

# A Survey of Fog Computing: Concepts, Applications and Issues

Shanhe Yi, Cheng Li, Qun Li  
Department of Computer Science  
College of William and Mary  
Williamsburg, VA, USA  
{syi,cli04,liqu}@cs.wm.edu

## ABSTRACT

Despite the increasing usage of cloud computing, there are still issues unsolved due to the inherent problem of cloud computing such as unreliable latency, lack of mobility support and location-awareness. Fog computing, also termed edge computing, can address those problems by providing elastic resources and services to end users at the edge of network, while cloud computing are more about providing resources distributed in the core network. This survey discusses the definition of fog computing and similar concepts, introduces representative application scenarios, and identifies various aspects of issues we may encounter when designing and implementing fog computing systems. It also highlights some opportunities and challenges, as direction of potential future work, in related techniques that need to be considered in the context of fog computing.

## Categories and Subject Descriptors

A.1 [General Literature]: Introduction and Survey; C.2.4 [Computer-Communication Networks]: Distributed Systems—*Cloud Computing*

## General Terms

Definition, Application, Performance, Design, Management

## Keywords

fog computing; edge computing; mobile cloud computing; mobile edge computing; cloud computing; review

## 1. INTRODUCTION

We are embracing the prevalence of ubiquitously connected smart devices, which are now becoming the main factor of computing. Along with the development of wearable computing, smart metering, smart home/city, connected vehicles and large-scale wireless sensor network, the Internet of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*Mobidata'15*, June 21, 2015, Hangzhou, China.  
Copyright © 2015 ACM 978-1-4503-3524-9/15/06 ...\$15.00.  
DOI: <http://dx.doi.org/10.1145/2757384.2757397>.

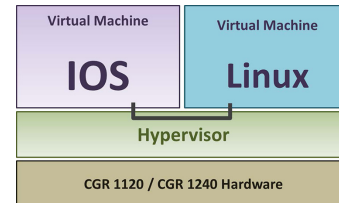


Figure 1: IOx architecture [7]

Things (IoT) has received attentions for years and is considered as the future of Internet. IDC (International Data Corporation) has predicted that in the year of 2015, “The IoT will continue to rapidly expand the traditional IT industry” up 14% from 2014 [14]. However, due to the limited computation/storage on smart devices, cloud computing is considered as a promising computing paradigm, which can provide elastic resources to applications on those devices. In spite of attempts of augmenting IoT applications with the power of cloud, there are still problems unsolved in that IoT applications usually require mobility support, geo-distribution, location-awareness and low latency.

*Fog computing* is proposed to enable computing directly at the edge of the network, which can deliver new applications and services especially for the future of Internet [3]. For example, commercial edge routers are advertising processor speed, number of cores and built-in network storage. Those routers have the potential to become new servers. In fog computing, facilities or infrastructures that can provide resources for services at the edge of the network are called *fog nodes*. They can be resource-poor devices such as set-top-boxes, access points, routers [52], switches, base stations, and end devices, or resource-rich machines such as Cloudlet and IOx. Cloudlet is a resource-rich computer like “cloud in a box”, which is available for use by nearby mobile devices. Satyanarayanan et al. [41] build Cloudlet, which is ahead of fog computing but coincides the concept of fog computing. IOx is a fog device product from Cisco, whose architecture is shown in Figure 1. “IOx works by hosting applications in a Guest Operating System (GOS) running in a hypervisor directly on the Connected Grid Router (CGR)” [7]. On IOx platform, developers can run python scripts, compile their own code, and even replace the operation system with their own.

This paper presents a survey on fog computing focusing on its concepts, applications and underlying issues one may encounter in designing and implementing fog computing system.

## 2. OVERVIEW OF FOG COMPUTING

In this section, we will first review the definition of fog computing and some similar concepts, and then present the characterization of fog computing.

### 2.1 Definition of Fog Computing

In the perspective of Cisco, fog computing is considered as an extension of the cloud computing paradigm from the core of network to the edge of the network. It is a highly virtualized platform that provides computation, storage, and networking services between end devices and traditional cloud servers [3]. While in the flavor of work [47], fog computing is defined as “*a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralised devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so.*” Although this definition is still debatable, we strongly agree that we need a definition to differ fog computing from related technologies since anyone of those underlying techniques may offer us a false view on fog computing.

