Article

Simulating Fog and Edge Computing Scenarios: an Overview and Research Challenges

Sergej Svorobej [1], Patricia Takako Endo [1], Malika Bendechache [1], Christos Filelis-Papadopoulos [2], Konstantinos M. Giannoutakis [2], George A. Gravvanis [2], Dimitrios Tzovaras [2], James Byrne [1], and Theo Lynn [1]

[1] Irish Institute of Digital Business, Dublin City University, Dublin, Ireland; {sergej.svorobej, patricia.endo, malika.bendechache, james.byrne, theo.lynn}@dcu.ie

[2] Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece ; {cfilpapadop, kgiannou, Dimitrios.Tzovaras}@iti.gr

Version June 10, 2020 submitted to Future Internet

Abstract: The fourth industrial revolution heralds a paradigm shift in how people, processes, things, data and networks communicate and connect with each other. Conventional computing infrastructures are struggling to satisfy dramatic growth in demand from a deluge of connected heterogeneous end points located at the edge of networks while, at the same time, meeting quality of service levels. The complexity of computing at the edge makes it increasingly difficult for infrastructure providers to plan for and provision resources to meet this demand. While simulation frameworks are used extensively in the modelling of cloud computing environments in order to test and validate technical solutions, they are at a nascent stage of development and adoption for fog and edge computing. This paper provides an overview of challenges posed by fog and edge computing in relation to simulation.

Keywords: Cloud computing; edge computing; fog computing; simulation; modelling; simulation challenges;

---

1. Introduction

It is commonly accepted that society is on the brink of what is termed the fourth industrial revolution (4IR), whereby cyber-physical systems will disrupt and transform entire industries and associated systems of production, management, and governance [1]. Schwab [1] posits that this revolution differs from the previous three revolutions because it is not based on breakthroughs in technology but rather communication and connectivity. At its core, 4IR is not a new concept and is couched in the concept of a networked society whose social structures and activities, to a greater or lesser extent, are organised around digital information networks that connect people, processes, things, data and networks [2,3]. As such, 4IR, Industry 4.0, and the Internet of Things (IoT) are often discussed in juxtaposition. 4IR is disruptive because it has the potential to radically redefine industrial performance trajectories and how, who, and where value is created, delivered and captured. Current exuberance about 4IR, Industry 4.0 and IoT is driven by both the advances and widespread adoption of a number of underlying technologies namely cloud computing, ubiquitous sensing, and mobile technologies, connected across a cloud-to-things (C2T) continuum. In conventional cloud computing, processing and storage typically takes place within the boundaries of a cloud and its underlying infrastructure. It is not designed to cater for the scale of geographically dispersed, heterogeneous end points and low latency required for many 4IR, Industry 4.0 and IoT use cases. As such, conventional paradigms of computing need to be rethought to cater for the scale of data processing and storage needed to support the requirements of 4IR, Industry 4.0 and IoT to function in a distributed, coordinated way at minimum latency [4].

Fog and edge computing are two relatively new paradigms of computing that have been proposed to address these challenges. NIST [4] recently defined fog computing as:

> *... a horizontal, physical or virtual resource paradigm that resides between smart end-devices and traditional cloud or data centers. This paradigm supports vertically-isolated, latency-sensitive applications by providing ubiquitous, scalable, layered, federated, and distributed computing, storage, and network connectivity.*

Edge computing, in contrast, is local computing at the network layer encompassing the smart end-devices and their users [4]. As such, edge computing in its narrow definition excludes both fog and cloud computing [5], as shown in Figure 1. Fog and edge computing provides significant advantages for processing data closer to the source and thus mitigate latency issues, lower costs of data transmission, and reduce network congestion [6,7].

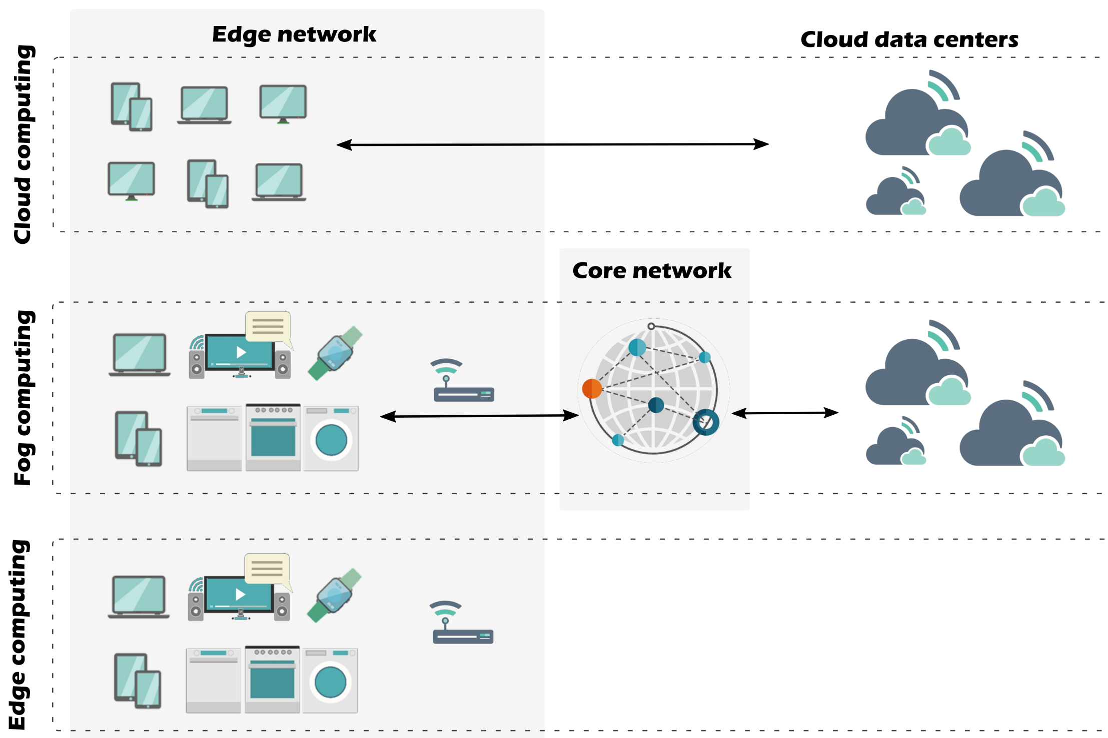

Figure 1. Cloud, fog and edge domains (based on Mahmud *et al.* [8])

