QoS-aware deployment of IoT applications through the fog. *IEEE Internet Things J.* **2017**, *4*, 1185–1192. [[CrossRef](#)]
65. Brogi, A.; Forti, S.; Ibrahim, A. How to best deploy your fog applications, probably. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017; pp. 105–114.
66. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008.
67. Tychalas, D.; Karatza, H. Simulation and performance evaluation of a fog system. In Proceedings of the 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, Spain, 23–26 April 2018; pp. 26–33.
68. Lera, I.; Guerrero, C.; Juiz, C. YAFS: A simulator for IoT scenarios in fog computing. *IEEE Access* **2019**, *7*, 91745–91758. [[CrossRef](#)]
69. Forti, S.; Ibrahim, A.; Brogi, A. Mimicking FogDirector application management. *Softw. Intensive Cyber Phys. Syst.* **2019**, *34*, 151–161. [[CrossRef](#)]
70. Forti, S.; Pagiaro, A.; Brogi, A. Simulating FogDirector Application Management. *Simul. Model. Pract. Theory* **2020**, *101*, 102021. [[CrossRef](#)]
71. Tuli, S.; Mahmud, R.; Tuli, S.; Buyya, R. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *J. Syst. Softw.* **2019**, *154*, 22–36. [[CrossRef](#)]
72. Liu, X.; Fan, L.; Xu, J.; Li, X.; Gong, L.; Grundy, J.; Yang, Y. FogWorkflowSim: An Automated Simulation Toolkit for Workflow Performance Evaluation in Fog Computing. In Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 11–15 November 2019; pp. 1114–1117.
73. Abbas, N.; Asim, M.; Tariq, N.; Baker, T.; Abbas, S. A mechanism for securing IoT-enabled applications at the fog layer. *J. Sens. Actuator Netw.* **2019**, *8*, 16. [[CrossRef](#)]
74. OPNET Network Simulator. Available online: <http://opnetprojects.com/opnet-network-simulator/> (accessed on 10 March 2020).
75. Shojafar, M.; Pooranian, Z.; Naranjo, P.G.V.; Baccarelli, E. FLAPS: Bandwidth and delay-efficient distributed data searching in Fog-supported P2P content delivery networks. *J. Supercomput.* **2017**, *73*, 5239–5260. [[CrossRef](#)]
76. Hong, K.; Lillethun, D.; Ramachandran, U.; Ottenwalder, B.; Koldehofe, B. Mobile fog: A programming model for large-scale applications on the internet of things. In Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, Hong Kong, China, 12–16 August 2013; pp. 15–20.
77. Cisco Fog Director. Available online: <https://www.cisco.com/c/en/us/products/cloud-systems-management/fog-director/index.html> (accessed on 3 March 2020).