**Similar Concepts** There are similar concepts such as mobile cloud computing (MCC) and mobile-edge computing (MEC) which have overlap with fog computing. MCC refers to an infrastructure in which both the data storage and the data processing happen outside of the mobile devices. Mobile cloud applications move the computing power and data storage from mobile phones to the cloud, providing applications and mobile computing to not only smartphone users but also a much broader range of mobile subscriber [9]. MEC can be seen as a cloud server running at the edge of a mobile network and performing specific tasks that could not be accomplished with traditional network infrastructure [13]. Fog computing seems like the combination of MCC and MEC, while it distinguishes itself as a more promising and well generalized computing paradigm in the context of Internet of Things.

### 2.2 Applications Scenarios

F. Bonomi, et al. [3] highlight three scenarios in connected vehicle, smart grid and wireless sensor and actuator networks (WSAN). I. Stojmenovic, et al [44, 45] elaborate previous scenarios and further expand this concept on smart building and software-defined networking (SDN). We will discuss three driving scenarios that will benefit from concept of fog computing.

**Augmented Reality (AR) and Real-time video analytics** Augment reality applications are popular on smartphone, tablet and smart glasses by overlaying an informative view on the real world (viewed on the device display system). Recent popular products or projects include Google Glass, Sony SmartEyeglass and Microsoft HoloLens. AR applications usually need high computation power to process video streaming and high bandwidth for data transmission. For example, a normal AR application needs to process real time video frame using computer vision algorithm and at the same time process other inputs such as voice, sensor and finally output timely informational content on displays. However, human are very sensitive to delays in a series of

consecutive interactions. A processing delay of more than tens of milliseconds will ruin the user experience and leads to negative user feedback. AR system supported by fog computing can maximize throughput and reduce latency in both processing and transmission. K. Ha, et al. [16] design and implement a wearable cognitive assistance spanning on Google Glass and Cloudlet, which can offer the wearer hints for social interaction via real-time scene analysis. The system achieves tight end-to-end latency constraint by offloading computation-intensive task to nearby Cloudlet. Network failure and unavailability of distant Cloudlets are also considered and provided automatic degrade services.

Largely-deployed camera sensors in city or along the road are important component of smart city and smart connected vehicle to support surveillance, traffic management etc. Fog computing can provide sufficient resource of computation and storage to store captured video streams, transcode and process video frame for tasks such as object recognition, object tracking and data mining etc. After that we can just send out notification, events, description or video summary to end users, central servers or databases. With the help of fog, we can achieve real-time processing and feedback of high-volume video streaming and scalability of service on low-bandwidth output data. Privacy-preserving techniques can also be applied at the fog side, to ease the concern of personal privacy leakage in public surveillance systems.

**Content Delivery and Caching** Traditional web content delivery technologies can not adapt to the requests from user after the web performance is optimized at server side. However some knowledge can only be known at the client side or near the client’s network such as local network conditions or traffic statistics, which can be leveraged to optimize the web performance. J. Zhu, et al. consider web optimization from this new perspective in the context of fog computing [58]. The fog server can provide dynamic customizable optimization based on client devices and local network conditions. And since fog server is in client’s vicinity, it can gather client side knowledge and user experience, to optimize the rendering of web page. Similarly, caching technique can be better implemented within the fog nodes to further save the bandwidth and reduce latency for content delivery.

**Mobile Big Data Analytics** Big data processing is a hot topic for big data architecture in the cloud and mobile cloud [55, 38]. Fog computing can provide elastic resources to large scale data process system without suffering from the drawback of cloud, high latency. In cloud computing paradigm, event or data will be transmitted to the data center inside core network and result will be sent back to end user after a series of processing. A federation of fog and cloud can handle the big data acquisition, aggregation and preprocessing, reducing the data transportation and storage, balancing computation power on data processing. For example, in a large scale environment monitoring system, local and regional data can be aggregated and mined at fog nodes providing timely feedback especially for emergency case such as toxic pollution alert. While detailed and thorough analysis as computational-intensive tasks can be scheduled in the cloud side. We believe data processing in the fog will be the key technique to tackle analytics on large scale of data generated by applications of IoT.

## 3. ISSUES

In this section, we will identify and discuss potential issues in the context of fog computing. Some of them would be the direction of future work.

