Future Edge Cloud and Edge Computing for Internet of Things Applications

Jianli Pan, *Member, IEEE,* James McElhannon, *Member, IEEE*

Abstract— **The Internet is evolving rapidly toward the future "Internet of Things" (IoT) which will potentially connect billions or even trillions of edge devices which could generate huge amount of data at a very high speed and some of the applications may require very low latency. The traditional cloud infrastructure will run into a series of difficulties due to centralized computation, storage, and networking in a small number of datacenters, and due to the relative long distance between the edge devices and the remote datacenters. To tackle this challenge, edge cloud and edge computing seem to be a promising possibility which provides resources closer to the resource-poor edge IoT devices and potentially can nurture a new IoT innovation ecosystem. Such prospect is enabled by a series of emerging technologies including Network Function Virtualization (NFV) and Software Defined Networking (SDN). In this survey paper, we investigate the key rationale, the state-of-the-art efforts, the key enabling technologies and research topics, and typical IoT applications benefiting from edge cloud. We aim to draw an overall picture of both ongoing research efforts and future possible research directions through comprehensive discussions.**

Index Terms— **Survey, Internet of Things, edge cloud, edge computing, Network Function Virtualization (NFV), Software Defined Networking (SDN), HomeCloud**

I. Introduction

The Internet is evolving rapidly toward the future "Internet of Things" (IoT), which will potentially network billions or even trillions of devices. As predicted by Ericsson Inc. [1], more than 50 billion devices will connect to Internet by the year 2025. Most of these devices will be located at the edge of Internet and could provide new applications, changing many aspects of both traditional industrial productions and our everyday living. Some devices that already appeared include Apple watches, Oculus Rift helmets [2], Google Nest [3], Fitbit sports trackers, and Google Glasses. The edge IoT devices actually can be any kind of sensors and chips with various capabilities made by different manufacturers, and many applications can be built to enable smart home, smart healthcare, smart transportation, smart buildings, and smart cities. For the current cloud computing and application infrastructure, it is very common that these large amounts of edge devices need to work closely with the application servers located at a small number of distributed large-size datacenters because most of the computation, storage, and networking resources are in these power datacenters that are owned by the Application Service Providers (ASPs) such as Google, Amazon, Microsoft, Facebook, and Apple.

A. The "Good and Bad" with Current Cloud Computing Model

The conventionally centralized cloud computing model favors several large-sized distributed datacenters. It has proved to be a huge success in the current Internet and was broadly adopted by the aforementioned giant corporations. The success can be attributed to several factors: (1) it provides an on-demand pay-as-you-go service to the users which lowers the owning cost for general customers; (2) it provides elasticity of computing, storage, and networking resources which is flexible and scalable; (3) it facilitates big-data analytics using machine learning technologies due to the highly centralized colocation of intensive computation and data. In short, it is through economics of scale in operations and system administration that the conventional cloud computing wins.

However, such a centralized model will face significant challenges toward the IoT world and we briefly discuss some.

(1) Volume and velocity of data accumulation of IoT devices. In current model, the new application delivery highly depends on giant companies' proprietary overlays and tools, and they generally have to transfer all the data from the edge devices to the remote datacenters, which will not be possible considering the volume and velocity of the data generated by the IoT devices in the future.

(2) Latency due to the distance between edge IoT devices and datacenters. The centralized cloud model also leads to a fact that the edge devices (often mobile) are usually relatively far away from the datacenters. In the future when the number of edge devices experiences exponential increase, it is imaginable that high latency can be a big challenge for quite a number of applications that involve end-to-end communications.

(3) Monopoly vs. open IoT competition. Current centralized cloud infrastructure is usually expensive to build and is only affordable to those giant companies that tend to define and use proprietary protocols. Customers are easily stuck to some specific infrastructures as the cost of switching to others could be dreadful. Such lack of openness could lead to a monopoly, ossification of the Internet, and further inhibit innovations.

In short, we need to address the deficiencies of the traditional cloud computing model. In our opinion, an open edge cloud infrastructure is inevitable and necessary to embrace the paradigm

J. Pan and J. McElhannon are with the Department of Mathematics and Computer Science, University of Missouri, St. Louis, MO 63121 USA email: pan@umsl.edu, and jlmx26@mail.umsl.edu.

This work was supported in part by grants H98230-17-1-0393 and H98230-17-1-0352 from National Security Agency (NSA), and a grant from University of Missouri System Research Board (UMRB) Award.

shift to the future IoT world.

B. *Why Edge Cloud?*

With open edge cloud infrastructures, firstly, the above challenge (1) can be addressed by providing local computing, storage, and networking resources to assist the often resource-poor IoT devices. The data generated by the edge devices at bewildering rates can be stored and preprocessed by the local edge cloud and only a small volume of processed data are required to be sent back to central datacenters. The networking load can be reduced. Secondly, for challenge (2), the IoT devices can offload [4] their tasks to the edge servers if the loads are beyond their capabilities. Since the edge cloud is closer to the devices, the latency can be well controlled compared to the conventional cloud computing model. Thirdly, for challenge (3), an open edge cloud innovation platform can break the monopoly and accommodate fairer competition among all stakeholders, no matter if they are giant corporations or small or medium-sized inventors, vendors or ASPs. Specifically, these small or medium-sized stakeholders are usually closer to the common users and are the most active and innovative groups for Internet community. Such an open environment would help nurture future innovations.

To show how the conventional cloud computing and the new edge cloud computing differ in various aspects, we summarize and compare their major characteristics in Table I.

Table I. Brief comparisons between conventional cloud computing and edge cloud and edge computing.

Characteristics	Conventional cloud computing	Edge cloud and edge computing
Major applications	Most of the current mainstream cloud-involved applications	Applications on IoT, VR, AR, smart homes, smart cities, smart energy, smart vehicles, etc.
Availability	A small number of large-sized datacenters	A large number of small-sized datacenters
Proximity of services and resources; Data processing location	Usually in remote datacenters and far from users	At the edge close to the users
End-to-end latency	High, due to the distance between the edge and remote datacenters	Low, due to proximity to the users
Backbone network bandwidth consumption	High, since huge data need to be transferred to the datacenters first	Low, since data are locally processed and stored in edge cloud
Scalability	Scalable at center	Scalable both center and edge
Security (e.g., attacks on data enroute)	Data subject to attack due to long-distance transmission; Physical security depends on large facilities	Lower risk for enroute attacks; Physical security varies and different mechanisms needed

Overall, the current edge cloud research is in an early stage and there are many challenges to be addressed. However, such trends and demands are being widely acknowledged, for example, in the three "Looking beyond the Internet" workshops [5] organized by the National Science Foundation (NSF) at the beginning of 2016. In this paper, we will present a comprehensive survey on the current research status and efforts regarding edge cloud. We will investigate the emerging key enabling technologies and efforts coming from both academia and industry. Example use cases will also be discussed and our unique perspectives and work will also be briefly presented. We aim to draw an overall picture of both ongoing research efforts and future possible research directions through comprehensive discussions. Comparing with two latest efforts addressing similar topic [6], [7], this paper presents a more comprehensive coverage of the state-of-the-art research and industry activities, and we focus more on the future key enabling technologies such as NFV and SDN, and in a future IoT application delivery new perspective. These are our unique contributions.

