



Resource Management and Simulation Tools in Fog Computing – A comparative Study

Vadde Usha¹, Dr Vijaya Sri Kompalli²

^{1,2} Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502

¹Email: vaddeusha@rediffmail.com

²Email: kompalliv@kluniversity.in

ABSTRACT

Internet of Things (IoT) evolved as a core technology in building smart, healthcare and industry automation applications. The massive data produced by these applications has to be processed. Present Cloud-IoT models offer cost effectiveness, scalability and reliability. Nevertheless, the large quantity of data and rate at which it is generated by IoT devices challenge the bottleneck between cloud and IoT devices. Therefore, processing the tasks always on cloud may not be an effective solution for latency sensitive IoT applications. Fog computing is a distributed paradigm that brings cloud resources closer to the devices (sensors/smart) that generate data there by addressing the above challenges. Resource management is a significant issue for enhanced utilization of fog resources and to provide best possible service to IoT applications. This survey paper emphasizes on state-of-the-art in resource management and simulation tools in line with comparison of various edge computing paradigms. The study concludes specifying the future challenges that needs to be addressed in the upcoming research.

Key words: edge computing, fog computing, fog simulation tools, resource management

1. INTRODUCTION

The Internet of Things paradigm interconnects various devices through the Internet, collect and share data from physical environments. The advancements in computing and networking technologies made IoT applications expand in many domains like industry automation, healthcare and gaming. The current cloud based IoT applications may send the data to distant cloud for storage, analysis or processing. As the data becomes voluminous, offloading the data to the cloud is not effective because of the bandwidth limitation. Also, the low latency demands of IoT applications cannot be handled by remote cloud.

CISCO has introduced fog computing to handle the issues of latency sensitive, high-bandwidth and geographically

distributed applications [1]. Fog computing facilitates processing and storage on the fog nodes in the vicinity of IoT devices. Any networking components like router, switch or gateway can act as a fog node. They can be placed anywhere between cloud to things. Figure 1 shows fog computing paradigm. Other similar computing paradigms such as mobile edge computing, mist computing, and cloudlets are also emerging alongside. In this paper, the fog computing is compared with these paradigms and perceive that fog suits best for IoT applications due to its close proximity.

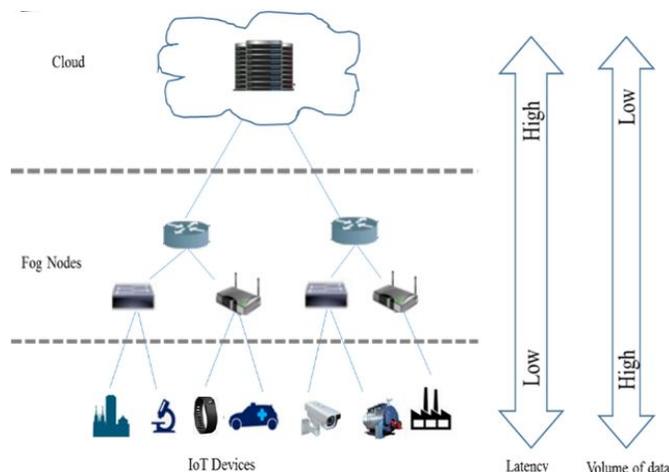


Figure 1: Fog Computing paradigm

Organization of this paper is as follows: Section 2 deals with fog computing and its contemporary technologies. State-of-the-art works are summarized in section 3. Review of simulation tools is presented in section 4. Section 5 deals with the fog computing challenges and in the last section the conclusion is presented.

2. Fog computing and related computing paradigms – An analogy

In this section, fog paradigm is compared with the cloud computing and its contemporary technologies. It also presents an insight into the role of these technologies with relevance to connected devices.

