# 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

There's no doubt that the internet of things (IoT) is a great resource. It comprises *things* that have unique identities and are connected to the Internet (e.g., vehicles, home appliances, wearable devices). The number of mobile devices are predicted to reach 11.6 billion by 2021, exceeding the world's projected population at that time (7.8 billion), where the subset of IoT ones are expected to be 929 million [1].

Although this kind of device has evolved radically in the last years, battery life, computation and storage capacity remain limited. This means that they are not suitable for running heavy applications, being necessary, in this case, to resort to third parties.

Cloud computing (CC) has been imperative in expanding the reach and capabilities of IoT devices. It enables that clients outsource the allocation and management of resources (hardware or software) that they rely upon to the cloud. In addition, to avoid over- or underprovisioning, cloud service providers also afford dynamic resources for a scalable workload, applying a pay-as-you-go cost model. This way, besides overcoming the aforementioned limitations, it also brings other advantages such as availability, flexibility, scalability, reliability, to mention a few.

Despite the benefits of using cloud computing, two main problems, linked to IoT applications, remain unresolved and they can not be underestimated. The first, and the most obvious, is the fact that cloud servers reside in remote data centers and, consequently, the end-to-end communication have long delays (characteristic of multi-hops transmissions over the Internet). Some applications, with ultra-low latency requirements, can't support such delays. Augmented reality applications that use head-tracked systems, for example, require end-to-end latencies to be less than 16 ms [2]. Cloud-based virtual desktop applications require end-to-end latency below 60 ms if they are to match QoS of local execution [3]. Remotely rendered video conference, on the other hand, demand end-to-end latency below 150 ms [4]. The other problem is related with the constantly growing number of IoT devices. In a sense-process-actuate model, where the processing is done in the cloud, core network traffic (i.e. bandwidth usage) grows depending on the number of IoT devices.

To overcome this limitations, the solution that has already been proposed is to bring the cloud closer to the end devices, where entities such as base-stations would host smaller sized clouds. This idea has been variously termed as Cloudlets [5], Fog Computing [6], Edge Computing [7], and Follow Me Cloud [8], to name a few.

Fog computing is a new computing architecture that 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. Fog nodes, also known as fog servers or cloudlets (smaller sized clouds with lower computational capacity), are virtual-machine (VM) based, which means that they promote flexibility and mobility. They can be placed close to IoT source nodes, due to low hardware footprint and low power consumption. This allows latency to be much smaller, through geographical distribution, compared to traditional cloud computing and still cut off a significant amount of core network traffic. Nevertheless, cloud is still more suitable than fog for massive data processing, when the latency constraints are not so tight. 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 to IoT applications. 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].

### 1.1 Motivation

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 efforts to overcome them.

There is lack of support for mobile fog computing. Most of the existing literature assumes that the fog nodes are fixed, or only considers the mobility of IoT devices [9]. Less attention has been paid to mobile fog computing and how it can improve the QoS, cost, and energy consumption. For instance, it does not foresee that a bus could have computational power; as a cloudlet, it cloud provide offloading support to elements (i.e. IoT devices and other cloudlets) inside and outside it. The same could be applied to cars that are nowadays getting increasingly better in terms of computational power. Both would be extremely useful in order to enhance the resources and capabilities of fog computing, for example, in large urban areas where traffic congestion is common or when they are parked. Apart from QoS enhancement, this would reduce the implementation costs since it would no longer require such computational power in the roadside cloudlets. Also, the costs in terms of latency to the client would be minimized. On top of that, it would minimize the energy consumption of mobile devices once the access points where they are connected may be even closer.

Another limitation of fog computing is to take into account few parameters in the decision making of migration. Most of the existing schemes that are proposed for fog systems, such as offloading, load balancing, or service provisioning, only consider few objectives (e.g., QoS, cost) and assume other objectives do not affect the problem [9]. Fog servers are less powerful than clouds due to the high deployment cost. If many requests are made 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: 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 seems to be the solution, however, migrate the VM that was initially one-hop away from the IoT device to a multi-hop away server, will increase the network distance. Consequently, raises the end-to-end latency and the bandwidth usage by the intermediate links. Besides, this decision still has to take into account the cost for both the client (e.g., migration time, computational delay) and the provider (e.g., computing and migration energy). Ignoring some of these parameters can lead to wrong decisions, what will both violate latency constraints of user's application and damage or defeat the credibility of fog computing.