The rest of this paper is organized as follows. We discuss some existing related research and efforts in Section II. Section III is about several key enabling technologies, research topics, and applications. In Section IV, we discuss some challenges and our perspectives and observations. Finally, a summary follows in Section V.

II. State-of-the-Art Efforts

In this section, we will discuss the status quo of the edge cloud related research and some existing efforts.

A. *Current Status*

Edge cloud related technologies are drawing increasing attention from both academia and industry. However, the concept and development is currently in a relatively early stage and many challenges are ahead to be solved from both academic and industry perspective. Most of the existing edge computing frameworks involve dedicated physical edge computing servers that work with the edge sensors for computation and storage, or involve simple dockers that provide very limited virtualization supports at the edge. They are mostly standalone deployments for applications such as video surveillance or video analytics. In these cases, the involved edge computing platforms are technically NOT an "edge cloud", and they are with limited scales and are rarely with multi-applications delivering capabilities. To enable a true "edge cloud" as a unified IoT application delivery platform for the future, first, the orchestration, application delivery mechanisms and processes for edge cloud could be significantly different from those traditionally centralized cloud applications. No mature business models are available and the "killer applications" are still yet to come. Second, the key enabling technologies such as Network Function Virtualization (NFV) [8] and Software Defined Networking (SDN) [9] for future edge cloud are still in early stage and their research and application are being carried out by different organizations. There are many unknown and uncertain things about them yet and they are also evolving respectively. There is no standard guideline on how they should interact for edge new application delivery. Furthermore, the research on synergistically integrating them to provide new IoT applications is just beginning.

B. *Cloudlet*

"Cloudlet" [10] is a project from a research group in Carnegie Mellon University. Its goal is to achieve the convergence of mobile computing and cloud computing by introducing a multi-tier hierarchical structure. The structure is approximately illustrated in Fig. 1. We can see that in the 3-tier hierarchy, the Cloudlet tier is standing in between the mobile devices and the central cloud tier. The Cloudlet presents as a small "datacenter in a box" close to the mobile devices and assists them with low end-to-end latency and high bandwidth. The mobile devices could potentially offload [4] the computation to the Cloudlets for various applications.

For the offloading and coordination between mobile devices and the “Cloudlets”, a Virtual Machine (VM) based approach is adopted instead of using process migration or software virtualization. Specifically, Cloudlet project proposed a dynamic VM synthesis mechanism, which means that the device-VMs interaction is user-driven and on-demand, and the mobile devices can negotiate with the Cloudlet infrastructure to dynamically request and launch VMs. The VMs can be created and discarded dynamically regardless the stability of Wide Area Network (WAN) connectivity. The Cloudlet prototype was implemented as an extension to the OpenStack [11] platform and was named “OpenStack++”, which was also used to build some example applications such as “GigaSight”, “QuiltView” and “Gabriel” as published by the research group. The current Cloudlet project mostly focuses on applications such as crowd-sourced video surveillance and cognitive assistance (such as using Google Glasses) which need intensive computation at the edge. Since Cloudlet is based on OpenStack, which is evolving quite fast (new release every 6 months), and some features may come and go, some researchers argued that such an open-source platform and model could potentially be inconsistent and unstable for future commercial application requirements.

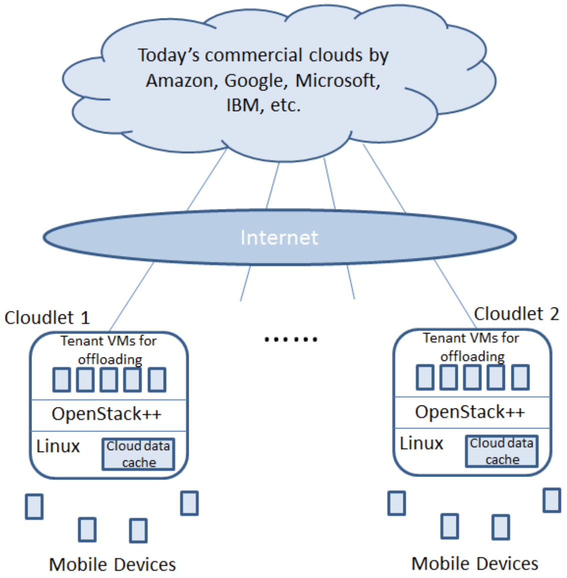

Fig. 1. The 3-tier structure of Cloudlet

From the structure and discussions we can see that the major design goal that Cloudlet’s approach trying to achieve is the convergence of mobile computing and cloud computing at a location closer to the users and IoT devices. Virtualization is used at the edge so that resources can be provisioned to assist the application-specific tasks offloaded from the mobile and wireless IoT devices. In terms of security, by pushing the cloud platform to the edge and providing computation closer to the users, the risk of data compromise in transmission can be significantly reduced.

C. Fog Computing

The “Fog Computing” [12] concept was originally proposed by some researchers from Cisco in 2012. The original idea was to extend the cloud computing and services to the edge of network to ease the wireless data transfer facing the Internet of Everything (IoE) trend. Fog Computing aims to provide data, compute, storage, and services to the end-users with proximity, dense geographical distribution, and mobility support. The claimed benefits also include reducing the data movement across the network, network congestion, end-to-end latency, and bottlenecks, and improving security and scalability to some extent. Fog model also claims benefits in advertising, entertainment, and big data analytical applications to the users. Even broader applications include IoT, connected vehicles (CV), Wireless Sensor and Actuator Networks (WSAN), Cyber-Physical Systems (CPSs), and distributed smart building control. Fig. 2 provides an example illustration of the Fog Computing layering architecture and its relationship with other related technologies. From the figure, we can see the Fog framework illustrates the cloud layering structure and the relationship between layers more clearly. It is also more general and applies to both WiFi-based Internet, mobile wireless telecommunication network, and even power-line communications network (PLC). It also shows the necessity of creating a synergistic distributed cloud platform between the traditionally centralized data centers and the tens of thousands of new Fog network at the edge.

Fog Computing also claims features such as improved security and elimination of the core computing environment. However, these issues are twofold. On one hand, Fog Computing could reduce network bandwidth and keep the data processing at the edge which reduces the possibility of attacks on data enroute. On the other hand, data and edge cloud infrastructure security at the edge could also be a challenge. Edge facilities may not be equipped with sophisticated security mechanisms and physical attacks can be relatively easier compared with centralized cloud computing. Also, it is not likely that Fog Computing will entirely eliminate the core computing environment. Quite the opposite, they are likely to coexist and be complementary with each other to fulfill their jobs for different applications and scenarios.

