How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions

Ahmet Cihat Baktir, *Student Member, IEEE*, Atay Ozgovde, *Member, IEEE*, and Cem Ersoy, *Senior Member, IEEE*

***Abstract*—A novel paradigm that changes the scene for the modern communication and computation systems is the Edge Computing. It is not a coincidence that terms like Mobile Cloud Computing, Cloudlets, Fog Computing, and Mobile-Edge Computing are gaining popularity both in academia and industry. In this paper, we embrace all these terms under the umbrella concept of "Edge Computing" to name the trend where computational infrastructures hence the services themselves are getting closer to the end user. However, we observe that bringing computational infrastructures to the proximity of the user does not magically solve all technical challenges. Moreover, it creates complexities of its own when not carefully handled. In this paper, these challenges are discussed in depth and categorically analyzed. As a solution direction, we propose that another major trend in networking, namely software-defined networking (SDN), should be taken into account. SDN, which is not proposed specifically for Edge Computing, can in fact serve as an enabler to lower the complexity barriers involved and let the real potential of Edge Computing be achieved. To fully demonstrate our ideas, initially, we put forward a clear collaboration model for the SDN-Edge Computing interaction through practical architectures and show that SDN related mechanisms can feasibly operate within the Edge Computing infrastructures. Then, we provide a detailed survey of the approaches that comprise the Edge Computing domain. A comparative discussion elaborates on where these technologies meet as well as how they differ. Later, we discuss the capabilities of SDN and align them with the technical shortcomings of Edge Computing implementations. We thoroughly investigate the possible modes of operation and interaction between the aforementioned technologies in all directions and technically deduce a set of "Benefit Areas" which is discussed in detail. Lastly, as SDN is an evolving technology, we give the future directions for enhancing the SDN development so that it can take this collaboration to a further level.**

***Index Terms*—Software-defined networking, edge computing, cloud computing, network virtualization, network management, service-centric network, Internet of Things, wearable computing.**

Manuscript received October 27, 2016; revised March 29, 2017 and May 17, 2017; accepted June 15, 2017. Date of publication June 20, 2017; date of current version November 21, 2017. *(Corresponding author: Ahmet Cihat Baktir.)*

A. C. Baktir and C. Ersoy are with the NETLAB, