

# Fog and Cloud Computing Optimization in Mobile IoT Environments

José Carlos Ribeiro Vieira  
josecarlosvieira@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa  
Advisors: Prof. António Manuel Raminhos Cordeiro Grilo  
Prof. João Coelho Garcia

**Abstract.** We introduce a xxxxx

**Keywords:** Cloud computing, fog computing, mobility, optimization, multi-objective

# Table of Contents

1	Introduction . . . . .	4
2	Objectives . . . . .	6
3	Related Work . . . . .	7
3.1	iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments	7
3.2	[123] 2018 When clones flock near the fog . . . . .	7
3.3	[143] 2018 Virtual fog: A virtualization enabled fog computing framework for internet of things . . . . .	7
3.4	[172] 2016 Vehicular fog computing: A viewpoint of vehicles as the infrastructures . . . . .	7
3.5	[186] 2018 Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning . . . . .	8
3.6	[197] 2016 Incremental deployment and migration of geo-distributed situation awareness applications in the fog . . . . .	8
3.7	[217] 2017 Mobile edge cloud network design optimization . . . . .	8
3.8	[231] 2015 Dynamic service migration and workload scheduling in edge-clouds	8
3.9	[232] 2016 Segue: Quality of service aware edge cloud service migration . . . .	9
3.10	[234] 2017 Optimizing service replication for mobile delay-sensitive applications in 5g edge network . . . . .	9
3.11	[237] 2013 Migcep: operator migration for mobility driven distributed complex event processing . . . . .	9
3.12	[243] 2017 Follow me fog: Toward seamless handover timing schemes in a fog computing environment . . . . .	9
3.13	[244] 2017 Efficient service handoff across edge servers via docker container migration . . . . .	9
3.14	[246] 2016 Primal: Profit maximization avatar placement for mobile edge computing . . . . .	10
3.15	[247] 2017 Towards efficient edge cloud augmentation for virtual reality mmogs . . . . .	10
3.16	[254] 2018 Move with me: Scalably keeping virtual objects close to users on the move . . . . .	10
3.17	[261] 2017 Virtual machine placement for backhaul traffic minimization in fog radio access networks . . . . .	10
3.18	[270] 2016 An adaptive cloudlet placement method for mobile applications over gps big data . . . . .	10
3.19	[285] 2018 Mobility support for fog computing: An sdn approach . . . . .	11
3.20	[385] 2017 Myifogsim: A simulator for virtual machine migration in fog computing . . . . .	11
3.21	2016 Scalable Fog Computing with Service Offloading in Bus Networks . . . .	11
3.22	2017 Fog Computing for the Internet of Mobile Things: issues and challenges	11
3.23	2018 Dynamic Mobile Cloudlet Clustering for Fog Computing . . . . .	11
4	Architecture . . . . .	11
5	Evaluation . . . . .	12
6	Schedule of Future Work . . . . .	12

7	Conclusion .....	12
---	------------------	----

## 1 Introduction

Cloud computing is a computing technology that became popular at the beginning of the twenty-first century, which provides users online access to services, employing large groups of computers, servers, disks, and routers interlinked together in a distributed and complex manner. Cloud computing has been imperative in expanding the reach and capabilities of computing, storage, data management, and networking infrastructure to the applications. The key idea in this model is that clients outsource the allocation and management of resources (hardware or software) that they rely upon to the cloud. Clouds can provide different service models according to the users' requirements, like infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). Since the demand for cloud resources will change over time, setting a fixed amount of resources results in either over- or under-provisioning, so cloud service providers (CSPs) afford dynamic resources for a scalable workload, applying a pay-as-you-go cost model where clients only pay for the amount of resources they actually use. Cloud computing brings many advantages to the end user applications like high availability, flexibility, scalability, reliability, to mention a few.

Although cloud computing has brought forth many advantages, it has certain limitations. Since cloud servers reside in remote data centers, end-to-end communication may have long delays (characteristic of multi-hops transmissions over the Internet), so the time required to access cloud-based services may not be suitable for some applications with ultra-low latency requirements (real-time). Augmented reality applications that use head-tracked systems, for example, require end-to-end latencies to be less than 16 ms [1]. Cloud-based virtual desktop applications require end-to-end latency below 60 ms if they are to match QoS of local execution [2]. Remotely rendered video conference, on the other hand, demand end-to-end latency below 150 ms [3].