Fog and edge computing provide new and significant architecture design challenges for all of those involved in the 4IR/IoT chain of service provision. Fog and edge computing use cases vary by the extent and degree of (i) contextual location awareness and low latency, (ii) geographic distribution, (iii) scale and coordination of end-point networks, (iv) heterogeneity, interoperability and functionality of end points, (v) real-time vs batch processing, (vi) mobility of end points, and (vii) interplay between the edge, the fog and the cloud layers [4]. These factors determine the extent to which quality of service levels can be met, performance bottlenecks avoided, energy consumption optimised, under-utilization reduced, and costs managed efficiently and effectively [9]. However, the complexity and scale of these use cases are orders of magnitude greater than conventional enterprise and cloud computing scenarios. Successful deployment of 4IR/IoT use cases requires optimal placement of computation and storage along the C2T continuum on a case by case basis and new approaches to test resource placement and management strategies at scale.

Like cloud computing, researchers and developers seeking to test resource placement and management strategies for fog and edge computing face a number of significant challenges. Firstly, commercial service providers typically do not give the necessary infrastructure access or control to third parties [10]. Secondly, establishing a test bed with a high degree of verisimilitude is both complex, costly, resource and time-intensive. Thirdly, from a research perspective, the use of commercial third party services and proprietary test beds limit the extent to which experiments can be validated and results reproduced.

Cloud computing researchers have attempted to overcome these issues through the application of a range of modelling techniques. For example, Petri Nets [11], Markov Chains [12], Fault Tree (FT), Reliability Graphs (RG) and Reliability Block Diagrams (RBD) [13] have all been used as analytical modelling techniques in distributed system research. A *Petri Net* is a mathematical modelling language that is commonly used to describe dynamic and parallel system behaviour in order to analyse the performance and availability metrics of systems capable of being clearly described using mathematical definitions [14]. In *Markov chains*, system models are defined as a sequence of stochastic events that can be used for estimating system behaviour using complex probability distributions. The increasing scale, complexity and heterogeneity of C2T systems renders the use of traditional mathematical modelling techniques difficult to utilise, however simulation is increasingly being adopted as a suitable approach due to its ability to model systems with such characteristics.

Given this, the use of simulation frameworks [15,16] has come to greater prominence in cloud computing research. Service configuration and resource placement and management strategies can be simulated prior to infrastructure deployment, performance can be optimised, technical and commercial hypotheses tested, and research results validated and reproduced in a low cost, low risk and often time-sensitive manner [10]. Research on fog and edge computing is still at a relatively early stage of development. Unsurprisingly, research on simulation frameworks to support fog and edge computing and the multitude of use cases that 4IR and IoT represent is lagging.

While there are a number of research surveys and articles exploring the challenges of fog computing (e.g. [17,18]), edge computing (e.g. [19,20]), simulation frameworks for cloud computing (e.g. [15,16]) and IoT data analytics (e.g. [21]), there is a paucity of publications addressing the challenges in modelling and simulating fog and edge computing specifically. This paper complements existing works on fog and edge computing by presenting the challenges and design considerations for simulation frameworks of fog and edge infrastructures. Our aim is to support both computing and simulation researchers entering this field in determining the requirements of the simulation platform that they would need to develop or employ to evaluate their technical proposals.

The remainder of this paper is organised as follows. The next section discusses challenges of fog and edge computing modelling and simulation. This is followed by a review of existing simulation frameworks for fog and edge computing and a gap analysis against support for four key criteria - *(a)* infrastructure level modelling, *(b)* application-level modelling, *(c)* resource management modelling, *(d)* mobility, and *(e)* scalability. Our analysis suggest significant gaps in relation to requirements to model mobility and scalability, after which the article concludes.

2. Fog and edge computing: modelling and simulation challenges

The increasing ubiquity of mobile technologies and low-cost connected sensors has resulted in a deluge of computational and networking end points at several orders of magnitude than previous decades. Conventional computing infrastructures, including cloud computing, leverage geographical centralised data centres using relatively homogeneous commoditized hardware. Such infrastructures were not designed to cater for the processing, storage and data generated by billions of distributed end points operating in often dynamic environments with intermittent network connectivity. Unsurprisingly, service providers have faced unprecedented challenges planning and managing for greater demands while meeting minimum service levels. Fog computing has emerged to complement cloud computing. As discussed earlier, fog computing is located between smart end-devices at the edge of networks and traditional cloud or data centres [4]. It plays an important role in reducing network congestion and facilitating location awareness, mobility support, real-time interactions, scalability and interoperability [8,22]. In contrast, edge computing, in its purest sense, defined by the exclusion of cloud and fog, and is limited to a small number of peripheral network layers [4]. Both fog and edge computing present significant challenges for service providers and researchers including application architecture design and deployment, infrastructure and network management, mobility, resource management, and scalability.

2.1. Application level modelling

There is an infinite range of potential applications for fog and edge computing, ranging from simple-IoT based sensor monitoring to the complex data processing systems inherent in Industry 4.0, e-health, smart cities etc. Consequently, underlying applications vary in their needs based on the degree of (i) contextual location awareness and low latency, (ii) geographic distribution, (iii) scale and coordination of end-point networks, (iv) heterogeneity, interoperability and functionality of end points, (v) real-time vs batch processing, (vi) mobility of end points, and (vii) interplay between the edge, the fog and the cloud layers [4,22]. Provisioning for such heterogeneity requires significant planning upfront and ongoing optimisation throughout the C2T continuum including application design.