### 1.2 Objectives

Summing up, this work 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 mobility support in fog computing environments, not exclusively to the end devices but also to the fog nodes and the other is to achieve multi-objective fog system design. These objectives shall be implemented in a toolkit allowing the simulation of resource management techniques in IoT, and mobile fog computing environments. In order to achieve the aforementioned goals, this work must follow the sequence of tasks presented below.

- Start with analysis of current mobility approaches publicly available with respect to IoT nodes or cloudlets;
- Propose mobile fog computing, where fog nodes can move, provisioning a method to keep the service always-available for IoT nodes;

- Analyze the most suitable optimization algorithms for fog computing and select the most appropriate ones;
- Design of mobility-aware task offloading when fog nodes are mobile, taking into account the related costs to the client (i.e. migration, communication and processing delay) and discovery;
- Propose a variation to achieve multi-objective (i.e. QoS, QoE, cost, handover, load balancing, energy, bandwidth and VMs or virtual objects migration);
- Study the current open source simulators for fog computing environments;
- Implement the proposed algorithms in the simulator and compare them.

The remainder of the document is structured as follows. Section 2 xxx. Section 3 describes xxxx. Section 4 defines the xxx. Finally, Section 5 presents xxxx and Section 6 xxxx.

# 2 Related Work

The solution proposed in this document leverages knowledge obtained from studying several concepts and systems from the current state-of-the-art. In this section an overview of those concepts and systems will be given, stating for each of them their advantages and disadvantages.
This section is structured as follows. Section 2.1 presents different methods to push intelligence and computing power closer to the source of the data and why this work adopted fog computing for this purpose. Section 2.2 presents mobility-aware systems xxx. Section 2.3 xxx. 2.4 xxx. 2.5 xxx. Section 2.6 xxx.

## 2.1 Computing paradigms

In what concerns about standardizing fog computing, there is a lack of unanimity. As aforementioned, fog has been variously termed as cloudlets, edge computing, etc. Different research teams are proposing many independent definitions of fog (and fog-related computing paradigms). As there is a research gap in the definitions and standards for fog computing, this work follows the definitions that Ashkan Yousefpour et al. [9] proposed. Below, are described some of the paradigms that were raised in order to bring cloud closer to the end devices, as well as their pros and cons. As a conclusion we show why fog computing is the natural platform for IoT.

### 2.1.1 Mobile computing

Mobile/nomadic computing (MC) is characterized by the processing being performed by mobile devices (e.g., laptops, tablets, or mobile phones). It raises to overcome the inherent limitations of environments where connectivity is sparse or intermittent and where there is low computing power. As this model only uses mobile devices to provide services to clients, there is no need for extra hardware, once they already have communication modules such as Bluetooth, WiFi, ZigBee, etc. As already mentioned, mobile devices have evolved in recent years, however their resources are more restricted, compared to fog and cloud. This computing paradigm has the advantage of being characterized by a distributed architecture, once mobile machines do not need a centralized location to operate. The disadvantages of MC are mainly due to their hardware nature (i.e. low resources, balancing between autonomy and the dependency of other mobile devices; characteristic that prevails in all distributed architectures) and the need of mobile clients to efficiently adapt to changing environments [10]. MC alone may not be able to meet the requirements of some applications. On the one hand it is limited due to autonomy constraints and in the order hand low computational power and low storage capacity further restricts the applications where this paradigm is feasible. For instance it is unsuitable for applications that require low-latency or that generates large amounts of data that needs to be stored or processed. Nonetheless, MC can use both fog and cloud computing to enhance its capacities, not being restricted to a local network; expanding the scope of mobile computing and the number of applications where it can be used.

### 2.1.2 Mobile Cloud Computing

Cloud and fog computing, as mentioned in 2.1.1, are key elements for validate the importance of MC. This interaction between them results in a new paradigm, called mobile cloud computing (MCC). MCC, differs from MC in a sence that mobile applications can be partitioned at runtime so that computationally intensive components of the application can be handled through adaptive offloading [11], from mobile devices to the cloud. This characteristic increases the autonomy of mobile devices (i.e battery lifetime), as both the data storage and data processing occur outside of the mobile device. Also, it enables a much broader range of mobile subscribers, rather then the previous laptops, tablets, or mobile phones. Opposed to resource-constrained in MC, MCC has high availability of computing resources, scaling the type of applications where it is possible to use, for instance, augmented reality applications. Unlike MC, MCC relies on cloud-based services, where its access is done through the network core by WAN connection, which means that applications running on these platforms require connection to the Internet all the time. On the one hand, both MCC an MC suffer from the intrinsic characteristics of mobility, such as frequent variations of network conditions, intensified under rapid mobility patterns. On the other hand, in MCC, even if the mobile devices remain fixed, it suffers from the inherent disadvantage of using cloud-based services (i.e. communication latency), which makes it unsuitable for some delay-sensitive applications.