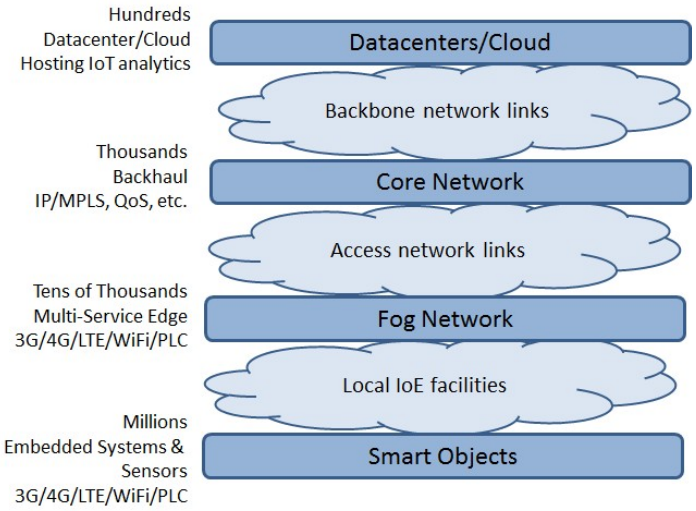

Fig. 2. Fog Computing layering architecture

A deeper look in to the Fog model and we can see it bears somewhat similar idea with Cloudlet. However, a notable difference is that Cloudlet is a project from academia trying to build some example applications using virtualization, while Fog Computing concept originates from industry and ambitiously tries to bring all the networks (including Internet and the 3G/4G/LTE networks) and everything (all smart objects, i.e., "Internet of Everything" or IoE) into the new perspective with a distributed and hierarchical cloud structure. In terms of security, it also reduce data attacks enroute. The edge Fog network is also more tolerant to network failure.

D. Mobile-Edge Computing (MEC) Initiative

The European Telecommunications Standard Institute (ETSI) launched a MEC Industry Specification Group (ISG) in late 2014 [13]. MEC is deemed as a critical technology to enable the transition toward future 5G and IoT world. Its major goal is to provide a network architecture that enables cloud computing capabilities and IT service environment at the edge of mobile cellular networks. It aims to provide a new ecosystem and value chain for application developers, content providers, network operators (carriers), and customers. In MEC perspective, the Radio Access Network (RAN) edge can be open to authorized third-parties for new application delivery. The MEC servers can be deployed at sites such as LTE macro base stations (eNodeB), 3G Radio Network Controllers (RNC), and multi-technology cell aggregation sites. The location of the cell aggregation sites can be very flexible: they can be located indoors within an enterprise, or indoors/outdoors in a large public building or arena. By deploying various new services at these edge sites that are close to the customers, the mobile cellular core network is alleviated of huge traffic burden and the edge can serve local demands more efficiently. Some typical use cases include video analytics, location services, IoT, augmented reality, optimized local content distribution, and data caching. An example MEC distributed video analytics use case is illustrated in Fig. 3. In this use case, video streams from multiple cameras arrive at the MEC server located at the LTE base station. The video management application transcodes and stores the video streams, and the video analytics application processes the video data and detects specific events. Only low-volume metadata are sent to the Core/IT servers for database searches。

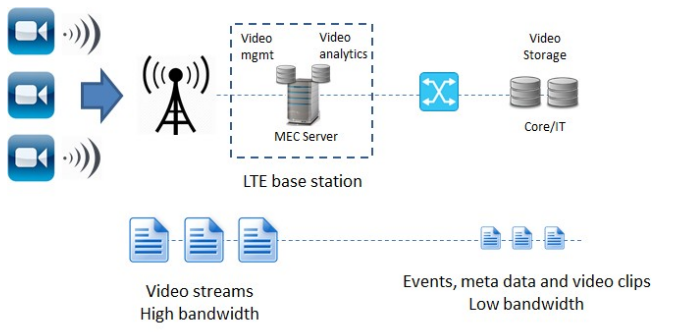

Fig.3. Example video analytics use case in MEC

As we can see a significant difference between the MEC architecture and Cloudlet or Fog Computing is that MEC is primarily focused on the mobile cellular network instead of the general Internet. Since the cellular network is a relatively "closed" environment compared to the "open" Internet, it could be more challenging to implement the edge computing idea in the mobile cellular network infrastructure. For example, interoperability is an important goal to achieve so that various devices and applications from multiple ASPs can run and coordinate on the mobile-edge computing platforms. Different stakeholders and players in the value chain need to actively participate and contribute to enable such a vision in the future. As of December 2015, three MEC Proofs of Concept (PoC) have been developed and demonstrated. They are: (1) RAN-aware video user experience optimization, (2) edge video orchestration and video clip replay, and (3) radio-aware video optimization in a fully virtualized network. Since 2016, the MEC group began working on platform services, APIs and interfaces. The MEC APIs should be transparent to the applications and will allow them to be portable on different edge servers across platforms with guaranteed Service Level Agreements (SLAs). In addition, as for security, the MEC requires the framework to fulfill the 3GPP security specifications. The applications also need to be provided with secure sandboxes for the deployment.

E. Central Office Re-architected as a Datacenter (CORD)

Central Office Re-architected as a Datacenter (CORD) [14] is a collaborative project between AT&T and ON.Lab, and it is under active development by the end of year 2017. An Open CORD effort has been formed to encourage the community participation and contributions in framework, new services, new hardware and building blocks. While the ETSI MEC initiative is primarily focused on mobile telecom network, CORD is more focused on the wireline access networks. Its goal is to transform the legacy Central Offices (C.O.) into CORD which integrates NFV, SDN, and Cloud into service providers' access networks. Today's telecommunication Central Offices are a huge source of Capital Expenditure (CAPEX) and Operational Expenditure (OPEX), and the infrastructure lacks programmability and flexibility. The CORD project aims to bring the cost, performance, and agility of Google or Facebook to the traditional telecommunication network providers. CORD virtualizes not just individual appliances; instead, it aims to holistically deliver end-to-end SDN/NFV/Cloud solution at the telecommunication Central Offices.