Despite the fact that mobile devices have evolved radically in the last years, battery life, computation and storage capacity remain limited. This means that, heavy application executions must be offloaded to cloud servers, which then return processed results. The solution that has already been proposed is to bring the cloud closer to the end users, where entities such as base-stations would host smaller sized clouds. This idea has been variously termed as Cloudlets [4], Fog Computing [5], Edge Computing [6], and Follow Me Cloud [7], to name a few.

Fog computing is a new computing architecture introduced in 2012 by Bonomi et al. [5], that later, in 2015, big companies such as Cisco Systems, ARM Holdings, Dell, Intel, Microsoft, and Princeton University, founded the OpenFog Consortium, to promote interests and development in this field [8]. It aims to enable computing, storage, networking, and data management not only in the cloud, but also along the cloud-to-thing path as data traverses to the cloud (preferably close to the IoT devices). Fog nodes, also known as fog servers or cloudlets (smaller sized clouds with lower computational capacity), can be placed close to IoT source nodes, due to low hardware footprint and low power consumption (e.g., small servers, routers, switches, gateways, set-top boxes, access points). This allows latency to be much smaller, through geographical distribution, compared to traditional cloud computing. Nevertheless, cloud is still more suitable than fog for massive data processing. So even though, fog computing has been proposed to grant support for IoT applications, it does not replace the needs of cloud-based services. In fact, fog and cloud complement each other, and one cannot replace the need of the other. Together, they offer services even further optimized. Allowing enhanced capabilities for data aggregation, processing, and storage, where cloudlets are fundamental to both improving latencies and reducing network traffic to the

cloud. Moreover, Internet connectivity is not essential for the fog-based services to work, what means that services can work independently and send necessary updates to the cloud whenever the connection is available [9].

Fog computing will be crucial in a diversity of scenarios. For instance, heterogeneous sensory nodes (e.g., sensors, controllers, actuators) on a self-driving vehicle, are estimated to generate about 1GB data per second [10]. As the number of features grow, the data deluge grows out of control. Moreover, these types of systems, where people's lives depends on it, are hard real-time what means that it is absolutely imperative that all deadlines are met. Offloading tasks to fog nodes will be the best solution, once a big effort in mobility support has been done through the migration of VMs using cloudlets [11]. Also, in this context, Puliafito et al. address three types of applications where fog is required, namely, citizen's healthcare, drones for smart urban surveillance and tourists as time travellers [12], addressing the needs of low latency and mobility support. On top of that, the number of mobile devices are predicted to reach 11.6 billion by 2021, where the subset of IoT ones are expected to be 929 million [13]. With more and more devices sending data to the cloud, leads to and increasing bandwidth usage, which is a bottleneck in terms of response time. To this end, fog computing is a good solution to prevent it.

Despite the benefits that fog promises to offer such as low latency, heterogeneity, scalability and mobility, the current model suffer from some limitations that still require more efforts to overcome them.

Fog servers are supposed to be closer to end devices so its location has to be ubiquitous. Consequently, less powerful than clouds due to the high deployment cost. If many IoT devices make requests to the same fog node at the same time, it will not have enough computational and storage power to give a prompt response. So it raises the question of should a service currently running in one fog node be migrated to another one, and if yes, where? While conceptually simple, it is challenging to make these decisions in an optimal manner. Offloading tasks to the next server appears to be the solution, however, migrate the related data and processing that initially was one-hop away to the end device to a multi-hop away server, will increase network distance. Consequently, increase the response delays and bandwidth usage by the intermediate links. Besides, this decision still has to take into account the cost for both the client, with respect to the time of migration, and the provider, in terms of energy. Ignoring some of these parameters can lead to make incorrect decisions, what will both violate acceptable latency QoS constraints user's application and damage or defeat the credibility of fog computing.

Another limitation, is that fog assumes that cloudlets are always fixed. For instance, it does not foresee that a bus could have computational power. A bus could serve as a fog server and provide support for the end devices inside or outside it. The same can be applied to cars that are nowadays getting better in terms of computational power. Moreover, in a near future, unmanned aerial vehicles (UAVs) could be used for the same purpose, providing support for some social phenomena where a large number of people join in the same place as, for example, in sports games, manifestations, etc.