2.1 Cloud Computing

Cloud computing enhances the facilities of storage, processing power and network infrastructure facilities to its users. A cloud data center provides on demand access to the resources and can be reconfigured based on the user requirement. Cloud providers offer different services[2]. The cloud users will make use of these resources and pay based on the utilization. The cloud users are able to perform ubiquitous computing by accessing the necessary resources from the cloud. But, accessing the resources may involve some delay. This delay is not tolerable for the mission critical and latency sensitive applications belong to smart grid, medical, and industrial fields. Also, the connected devices are producing voluminous data, necessitates cloud resources must be closer to devices that are generating the data. Therefore, the requirements of smart applications can be met. CISCO has come up with fog computing paradigm to deal with the above issues

2.2 Fog Computing

Fog Computing uses a distributed paradigm in which fog nodes offer cloud like services within the vicinity of devices that generate the data. The close proximity of fog nodes results in faster computation, storage and data management. Fog nodes can be situated anywhere in the IoT-to-Cloud path. Accordingly multitier (three or more) application deployment is possible [4]. As the latency aware computation is mandatory for IoT applications, physical proximity also comes into play [5]. Because of this close proximity fog computing better caters the needs of IoT applications in comparison to other similar models. Manufacturing, transportation, health care, smart cities are popular use cases of fog paradigm [6].

2.3 Cloudlet Computing

Cloudlet is a server with lower scale capabilities compared to cloud[7]. It is available in close proximity of the user. Cloudlet supports three tier (device, edge and cloud) hierarchical framework [4]. It addresses the major drawback of Mobile Cloud Computing (MCC). MCC amalgamate cloud computing and mobile computing. In MCC, storage and processing takes place outside of the mobile device. MCC often offloads greater part of the computation to cloud to minimize the battery usage of mobile device. Thus, allowing mobile devices to run computational intensive applications like crowd sourcing, image processing, language translators. But MCC faces a major challenge in the form of latency because of the mobile device constraints [8]. To handle this challenge, the research community has provided cloudlet computing. Cloudlet has high potential in domains such as wearable cognitive assistance and mobile applications.

In spite of the fact that cloudlet addresses very well the latency issues of MCC, it supports exactly 3 tier architecture (mobile-cloudlet-cloud). But fog inherently supports huge

traffic and permits 3 or more tier architecture, as a result close proximity of fog nodes can be achieved.

2.4 Mobile Edge Computing(MEC)

MEC provides facilities of cloud computing and information technology services at the edge. Operators locate their MEC servers at Radio Access Network (RAN). Hence, services are also offered within the coverage region of RAN [9]. The closeness of RAN can aid the mobile subscribers with low latency and high bandwidth there by achieving Quality of Service (QoS). MEC supports two (device and edge location) or three (device, edge and cloud) tier hierarchical framework for application deployment [4]. MEC is better suited for content delivery and mobile big data analytics and mobile video processing applications. In MEC topology of the servers is restricted due to the coverage region of RAN where as in fog computing the topology is more flexible.

2.5 Mist Computing

Mist Computing capture a more extreme edge of connected devices [10]. Mist computing takes place in sensors and actuators only. This is proposed with keeping more advanced future systems like autonomic and self-aware systems in mind [11]. Micro controllers and microcomputers will be used as first level infrastructure in the IoT-fog-cloud path. The data can be forwarded to the nearby fog nodes and finally to a cloud. In mist computing, computations can be done on the sensor itself. But, there is a great limitation in their processing ability. Mist computing fits well in the situation where extended battery life is necessary.

From the previous discussion, it can be concluded that each paradigm has its own benefits and limitations resulting in their suitability to different use cases. Flexibility of fog computing allows it to suitable for a wide range of latency sensitive and data centric applications [12]. This does not mean that fog suits for all domains. To say, mist computing is a better choice in disaster regions. Nevertheless, most of the times fog computing is considered because of its multi-tier hierarchy. Table 2 presents the comparison of various paradigms.

3. State-of- the-art works

This section gives an insight into the recent works carried in fog computing paradigm focusing various aspects of resource management.