The majority of current fog and edge services that support applications can be further divided into three main categories - Content Distribution Networks (CDN), IoT and Virtual Network Functions (VNF). While all three use the same infrastructure, the functional aspects of each type of service are fundamentally different. CDN services focus mostly on static content replication and distribution across multiple locations. IoT services are used to offload data processing and storage from sensors to edge locations selectively pushing some of the data up the network stack to the cloud. VNFs are chains of network functions that handle mobile network protocol traffic (e.g. LTE stack) or provide network traffic filtering and routing functions, such as enterprise, firewall and VPN services. Law [23] suggests "*a simulation model should always be developed for a particular set of objectives. In fact, a model that is valid for one objective may not be for another.*". Modelling all of the applications deployed within a fog/edge network can be beneficial for infrastructure providers but constructing a simulation solution that can efficiently handle a set of such broad objectives is a challenge and needs careful consideration.

2.2. Infrastructure and network level modelling

Due to cloud communication dependency and large volumes of data generated by fog and edge applications, network connectivity and capacity can be be a significant limitation, especially in the case of real-time delay-sensitive applications. This is particularly the case at the mobile edge. Mobility management is critical in mobile edge computing (MEC), especially in highly dynamic environments. To manage demand in MEC scenarios, massive numbers of small cells are deployed. In this scenario, the user range can be very limited and therefore handovers are more frequent, resulting in a heavy burden on the network [24].

Fog and edge devices make use of a wide variety of communication technologies, ranging from traditional low cost protocols, such as IEEE 802.11 to energy efficient protocols, such as IEEE 802.15.4

(ZigBee/6LoWPAN) and Bluetooth Low Energy (LE). Each of these technologies has an impact on the end point performance directly whether data processing, service time, data transfer delay etc. Arriving at the optimal network access technology, typically involves a trade-off between performance and cost and is often outside of the control of the service provider [17].

Fog and edge system models can extend to thousands of distributed site locations creating a network of resources spanning multiple countries. Each site can comprise of compute and network equipment hosting multiple applications that can be accessed by edge service users. Creating such a model by hand, even at a higher abstraction level, is no longer practical from a time and effort perspective. To solve the problem an automated approach is required for model building. Integration with a monitoring data collection system can partially address the challenge by taking snap-shot of an existing infrastructure state [25]. However, in order to build meaningful system behaviour models the monitoring data has to undergo additional processing to extract behaviour trends of workload and application resource demands. Such a process brings big data management and processing challenges into play that require further development within the scope of the simulation domain.

2.3. Mobility

Recent studies focus on the emergence of 5G networks and the interaction between these networks and fog and edge computing. 5G networks offer network improvements through optimization of mobile resource usage, large data pre-processing, and context-aware services (using cell load, user location, and allocated bandwidth as information) [20]. Notwithstanding these improvements, as each fog and edge application may have different latency requirements and may generate different types of data and network traffic, a mechanism may be required to differentiate delay-sensitive flows such as network slicing [26].

Modelling user mobility aspects requires the implementation of geographic awareness logic, for example calculation of the nearest mobile access point based on user coordinates at each simulation timestep. Furthermore, availability and access to real-world data on end user mobility is problematic both legally and technically. Additional calculations further increase the complexity and computational resource demands of a given simulation platform. Intelligent model generators are one solution for creating fog and edge infrastructure workload models based on 3rd party socio-demographic and geographic data that can be used for simulation purposes [27].

2.4. Resource management

The majority of 4IR, Industry 4.0 and IOT scenarios assume the generation, capture and analysis of data in volumes, variety and velocities orders of magnitude greater than before. This data may include useful information if such information can be identified [28]. For example, a basic connected vehicle system can generate tens of megabytes of data per second [5]. To provision infrastructure efficiently and effectively requires a number of key decisions, not least how the data will be collected, where and how data will be processed (edge, fog or cloud), and how often the data should be sent to the cloud for long-term storage or further analysis. There are two competing pressures informing these decisions - utilisation of infrastructure and end user quality of service.

Complex Event Processing (CEP) systems are increasingly cited for processing and analysing high volumes of data and detecting events of interest when they occur [28,29]. However, the CEP task can be time-consuming, and commonly fog or edge devices present computational and storage capacity limitations, when compared to cloud capacity and capability. As such, caching is used widely to bring storage functionality to network edges with lower latency, less excessive bandwidth consumption, and reduced streaming times [19,30,31]. This is particularly the case for content distribution use cases such as IP video, forecast to account for a significant portion of all IP traffic in the coming years[32]. Wang *et al.*

[19] summarise the main challenges in content caching in edge networks as caching placement, content popularity, caching policies and algorithms, and mobility awareness.

Understanding data load generation and its propagation through a given system is a worthwhile approach for deciding on (optimal) resource placements. Data load prediction has been presented as a solution for proactive system remediation [33]. In this case, historical (big) data stored at cloud and live data collected in the fog and edge devices are used to feed models and predict important metrics, such as resource usage and content popularity distribution [34]. Machine learning techniques have also been widely used to solve this problem. For instance, Zeydan *et al.* [35] use machine learning to predict the spatio-temporal user behaviour for proactive caching decisions with a goal of satisfying user demand by delivering low latency and higher QoE.

Noor *et al.* [36] identify energy efficiency as one of the significant challenges for mobile cloud resource management. For example, while offloading data processing to the cloud can reduce device battery consumption, it increases network bandwidth usage and power consumption, a significant contributor to rising energy consumption [37,38]. Unsurprisingly, energy efficiency is a major focus of cloud, fog and edge computing research including the optimisation of resource allocations under energy, performance, and QoS constraints [39–41]. Fog computing can introduce management flexibility by providing more options for data processing within the distributed network hierarchy. From the cloud perspective, deciding whether to cache or process the data offload is one way to alleviate network congestion and reduce data transfer costs; from an edge device perspective, offloading some computing tasks could enhance the service performance and be more energy efficient [42,43], since some edge devices are very energy constrained (and, at the same time, hungry energy consumers). As such, the performance profile of the edge device should be taken in to consideration, especially for real-time sensitive applications, such as vehicle-to-vehicle communications, vehicle-to-roadside communications and real-time financial trading applications, that may require latencies below tens of milliseconds [44]. Intensive benchmark performance experiments are typically required to decide the best fog-to-edge configuration considering a range of factors including computational and storage capacity, battery life, mobility, communication interface etc. The importance of system availability for such devices can be an important consideration particularly in use cases where data loss or service outage can result in adverse outcomes for end users, for example, in health care monitoring systems.