Summing up, we intend to tackle two of the current limitations which are, to the best of our knowledge, untreated problems in the literature. One is to provide support mobility in fog computing, not exclusively to the end devices but also to the fog nodes and the other is to achieve multi-objective fog system design.

This research will, therefore, focus on (...)

The remainder of the document is structured as follows. Section 2 xxx. Section 3 xxx.

Section 4 describes xxxx. Section 5 defines the xxx. Finally, Section 6 presents xxxx and Section 7 xxxx.

## **2 Objectives**

This work has the main objective to present

### 3 Related Work

In this section we will give some contextual information about concepts and techniques that are relevant to our work.

#### 3.1 iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments

[14] In this paper they propose a simulator, called iFogSim, to model IoT and Fog environments and measure the impact of resource management techniques in latency, network congestion, energy consumption, and cost.

#### 3.2 [123] 2018 When clones flock near the fog

Abdelwahab et al. [15] design FogMQ, a self-deploying brokering clones that discover cloud/edge hosting platforms and autonomously migrate clones between them according to self-measured weighted tail end-to-end latency without the need of a central monitoring and control unit, not having to sacrifice computation offloading gain in cloud platforms. Finally, it is simple and requires no change in existing cloud platforms controllers.

#### 3.3 [143] 2018 Virtual fog: A virtualization enabled fog computing framework for internet of things

Jianhua Li et al. [16] propose a layered Fog framework to better support IoT applications through virtualization. The virtualization is divided into object virtualization (VOs), network function virtualization and service virtualization. VOs to address the protocol inconsistency (lack of unified networking protocols that leads to exaggerated overhead); Network function virtualization maps standard networking services to VOs, thus, minimize the communication process between consumers and producers by minimizing latency, improving security and scalability; Service virtualization that composes the community and Cloud Apps from various vendors to serve local Fog users with high quality of experience (QoE) but at low cost. At last, Foglets are involved to seamless aggregate multiple independent virtual instances, Fog network infrastructures, and software platforms.

#### 3.4 [172] 2016 Vehicular fog computing: A viewpoint of vehicles as the infrastructures

[17] This paper presents the idea of utilizing vehicles as the infrastructures for communication and computation, named vehicular fog computing (VFC), which is an architecture that utilizes a collaborative multitude of end-user clients or near-user edge devices to carry out communication and computation, based on better utilization of individual communication and computational resources of each vehicle.

### **3.5 [186] 2018 Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning**

[18] Unmanned aerial vehicles (UAVs) have been considered as means to provide computing capabilities. In this model, UAVs act as fog nodes and provide computing capabilities with enhanced coverage for IoT nodes. The system aims at minimizing the total mobile energy consumption while satisfying QoS requirements of the offloaded mobile application. This architecture is based on a UAV-mounted cloudlet which provides the offloading opportunities to multiple static mobile devices. They aim to minimize the mobile energy and optimize UAV's trajectory.

### **3.6 [197] 2016 Incremental deployment and migration of geo-distributed situation awareness applications in the fog**

[19] They propose Foglets, a programming model that facilitates distributed programming across fog nodes. Foglets provides APIs for spatio-temporal data abstraction for storing and retrieving application-generated data on the local nodes. Through the Foglets API, Foglets processes are set for a certain geospatial region and Foglets manages the application components on the Fog nodes. Foglets is implemented through container-based visualization. The Foglets API takes into account QoS and load balancing when migrating persistent (stateful) data between fog nodes.

### **3.7 [217] 2017 Mobile edge cloud network design optimization**

[20] This paper presents link-path formulations supported by heuristics to compute solutions in reasonable time. Qualify the advantage in considering mobility for both users and VMs. Compare two VM mobility modes, determining that high preference should be given to live migration and bulk migration seem to be a feasible alternative on delay-stringent tiny-disk services, such as augmented reality support, and only with further relaxation on network constraints. Also, they focus on the potential medium-term planning of an edge cloud network in mobile access networks. They study two design cases: 1) network in a static state 2) network state variations in terms of load and service level, caused by user mobility.

### **3.8 [231] 2015 Dynamic service migration and workload scheduling in edge-clouds**

[21] This paper presents a model to optimize operational costs while providing rigorous performance guarantees as a sequential decision-making Markov Decision Problem (MDP). This model is different from the traditional solution methods (such as dynamic programming) that require extensive statistical knowledge and are computationally prohibitive. First they establish a decoupling property of the MDP that reduces it to two independent MDPs. Then, using the technique of Lyapunov optimization over renewals they design an online control algorithm that is provably cost-optimal.