OlenaSkarlat et al. suggested a framework for fog resource provisioning [13]. Their work introduces fog colonies, fog cells and fog orchestration control node. Fog cell is a software components running on fog device. Fog orchestration control node manages these fog cells. Fog cells, receive the task request and allocate resources if available, otherwise it passes

Table 2: Comparison of fog computing and related paradigms

Feature	Cloud Computing	Fog Computing	Cloudlet Computing	Mobile Edge Computing	Mist Computing
Hardware	Large scale data centers	Access points, Routers, Switches and Gateways	Low scale data center	Servers at base station	Sensors, cell phones, home appliances
Location of servers	Installed in large dedicated buildings	Anywhere between end device and cloud	Near mobile devices (Close proximity to the user)	Radio network controller (at the edge)	–
Architecture	Centralized/ hierarchical	Decentralized/ hierarchical	Localized	Localized/ hierarchical	Localized/ distributed
Distance from the user	Far	Relatively close	Close	Close	Very close
Access Mechanism	WAN	WAN, Cellular, Wi-Fi	WLAN, LAN, Wi-Fi	WAN, Cellular	LAN, Bluetooth, Wi-Fi, Cellular
Latency	Relatively High	Low	Low	Low	Moderate
Location Awareness	No	Yes	Yes	Yes	Yes
Heterogeneity Support	Yes	Yes	No	No	Yes
Mobility Support	No	Yes	Yes	Yes	Yes

the request to fog orchestration control node in turn to the cloud. With this concept the authors provide a suitable resource provisioning solution for distributing the tasks. They simulated their work using cloudsims.

OlenaSkarlat et al. proposed optimized model for service placement problem[14]. The authors have adopted the fog framework from their earlier work[13]. While placing the service, their model prioritizes the applications that have closer deadline. The results have shown that their model places 70% of services on fog for processing thereby reducing execution cost by 35%. No violation of application deadlines was observed.

Wen et al. emphasizes the significance of fog orchestration to maintain deployed applications in distributed systems[15]. The authors focused on scalability, reliability and security challenges in developing the methodologies.

W.Bao et al. proposed a handover framework, ‘Follow Me Fog’[16]. This framework allows seamless handover of mobile IoT devices between different access points. This framework offers guarantee for service continuity and reduced latency during handovers. The idea behind this is to constantly monitor the received signal strength of the fog nodes and whenever the handover is expected then follow me fog, pre-migrates the computation jobs.

Gu et al. provided an ideal solution for fog service placement[17]. They developed a two-phase heuristic algorithm based on linear programming. The different factors considered by the authors while formulating service placement problem are task distribution, user association, QoS constraints and VM placement. Yousefpour et al. recommended a frame work for IoT applications[18]. Their aim was to reduce the service delay. They

suggested a delay-minimizing collaboration and offloading policy for fog-capable devices. They have evaluated their policy by developing an analytical model. The results show that the proposed framework helps to reduce IoT service delay.

Lee et al. study online fog network formation and task distribution in a hybrid cloud-fog environment [19]. Their framework dynamically constructs a fog network for any given node by selecting most suitable nearby fog nodes. The fog nodes may join and leave the network dynamically, an online approach is proposed for quickly obtaining the details of the fog network thereby minimizing computational latency. Their online k-secretary algorithm let a given fog node to monitor its unknown environment and determine how to offload computational tasks.

Mahmud et al. suggested fuzzy logic model and linear optimization technique to place the application in fog environment based on user expectations and status of fog instances[20]. With the proposed policy the authors noticed considerable improvement in quality of experience gain of the user. Mahdi Abbasi et al. presented learning classifier systems to allocate workloads between fog and cloud[21]. The idea is to minimize the energy consumption at the fog layer as well to reduce processing time. The authors, with their proposed system observed 42% reduction in processing time and reasonable energy consumption at fog layer

R. Beraldi et al. proposed power of random choices, a load balancing algorithm for fog computing[22]. The node that receives a task triggers the algorithm only when its current load is above a threshold value. This algorithm probes only a small set of nodes for job execution. Among them least loaded node will be allocated the task. If two nodes have