The runtime, programmability, and interoperability of edge devices often differ due their heterogeneity resulting in data offloading issues [42]. The European Telecommunications Standards Institute (ETSI), the OpenFog Consortium, and others are seeking to address standardisation for multi-access edge computing [1] however such initiatives are at an early stage of development and face uphill challenges against the onslaught of new connected end points being introduced.

The simulation approach has proven to be a worthwhile endeavour in testing resource allocation and management in cloud. In their study, Stier *et al.* [45] presents direct integration between the simulation and optimisation framework which was implemented in order to test available resource management algorithms that can be directly used within a real system. In addition simulation can also be used as part of resource management algorithms to narrow search space for the optimal solutions in an optimisation technique known as simulated annealing (SA). SA is an optimisation algorithm that uses local search approach of moving around the neighbouring values in a defined search space until the optimal solution is found [46]. Even though the simulated annealing technique goes outside the traditional scope of discrete event simulation (DES), the general annealing approach can be useful in testing different parameter

[1] https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

variations within edge computing such as virtual Content Delivery Network (vCDN) deployments [47] or exploring infrastructure provisioning options [48]. Moving to the fog and edge domain the need for simulating resource management approaches remains one of the main simulation analysis features [49].

2.5. Scalability

The choice of a simulation tool depends significantly on the type of applications. This fact also dictates the granularity of the simulation. For example, macroscopic phenomena, such as routing strategies, can be studied by packet-level, using the discrete event simulation approach. Notwithstanding this, a very accurate simulation might substantially hinder performance leading to similar results as other faster performing methods. Another key point when considering a simulation framework is the generality of the range of phenomena and applications that can be simulated. More general simulation frameworks are usually not focused on specific characteristics, but rather on a large number of parameters that may not be required by the user and may be very complex to setup and operate. These frameworks tend to cover a wide variety of applications and phenomena. On the other hand, dedicated simulation solutions are usually easier to use, tailored and optimised to specific applications and their complexity. However, dedicated solutions are not easily adaptable to other applications, without significant development effort. In our analysis of extant simulation frameworks for fog and edge computing, many are extensions of *CloudSim* and suffer from its limitations in terms of scale and focus. Others focus on specific use case scenarios. These present short term limitations for fog and cloud computing researchers but also opportunities for simulation research.

Experimenting with large scale systems requires compute resources to be available for a simulation framework to use. DES is the most popular approach used in cloud computing, as reflected in the above analysis. However, the sequential nature of the event queue is notoriously difficult to parallelize as each event can change the state of the system. Therefore, if an event is processed out of order the calculation may be incorrect. Having said that, where one can make clear divisions within the model, simulation events can be processed in independent clusters increasing the degree of parallelism. For example Varga and Sekercioglu [50] discuss a parallel discrete event simulation (PDES) approach that is capable of distribute simulation over multiple processors and machines, also avoiding memory bottlenecks by dividing the model across machines. Another example is *Cloud2Sim*, an extension of popular CloudSim [51] framework using Hazelcast and Infinispan in memory distributed data stores [52]. As seen in Table 1 parallel DES execution is not widely adopted in newly released edge simulation tools limiting their application range.

The Discrete Time Simulation (DTS) approach can be used to attempt to combat such parallelization difficulties associated with DES. DTS uses the concept of time-step to update the state of the system components, avoiding the need of pre-computation and storage of future events. This approach presents a significant reduction of the simulator's memory requirements and enhances performance while enabling parallel processing [53], and, along with reduced memory requirements, provides a mechanism for the simulation of very large networks. The state of all components involved in a simulation e.g. sites, nodes, VMs, etc, can be updated in parallel, since, there are no dependencies between components thus enhancing scalability. The change of state of the constituent components is only affected by input requests. This approach substantially simplifies the design and incorporation of advanced power consumption models and strategies for path formation on networks. Furthermore, the granularity of the simulation is controlled by the choice of timestep - choosing a smaller timestep results in a very large number of timesteps, potentially increasing the accuracy of the simulation, while hindering performance. A large timestep typically results in the undersampling of the studied phenomenon neglecting transient phenomena that might substantially affect the result of the simulation. DTS simulation has been used in the context of very large scale simulation of traditional and self* based cloud environments in supercomputing environments [54,55].

DES and DTS approaches have both strong and weak points when compared to one another. DES is generally considered to be a suitable tool when applied to a problem requiring a more granular modelling approach which is more difficult to scale, whereas DTS typically enables the modelling of large scale systems with relatively less effort, but with the possibility of a higher degree of inaccuracy [56]. Both paradigms can be applied to the fog and edge domain depending on the experimentation objectives.

In terms of accuracy, there are challenges in relation to the validation of large scale systems as simulation represents an abstracted model of a real environment, hence it is imperative that the level of abstraction does not impede simulation result accuracy. The simulation validation process ensures that simulation experiments produce reliable estimations of system behaviour. Existing approaches suggest to validate simulation models with domain experts to ensure model face validity attributes and behaviour constraints [57]. Furthermore, simulation results should be validated by comparing simulated results with monitored real system data e.g. by visually comparing simulated and monitored results plotted side by side [57] or statistically comparing data distribution e.g. using a t-test approach [58]. Validating simulation models and results is not a new challenge, however applying validation techniques to fog and edge computing simulations can be challenging. Firstly, it is difficult to inspect the target environment due to size and complexity. Secondly, lack of access to real data impacts validation by comparison. Automatic or a semi-automatic validation methodologies capable of processing high volumes of data can potentially resolve or alleviate validation challenges by checking model data for consistency and result anomalies.

3. Fog and Edge Modelling and Simulation Tools

According to Dastjerdi and Buyya [59], in order to enable real-time analytics in fog and edge computing at the software-level, we must be concerned about different resource management and scheduling techniques including resource distribution, load balancing, migration, and consolidation. At the physical layer, fog and edge systems have many additional requirements that need to be addressed, such as network connectivity and capacity. This scale and complexity of C2T systems makes the use of realistic prototypes unfeasible. Similarly, commercial service providers typically do not give the necessary infrastructure access or control to third parties to test aforementioned techniques [10] and constructing a test bed with a high degree of verisimilitude is both complex, costly, resource and time-intensive. To overcome these issues, simulation frameworks provide a relatively low cost means to understand and evaluate fog and edge systems and eliminate ineffective policies and strategies [60].