### **3.9 [232] 2016 Segue: Quality of service aware edge cloud service migration**

[22] This paper proposes SEGUE, a service that achieves optimal migration decisions by providing a long-term optimal QoS to mobile users. This model arises to overcome the limitations of previous studies that propose a distance-based Markov Decision Process (MDP) for optimizing migration decisions that fails to consider dynamic network and server states. SEGUE is a MDP-based model which incorporates the two dominant factors in making migration decisions: 1) network state, and 2) server state.

### **3.10 [234] 2017 Optimizing service replication for mobile delay-sensitive applications in 5g edge network**

[23] Define two integer linear programming (ILP) optimization schemes to minimize QoE degradation and cost of replica deployment in service replication for MEC. They distinguish classic reactive service migration from proactive migration: reactive service migration is dependent on user movement and accommodates this movement by locating the most suitable target edge and then starting the process for migration; however, proactive service migration deploys multiple replicas of the user service to neighbouring nodes.

### **3.11 [237] 2013 Migcep: operator migration for mobility driven distributed complex event processing**

[24] This paper presents a placement and migration method for providers of infrastructures that incorporate cloud and fog resources. It ensures application-defined end-to-end latency restrictions and reduces the network utilization by planning the migration ahead of time. Furthermore, present how the application knowledge of the CEP system can be used to improve current live migration techniques for Virtual Machines (VMs) to reduce the required bandwidth during the migration.

### **3.12 [243] 2017 Follow me fog: Toward seamless handover timing schemes in a fog computing environment**

[25] The authors in observe that traditional mobile network handover mechanisms cannot handle the demands of fog computation resources and the low-latency requirements of mobile IoT applications. The authors propose Follow Me Fog framework to guarantee service continuity and reduce latency during handovers. The key idea proposed is to continuously monitor the received signal strength of the fog nodes at the mobile IoT device, and to trigger pre-migration of computation jobs before disconnecting the IoT device from the existing fog node.

### **3.13 [244] 2017 Efficient service handoff across edge servers via docker container migration**

[26] Present a novel service handoff system which seamlessly migrates offloading services to the nearest edge server, while the mobile client is moving. Service handoff is achieved via container migration. They have identified an important performance problem during Docker container migration, proposing a migration method which leverages the layered storage system to reduce file system synchronization overhead, without dependence on the distributed file system.

### **3.14 [246] 2016 Primal: Profit maximization avatar placement for mobile edge computing**

[27] By considering the gain (i.e., the end-to-end delay reduction) and the cost (i.e., the migration overheads) of the live Avatar (a software clone located in a cloudlet) migration, they propose a PProft Maximization Avatar pLacement (PRIMAL) strategy for the cloudlet network in order to optimize the trade-off between the migration gain and the migration cost by selectively migrating the Avatars to their optimal locations.

### **3.15 [247] 2017 Towards efficient edge cloud augmentation for virtual reality mmogs**

[28] Propose a hybrid gaming architecture that achieves clever work distribution. It places local view change updates on edge clouds for immediate responses, frame rendering on edge clouds for high bandwidth, and global game state updates on the center cloud for user scalability. In addition, they propose an efficient service placement algorithm based on a Markov decision process. This algorithm dynamically places a user's gaming service on edge clouds while the user moves through different access points. It also co-places multiple users to facilitate game world sharing and reduce the overall migration overhead. Also, they derive optimal solutions and devise efficient heuristic approaches and study different algorithm implementations to speed up the runtime.

### **3.16 [254] 2018 Move with me: Scalably keeping virtual objects close to users on the move**

[29] It is proposed a VO clustering and migration policy that jointly considers user proximity and inter-VO affinity to scalably support user mobility, while allowing service differentiation among users.

### **3.17 [261] 2017 Virtual machine placement for backhaul traffic minimization in fog radio access networks**

[30] Analyses the VM placement problem in fog radio access networks (F-RANs) with the objective to minimize the overall back-haul traffic. The back-haul traffic is incurred in two ways: the VM replication and data transmission to the cloud. When a user connects to a fog node and requests an application service, there is no back-haul bandwidth consumption if the fog node has the application VM. Otherwise, the VM has to be replicated on the fog node, or the request is forwarded to the cloud. They formulate the replica-based VM placement problem by considering the computing and storage of fog nodes, the user service constraint, as well as the edge bandwidth constraint.