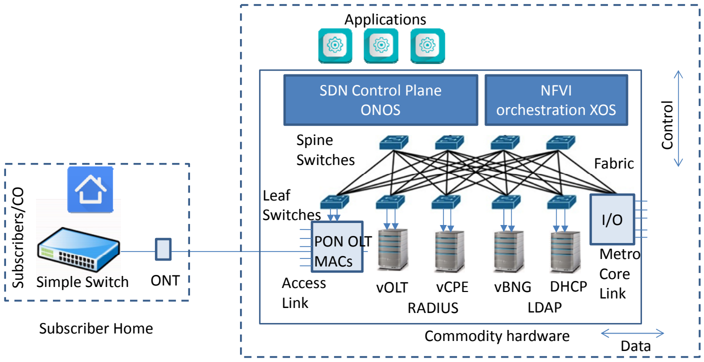

Fig.4. CORD building blocks

In CORD, the closed and proprietary hardware in the Central Offices is replaced by separate commodity hardware and software. This means software is decoupled from hardware and can be provided by different vendors in an open environment. CORD leverages open source software by combining multiple types of open projects including OpenStack [11], Open Network Operating System (ONOS) by Open Networking Lab (ON.Lab), and XOS (by ON.Lab). OpenStack is used to manage virtualized infrastructure where Virtual Machines (VMs) are instantiated on industry-standard commodity servers. ONOS is the SDN control plane used to manage the virtual networks and configure and control the virtualized network functions. XOS is used to orchestrate and manage high-level services running on OpenStack and ONOS. A conceptual illustration of the CORD building blocks is shown in Fig. 4. CORD can build white boxes and hence various services on top of underlying open commodity infrastructure. Some example services are vCPE (virtualized Customer Premises Equipment), vOLT (virtualized Optical Line Termination), and vBNG (virtualized Broadband Network Gateway). The SDN control plane functionalities of these virtualized entities are done by ONOS and the NFV orchestration and service management are done by XOS and OpenStack.

CORD envisions a big picture of "Everything-as-a-Service" for the future. Basically, in the Central Offices, multiple appliances with different functionalities can be virtualized to provide end-to-end services. For example, Access-as-a-Service (ACCaaS) is implemented by vOLT control application running on ONOS in which tenants are the subscriber VLANs. Subscriber-as-a-Service (SUBaaS) is implemented by vCPE running in a Linux box where tenants are the subscriber bundles. Internet-as-a-Service (INTaaS) is implemented by vBNG control application running on ONOS in which routable subnets are the tenant abstraction. A CORD proof of concept prototype was built and demonstrated in the Open Networking Summit in June 2015. The demo showed multiple scenarios of virtualizing the CPEs, OLTs, and BNGs, and working with G.fast and GPON (Gigabit Passive Optical Networks) high-speed access network technologies. In CORD's strategic roadmap, it was under lab trial of the CORD "POD", which is a bundle of all software and hardware building blocks as a ready-to-use system in 2016 and 2017. After several planned trial deployments, full development and deployment of CORD is currently underway.

One of the important traits of CORD that makes it different from academic projects such as Cloudlet is that it is driven by industry demands and it uses real access networks for trials, which is a big step forward. While it will not be an easy job given the long history of the relatively "closed" telecommunication networks, it is a good start moving to the future after all and many more exciting changes can be expected. Regarding security, in addition to the general security concerns related to the edge cloud, CORD also requires telecommunication network level security assurance.

F. Nebula

Nebula [15] is one of the several collaborative projects funded by the Future Internet Architecture (FIA) program of the National Science Foundation (NSF). It is led by a University of Minnesota research group. As of now, Nebula presents as a location- and context-aware distributed edge cloud infrastructure which allows volunteers from the edge to carry out distributed MapReduce tasks for data-intensive computing. The Nebula system architecture is shown in Fig. 5. We can see that at the center are a set of global and application-specific Nebula functionalities. The four components are Nebula Central, Nebula monitor, ComputePool Master, and DataStore Master. They work together to enable the data-intensive application on Nebula.

Overall, Nebula is somewhat different from the aforementioned other edge cloud infrastructures and it is more like a special case or application for edge computing. In Nebula, only the volunteer nodes come from the edge and their roles over the architecture is temporary and specific for the MapReduce tasks. For example, if you want to donate the compute resources, you can run it on Chrome web browser by enabling native client programs. There is no specific edge infrastructure that provides dedicated virtualization services for the mobile devices. For security consideration, because of the no-infrastructure-server design of Nebula, the compute nodes are able to offload computation to each other; hence it is more tolerant to compute node failure and data node failure.

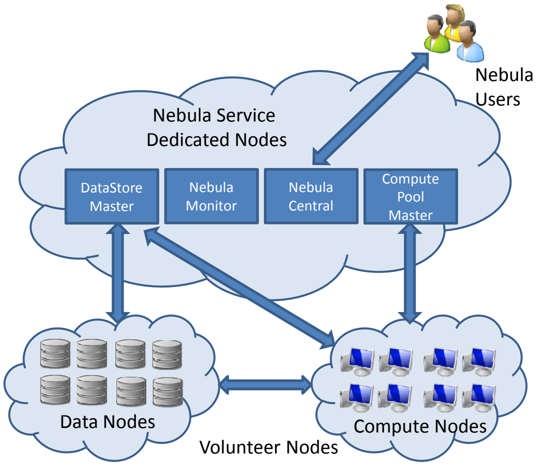

Fig.5. Nebula system architecture

G. FemtoCloud

FemtoCloud [16] is a project by a research group from Georgia Institute of Technology. It is a refactoring of the Cloudlet concept. The basic idea of FemtoCloud is that a group of mobile devices (e.g., in a coffee shop, a classroom, or a theaters) can be grouped and controlled by a controller to function as a cluster. The idle computing resources from these mobile devices can be shared within this group for a specific task managed by a controller. Such systems can present as mobile devices and applications in coffee shops, classrooms, and theaters.

The claimed benefits include better scalability and not relying too much on infrastructure. While the FemtoCloud concept is of experimentation value, some unavoidable challenges remain due to the mobile devices' high volatility, dynamicity, and instability to allow them to fulfil the offloading with each other in this environment. In comparison, an infrastructure server at the edge may be a more stable option to make the offloading more effective and predictable. Furthermore, due to the mobile devices' relatively high volatility, dynamicity, and instability, security may become challenges in such application environments.

H. Mobile-Edge Offloading and Foraging

Mobile-edge offloading [4] and foraging are two related key research topics for the collaboration and coordination between the mobile devices and the edge servers. Mobile-edge offloading allows the mobile devices to offload computation and storage to the edge servers for those tasks that need more resources. Typical research projects on mobile-edge offloading include MAUI (by Duke University in 2010) [17], CloneCloud (by Intel Labs Berkeley in 2011) [18], Odessa (by University of Southern California in 2011) [19], and COMET (by University of Michigan in 2012) [20]. More discussions about these technologies can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2767608, IEEE Internet of Things Journal