### 3.1 Fog networking

Due to located at the edge of Internet, fog network is heterogeneous. The duty of fog network is to connect every component of the fog. However, managing such a network, maintaining connectivity and providing services upon that, especially in the scenarios of the Internet of Things (IoT) at large scale, is not easy. Emerging techniques, such as software-defined networking (SDN) and network function virtualization (NFV), are proposed to create flexible and easy maintaining network environment. The employment of SDN and NFV can ease the implementation and management, increase network scalability and reduce costs, in many aspects of fog computing, such as resource allocation, VM migration, traffic monitoring, application-aware control and programmable interfaces.

**SDN** “*When SDN concept is implemented with physically (not just logically) centralized control, it resembles the fog computing concepts, with fog device acting as the centralized controller.*” [44]. In the fog, each node should be able to act as a router for nearby nodes and resilient to node mobility and churn, which means controller can also be put on the end nodes in fog network. The challenges of integrating SDN into fog network is to accommodate dynamic conditions as mobility and unreliable wireless link. The closest work [27] proposes several designs for SDN-based mobile cloud architectures for mobile/vehicular ad hoc network (MANET/VANET), and shows the feasibility by achieving high packet delivery ratio with acceptable overhead. The proposed SDN-based frequency selection architecture adapts the changes from wired ports to heterogeneous wireless interfaces, to support applications such as wireless network virtualization, privilege traffic reservation, and frequency hopping communication. There are other interesting questions like how to deal with node churn, updating, predicting and maintaining the connectivity graph of network in different granularity; how to cooperate different controllers such as constantly connected controller (at the edge infrastructures) or intermittently connected controller (at the end devices) and where to place controllers in fog network [21]; how to design distributed SDN system that meet the harsh requirement of fog computing such as latency, scalability and mobility.

**NFV** NFV replaces the network functions with virtual machine instances. Since the key enabler of fog computing is virtualization and those VMs can be dynamically created, destroyed and offloaded, NFV will benefit fog computing in many aspects by virtualizing gateways, switches, load balancers, firewalls and intrusion detection devices and placing those instances on fog nodes. NFV, however, is not studied in the context of fog computing yet. In cellular core network, work [2] proposes function placement problem of virtualized gateway and SDN-decomposed gateway, minimizing the network overhead against constraints in data-plane latency, data center utilization and control-plane overhead. Several NFV researches are about how to design high-performance, virtualized software middlebox platform [24, 32]. For NFV in fog computing, the performance of virtualized network appliances is still the first concern [17]. This problem has two aspects: one is the throughput or latency of virtualized net-

work appliances (middlebox) in fog network, and the other is how to achieve efficient instantiation, placement and migration of virtual appliances in a dynamic network, together to meet low latency and high throughput requirements.

### 3.2 Quality of Service (QoS)

QoS is an important metric for fog service and can be divided into four aspects, 1) connectivity, 2) reliability, 3) capacity, and 4) delay.

**Connectivity** In a heterogeneous fog network, network relaying, partitioning and clustering provide new opportunities for reducing cost, trimming data and expanding connectivity. For example, an ad-hoc wireless sensor network can be partitioned into several clusters due to the coverage of rich-resource fog nodes (cloudlet, sink node, powerful smartphone, etc.). Work [53] proposes an online AP association strategy that not only achieves a minimal throughput, but efficiency in computational overhead. Similarly, the selection of fog node from end user will heavily impact the performance. We can dynamically select a subset of fog nodes as relay nodes for optimization goals of maximal availability of fog services for a certain area or a single user, with constraints such as delay, throughput, connectivity, and energy consumption.

**Reliability** Madsen et al. review the reliability requirement of clustering computing, grid computing, cloud and sensor network towards a discussion of reliability of fog computing [30]. Normally, reliability can be improved through periodical check-pointing to resume after failure, rescheduling of failed tasks or replication to exploit executing in parallel. But checkpointing and rescheduling may not suit the highly dynamic fog computing environment since there will be latency, and cannot adapt to changes. Replication seems more promising but it relies on multiple fog nodes to work together.