### 2.1.3 Mobile ad hoc Cloud Computing

In some scenarios there exists lack of infrastructure or a centralized cloud, so implement a network based on MCC may not always be suitable. To overcome dependence on an infrastructure, raises mobile ad hoc cloud computing (MACC). It consists on a set of mobile nodes that form a dynamic and temporary network through routing and transport protocols. These nodes are composed by mobile ad hoc devices which may continuously join or leave the network. In order to counteract the, aforementioned, characteristics inherent to this type of networks, and unlike MC, a set of ad hoc devices may form a local cloud that can be used in the network for purposes of storage and computation. As mobile ad hoc networks (MANET), it can play a big part in use cases such as disaster recovery, car-to-car communication, factory floor automation, unmanned vehicular systems, etc. Although it does not rely in external cloud-based services as MCC does, which mitigates the latency problem, it shares some of the limitations inherent to MC and ad hoc networks such as the power consumption constraints. Moreover, the formed local cloud may still be computationally weak and as both network and cloud are dynamic it is more challenging to achieve an optimal performance. For instance, as there is no infrastructure, mobile devices are also responsible for routing traffic among themselves.

### 2.1.4 Edge Computing

Edge computing (EC), enhances their capabilities (i.e. management, storage, and processing power) by connected devices at the edge of the network. Edge computing is located in the local IoT network, being ideally at one hop away from the Iot device and at most located few hop away. Open Edge Computing defines edge computing as computation paradigm that provides small data centers (edge nodes) in close proximity to the users, enabling a dramatic improvement in customer experience through low latency interaction with compute and storage resources just one hop away from the user [12]. As in EC the connected devices do not have to wait for a centralized platform to provide a requested service, nor are so limited in terms of resources as in the traditional MC, their service availability is relatively high. Nonetheless EC has some drawbacks. As latency, in this context, is composed by three components: data sending time, processing time and result receiving time, even though the communication latency is negligible, processing time may not be. This computing paradigm only uses edge devices, and their computation and storage power may be poor (e.g., routers, switches). Consequently, this processing latency may still be too high for some applications with low latency requirements.

OpenFog Consortium states that fog computing is often erroneously called edge computing, but there are key differences [13]. Although they have similar concepts, edge computing tends to be limited to the edge devices (i.e. located in the IoT node network), excluding the cloud from its architecture. Whereas, fog computing is hierarchical and unlike EC, it is not limited a local network, but instead it provides services anywhere from cloud to things. It is worth noting that the term edge used by the telecommunications industry usually refers to 4G/5G base stations, radio access networks (RANs), and internet service provider (ISP) access/edge networks. Yet, the term edge that is recently used in the IoT landscape refers to the local network where sensors and IoT devices are located [9].

### 2.1.5 Multi-access Edge Computing

Analogously, MCC is an extension of MC through cloud computing, as multi-access edge computing (MEC) is an extension of MC through EC. ETSI defines MEC as computation paradigm that offers application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network. This environment is characterized by ultra-low latency and high bandwidth as well as real-time access to radio network, information that can be leveraged by applications [14]. In MEC, operators can open their RAN edge to authorized third parties, allowing them to deploy applications and services towards mobile subscribers through 4G/5G base stations. The first approach to the edge of a network meant the edge of a mobile network, hence the name mobile edge computing. As MEC research progressed, was noticed that the term leaves out several access points that may also construct the edge of a network. Thus, prompted the change from mobile edge computing to multi-access computing in order to reflect that the edge is not solely based on mobile networks [15]. Now it include a broader range of applications beyond mobile device-specific tasks, such as video analytics, connected vehicles, health monitoring, augmented reality, etc.