Table II. Summary and comparisons of the related efforts

Characteristics	Cloudlet	Fog Computing	ETSI MEC Initiative	CORD	NEBULA	FemtoCloud	Cloud offloading and Foraging	HomeCloud
Advocates and Sponsors	Academia	Industry; multiple vendors	Industry; Wireless telecom providers	Industry; Wireline service providers	Academia	Academia	Academia	Academia
Key design goals	Mobile computing and cloud computing convergence	Reduce data movement across networks; Bring all "networks" and all "things" in.	Transition mobile cellular networks toward 5G with edge cloud capabilities	Wireline telecom access networks	Location and context-aware edge cloud for data-intensive computing	Share idle resources among mobile devices	Mobile-edge offloading and foraging	Automated, open, and portable new IoT app delivery
Key design features	1. 3-tier cloud structure; 2. "datacenter in a box"; 3. Dynamic VM synthesis	Near-edge servers; Control plane and data plane separation	Radio Access Network (RAN) open to third-parties; MEC servers in flexible locations (eNodeB, RNC, etc)	Software and hardware separation in Central Offices (COs); using open source software on "whiteboxes"	Allows edge volunteers to contribute for distributed MapReduce apps	A group of mobile devices function as a cluster	Program partitioning and coordination between mobile and edge	NFV+SDN; Automated orchestration; multiple apps on same infrastructure; portability
Key applications	crowd-sourced video surveillance and cognitive assistance	Potentially all kinds of IoE, connected vehicles, WSN, CPS, smart buildings, etc.	Mobile cellular apps; Location tracking; Radio aware video optimization, etc.	Virtualize COs and deliver end-to-end SDN/NFV/Cloud solutions	Distributed data-intensive applications such as MapReduce	Mobile device apps in Coffee shops, classrooms, theaters, etc	Mobile programs partitioned to allow offloading parts to servers.	Future IoT apps, smart home, smart energy, VR, AR, etc
Infrastructure sever support	Yes, edge "boxes"	Yes, edge servers	Yes, at base stations, etc	Yes, at central offices (COs)	No dedicated edge servers	No, but with a task controller	Yes, at edge servers	Yes, at edge servers
Virtualization at the edge	Yes, extends OpenStack	Not specified	Yes	Yes	Not specified	No	No, but by program partitioning	Yes
SDN at the edge	Not specified	Not specified	Not specified	Yes	Not specified	No	No	Yes
Mobility support	Possibly VM migration	Not specified	Yes, but depends on the specific apps	Not specified	Dynamic and mobile volunteers at edges	Not specified	Process migration and code partitioning	Support high level mobility (user, data, VMs)
Application portability	Not specified	Not specified	Possibly yes, mechanisms unclear yet	Not specified	Not specified	Not specified	Applications not platform-independent	Yes, as principal designs goals
Security considerations	Reduce data transmission risks; edge apps failover	Reduce data attacks enroute; edge tolerant to network failure	Need to fulfill 3GPP security and provide secure sandbox for apps	Telecom level security requirements	Tolerant to compute node failure and data node failure	Difficult due to mobile devices' high volatility, dynamicity, and instability	Software security concerns related to code partitioning and coordination	Reduce data transmission risks; Reduce data attacks enroute; apps isolation and dynamic start/end

found at the survey paper [4]. To enable offloading, the first method is program partitioning that decides which parts can be run on mobile devices and which parts can be run on edge servers. The partitioning can be done manually by the programmers or be automated. MAUI reduces the burden on programmers and automates many steps for program partitioning which saves energy for the mobile devices. CloneCloud uses static analysis to decide automatically what could be offloaded. COMET focuses on "how to offload" instead of "what to offload" and uses the Distributed Shared Memory (DSM) systems to reduce the communication while still supporting multi-threaded applications. Odessa adopts an incremental greedy strategy to structure parallelism across mobile devices and edge servers for better partitioning. The second method is based on process or Virtual Machines (VMs) migration instead of program partitioning. Process migration needs Operating System (OS) support for checkpoint and restart, while live VM migration enables moving the entire OS and all its running applications in a mobile environment. CloneCloud and Cloudlets use the migration method, which reduces the burden of the programmers. These research systems can be summarized and categorized based on the following criteria: (1) where to offload; (2) when to offload; (3) what to offload. Mobile-edge offloading can be further traced back to cyber foraging technologies in mid-1990s that were used by more than 50 related solutions. There are two types of cyber foraging: computation offloading to extend battery life and increase computational capacity, and data staging to improve data transfers between mobile devices and edge servers by temporarily staging data in transit.

I. *Summary and Comparison*

To better show and compare these different efforts on edge cloud and edge computing, we provide Table II summarizing and comparing them. The characteristics we use for comparison include major advocates, design goals, design features, applications, infrastructure server support, virtualization at edge or not, SDN at edge or not, mobility support, application portability, and security.

From the summarizing table we can see that what unites the different approaches is the basic idea of providing computation, storage, and networking assistance to the wireless IoT devices from sources closer to them. The varying components of these approaches can be from different angles. For example, either the assistance can be provided through a dedicated virtualized edge cloud, which has a much bigger pool of resources, or from nearby other IoT peers that has available resources. Either the assistance can be provided through a group of virtualized machines to carry out tasks on behalf of the IoT devices, or the tasks of the IoT devices are partitioned and some parts of them are offloaded to others to accomplish.

Due to the length limitation, we are not able to enumerate all the references for the projects discussed. However, we are working a longer version of the survey, which includes a more complete reference list for further reading.

III. Key Enabling Technologies, Research Topics, and Typical Benefited IoT Applications

In this section, we discuss some key enabling technologies and potential research topics that can affect the future prospects of edge cloud and edge computing in facilitating the IoT prospect. We will explain the rationale and present some very typical IoT applications that can benefit from edge cloud or edge computing most.

A. *Key Enabling Technologies and Research Topics*

We will focus on three key enabling technologies and research topics for IoT application delivery.

1) Convergence of NFV and SDN in Edge Cloud

To push computing, storage, and networking resource to the edge and enable future IoT applications, a small-scale cloud computing platform is needed. NFV and SDN seem to be two key synergistic enabling technologies that enable such a vision.

Using NFV at the edge, local computing, storage, and networking resources are available and closer to edge devices for those applications (video monitoring, face recognition, and augmented reality) that generate intensive data or require low latency. In edge clouds, NFV can build on top of affordable industry standard servers, switches and storage, and create VNFs replacing traditional and specialized equipment from proprietary vendors. The VNFs can be launched or terminated dynamically according to demands, and can be placed in much more flexible positions. They can also be chained and scaled up or down for complex functions and applications. All the aforementioned benefits of NFV can be employed by the edge cloud applications if adopting NFV.

SDN, on the other hand, is very suitable to work with NFV in the edge cloud to network, configure, control, and manage the VNFs created by NFV. SDN could greatly reduce the costs and increase the flexibility and programmability of the VNFs in the edge cloud because of the separation of control from the data forwarding and the usage of centralized network control and configuration. A simple illustration of the relationship between NFV, SDN, and open innovation is shown in Fig 6.

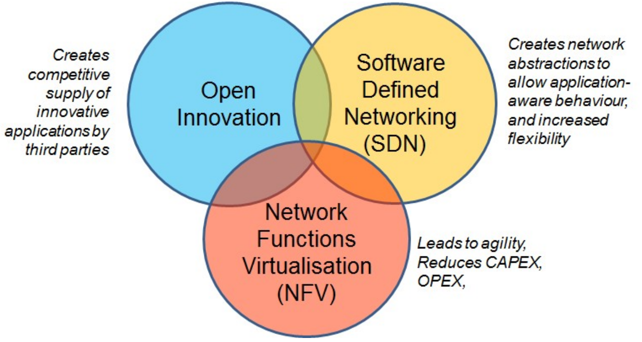
Creates competitive supply of innovative applications by third parties
Open Innovation
Software Defined Networking (SDN)
Creates network abstractions to allow application-aware behaviour, and increased flexibility
Network Functions Virtualisation (NFV)
Leads to agility, Reduces CAPEX, OPEX,

Fig. 6: Convergence of NFV and SDN in a open innovative environment.