**Capacity** Capacity has two folds: 1) network bandwidth, 2) storage capacity. In order to achieve high bandwidth and efficient storage utilization, it is important to investigate how data are placed in fog network since data locality for computation is very important. There are similar works in the context of cloud [1], and sensor network [42]. However, this problem faces new challenges in fog computing. For example, a fog node may need to compute on data that is distributed in several nearby nodes. The computation cannot start before the finish of data aggregation, which definitely adds delay to services. To solve this, we may leverage user mobility pattern and service request pattern to place data on suitable fog nodes to either minimize the cost of operation, the latency or to maximize the throughput. Data placement in federation of fog and cloud also needs critical thinking. The challenges come from how to design interplay between fog and cloud to accommodate different workloads. Due to the dynamic data placement and large overall capacity volume in fog computing, we may also need to redesign search engine which can process search query of content scattered in fog nodes [49, 50]. It is also very interesting to redesign cache on fog node to exploit temporal locality and broader coverage to save network bandwidth and reduce delay, while there is existing work of cache on end device [57] and cache on edge router [51].

**Delay** Latency-sensitive applications, such as streaming mining or complex event processing, are typical applications which need fog computing to provide real-time stream-

ing processing rather than batch processing. K. Hong, et al. [23] propose a fog-based opportunistic spatio-temporal event processing system to meet the latency requirement. Their system predicts future query region for moving consumers and starts the event processing early to make timely information available when consumers reaches the future locations. Work [36] proposes RECEP, which exploits overlapping interests in data and acceptable inaccurate results to reuse computation and reduce resource requirement. RECEP increases the scalability and amortizes the delay of mobile CEP systems.

### 3.3 Interfacing and programming model

In order to ease the effort for developers to port their applications to fog computing platform, we need unified interfacing and programming model. The reasons are 1) application-centric computing will be an important fog computation model, in which components in the environment will be application-aware and allow suitable optimizations for different kinds of applications; 2) it is hard for developer to orchestrate dynamic, hierarchical, and heterogeneous resources to build compatible applications on diverse platforms. Hong et al. [22] propose a high-level programming model for future Internet applications with on-demand scaling, which are large-scale geospatially distributed and latency sensitive. However, their scheme is dedicated on a tree-based network hierarchy in which fog nodes have fixed locations. Therefore, we may need more general schemes for diverse networks where fog nodes are nodes with dynamic mobility.

### 3.4 Computation Offloading

Computation offloading can overcome the resource constraints on mobile devices since some computation-intensive tasks can benefit from offloading in performance of applications, saving storage and battery lifetime. Existing work of computation offloading for mobile cloud computing can be classified into six metrics: objectives, granularity, scheme, adaptation, distributed execution and communication [12]. While there are plenty of search in computation offloading in the context of cloud computing and mobile computing [41, 8, 6, 26], we review a few of them in this paper. MAUI [8] propose code offloading and profile offloaded method to make decisions on future invocations adapting to the change of network connectivity, bandwidth and latency. It requires the developers to manually annotate methods that can be offloaded. CloudCloud [6] use static code analyzer to automatically mark possible migrate/merge point in program bytecode. ThinkAir [26] moves on to elasticity and scalability of the cloud and enhances the power of mobile cloud computing by parallelizing method execution using multiple virtual machine (VM) images. COMET [15] leverages the distributed shared memory and VM synchronization primitives to augment smartphones or tablets with machines available in the network.

The main challenges in offloading in fog computing are how to deal with dynamic. The dynamic has three fold 1) radio/wireless network access is highly dynamic 2) nodes in the fog network are highly dynamic 3) resources in the fog are highly dynamic. The federation of fog and cloud actually present us a three-layering construction: device-fog-cloud. Computation offloading in such infrastructure faces new challenges and opportunities. There are questions such

as which granularity to choose for offloading at different hierarchy of fog and cloud; how to dynamically partition application to offload on fog and cloud; and how to make offloading decisions to adapt dynamic changes in network, fog devices, and resources etc.

### 3.5 Accounting, billing and monitoring

Fog computing cannot be prosperous without a sustainable business model. According to current researches and proposals, the fog computing providers can consist of the following parties: 1) Internet service providers or wireless carriers, who can construct fog at their infrastructures. 2) Cloud service providers, who want to expand their cloud service to the edge of the network. 3) End users, who want to trade their spare computation, storage of their local private cloud to reduce the cost of ownership. Therefore, in order to do “Pay-as-you-go”, we need to resolve many issues. For example in terms of billing, we need to figure out how to set the price for different resources and how to set the fraction of the payment goes to different parties of fog. To enforce those pricing policies, we need accounting and monitoring the fog in different granularity. It is also interesting that how we dynamically do pricing in fog computing services to maximize revenue and utilization, just like what traditional industrialise do in airline ticketing, car rental and hotels [25, 54].