Table 3: Summary of state-of-the-art works in resource management

Author	Proposed Algorithm /Method	Evaluation Process	Workload	Method used for	Other considerations
Olena Skarlat et.al [13]	System Model for resource provisioning	Simulation (Cloudsim)	Synthetic	Resource Provisioning	Minimization of delay
Olena Skarlat et.al [14]	Linear Programming model	Simulation (iFogsim)	Synthetic	Service placement	Deadlines of applications
Wen et al. [15]	Parallel Genetic algorithm – framework for fog orchestration	Simulation (SPARK)	Synthetic	Management-Deployment and scheduling	Reliability, scalability
Wei Bao et al. [16]	Follow Me Fog-Framework for seamless handover	Prototy		Handover	Mobility, Latency
Gu et al. [17]	2-Phase heuristic algorithm based on linear programming	Simulation	Synthetic	Service Placement	Latency and min. cost for deploying
Yousefpour et al.[18]	Framework	Simulation	Synthetic	Offload	Reduced Service delay
Lee et al. [19]	Online k-secretary algorithm	Simulator	Synthetic	Offload	latency
Mahmud et al. [20]	Fuzzy logic based algorithm for prioritization of the task and linear optimization for mapping application requests to fog instances	Simulator (iFogsim)	Synthetic	Placement of applications	QoE & latency
Mahdi Abbasi et al.[21]	Learning classifier system- XCS and BCM-XCS	MATLAB R2014	Synthetic	Resource Management & Workload allocation	Power consumption at fog and reducing delays
R. Beraldi et al[22]	Power-of-d random choices load balancing algorithm	Simulation & P2PFaaS (Own framework)	Synthetic	Load balancing	Latency
Aazam et al. [23]	Dynamic resource estimation	Simulation (Cloudsim)	Real IoT traces	Resource allocation	Minimizing underutilization of resources

same workload, then the node will be chosen randomly.

Aazam et al.[23] provided a dynamic resource estimation model. The objective of this model is to minimize resource underutilization. While estimating, the model considers relinquish probabilities of the users, the quality of experience and historical records of customers.

The summary of reviewed papers is presented in table 3 and from the summary, we can conclude that most of the works addressed service placement in fog. Hence, future works can address resource scheduling and resource allocation. Further, they can focus on energy efficiency, cost, QoS in the fog.

4. Simulation tools for fog computing

Real time implementation of fog computing is not practical in most cases and will be expensive. To overcome these limitations, simulation frameworks facilitate simulation of fog environment and analyze various parameters like energy consumption, latency and network congestion[24]. Next, we briefly explain and compare different simulators that are popular.

Qayyum et al. proposed FogNetSim++, a fog simulation tool with detailed configuration options to simulate a large fog network [25]. OMNeT++ is an open source tool and has an extensive library to simulate network characteristics[26]. FogNetSim++ is designed on top of this tool. FogNetSim++

Table 4: Comparison of simulation tools

Attribute	FogNetSim++	iFogSim	FogTorchII	Emufog	Fogbed
Computation Paradigm	Fog Computing	Fog Computing	Fog Computing	Fog Computing	Fog Computing
Network level modeling/ Infrastructure	Handover, Delay and Bandwidth. Distributed data centers, Fog nodes, Sensors, Brokers,	Network usage, Delay and energy consumption. Cloud data centers, Fog devices, Sensors, Actuators	Latency and Bandwidth	Network links, routers and fog nodes	Network links, virtual nodes and switches
Resource Management	Resource consumption (RAM and CPU)	Resource consumption, Power consumption	Resource consumption (RAM and Storage)	Workload	Resource consumption (RAM and CPU), Workload, bandwidth
Mobility	Yes	No	No	No	No
Scalability	Yes	No	No	No	No
Availability of Version	Open source	Open source	Open source	Open source	Open source

allows researchers to incorporate fog node scheduling algorithms, customized mobility models and managing handover mechanisms. The efficiency of the FogNetSim++ in connection with CPU and memory usage is illustrated by evaluating a traffic management system. The authors have given a benchmark of network parameters like packet error rate, execution delay, and latency. Nevertheless, among fog nodes VM migration is not supported by FogNetSim++.