From a technical view, NFV and SDN are highly complementary for edge cloud prospects. The separation of the control and data forwarding in SDN can simplify the compatibility of NFV with existing deployments. NFV can support SDN by providing the infrastructure on top of which SDN can run. The NFV and SDN convergence in edge cloud potentially opens a new door for innovative, fast, and cost-effective new service and application delivery and deployment. From a non-technical view, the stakeholders in future edge cloud and application market would include ISPs, ASPs, device vendors, and software vendors. The convergence of NFV and SDN would allow them to be treated fairly and equally benefit from future architecture and applications.

Putting NFV and SDN convergence into a broader context, it is the incoming 5G networks and the trend of “Network Softwarization”. The 5G networks aim at providing a significantly improved and programmable network infrastructure by 2020 when video traffic could dominate the mobile networks, the IoT and big data processing boom, and Virtual Reality (VR) becomes widespread and is provided with short delay. The International Telecommunication Union (ITU) formed a working group on Network Softwarization and the major deliverables for IMT-2020 networks had been defined in the draft [23] in October of the year 2015. Network Softwarization means an overall trend for designing, deploying, and managing network components by software programming along the whole life cycle of network. This enables redesign of networks, optimization of costs and processes, and enables automated management to the networks. Harmonious convergence of NFV and SDN is an integral part of realizing Network Softwarization and to enable high programmability of the networks. The core concept of Network Softwarization is called “Slicing”, which turns networks into logically isolated units of programmable resources such as networking, computation and storage. Synergy between NFV and SDN technologies becomes indispensable for such a perspective.

2) Automated Orchestration and Dynamic Offloading

(a) Automated Orchestration. Edge cloud research and its application in IoT are still in a relatively early stage and many challenges remain. One of them is effective cloud orchestration which is also an essential element and a key process for the edge cloud vision. Orchestration is defined as a set of methods and operations that the cloud providers and application owners undertake to either manually or automatically select, deploy, monitor, and control the configuration of hardware and software resources for application delivery [26]. However, the current orchestration approaches, from both open-source cloud control platforms [11] and commercial cloud providers (like Microsoft Azure and Amazon EC2), are not for edge cloud IoT applications. They highly depend on either manual or simple conditional check (if-then-else) method that is usually complex and error-prone to orchestrate the increasingly sophisticated cloud services. In addition, most of the orchestration methods are application-specific and are highly customized for certain types of applications, and using such method for edge cloud orchestration (where it is common that multiple IoT applications are required to be delivered and deployed in a shared edge cloud infrastructure) has significant limitations. Thus, there is a demand for automated tools and appropriate abstractions to turn the application requirements more effectively into orchestration schemes optimizing the resource allocation and provisioning for the IoT applications under deployment.

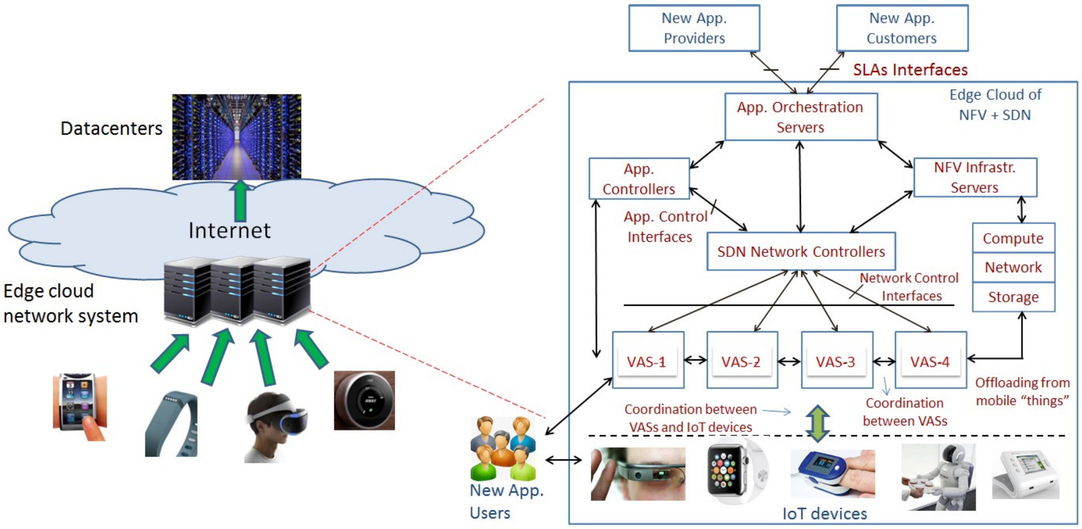

Fig. 7: Homecloud [22, 27] architecture for IoT application delivery.

(2) Dynamic Offloading. Meanwhile, in edge cloud, the orchestrator needs to work closely with various types of IoT devices to offload their data and computation to the edge cloud, and to dynamically and optimally commit appropriate resources to carry out these offloaded tasks matching the demands. There is also a lack of systematic configuration and integration method and framework to deliver various IoT applications and manage them efficiently over a unified edge cloud platform. We discussed some existing offloading approaches in Section II.H. Most of these traditional "program partitioning" or "process migration" based offloading methods depend on the programmers and are usually error-prone and hard to manage. As an alternative, similar to the Cloudlet, a Virtual Machine (VM) based offloading framework is relatively easier to control and manage, and with higher reliability. However, more than what Cloudlet provides, there is a demand to launch, configure, and manage these VMs effectively for a specific IoT application delivered in the edge cloud. To enable such a vision, a typical method would depend on the NFV and SDN integrated edge cloud platform to orchestrate the resources to fulfill the offloaded tasks from the battery-constrained mobile IoT devices. The edge cloud platform also configures the launched VMs to install and perform application-specific tasks.

B. *Example Effort—"HomeCloud"*

HomeCloud [22, 27] is one of the typical efforts that works in the direction of converged NFV and SDN, and focuses on the two key research issues of automated orchestration and dynamic offloading. It aims at developing an open framework for portable and automated IoT application delivery in future edge clouds. The architecture framework is shown in Fig. 7. It coherently integrates NFV and SDN in edge datacenter for efficient edge cloud orchestration and dynamic offloading for IoT application delivery and functioning.