Simulation has been used extensively to simulate traditional network infrastructures, such as the mainstream Wireless Sensor Networks (WSNs). Some examples of these simulators are *NS-2*, *TOSSIM*, *EmStar*, *OMNeT++*, *J-Sim*, *ATEMU*, and *Avrora*. These simulators are universally used to develop and test network protocols, especially in the initial design stage. They were not designed with fog and edge computing environments in mind; as such, they are outside the scope of this paper. We redirect the reader to a detailed survey by [61] for further information on these simulators.

While there are a wide range of simulators for cloud computing, there are relatively few that can be used to simulate fog and edge computing scenarios. Next, we briefly describe a selection of prominent simulators used for fog and edge modelling and compare them in qualitative terms.

FogNetSim++[62] is a fog simulator tool that provides users with detailed configuration options to simulate a large fog network. It is designed on the top of *OMNeT++* [63] which is an open source tool that provides an extensive library to simulate network characteristics using discrete event simulation. *FogNetSim++* enables researchers to incorporate customised mobility models and fog node scheduling algorithms as well as managing handover mechanisms. A traffic management system is evaluated to demonstrate the scalability and effectiveness of the *FogNetSim++* simulator in terms of CPU and memory usage. The authors provide a benchmark of network parameters, such as execution delay, packet error rate, handovers, and latency. However, *FogNetSim++* does not yet support VM migration among fog nodes.

iFogSim [64] is a fog computing simulation toolkit that allows users to simulate fog computing infrastructures and execute simulated applications in order to measure performance in terms of latency, energy consumption and network usage. *iFogSim* is based and implemented over *CloudSim* [51]. *iFogSim* enables the modelling and simulation of fog computing environments for evaluating resource-management and scheduling policies. It measures performance metrics and simulates edge devices, cloud data-centres, sensors, network links, data streams, and stream-processing applications. In addition, *iFogSim* integrates simulated services for power monitoring and resource management at two separate levels i.e. 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 centres and (b) edge-ward placement, where application modules run on fog nodes close to edge devices [60]. Furthermore, extensions are available to support the design of data placement strategies according to specific objectives such as minimisation of service latency, network congestion, and energy consumption [65]. It is also worth noting that as the fog computing paradigm has many similarities to cloud computing, *CloudSim* can also be used as a standalone application to implement many features of fog computing. *iFogSim* is not without its limitations. While it enables the definition of the location of devices getting service from the fog servers, this information is static and not updated by any mobility model. In addition, while being based on *Cloudsim* provides advantages, *iFogSim* is limited to DES and its scalability is limited.

Both *EdgeCloudSim* and *IOTsim*, like *iFogSim*, are also based on *CloudSim*. *EdgeCloudSim* is specifically design to evaluate the computational and networking needs of edge computing. Unlike *iFogSim*, *EdgeCloudSim* supports mobility. In fact, it provides the mobility model, network link model, and edge server model to evaluate the various facets of edge computing. In addition to its simulation capabilities, *EdgeCloudSim* is relatively user-friendly providing a mechanism to obtain the configuration of devices and applications from the XML files instead of defining them programmatically. *IOTSim* was designed to simulate edge computing environments where large data volumes are sent to a big data processing system by the IoT application [66]. As such, it adds a storage and the big data processing layer in to *CloudSim*. In the storage layer, the network and storage delays are simulated for IoT applications. The big data processing layer simulates MapReduce to support the batch-oriented data processing paradigm. Both *EdgeCloudSim* and *IOTsim* inherit the same scalability and DES limitations as *iFogSim*.

Brogi *et al.* [67] recently presented a prototype simulator, *FogTorchII*, that extends their previous work, *FogTorch*) [68]. Primarily designed to support application deployment in the fog, *FogTorchII* is an open source simulator developed in Java. It is capable of evaluating fog computing infrastructure deployments, it models software capabilities (operating system, programming languages, frameworks etc.), hardware capabilities (CPU cores, RAM and storage), and QoS attributes including latency and bandwidth. *FogTorchII* uses Monte Carlo simulations to implement variations in communications links used as inputs. The final output consists of the aggregated results in terms of QoS-assurance and fog resource consumption through an indicator of the percentage of consumed RAM and storage. An acknowledged and major limitation of *FogTorchII* is scalability, an issue that Brogi *et al.* [67] hope to address by exploiting heuristics to reduce the search space [67].

Simulations make a number of simplifications that may not always hold true, especially with an infrastructure as dynamic as fog and edge computing. As such, a number of emulation frameworks were developed to address this limitation. *EmuFog* is an extensible emulation framework tailored for fog computing scenarios [69]. *EmuFog* enables the design of fog computing infrastructures *ab initio* and the emulation of real large scale applications and workloads which allows developers to implement and evaluate their behaviour as well as the induced workload in the network topology. The implementation process in *EmuFog* consists of four stages:

1. A network topology is either generated or loaded from a file, supporting thus real-world topology datasets.
2. 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.
3. 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.
4. Fog nodes are emulated from the network emulated environment, while the applications in any individual fog node are running under Docker containers.

Despite the usefulness of the *EmuFog*, the framework does not support mobility both for clients and fog nodes. Furthermore, *EmuFog* does not support hierarchical fog infrastructures.

Fogbed [70] is another emulator which extends the network emulator *Mininet*[71] framework to allow the use of Docker containers as virtual nodes. It provides capabilities to build cloud and fog testbeds. The *Fogbed* API enables adding, connecting and removing containers dynamically from the network topology. These features allow for the emulation of real-world cloud and fog infrastructures in which compute instances can be started and terminated at any point in time. Also, it is possible to change the run-time resource limitations for a container, such as CPU time and memory available. However, *Fogbed* does not yet support key aspects of fog computing including security, fault tolerance, scalability and reliability management.

Table 1 summarises the above simulator tools against six key qualitative attributes: (i) computing paradigm (target system), (ii) infrastructure-level modelling, (iii) application-level modelling, (iv) resource management modelling, (v) mobility, and (vi) scalability.

Table 1. Fog and Edge Simulator Tools: Comparative Study

Attributes	FogNetSim++	iFogSim	FogTorchII	EdgeCloudSim	IOTSim	EmuFog	Fogbed
Computing paradigm (target system)	Fog computing (general)	Fog computing (general)	Fog computing (general)	Edge computing (IoT)	Edge computing (IoT)	Fog computing (general)	Fog computing (general)
Infrastructure and network level modelling	Distributed data centres Sensors Fog nodes Broker Network links Delay Handovers Bandwidth	Cloud data centres Sensors Actuators Fog devices Network links Delay Network usage Energy consumption	Latency Bandwidth	Cloud data centres Network links Edge servers WLAN and LAN delay Bandwidth	Cloud data centre Latency Bandwidth	Network links Fog nodes Routers	Virtual nodes Switches Instance API Network links
Application level modelling	Fog network	Data stream Stream-processing	Fog applications	Mobile edge	IoT	Fog	Fog network
Resource management modelling	Resource consumption (RAM and CPU)	Resource consumption Power consumption Allocation policies	Resource consumption (RAM and storage)	Resource consumption (RAM and CPU) Failure due to mobility	Resource consumption (RAM, CPU and storage)	Workload	Resource consumption (RAM and CPU) Bandwidth Workload
Mobility	Yes	No	No	Yes	No	No	No
Scalability	Yes	No	No	No	Yes (MapReduce)	No	No

In summary, despite an increase interest on fog and edge computing, research on suitable simulation frameworks to support the requirements of this domain is lagging (see Table 1). Most of the existing simulation tools, albeit a small number, place a greater emphasis on fog computing. They have significant limitations in scalability and mobility support. All the existing simulators use DES at their core and the dependence on *CloudSim* for three of the simulators places an additional limitation to them particularly in terms of scalability. Therefore, there is an urgent need for simulation tools with greater coverage of characteristics of fog and edge computing.

Ficco *et al.* [72] argue that purely simulated environments and real testbeds are not sufficiently representative of real world scenarios and/or are unacceptably expensive. As such, they suggest that a hybrid *pseudo-dynamic testing* approach may increase verisimilitude by simulating a portion of the experimental scenario, while either emulating the edge and fog nodes under test or executing them in a real environment.

4. Conclusions

The emergence of the fourth industrial revolution and the Internet of Things is quickly becoming a reality. For the developer and research community, the availability of a means to test, validate, compare and reproduce technical proposals efficiently and cost-effectively is central to commercialisation and the scientific method. Like cloud computing, public clouds and test beds do not provide sufficient control of resources and infrastructure to validate technical solutions for fog and edge computing at the appropriate level of granularity. While modelling and simulation can address these issues, early attempts at simulation frameworks have significant gaps in their capability to model the complexity and specific requirements of fog and edge computing scenarios at the scale facing key stakeholders in the chain of service provision today, let alone the future.

Indeed, many existing fog and edge computing simulators derive from cloud computing simulation frameworks and may be inflected towards the cloud layer rather than the nuances of a multi-layered C2T continuum. This review of existing simulation frameworks and challenges in modelling and simulating fog and edge computing use cases provides a landscape of existing options but also a roadmap for future research in both fog and edge computing and the design of associated simulation frameworks.

Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732667 (RECAP).

References

1. Schwab, K. Will the fourth industrial revolution have a human heart. World Economic Forum, 2015.
2. Castells, M. *The Rise of the Network Society. The Information Age: Economy, Society, and Culture Volume I (Information Age Series)*; London: Blackwell, 1996.
3. Van Dijk, J. *The network society*; Sage Publications, 2012.
4. Iorga, M.; Goren, N.; Feldman, L.; Barton, R.; Martin, M.; Mahmoudi, C. *Fog Computing Conceptual Model*; Maryland: NIST, 2018.
5. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE IoT Journal* **2016**, *3*, 854–864.
6. Jain, K.; Mohapatra, S. Taxonomy of Edge Computing: Challenges, Opportunities, and Data Reduction Methods. In *Edge Computing*; Springer, 2019; pp. 51–69.
7. Mann, Z.Á. Optimization Problems in Fog and Edge Computing. *Fog and Edge Computing: Principles and Paradigms* **2019**, pp. 103–121.
8. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of everything*; Springer, 2018; pp. 103–130.
9. Cherrueau, R.A.; Lèbre, A.; Pertin, D.; Wuhib, F.; Soares, J. Edge Computing Resource Management System: a Critical Building Block! Initiating the debate via OpenStack. The USENIX Workshop on Hot Topics in Edge Computing (HotEdge'18), 2018.
10. Buyya, R.; Ranjan, R.; Calheiros, R.N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. IEEE, 2009, pp. 1–11.
11. Petri, C.A.; Reisig, W. Petri net. *Scholarpedia* **2008**, *3*, 6477.
12. Meyn, S.P.; Tweedie, R.L. *Markov chains and stochastic stability*; Springer Science & Business Media, 2012.

13. Distefano, S.; Puliafito, A. Dependability Evaluation with Dynamic Reliability Block Diagrams and Dynamic Fault Trees. *IEEE Transactions on Dependable and Secure Computing* **2009**, *6*, 4–17. doi:10.1109/TDSC.2007.70242.
14. Thong, W.J.; Ameedeen, M. A survey of Petri net tools. In *Advanced Computer and Communication Engineering Technology*; Springer, 2015; pp. 537–551.
15. Lynn, T.; Gourinovitch, A.; Byrne, J.; Byrne, P.J.; Svorobej, S.; Giannoutakis, K.; Kenny, D.; Morrison, J. A Preliminary Systematic Review of Computer Science Literature on Cloud Computing Research using Open Source Simulation Platforms. Proceedings of the 7th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER,. INSTICC, SciTePress, 2017, pp. 565–573. doi:10.5220/0006351805650573.
16. Byrne, J.; Svorobej, S.; Giannoutakis, K.M.; Tzovaras, D.; Byrne, P.J.; Östberg, P.; Gourinovitch, A.; Lynn, T. A Review of Cloud Computing Simulation Platforms and Related Environments. Proceedings of the 7th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER,. INSTICC, SciTePress, 2017, pp. 679–691. doi:10.5220/0006373006790691.
17. Kitanov, S.; Monteiro, E.; Janevski, T. 5G and the Fog—Survey of related technologies and research directions. Electrotechnical Conference (MELECON), 2016 18th Mediterranean. IEEE, 2016, pp. 1–6.
18. Hao, Z.; Novak, E.; Yi, S.; Li, Q. Challenges and software architecture for fog computing. *IEEE Internet Computing* **2017**, *21*, 44–53.
19. Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J.; Wang, W. A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access* **2017**, *5*, 6757–6779.
20. Tran, T.X.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine* **2017**, *55*, 54–61.
21. Kecskemeti, G.; Casale, G.; Jha, D.N.; Lyon, J.; Ranjan, R. Modelling and simulation challenges in internet of things. *IEEE cloud computing* **2017**, *4*, 62–69.
22. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog computing: A platform for internet of things and analytics. In *Big data and internet of things: A roadmap for smart environments*; Springer, 2014; pp. 169–186.
23. Law, A.M. How to build valid and credible simulation models. 2008 Winter Simulation Conference, 2008, pp. 39–47. doi:10.1109/WSC.2008.4736054.
24. Hung, S.C.; Hsu, H.; Lien, S.Y.; Chen, K.C. Architecture harmonization between cloud radio access networks and fog networks. *IEEE Access* **2015**, *3*, 3019–3034.
25. Svorobej, S.; Byrne, J.; Liston, P.; Byrne, P.; Stier, C.; Groenda, H.; Papazachos, Z.; Nikolopoulos, D. Towards Automated Data-Driven Model Creation for Cloud Computing Simulation. *EAI Endorsed Transactions on Cloud Systems* **2015**, *2*. doi:10.4108/eai.24-8-2015.2261129.
26. Ku, Y.J.; Lin, D.Y.; Lee, C.F.; Hsieh, P.J.; Wei, H.Y.; Chou, C.T.; Pang, A.C. 5G radio access network design with the fog paradigm: Confluence of communications and computing. *IEEE Communications Magazine* **2017**, *55*, 46–52.
27. Greasley, A.; Owen, C. Modelling people's behaviour using discrete-event simulation: a review. *International Journal of Operations & Production Management* **2018**, *38*, 1228–1244, [https://doi.org/10.1108/IJOPM-10-2016-0604]. doi:10.1108/IJOPM-10-2016-0604.
28. Flouris, I.; Giatrakos, N.; Deligiannakis, A.; Garofalakis, M.; Kamp, M.; Mock, M. Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software* **2017**, *127*, 217–236.
29. Zacheilas, N.; Kalogeraki, V.; Zygouras, N.; Panagiotou, N.; Gunopulos, D. Elastic complex event processing exploiting prediction. Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015, pp. 213–222.
30. Abdullahi, I.; Arif, S.; Hassan, S. Ubiquitous shift with information centric network caching using fog computing. In *Computational Intelligence in Information Systems*; Springer, 2015; pp. 327–335.
31. Huo, R.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Leung, V.C.; Liu, Y. Software defined networking, caching, and computing for green wireless networks. *IEEE Communications Magazine* **2016**, *54*, 185–193.
32. Cisco Systems, I. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. *Cisco, San Jose, CA, USA, White Paper C11-738085-02* **2018**.

33. Östberg, P.O.; Byrne, J.; Casari, P.; Eardley, P.; Anta, A.F.; Forsman, J.; Kennedy, J.; Le Duc, T.; Marino, M.N.; Loomba, R.; others. Reliable capacity provisioning for distributed cloud/edge/fog computing applications. 2017 European conference on networks and communications (EuCNC). IEEE, 2017, pp. 1–6.
34. Sharma, S.K.; Wang, X. Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks. *IEEE Access* **2017**, *5*, 4621–4635.
35. Zeydan, E.; Bastug, E.; Bennis, M.; Kader, M.A.; Karatepe, I.A.; Er, A.S.; Debbah, M. Big data caching for networking: Moving from cloud to edge. *IEEE Communications Magazine* **2016**, *54*, 36–42.
36. Noor, T.H.; Zeadally, S.; Alfazi, A.; Sheng, Q.Z. Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications* **2018**, *115*, 70–85. doi:10.1016/j.jnca.2018.04.018.
37. Peter Judge. Cloud Wireless Access Burns More Energy Than Data Centres – Report | Silicon UK, 2013.
38. Corcoran, P.; Andrae, A. Emerging Trends in Electricity Consumption for Consumer ICT. *Electrical and Electronic Engineering (Reports)* **2013**, pp. 1–56.
39. Wang, K.; Yang, K.; Magurawalage, C.S. Joint Energy Minimization and Resource Allocation in C-RAN with Mobile Cloud. *IEEE Transactions on Cloud Computing* **2018**, *6*, 760–770, [1509.00374]. doi:10.1109/TCC.2016.2522439.
40. Hirsch, M.; Rodríguez, J.M.; Mateos, C.; Zunino, A. A Two-Phase Energy-Aware Scheduling Approach for CPU-Intensive Jobs in Mobile Grids. *Journal of Grid Computing* **2017**, *15*, 55–80. doi:10.1007/s10723-016-9387-6.
41. Chunlin, L.; Layuan, L. Exploiting composition of mobile devices for maximizing user QoS under energy constraints in mobile grid. *Information Sciences* **2014**, *279*, 654–670. doi:10.1016/j.ins.2014.04.018.
42. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal* **2016**, *3*, 637–646.
43. Liang, K.; Zhao, L.; Chu, X.; Chen, H.H. An integrated architecture for software defined and virtualized radio access networks with fog computing. *IEEE Network* **2017**, *31*, 80–87.
44. Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal* **2016**, *3*, 854–864.
45. Stier, C.; Domaschka, J.; Koziolek, A.; Krach, S.; Krzywda, J.; Reussner, R. Rapid Testing of IaaS Resource Management Algorithms via Cloud Middleware Simulation. Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering; ACM: New York, NY, USA, 2018; ICPE '18, pp. 184–191. doi:10.1145/3184407.3184428.
46. Fanian, F.; Bardsiri, V.K.; Shokouhifar, M. A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing. *International Journal of Advanced Computer Science and Applications* **2018**, *9*. doi:10.14569/IJACSA.2018.090228.
47. Kolisch, R.; Dahlmann, A. The dynamic replica placement problem with service levels in content delivery networks: a model and a simulated annealing heuristic. *OR Spectrum* **2015**, *37*, 217–242. doi:10.1007/s00291-013-0358-z.
48. Ghosh, R.; Longo, F.; Xia, R.; Naik, V.K.; Trivedi, K.S. Stochastic Model Driven Capacity Planning for an Infrastructure-as-a-Service Cloud. *IEEE Transactions on Services Computing* **2014**, *7*, 667–680. doi:10.1109/TSC.2013.44.
49. Östberg, P.; Byrne, J.; Casari, P.; Eardley, P.; Anta, A.F.; Forsman, J.; Kennedy, J.; Duc, T.L.; Mariño, M.N.; Loomba, R.; Peña, M.A.L.; Veiga, J.L.; Lynn, T.; Mancuso, V.; Svorobej, S.; Torneus, A.; Wesner, S.; Willis, P.; Domaschka, J. Reliable capacity provisioning for distributed cloud/edge/fog computing applications. 2017 European Conference on Networks and Communications (EuCNC), 2017, pp. 1–6. doi:10.1109/EuCNC.2017.7980667.
50. Varga, A.; Sekercioglu, A. Parallel simulation made easy with omnet++. 15th European Simulation Symposium, 2003.
51. 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. *Software: Practice and experience* **2011**, *41*, 23–50.