### **3.18 [270] 2016 An adaptive cloudlet placement method for mobile applications over gps big data**

[31] Introduces the concept of movable cloudlets and explores the problem of how to cost-effectively deploy these movable cloudlets to enhance cloud services for dynamic context-aware mobile applications. To this end, the authors propose an adaptive cloudlet placement (via GPS) method for mobile applications.

### **3.19 [285] 2018 Mobility support for fog computing: An sdn approach**

[32] Study of mobility support issue in fog computing for guaranteeing service continuity. Propose a novel SDN enabled architecture that is able to facilitate mobility management in fog computing by decoupling mobility management and data forwarding functions. Design an efficient handover scheme by migrating mobility management and route optimization logic to the SDN controller. By employing link layer information, the SDN controller can pre-compute the optimal path by estimating the performance gain of each path.

### **3.20 [385] 2017 Myifogsim: A simulator for virtual machine migration in fog computing**

[11] An extension of iFogSim to support mobility through migration of VMs between cloudlets.

### **3.21 2016 Scalable Fog Computing with Service Offloading in Bus Networks**

[33] In this paper, they leverage the characteristics of buses and propose a scalable fog computing paradigm with servicing offloading in bus networks. The bus fog servers not only provide fog computing services for the mobile users on bus, but also are motivated to accomplish the computation tasks offloaded by roadside cloudlets. By this way, the computing capability of roadside cloudlets is significantly extended. They consider an allocation strategy using genetic algorithm (GA). With this strategy, the roadside cloudlets spend the least cost to offload their computation tasks. Meanwhile, the user experience of mobile users are maintained.

### **3.22 2017 Fog Computing for the Internet of Mobile Things: issues and challenges**

[12] This paper analyse what means provide mobility in Fog computing, the main challenges and three good examples where it is necessary. Also point as future work Proactive vs. reactive service migration, Exploit context information to trigger service migration, Fog federation to enable mobile roaming, Virtualization and migration techniques, Compliance with existing interoperability platforms and Integration with mobile networks towards 5G.

### **3.23 2018 Dynamic Mobile Cloudlet Clustering for Fog Computing**

[34] Fog Computing is one of the solutions for offloading the task of a mobile. However the capability of fog server is still limited due to the high deployment cost. In this paper, is proposed a dynamic mobile cloudlet cluster policy (DMCCP) to use cloudlets as a supplement for the fog server for offloading. The main idea is that by monitoring each mobile device resource amount, the DMCCP system clusters the optimal cloudlet to meet the requests of different tasks from the local mobile device.

## **4 Architecture**

XXXXX  
XXXX

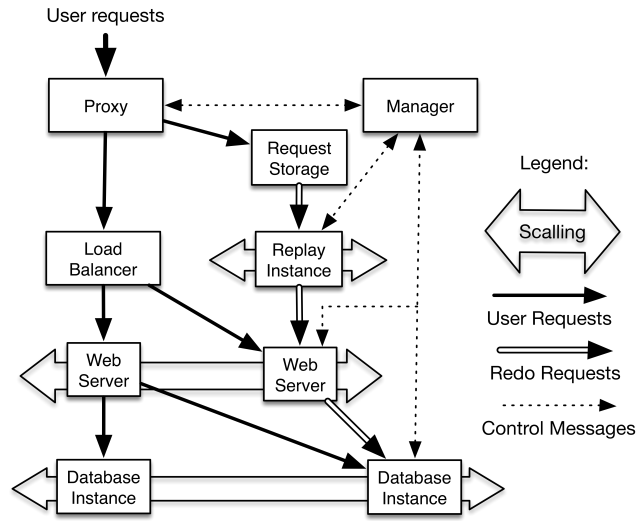


Fig. 1: Overview of the proposed service

## 5 Evaluation

The evaluation of the proposed architecture will be done xxxx

## 6 Schedule of Future Work

Future work is scheduled as follows:

- xxxx
- xxxx

## 7 Conclusion

xxxxxx

## References

1. S. R. Ellis, K. Mania, B. D. Adelstein, and M. I. Hill, “Generalizeability of latency detection in a variety of virtual environments,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 48, no. 23. SAGE Publications Sage CA: Los Angeles, CA, 2004, pp. 2632–2636.
2. B. Taylor, Y. Abe, A. Dey, M. Satyanarayanan, D. Siewiorek, and A. Smailagic, “Virtual machines for remote computing: Measuring the user experience,” *Carnegie Mellon University*, 2015.
3. T. Szigeti and C. Hattingh, *End-to-end qos network design*. Cisco press, 2005.