**User Incentives** An interesting business model to accelerate the deployment of fog computing is “Join Fog computing with private local cloud at the edge”. Local private clouds are also deployed at the edge of Internet, with computation and storage capacity. Though private cloud is aiming at provide cloud service to private party only. From the technique perspective of cloud computing and virtualization, it is possible to lease spare computation and storage to fog service provider and they will pay the owner of private cloud to reduce cost.

### 3.6 Provisioning and resource management

Cloud provisioning and resource management are still interesting topics in fog computing environment.

**Application-aware provisioning** The challenges lie in the mobility of end node since metrics such as bandwidth, storage, computation and latency will be changed dynamically. For example, in a connected vehicle scenario, we can track an in-duty ambulance and tune smart traffic light to ensure green traffic wave and give warning to all the nearby vehicles to clear the road. In order to meet the QoS requirement such as delay, we need to do provisioning in order to prepare resources to provide service mobility. Work [37] proposes MigCEP, a placement and migration method for both fog and cloud resources. By planning operator migration ahead, it ensures end-to-end latency restrictions and reduce network utilization. We feel like with Internet of Things, fog computing will play an important role in mobile crowd-sourcing/sensing applications by providing application-aware provisioning.

**Resource discovery and sharing** Resource discovery and sharing is critical for application performance in fog. Work [28] propose method dynamically select centralized and flooding strategies to save energy in heterogeneous networks, while there are more constraints to take into consideration in fog computing, such as latency, density and mobility. N. Takayuki, et al. propose a framework for heterogeneous

resource sharing in fog computing [34] by mapping heterogeneous resources such as CPUs, communication bandwidth, and storage all to “time” resources. The resource sharing optimization problems can be formulated for maximizing the sum or product of service-oriented utility functions. However, the utility function is only about service latency which can be further expanded to include metrics such as service availability, energy consumption or even revenue.

### 3.7 Security and Privacy

Currently, there are few works focusing on security or privacy issues in fog computing. However, some topics have been studied extensively in the context of virtual machine and hypervisor [20], and cloud computing [46].

**Authentication** As the emergence of biometric authentication, such as fingerprint authentication, face authentication, touch-based or keystroke-based authentication etc, in mobile computing and cloud computing, applying biometric-based authentication in fog computing will be beneficial. I. Stojmenovic, et al [45] consider the main security issue of fog computing as the authentication at different levels of fog nodes. While public key infrastructure (PKI) based technique could solve this problem, we think trusted execution environment (TEE) technique may have its potential in fog computing [5, 31]. We may also leverage measurement-based method to filter fake or unqualified fog node that is not in end users’ vicinity to reduce the authentication cost [18, 19].

**Access control** Access control has been a reliable tool on smart devices [43], and cloud [56], ensuring the security of the system. To expand access control of data owner into the cloud, S. Yu, et al. [56] achieve this by exploiting techniques of several encryption schemes together to build an efficient fine-grained data access control in the context of Cloud Computing. Work [10] proposes a policy-based resource access control in fog computing, to support secure collaboration and interoperability between heterogeneous resources. In fog computing, we can also raise questions like how to design access control spanning client-fog-cloud, to meet the goals and resource constraints at different levels.

**Intrusion detection** Intrusion detection techniques have been applied to cloud infrastructures to mitigate attacks such as insider attack, flooding attack, port scanning, attacks on VM or hypervisor [33]. Those intrusion detection systems can be deployed on either host machine, VM and hypervisor to detect intrusive behavior by monitoring and analyzing log file, access control policies and user login information. They can also be deployed at network side to detect malicious activities such as denial-of-service (DoS), port scanning etc. In fog computing, it provides new opportunities to investigate how fog computing can help with intrusion detection on both client side and the centralized cloud side. There are challenges such as implementing intrusion detection in geo-distributed, large-scale, high mobility fog computing environment.