Guptha et al. proposed iFogSim, an extension of cloudsim[27], to build fog infrastructure and execute the applications[28]. The performance metrics like terms of latency, network and energy usage can be measured and analyzed. Mainly, iFogSim provides the fog environment for evaluating scheduling and resource management. It has different classes corresponding to Fog devices, applications, tuples, sensors and actuators. To facilitate multiple deployments, two modules, namely, cloud-only placement and edge-ward placement are available. Furthermore, to support data placement strategies according to specific objectives like minimization of service latency, energy consumption and network congestion extensions are available [29]. However, iFogSim is having its own limitations like static device location and limited scalability.

Brogi et al. presented a prototype simulator, FogTorchII[30], an extension of FogTorch [31]. The main purpose of this simulator is to support composite application deployment in fog environment. FogTorchII uses Monte Carlo simulations to consider QoS variations in communications links. Finally, it provides most eligible deployment considering resource consumption and QoS-assurance. The major limitation of FogTorchII is scalability.

Mayer et al. presented EmuFog, an extensible emulation framework adapted for fog computing [32]. This framework facilitates in building fog computing infrastructure, emulation of Real applications and workloads. This permits developers to evaluate generated workload in the network topology. EmuFog implementation will begin by generating a network topology and converting this topology into an undirected graph. Then, arrangement of fog nodes will be according to the placement policy. Finally, applications in fog nodes will run under docker containers. The EmuFog does not support mobility. Furthermore, the hierarchical fog infrastructure is not supported by EmuFog.

Coutinho et al. presented Fogbed [33], another emulator that extends Mininet [34], the network emulator and docker open source software. Cloud and fog testbeds can be provided by Fogbed. Using Fogbed API, fog nodes can be deployed as software containers. Because of this ability real world infrastructure (cloud/fog) can be emulated. Also, it is possible to change run-time resource limitations for a container at run time, such as memory available and CPU time. Fogbed doesn't support fault tolerance, scalability and reliability.

In summary, in spite of having lots of demand for fog and edge computing applications, research on simulation frameworks in this domain are lagging. The majority of the simulation tools have constraints like scalability and mobility support (Table 4). Hence, the research community is in great need of tools which supports all the features of edge and fog computing.

5. Research Challenges

Fog frameworks face new challenges with the current service provisioning methods when reliability and availability are considered. For example, to process streams of data, if a fog service needs some functions, providing extra replicas of those functions can enhance the availability and reliability of the service. Nevertheless, allocation of extra function replicas is not a good choice because of the fog nodes limited resources. Fault tolerance and reliability can be given focus in the future to realize high performance fog computing [35]

Presently, in fog computing Service Level Agreement (SLA) is not defined. A future research direction to identify QoS parameters to consider while defining SLA[36]

Previous works that are addressing load balancing, offloading or service provisioning have considered only a few objectives like QoS, cost and assume other objectives do not affect the problem[37]. A new research direction will be to design schemes that consider multi objectives (QoS, bandwidth, energy, and cost) simultaneously.

A simple client(IoT devices)-server(fog or cloud) model is assumed for current fog computing resource models to process their requests. Fog nodes either offload computation to another fog node in the vicinity or to the cloud. In the future, researchers may work towards applying peer-to-peer (P2P) architecture to fog environment [38]. This allows fog nodes to share their resources in a P2P manner without an intermediary third-party. This architecture best suits when there is no connectivity to the cloud due to the disasters.

6. Conclusion

The IoT accelerates the digital transformation, thereby automation in diversified fields such as smart cities, education, transportation and, healthcare. The connected devices in these fields are producing massive data. Fog computing paradigm is best suited to handle and process this data. In this paradigm cloud facilities are brought to close vicinity of IoT devices, thus it satisfies the latency requirements of IoT applications. The first section of this study presented the captivating factor of fog computing and followed by a comparative study with its contemporary paradigms. Next, a survey of state-of-the-art resource management is tabulated. Research challenges are highlighted at the end.

REFERENCES

[1] Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are Cisco white paper 2015.
 [2] Syed.Karimunnisa, Dr.Vijaya Sri Kompalli. **Cloud Computing: Review on Recent Research Progress and Issues**, April 2019, International Journal of

Advanced Trends in Computer Science and Engineering Volume 8, No.2, ISSN 2278-3091.
<https://doi.org/10.30534/ijatcse/2019/18822019>
 [3] L.M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner. **A break in the clouds: towards a cloud definition**, ACM SIGCOMM Comput. Commun. Rev. 39 (1) (2008) 50–55.
<https://doi.org/10.1145/1496091.1496100>
 [4] Guenter I. Klas. **Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium**, ETSI MEC and Cloudlets, 2015
 [5] A Ahmed, E Ahmed. **A survey on mobile edge computing** in 10th international conference on Intelligent Systems and Control (ISCO), IEEE, 2016.
 [6] Stefan Gendita Bunawan, Muhammad Aldenny, Dina Ikramina Setiani, Gunawan Wang. **Architecture Internet of Things Based on Cluster Housing Security System Using Fog Computing**, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.6, December 2019 , ISSN 2278-3091.
<https://doi.org/10.30534/ijatcse/2019/68862019>
 [7] M Satyanarayanan, P Bahl, R Caceres. **The Case for VM-Based Cloudlets in Mobile Computing**, Pervasive Computing, IEEE, vol. 8, no:4, pp. 14-23,2009
<https://doi.org/10.1109/MPRV.2009.82>.
 [8] H Qi, A Gani. **Research on Mobile Cloud Computing: Review, Trends and perspectives**, in second international conference on Digital Information and Communication Technology and its applications, 2012.
 [9] ETSI PORTAL, “Mobile-Edge Computing – Introductory Technical White Paper” 2014.
 [10] A. Davies. **Cisco pushes iot analytics to the extreme edge with mist computing**. [Online]. Available: <http://rethinkresearch.biz/articles/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2>, Blog, ethink Research.
 [11] J.S. Preden , K. Tammemaie , A. Jantsch , M. Leier , A. Riid , E. Calis. **The benefits of self-awareness and attention in fog and mist computing**, *Comput. (Long Beach Calif)* 48 (7) (2015) 37–45 .
<https://doi.org/10.1109/MC.2015.207>
 [12] Siwoo Byun. **Gateway-based Resource Control for Reliable IoT Environments**, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.5,October 2019 , ISSN 2278-3091.
<https://doi.org/10.30534/ijatcse/2019/69862019>
 [13] O. Skarlat, S. Schulte, M. Borkowski, P. Leitner. **Resource Provisioning for IoT Services in the Fog"**, *ninth IEEE International Conference on Service Oriented Computing and Applications (SOCA 2016)*, pp. 32-39, 2016.
<https://doi.org/10.1109/SOCA.2016.10>
 [14] O Skarlat, S Schulte, M Borkowski. **Towards QoS-Aware Fog Service Placement**, in 1st