In current cloud computing, various cloud companies have their own ways of delivering new services and they generally use different proprietary protocols and mechanisms that are usually closed, private, time-consuming, specialized designs and implementations. As such, these applications are also not portable across platforms. In the HomeCloud orchestration and application delivery framework, an objective oriented northbound mechanism is investigated to enable the application providers to turn the high-level Service-level Agreements (SLAs) into a series of implementable objectives that can be further parsed into machine-understandable schemes for resource allocation, control, configuration and management of VNFs. In other words, HomeCloud provides key mechanisms that sit between the edge cloud providers and the application providers to enable edge IoT applications to be efficiently delivered and orchestrated. Such objectives-oriented northbound approach would provide benefits in terms of portability, composability, and scalability, which are not available in any current centralized cloud model. HomeCloud also allows that different applications by different vendors are isolated in a shared edge cloud infrastructure with best cost-efficiency.

C. *Typical IoT Applications Benefiting from Edge Cloud*

Edge clouds or edge computing infrastructure will be important places for future innovations where potentially there will be many IoT applications. Some examples include smart home/buildings, home robotics, smart cities, smart health, augmented reality (AR) or virtual reality (VR), cognitive assistance, autonomous driving, video crowd sourcing, and M2M communications. We will not try to enumerate all such applications. Instead, we will discuss three types of future IoT applications could directly benefit from the edge cloud vision we discussed above. Moreover, since currently the key enabling technologies (like NFV and SDN) and the application delivery methods are still not mature yet, there seems to be a relatively long way to go before some true "killer" applications emerge in the market.

(1) Applications requiring low latency. For such applications, the traditionally centralized cloud computing would be vulnerable due to the large data volume generated and the long distance between clients and the backend datacenters. For example, "Foursquare" and "Google Now" applications require fast response to the users. Some wearable camera applications, or industrial monitoring and controlling applications require response time to be as low as 10 to 50 milliseconds. Some multimedia video or gaming applications also have high constraints on delays without significant downgrade of user experience.

(2) Applications requiring high data bandwidth. The number of edge IoT devices is experiencing exponential increase and it is predicted that by year 2020 these devices will generate 2.3 trillion gigabytes of data each day. Such scale of devices and data will post significant pressure to the Internet. It is necessary that most of the data are processed by the edge cloud or edge computing first and reduce the volume of data sent to the remote data center. This is also an economic and sustainable approach reducing costs and saving energy. The computing and storage resources can also be assigned and utilized more efficiently. Examples include those augmented reality (AR) or virtual reality (VR) applications using Oculus Rift helmets and Google Glasses.

(3) Applications involving large amount of IoT devices with limited capacities. Comparing with the servers in datacenters, most of the IoT devices or sensors at the edge are somewhat limited in both computing power and battery capacity. Limited by the hardware constraints, complex and intense computation are not suitable for these devices. Instead, most of such computation and data processing tasks can be offloaded to edge cloud, which could save energy on IoT devices and get the tasks done faster in the edge cloud. Example IoT applications include structure or agricultural field monitoring, and industrial monitoring, which usually involve a lot of small-sized and distributed sensors and devices. In these cases, the sensors and IoT devices mostly generate and send data but with limited capacity in processing and analyzing data.

IV. Challenges, Discussions, and Perspectives

Having presented a variety of related research projects, we find that there are some key issues worth further discussion. In this section, we present our perspectives and observations. Of course, it does not mean or imply any agreement among researchers.

(1) NFV and SDN coherency. While converging NFV and SDN for applications in the edge cloud context has huge potential, the research is still in an early stage since NFV and SDN technologies individually are still not mature and research on effective synergy between them is just beginning. Some typical example research questions to be answered in this regard include: (a) What necessary functional interfaces should be opened to each other between NFV and SDN for coherent interaction? (b) How to enable effective coordination among various VNFs and between mobile IoT devices ("Things") and edge clouds (the VNFs)? (c) How would the edge cloud orchestrator interact with NFV and SDN modules to create multiple applications through both northbound and southbound interfaces and mechanisms?

(2) 4G/5G Mobile Networks vs. Internet. The edge cloud trend and virtualization technologies are transforming networks everywhere. Future IoT applications can be deployed in various networks including both 4G/5G mobile telecommunication networks and traditional Internet. However, there could be significant differences on how and where they implement the edge cloud idea based on their own existing network architecture. For 4G/5G networks, edge cloud implementation can be in the cell towers (NodeB) to provide services close to the mobile phone users for better user experience or new applications. For Internet, the location of the edge servers can be in a Central Office (as in CORD), in a building or community, or near the Access Point of a smart home. Due to the network architecture difference, their implementations and corresponding participating vendors may also vary significantly. An open challenge for the 4G/5G carriers is to re-architect their networks from traditionally closed and proprietary platforms and devices to be open to third-party software and hardware vendors for new innovations, since the key technologies such as NFV and SDN are all open-source endeavors.

(3) Coordination between IoT devices and the Edge. The key motivation of edge computing is to allow the edge servers to get involved to help the IoT devices with computing, storage, and networking. How the things and the edge coordinate with each other to achieve the goals will have significant impacts on the effectiveness of such methods. In traditional offloading and cyber foraging, program/process partitioning technologies had been very broadly studied. However, they generally add complexity in programming and put an extra burden on the application developers. A typical alternative which is also advocated by the Cloudlet project is that instead of programming the applications differently to be adaptive, a whole Virtual Machine (VM) can be dynamically launched in the edge cloud and the computation tasks can be done in the VM as a whole until the results are sent back to the "things". VMs can be launched and deleted dynamically on demand. Such method could simplify the developers' work and further reduce complexity. Understandably, such an implementation may involve extra delay while the NFV platform manages the VMs. However, the good news is that for most of the resource-intensive tasks involving offloading, the benefits normally out-number the costs. This VM based method is also facilitated by the VM live migration technologies that had been relatively well studied.

(4) Southbound Interfaces (SBIs) and Northbound Interfaces (NBIs) [24, 25]. Integrating NFV and SDN is not an easy task as it involves multiple stakeholders that may implement the concepts differently. These stakeholders may not be motivated to work together and may not necessarily provide enough and clear interfaces to integrate with other software vendors. SBIs and NBIs are two important types of interfaces to make everything happen fluently and synergistically. NBIs [24, 25] generally refer the interfaces between the application plane and control plane and SBIs refer to interfaces between the control plane and data plane. While the SBIs have been well defined by the protocols such as OpenFlow [21], the NBIs have not. According to a recent post on the Open Networking Foundation (ONF) blog, more than 20 SDN controllers and hence preliminary NBIs are currently available for SDN in the marketplace. It is unlikely the NBIs will be standardized in the short term, which has the possibility to stifle innovations. To allow deploying and delivering scalable and portable future applications over the dynamic NFV and SDN infrastructures, appropriate NBIs are very important and more research efforts are needed. An example ongoing effort is that the ONF northbound API working group is working on developing an "intent-based" NBIs system.

(5) SDN Multi-cloud Scenarios. From the discussions on the three NSF workshops [5], SDN is evolving to SDX ("Software-Defined Exchange"). The traditional SDN concept is for inside a network, while SDX refers to applying the concept to inter-domain networking. One of SDX's goals is to enable large-scale interconnections of Software-Defined Internet (SDI) owned and operated by various organizations while gaining similar benefits in flexibility and programmability as in SDN for an individual network. Moreover, with SDX, a series of new features (impossible or difficult to achieve in the current inter-domain routing system) can be provided, including application-specific peering, blocking Denial-of-Service (DoS) traffic, load balancing, steering through network functions, and inbound traffic engineering. Applying SDN in the inter-domain scenario, even at single IXP (Internet Exchange Point), could benefit tens of hundreds of providers without deploying new equipment. The SDX perspective may affect future edge cloud applications as well. For example, how can SDN in the local edge clouds better interact and coordinate with the SDX in the future Internet for more benefits and features? Many interesting research topics may emerge.

(6) Security. Security remains one of the most important challenges for the future edge cloud infrastructure and applications. Due to the fact that the future edge cloud could involve multiple technologies (such as NFV, SDN, and IoT), security concerns will be multi-fold. First, because of the adoption of virtualization technology in edge cloud, security concerns with all the traditional cloud computing model (such as the VM security) will also exist for edge clouds. Second, because the edge cloud servers are sparsely located and are close to the users' premises, they may be more fragile to physical attacks. Third, security issues for individual technologies such as NFV, SDN, and IoT will continue to exist in the edge clouds. Since the future edge clouds could be a synergistic effort and all of these technologies may play respective roles, additional security issues may also come up from the interfaces or interactions among them. Fourth, multiple applications could run on the shared infrastructure in edge clouds, so it is important to address the application-level security issues such as appropriate application isolation and shared traffic and data access for multiple applications. Lastly, software security can also be a challenge. Since the future edge clouds will enable more programmability and the open platform will allow more third-party software and hardware vendors to weigh in and contribute, it is important to control and manage the potential risks among different stakeholders. Also, appropriate authentication, authorization, and auditing mechanisms may be required to identify and protect the trusted parties and defend from potential malicious attacks and misuses.

V. Summary

Empowered by the emerging technologies such as NFV and SDN, edge cloud and edge computing technologies are promising to address multiple challenges with the current cloud computing model facing with the future IoT world. In this survey paper, we investigated the motivations, state-of-the- art research efforts, key enabling technologies, and possible future use cases for the edge cloud environment. We aim to draw an overall picture of the topic through comprehensive discussions.

References

[1] Ericsson Inc, "Ceo to shareholders: 50 billion connections 2020," 2010, available at: http://www.ericsson.com/thecompany/press/releases/2010/04/1403231.

[2] Oculus, "Oculus Rift helmet: next generation virtual reality," 2016, available at: https://www3.oculus.com/en-us/rift/.

[3] Google, "Nest IoT devices," 2016, available at: https://nest.com/.

[4] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," GetMobile: Mobile Computing and Communications, vol. 18, no. 4, pp. 19–23, 2015.

[5] NSF Workshop, "Looking beyond the internet," March 2016, available at: https://lookingbeyondtheinternetblog.wordpress.com/.

[6] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," IEEE Internet of Things Journal, 2016.

[7] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in the Proceedings of the 10th IEEE International Conference on Intelligent Systems and Control (ISCO 2016), Coimbatore, India, 2016.

[8] ETSI, ISG, "Network Functions Virtualization (NFV): Introductory White Paper," Virtualization Requirements, 2014.

[9] Open Networking Foundation (ONF), "Software-Defined Networking: The New Norm for Networks, White Paper," 2014.

[10] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," IEEE pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[11] "Openstack: free and open-source software cloud computing platform," 2016, [Online resource], available at: http://www.openstack.org/. [Online]. Available: http://www.openstack.org/

[12] F. Bonomi, R. Milito, J. Zhu et al., Fog computing and its role in the internet of things. Proceedings of the first edition of the ACM MCC workshop on Mobile cloud computing, 2012.

[13] European Telecommunications Standards Institute (ETSI), "Mobile- edge computing initiative," 2016, available at: http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing.

[14] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central office re-architected as a data center (cord)," IEEE Communications Magazine, vol. 54, no. 10, pp. 96–101, 2016.

[15] M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: Distributed edge cloud for data intensive computing," in Cloud Engineering (IC2E), 2014 IEEE International Conference on. IEEE, 2014, pp. 57–66.

[16] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: Leveraging mobile devices to provide cloud service at the edge," in 2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015, pp. 9–16.

[17] 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 Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 49–62.

[18] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in Proceedings of the sixth conference on Computer systems. ACM, 2011, pp. 301–314.

[19] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govin- dan, "Odessa: enabling interactive perception applications on mobile devices," in Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM, 2011, pp. 43–56.

[20] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen, "COMET: code offload by migrating execution transparently," in Pre- sented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12), 2012, pp. 93–106.

[21] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

[22] J. Pan, L. Ma, R. Ravindran, and P. TalebiFard, "Homecloud: An edge cloud framework and testbed for new application delivery," in Telecommunications (ICT), 2016 23rd International Conference on. IEEE, 2016, pp. 1–6.

[23] Network Softwarization Group, International Telecommunication Union (ITU) standardization sector, "Draft deliverable of network softwariza- tion for imt-2020 networks, imt-i-096r1," October 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2767608, IEEE Internet of Things Journal

[24] Open Networking Foundation (ONF), “Northbound interfaces work- ing group charter,” 2015, available at: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf.

[25] Wikipedia, “Northbound interface – wikipedia, the free encyclo- pedia,” 2014, available at: https://en.wikipedia.org/w/index.php?title=Northbound_interface&oldid=636148667.

[26] R. Ranjan, B. Benatallah, S. Dustdar, and M. P Papazoglou, “Cloud resource orchestration programming: Overview, issues, and directions. IEEE Internet Computing, 19(5):46–56, 2015.

[27] J. Wang, J. Pan and F. Esposito, "Elastic Urban Video Surveillance System Using Edge Computing," ACM Workshop on Smart Internet of Things 2017 (SmartIoT 2017), San Jose, CA, October 14, 2017.

Jianli Pan is currently an Assistant Professor in the Department of Mathematics and Computer Science at the University of Missouri, St. Louis. He obtained his Ph.D. degree from the Department of Computer Science and Engineering of Washington University in St. Louis. He also holds a M.S. degree in Computer Engineering from Washington University in Saint Louis and a M.S. degree in Information Engineering from Beijing University of Posts and Telecommunications (BUPT), China. He is currently the associate editor for IEEE Communication Magazine and IEEE Access. His current research interests include edge clouds, Internet of Things (IoT), Cybersecurity, Network Function Virtualization (NFV), and smart energy.

James McElhannon is a Ph.D. student with the Department of Mathematics and Computer Science at the University of Missouri, St. Louis. He holds an M.S. in Computer Science from the University of Illinois-Springfield, an MBA from Webster University, and a B.S. in Computer Science from Webster University. His current research interests include Network Function Virtualization (NFV) and Internet of Things.