**Privacy** Users are concerned about the risk of privacy leakage (data, location or usage) on the Internet nowadays. Privacy-preserving techniques have been proposed in many scenarios including cloud [48, 4], smart grid [40], wireless network [39], and online social network [35]. In the fog network, privacy-preserving algorithms can be run in between the fog and cloud since computation and storage are sufficient for both sides while those algorithms are usually resource-prohibited at the end devices. Fog node at the edge

usually collects data generated by sensor and end devices. Techniques such as homomorphic encryption can be utilized to allow privacy-preserving aggregation at the local gateways without decryption [29]. For aggregation and statistical queries, *differential privacy* [11] can be applied to ensure non-disclosure of privacy of an arbitrary single entry in the data set.

## 4. CONCLUSION

This survey discusses definitions of fog computing with similar concepts, gives representative applications which will promote fog computing, and mentions various aspects of issues we may encounter when design and implement fog computing systems. Besides, new opportunities and challenges in fog computing for related techniques are discussed and issues related to QoS, interfacing, resource management, security and privacy are highlighted. Fog computing will evolve with the rapid development in underlying IoT, edge devices, radio access techniques, SDN, NFV, VM and Mobile cloud. We think fog computing is promising but currently need joint efforts from underlying techniques to converged at “fog computing”.

## 5. ACKNOWLEDGMENTS

This work was supported in part by US National Science Foundation grants CNS-1320453 and CNS-1117412.

## 6. REFERENCES

- [1] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan. Volley: Automated data placement for geo-distributed cloud services. In *NSDI*, 2010.
- [2] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann. Applying nfv and sdn to lte mobile core gateways, the functions placement problem. In *workshop on All things cellular*. ACM, 2014.
- [3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *workshop on Mobile cloud computing*. ACM, 2012.
- [4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *TPDS*, 2014.
- [5] C. Chen, H. Raj, S. Saroiu, and A. Wolman. ctpm: a cloud tpm for cross-device trusted applications. In *NSDI*, 2014.
- [6] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: elastic execution between mobile device and cloud. In *Eurosys*. ACM, 2011.
- [7] Cisco. Iox overview. <http://goo.gl/n2mfiw>, 2014.
- [8] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In *Mobisys*. ACM, 2010.
- [9] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *WCMC*, 2013.
- [10] C. Dsouza, G.-J. Ahn, and M. Taguinod. Policy-driven security management for fog computing: Preliminary framework and a case study. In *IRI*. IEEE, 2014.
- [11] C. Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*. 2011.
- [12] N. I. M. Enzai and M. Tang. A taxonomy of computation offloading in mobile cloud computing. In *MobileCloud*. IEEE, 2014.
- [13] ETSI. Mobile-edge computing. <http://goo.gl/7NwTLE>, 2014.
- [14] Gil Press. Idc: Top 10 technology predictions for 2015. <http://goo.gl/zFujnE>, 2014.