52. Kathiravelu, P.; Veiga, L. Concurrent and Distributed CloudSim Simulations. 2014 IEEE 22nd International Symposium on Modelling, Analysis Simulation of Computer and Telecommunication Systems, 2014, pp. 490–493. doi:10.1109/MASCOTS.2014.70.
53. Filelis-Papadopoulos, C.K.; Giannoutakis, K.M.; Gravvanis, G.A.; Endo, P.T.; Tzovaras, D.; Svorobej, S.; Lynn, T. Simulating large vCDN networks: A parallel approach. *Simulation Modelling Practice and Theory* **2019**, *92*, 100 – 114. doi:https://doi.org/10.1016/j.simpat.2019.01.001.
54. Filelis-Papadopoulos, C.K.; Gravvanis, G.A.; Kyziropoulos, P.E. A framework for simulating large scale cloud infrastructures. *Future Generation Computer Systems* **2018**, *79*, 703 – 714. doi:https://doi.org/10.1016/j.future.2017.06.017.
55. Filelis-Papadopoulos, C.K.; Giannoutakis, K.M.; Gravvanis, G.A.; Tzovaras, D. Large-scale simulation of a self-organizing self-management cloud computing framework. *The Journal of Supercomputing* **2018**, *74*, 530–550. doi:10.1007/s11227-017-2143-2.
56. Buss, A.; Rowaei, A.A. A comparison of the accuracy of discrete event and discrete time. Proceedings of the 2010 Winter Simulation Conference, 2010, pp. 1468–1477. doi:10.1109/WSC.2010.5679045.
57. Sargent, R.G. Verification and validation of simulation models. Proceedings of the 2009 Winter Simulation Conference (WSC), 2009, pp. 162–176. doi:10.1109/WSC.2009.5429327.
58. Sargent, R.G. A new statistical procedure for validation of simulation and stochastic models. *ELECTRICAL ENGINEERING AND COMPUTER SCIENCE* **2010**, *SYR-EECS-2010-06*.
59. Dastjerdi, A.V.; Buyya, R. Fog computing: Helping the Internet of Things realize its potential. *Computer* **2016**, *49*, 112–116.
60. Moysiadis, V.; Sarigiannidis, P.; Moscholios, I. Towards Distributed Data Management in Fog Computing. *Wireless Communications and Mobile Computing* **2018**, *2018*.
61. Yu, F.; Jain, R. A survey of wireless sensor network simulation tools. *Washington University in St. Louis, Department of Science and Engineering* **2011**.
62. 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.
63. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2008, p. 60.
64. 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. *Software: Practice and Experience* **2017**, *47*, 1275–1296.
65. Naas, M.I.; Boukhobza, J.; Parvedy, P.R.; Lemarchand, L. An Extension to iFogSim to Enable the Design of Data Placement Strategies. Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on. IEEE, 2018, pp. 1–8.
66. Zeng, X.; Garg, S.K.; Strazdins, P.; Jayaraman, P.; Georgakopoulos, D.; Ranjan, R. IOTSim: a Cloud based Simulator for Analysing IoT Applications. *arXiv preprint arXiv:1602.06488* **2016**.
67. Brogi, A.; Forti, S.; Ibrahim, A. How to best deploy your Fog applications, probably. Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on. IEEE, 2017, pp. 105–114.
68. Brogi, A.; Forti, S. QoS-aware deployment of IoT applications through the fog. *IEEE Internet of Things Journal* **2017**, *4*, 1185–1192.
69. Mayer, R.; Graser, L.; Gupta, H.; Saurez, E.; Ramachandran, U. EmuFog: extensible and scalable emulation of Large-Scale fog computing infrastructures. Fog World Congress (FWC), 2017 IEEE. IEEE, 2017, pp. 1–6.
70. Coutinho, A.; Greve, F.; Prazeres, C.; Cardoso, J. Fogbed: A rapid-prototyping emulation environment for fog computing. 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–7.
71. De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. Using mininet for emulation and prototyping software-defined networks. 2014 IEEE Colombian Conference on Communications and Computing (COLCOM). IEEE, 2014, pp. 1–6.

72. Ficco, M.; Esposito, C.; Xiang, Y.; Palmieri, F. Pseudo-Dynamic Testing of Realistic Edge-Fog Cloud Ecosystems. *IEEE Communications Magazine* **2017**, *55*, 98–104.