4. M. Satyanarayanan, "Cloudlets: at the leading edge of cloud-mobile convergence," in *Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures*. ACM, 2013, pp. 1–2.
5. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.
6. S. Davy, J. Famaey, J. Serrat-Fernandez, J. L. Gorricho, A. Miron, M. Dramitinos, P. M. Neves, S. Latré, and E. Goshen, "Challenges to support edge-as-a-service," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 132–139, 2014.
7. T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.
8. J. MSV, "Is fog computing the next big thing in internet of things?" Apr 2016. [Online]. Available: <https://www.forbes.com/sites/janakirammsv/2016/04/18/is-fog-computing-the-next-big-thing-in-internet-of-things/#6bc7b2c4608d>
9. A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *arXiv preprint arXiv:1808.05283*, 2018.
10. A. Angelica, "Google's self-driving car gathers nearly 1gb/sec," 2013. [Online]. Available: <http://www.kurzweilai.net/googles-self-driving-car-gathers-nearly-1-gbsec>
11. M. M. Lopes, W. A. Higashino, M. A. Capretz, and L. F. Bittencourt, "Myifogsim: A simulator for virtual machine migration in fog computing," in *Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*. ACM, 2017, pp. 47–52.
12. C. Puliafito, E. Mingozzi, and G. Anastasi, "Fog computing for the internet of mobile things: issues and challenges," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.
13. "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper," Mar 2017. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>
14. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.
15. S. Abdelwahab, S. Zhang, A. Greenacre, K. Ovesen, K. Bergman, and B. Hamdaoui, "When clones flock near the fog," *IEEE Internet of Things Journal*, 2018.
16. J. Li, J. Jin, D. Yuan, and H. Zhang, "Virtual fog: A virtualization enabled fog computing framework for internet of things," *IEEE Internet Things J*, vol. 5, pp. 121–131, 2018.
17. X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.
18. S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.
19. E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*. ACM, 2016, pp. 258–269.
20. A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 3, pp. 1818–1831, 2017.
21. R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Performance Evaluation*, vol. 91, pp. 205–228, 2015.
22. W. Zhang, Y. Hu, Y. Zhang, and D. Raychaudhuri, "Segue: Quality of service aware edge cloud service migration," in *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*. IEEE, 2016, pp. 344–351.

23. I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing service replication for mobile delay-sensitive applications in 5g edge network," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.
24. B. Ottenwalder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Migcep: operator migration for mobility driven distributed complex event processing," in *Proceedings of the 7th ACM international conference on Distributed event-based systems*. ACM, 2013, pp. 183–194.
25. W. Bao, D. Yuan, Z. Yang, S. Wang, W. Li, B. B. Zhou, and A. Y. Zomaya, "Follow me fog: Toward seamless handover timing schemes in a fog computing environment," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 72–78, 2017.
26. L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, p. 11.
27. X. Sun and N. Ansari, "Primal: Profit maximization avatar placement for mobile edge computing," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.
28. W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, "Towards efficient edge cloud augmentation for virtual reality mmogs," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, p. 8.
29. R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "Move with me: Scalably keeping virtual objects close to users on the move," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.
30. Y.-J. Yu, T.-C. Chiu, A.-C. Pang, M.-F. Chen, and J. Liu, "Virtual machine placement for backhaul traffic minimization in fog radio access networks," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.
31. H. Xiang, X. Xu, H. Zheng, S. Li, T. Wu, W. Dou, and S. Yu, "An adaptive cloudlet placement method for mobile applications over gps big data," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.
32. Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility support for fog computing: An sdn approach," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 53–59, 2018.
33. D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference on*. IEEE, 2016, pp. 247–251.
34. Y. Li, N. T. Anh, A. S. Nooh, K. Ra, and M. Jo, "Dynamic mobile cloudlet clustering for fog computing."