- [15] M. S. Gordon, D. A. Jamshidi, S. A. Mahlke, Z. M. Mao, and X. Chen. Comet: Code offload by migrating execution transparently. In *OSDI*, 2012.
- [16] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan. Towards wearable cognitive assistance. In *Mobisys*. ACM, 2014.
- [17] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. Network function virtualization: Challenges and opportunities for innovations. *Communications Magazine, IEEE*, 2015.
- [18] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu. A measurement based rogue ap detection scheme. In *INFOCOM*. IEEE, 2009.
- [19] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu. A timing-based scheme for rogue ap detection. *TPDS*, 2011.
- [20] Z. Hao, Y. Tang, Y. Zhang, E. Novak, N. Carter, and Q. Li. SMOC: a secure mobile cloud computing platform. In *INFOCOM*, 2015.
- [21] B. Heller, R. Sherwood, and N. McKeown. The controller placement problem. In *HotSDN*. ACM, 2012.
- [22] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe. Mobile fog: A programming model for large-scale applications on the internet of things. In *ACM SIGCOMM workshop on Mobile cloud computing*, 2013.
- [23] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe. Opportunistic spatio-temporal event processing for mobile situation awareness. In *DEBS*. ACM, 2013.
- [24] J. Hwang, K. Ramakrishnan, and T. Wood. Netvm: high performance and flexible networking using virtualization on commodity platforms. In *NSDI*. ACM, 2014.
- [25] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki. Optimal service pricing for a cloud cache. *TKDE*, 2011.
- [26] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *INFOCOM*. IEEE, 2012.
- [27] I. Ku, Y. Lu, and M. Gerla. Software-defined mobile cloud: Architecture, services and use cases. In *IWCMC*. IEEE, 2014.
- [28] W. Liu, T. Nishio, R. Shinkuma, and T. Takahashi. Adaptive resource discovery in mobile cloud computing. *Computer Communications*, 2014.
- [29] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *TPDS*, 2012.
- [30] H. Madsen, G. Albeanu, B. Burtschy, and F. Popentiu-Vladicescu. Reliability in the utility computing era: Towards reliable fog computing. In *IWSSIP*. IEEE.
- [31] C. Marforio, N. Karapanos, C. Soriente, K. Kostianen, and S. Cegapkun. Smartphones as practical and secure location verification tokens for payments. In *NDSS*, 2014.
- [32] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici. Clickos and the art of network function virtualization. In *NSDI*. ACM, 2014.
- [33] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan. A survey of intrusion detection techniques in cloud. *JNCA*, 2013.
- [34] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam. Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud. In *workshop on Mobile cloud computing & networking*. ACM, 2013.
- [35] E. Novak and Q. Li. Near-pri: Private, proximity based location sharing. In *INFOCOM*. IEEE, 2014.
- [36] B. Ottenwalder, B. Koldehofe, K. Rothermel, K. Hong, and U. Ramachandran. Recep: selection-based reuse for distributed complex event processing. In *DEBS*. ACM, 2014.
- [37] B. Ottenwalder, B. Koldehofe, K. Rothermel, and U. Ramachandran. Migcep: operator migration for mobility driven distributed complex event processing. In *DEBS*. ACM, 2013.
- [38] Z. Qian, Y. He, C. Su, Z. Wu, H. Zhu, T. Zhang, L. Zhou, Y. Yu, and Z. Zhang. Timestream: Reliable stream computation in the cloud. In *Eurosys*. ACM, 2013.
- [39] Z. Qin, S. Yi, Q. Li, and D. Zamkov. Preserving secondary users’ privacy in cognitive radio networks. In *INFOCOM, 2014 Proceedings IEEE*, pages 772–780. IEEE, 2014.
- [40] A. Rial and G. Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011.
- [41] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *Pervasive Computing*, 2009.
- [42] B. Sheng, Q. Li, and W. Mao. Data storage placement in sensor networks. In *Mobihoc*. ACM, 2006.
- [43] S. Smalley and R. Craig. Security enhanced (se) android: Bringing flexible mac to android. In *NDSS*, 2013.
- [44] I. Stojmenovic. Fog computing: A cloud to the ground support for smart things and machine-to-machine networks. In *ATNAC*. IEEE, 2014.
- [45] I. Stojmenovic and S. Wen. The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, 2014.
- [46] H. Takabi, J. B. Joshi, and G.-J. Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security and Privacy*, 2010.
- [47] L. M. Vaquero and L. Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, 2014.
- [48] C. Wang, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. Ieee, 2010.
- [49] H. Wang, C. C. Tan, and Q. Li. Snoogle: A search engine for the physical world. In *INFOCOM*. IEEE, 2008.
- [50] H. Wang, C. C. Tan, and Q. Li. Snoogle: A search engine for pervasive environments. *TPDS*, 2010.
- [51] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung. Cache in the air: exploiting content caching and delivery techniques for 5g systems. *Communications Magazine, IEEE*, 2014.
- [52] D. F. Willis, A. Dasgupta, and S. Banerjee. Paradrop: a multi-tenant platform for dynamically installed third party services on home gateways. In *SIGCOMM workshop on Distributed cloud computing*. ACM, 2014.
- [53] F. Xu, C. C. Tan, Q. Li, G. Yan, and J. Wu. Designing a practical access point association protocol. In *INFOCOM*. IEEE, 2010.
- [54] H. Xu and B. Li. Dynamic cloud pricing for revenue maximization. *Cloud Computing, IEEE Transactions on*, 2013.
- [55] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan. A framework for partitioning and execution of data stream applications in mobile cloud computing. *ACM SIGMETRICS Performance Evaluation Review*, 2013.
- [56] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM*, 2010.
- [57] Y. Zhang, C. Tan, and Q. Li. Cachekeeper: a system-wide web caching service for smartphones. In *UbiComp*. ACM, 2013.
- [58] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi. Improving web sites performance using edge servers in fog computing architecture. In *SOSE*, 2013.