- international conference on Fog and Edge Computing (ICFEC), IEEE, 2017
- [15] Z. Wen , R. Yang , P. Garraghan , T. Lin , J. Xu , M. Rovatsos. **Fog orchestration for internet of things services**, IEEE Internet Comput. 21 (2) (2017) 16–24 .
- [16] W. Bao , D. Yuan , Z. Yang , S. Wang , W. Li , B.B. Zhou , A.Y. Zomaya. **Follow me fog: toward seamless handover timing schemes in a fog computing environment**, IEEE Commun. Mag. 55 (11) (2017) 72–78 .
- [17] L. Gu , D. Zeng , S. Guo , A. Barnawi , Y. Xiang. **Cost efficient resource management in fog computing supported medical cyber-physical system**, IEEE Trans. Emerg. Top Comput. 5 (1) (2017) 108–119 .
- [18] A. Yousefpour , G. Ishigaki , R. Gour , J.P. Jue. **On reducing iot service delay via fog offloading**, IEEE Internet Things J. 5 (2) (2018) 998–1010 .
- [19] G. Lee , W. Saad , M. Bennis. **An online secretary framework for fog network formation with minimal latency**, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6 .
<https://doi.org/10.1109/ICC.2017.7996574>
- [20] Mahmud R, Srirama S.N, Ramamohanarao K, Buyya R .**Quality of experience (QoE)-aware placement of applications in Fog computing environments**, J. Parallel Distributed Computing, 132 (2019), pp. 190-203
- [21] Mahdi Abbasi, Mina Yaghoobikia, Milad Rafiee, Alireza Jolfaei, Mohammad R. Khosravi. **Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems**, Computer Communications, Volume 153, 2020, Pages 217-228.
- [22] R. Beraldi and G. Proietti Mattia. **Power of random choices made efficient for fog computing**, in IEEE Transactions on Cloud Computing. 2020, doi: 10.1109/TCC.2020.2968443
- [23] Aazam M., St-Hilaire M., Lung CH., Lambadaris I., Huh EN. (2018). **IoT Resource Estimation Challenges and Modeling in Fog**. In: Rahmani A., Liljeberg P., Preden JS., Jantsch A. (eds) Fog Computing in the Internet of Things. Springer, Cham
- [24] Moysiadis, V.; Sarigiannidis, P.; Moscholios, I. **Towards Distributed Data Management in Fog Computing**. Wirel. Commun. Mob. Comput. 2018. <https://doi.org/10.1155/2018/7597686>
- [25] 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.
- [26] 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; p. 60.
- [27] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya. **CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms**, Software - Practice and Experience, vol. 41, pp. 23-50, 2011. <https://doi.org/10.1002/spe.995>.
- [28] 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.
- [29] Naas, M.I. Boukhobza, J. Parvedy, P.R. Lemarchand, L. **An Extension to iFogSim to Enable the Design of Data Placement Strategies**. In Proceedings of the 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC), Washington, DC, USA, 1–3 May 2018; pp. 1–8.
- [30] 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.
<https://doi.org/10.1109/ICFEC.2017.8>
- [31] Brogi, A.; Forti, S. **QoS-aware deployment of IoT applications through the fog**. IEEE Internet Things J. 2017, 4, 1185–1192.
- [32] Mayer, R.; Graser, L.; Gupta, H.; Saurez, E.; Ramachandran, U. **EmuFog: extensible and scalable emulation of Large-Scale fog computing infrastructures**. In Proceedings of the 2017 IEEE Fog World Congress (FWC), Santa Clara, CA, USA, 30 October–1 November 2017; pp. 1–6.
- [33] Coutinho, A.; Greve, F.; Prazeres, C.; Cardoso, J. **Fogbed: A rapid-prototyping emulation environment for fog computing**. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.
- [34] De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. **Using mininet for emulation and prototyping software-defined networks**. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia, 4–6 June 2014; pp. 1–6.
- [35] Mahmud R, Kotagiri R, Buyya R. **Fog Computing: A Taxonomy, Survey and Future Directions** In: Di Martino B., Li KC., Yang L., Esposito A. (eds) Internet of Everything. Pp. 103-130, Internet of Things
- [36] R. K. Naha *et al.*, **Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions**, in IEEE Access, vol. 6, pp. 47980-48009, 2018.
- [37] Z. Chang , Z. Zhou , T. Ristaniemi , Z. Niu. **Energy efficient optimization for computation offloading in fog computing system**, in: GLOBECOM 2017–2017

IEEE Global Communications Conference, IEEE, 2017, pp. 1–6 .

- [38] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, Jason P. Jue. **All one needs to know about fog computing and related edge computing paradigms: A complete survey**, Journal of Systems Architecture, Volume 98, 2019, Pages 289-330.
<https://doi.org/10.1016/j.sysarc.2019.02.009>