MEC supports low-latency applications once it benefits from real-time radio and network information. Similar to edge computing, MEC can operate with little to no Internet connectivity and use small-scale data centers with virtualization capacity to provide services. Unlike EC, MEC establishes connectivity through a WAN, WiFi, and cellular connections, whereas edge computing generally can establish any form of connectivity (e.g., LAN, WiFi, cellular). MCC focuses on the relationship between cloud service users (on mobile devices) and cloud service providers, whereas research in MEC focuses on (RAN-based) network infrastructure providers. MEC allows edge computing to be accessible to a wide range of mobile devices with reduced latency and more efficient mobile core networks [16].SDN allows for virtual networking devices to be easily managed through software APIs [17], and NFV allows for reduced deployment times for networking services through virtualized infrastructure. MEC is expected to benefit significantly from the up-and-coming 5G platform as it allows for lower latency and higher bandwidth among mobile devices, and it supports a wide range of mobile devices with finer granularity.

### 2.1.6 Cloudlet Computing

Cloudlet computing is another direction in mobile computing that aims to bring cloud closer to end devices through the use of cloudlets. M. Satyanarayanan et al. states that a cloudlet is a trusted, resource-rich computer or cluster of computers that's well-connected to the Internet and available for use by nearby mobile devices [18]. Cloudlet is, as the name suggests, a smaller sized clouds with lower computational capacity. It can be seen as a "data center in a box", where mobile users can exploit their virtual machines (VM) to rapidly instantiate customized-service software in a thin client fashion. This way, it is possible to offload computation from mobile devices to VM-based cloudlets located on the network edge. Through those VMs, cloudlets are capable of providing resources to end devices in real time over a WLAN network. The relatively low hardware footprint, results in moderate computing resources, but lower latency, energy consumption and higher bandwidth compared to cloud computing. The characteristics of this computing paradigm make possible to handle applications with low-latency requirements, supporting real-time IoT applications. Y Jararweh et al. [19] propose an architecture mobile-cloudlet-cloud, where they present three reasons which indicate that even though cloudlets are computationally powerful, they still need a connection to the cloud and its services: (1) Heavy non real time jobs might process in the enterprise cloud whiles the real time ones processed by the cloudlet, (2) Accessing a file stored in the Enterprise Cloud, (3) Accessing some services that are not available inside the Cloudlet. Although cloudlet computing fits well with the mobile-cloudlet-cloud architecture, fog computing offers a more generic alternative that natively supports large amounts of traffic, and allows resources to be anywhere along the thing-to-cloud continuum.

### 2.1.7 Mist Computing

Mist computing emerges to push IoT analytics to the extreme edge. This computing paradigm is an even more dispersed version of fog. That means locating analytics tools not just in the core and edge, but at the "extreme edge" - actually in many of the devices (the IoT devices themselves; the sensor and actuators) [20]. It has been proposed with future self-aware and autonomic systems in mind [21]. Therefore, mist computing can be seen as the first computing location in the IoT-fog-cloud continuum. Mist computing extends compute, storage, and networking across the fog through the things. This decreases latency and increases subsystems' autonomy. The challenge with implementing mist computing systems lies in the complexity and interactions of the resulting network, which must be managed by the devices themselves as central management of such systems is not feasible.

### 2.1.8 Fog computing

Fog computing bridges the gap between the cloud and IoT devices by enabling computing, storage, networking, and data management on the network nodes within the close vicinity of IoT devices. OpenFog Consortium [22] defines fog computing as "a horizontal system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum." The decentralized nature of fog computing allows devices to either serve as fog computing nodes themselves (e.g. a car acts as a fog node for on-board sensors) or use fog resources as the clients of the fog (end device). Compared to the cloud, fog computing offers moderate computing capacity and low power consumption. Clouds are composed by large datacenters whereas fog uses small servers, routers, switches, gateways, set-top boxes or access-points (AP). Fog has lower hardware footprint, so it can be closer to the end devices. Fog computing can be accessed through connected devices from the edge of the network to the network core, whereas cloud computing must be accessed through the network core. Moreover, continuous Internet connectivity is not essential for the fog-based services to work. That is, the services can work independently with low or no Internet connectivity and send necessary updates to the cloud whenever the connection is available. Cloud computing, on the other hand, requires devices to be connected when the cloud service is in progress. There are clear differences and tradeoffs between cloud and fog computing, and one might ask which one to choose. However, fog and cloud complement each other; one cannot replace the need of the other. By coupling cloud and fog computing, the services that connected devices use can be optimized even further.

### 2.1.9 Other Similar Computing Paradigms

- **Micro Data Center** (MDC) - Cloudlet são por vezes referidas como MDC. Um MDC pode ser um edge node ou uma cloudlet que é implementada entre os dispositivos IoT e a cloud.
- **Cloud of Things** (CoT) - Parecido a mist computing, no entanto em mist a computação é feita nos dispositivos IoT e não necessariamente numa cloud de pooled resources. Em CoT, a computação é feita sobre a cloud que é formada por pooled resources de IoT devices.
- **Edge Cloud** - Quando falamos de cloud computing, falamos de ”core” ou ”distant” clouds que estão distantes do end users que são responsáveis pela computação ”pesada”. Ao contrário, ”edge” clouds são mais pequenas e estão mais próximas. Edge cloud é parecido a edge computing. Edge cloud extende capacidades da cloud na edge aproveitando os compute nodes (user ou operator-contributed) na edge da rede. À semelhança do fog, em edge clouds a habilidade de correr uma aplicação numa forma coordenada tanto na edge com na cloud é prevista. Edge clouds são nós na edge como micro data centers, cloudlets, e MEC.

### 2.1.10 Concluding Remarks

As the above mentioned, there are some other similar computing paradigms such as follow me cloud (FMC), follow me edge (FME), Follow Me edge-Cloud (FMeC), etc. However, this state-of-the-art had as objective to investigate the most treated ones in the literature. The purpose was to understand their characteristics and identify where to tackle the current limitations of delay sensitive IoT applications and consequently their deployment.

The previous discussion about fog computing and related paradigms demonstrate the importance of understanding the characteristics of these platforms in the changing IT landscape. As demonstrated by the strength and weaknesses attributed to these computing paradigms, some paradigms may be better suited for a particular use case than others. Even so, fog computing is suited for a large number of use cases in the current landscape of IoT and connected devices. The versatility of fog computing makes it suitable for many cases of data-driven computing and low-latency applications, even though it may not be suitable for a few extreme applications, such as disaster zones or sparse network topologies where ad hoc computing (e.g., MACC) or extreme edge clouds (e.g., mist, CoT) may be a better fit. Nonetheless, fog computing is considered a more general form of computing when compared to other similar paradigms (e.g., EC, MEC, cloudlet), because of its comprehensive definition scope, generality, and extensive presence along the thing-to-cloud continuum. Tables 2 and 3 summarize these characteristics. Fog computing offers a bright future for an open-standards environment of connected devices, as it is evident by IEEE Standard's adoption of OpenFog Reference Architecture [69]. There does not yet exist a globally unanimous distinction between fog computing and related computing paradigms, such as edge computing, mist computing, and cloudlets across researchers and industries, as shown in the previous sections of this paper. We attempt in this survey paper to clarify the distinctions between fog computing and the related computing paradigms. A comparison of the underlying infrastructure of fog computing and its related computing paradigms from the networking perspective is shown in Fig. 7. In the rest of this paper, we will mainly survey and discuss the recent literature on fog computing, but mention the studies on other related computing paradigms that could be easily extended or directly applied in fog.

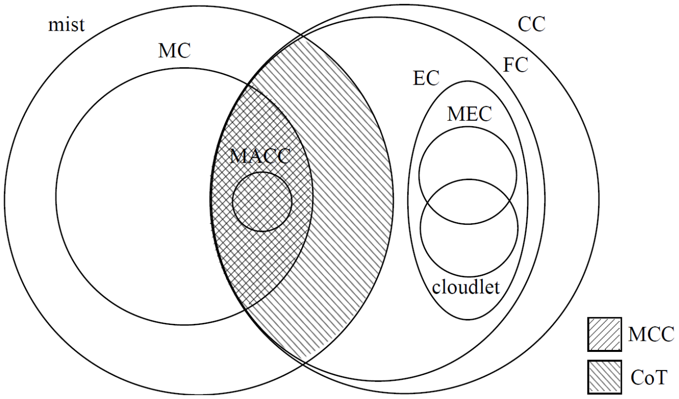

Fig. 1: A classification of scope of fog computing and its related computing paradigms.

### 2.2 Mobile Fog Computing

Mobile Fog Computing is the ability to provide mobility support for both IoT and Fog nodes. This means that users holding end devices, can outsource the allocation and management of resources to the fog infrastructure, where both iot and fog nodes are able to move, ensuring that QoS requirements of the IoT applications are met.
Several studies were already done in order to provide mobile support for IoT devices, however, the purpose of this study is to support mobile Fog Computing, once in smart cities, fog nodes cloud be anything from things to the cloud.

This distributed middle tier, in a 3-tier architecture, things-fog-cloud, can use as fog nodes any physical device that has facilities or infrastructures that can provide resources and visualization capabilities. This, may include movable fog nodes, such as cars, buses,

unmanned aerial vehicles (UAVs), etc.
In this field there are already some early efforts

falar dos vários exemplos que o stor gosou. no final referir que a cloud utiliza VMs para .... (wikipedia), e que fog para além disto utiliza as VMs para sportar a mobilidade. Falar das possibilidades VM num todo ou floglets...........

## 2.3 Data placement

## 2.4 Migration Optimization

## 2.5 Multi-objective

QoS, QoE, Cost, Energy, Handover, Mobility, Bandwidth

## 2.6 Toolkits

## 3 Architecture

xxxxx

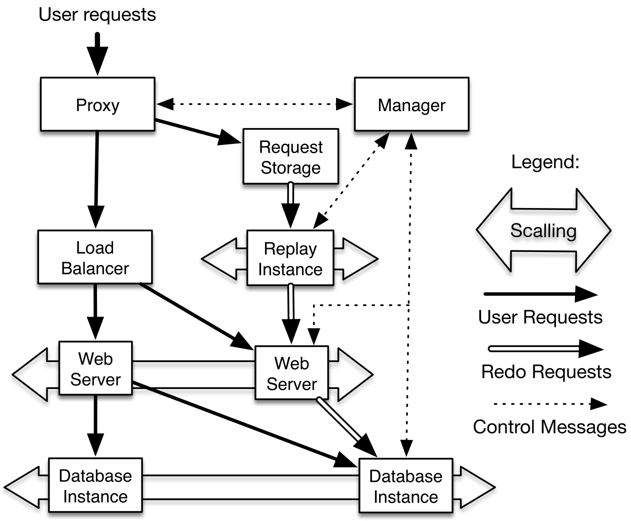

Fig. 2: Overview of the proposed service

xxxx

## 4 Evaluation

The evaluation of the proposed architecture will be done xxxx

## 5 Schedule of Future Work

Future work is scheduled as follows:

- xxxx
- xxxx

## 6 Conclusion

xxxxx

## References

1. "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper - cisco," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper- c11-520862.html, (Accessed on 10/25/2018).
2. 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.
3. 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.
4. T. Szigeti and C. Hattingh, *End-to-end qos network design.* Cisco press, 2005.
5. 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.
6. 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.
7. 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.
8. 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.
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. M. Satyanarayanan, "Fundamental challenges in mobile computing," in *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing.* Acm, 1996, pp. 1–7.
11. M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.
12. "Open edge computing," http://openedgecomputing.org/about.html, (Accessed on 10/24/2018).
13. P. by the OpenFog, C. Architecture, and W. Group, "Openfog reference architecture for fog computing," 0208.
14. "Etsi - multi-access edge computing," https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing, (Accessed on 10/25/2018).
15. "Mobile edge computing vs. multi-access edge computing," https://www.sdxcentral.com/mec/definitions/mobile-edge-computing-vs-multi-access-edge-computing/, (Accessed on 10/25/2018).
16. T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
17. K. P. Kadiyala and J. A. Cobb, "Inter-as traffic engineering with sdn," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on.* IEEE, 2017, pp. 1–7.
18. M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, 2009.
19. Y. Jararweh, L. Tawalbeh, F. Ababneh, and F. Dosari, "Resource efficient mobile computing using cloudlet infrastructure," in *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on.* IEEE, 2013, pp. 373–377.

20. “Cisco pushes iot analytics to the extreme edge with mist computing - rethink,” https://rethinkresearch.biz/articles/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2/, (Accessed on 10/25/2018).
21. J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, “The benefits of self-awareness and attention in fog and mist computing,” *Computer*, vol. 48, no. 7, pp. 37–45, 2015.
22. “What is fog computing? : Openfog consortium,” https://www.openfogconsortium.org/page-section/definition-of-fog-computing/, (Accessed on 10/24/2018).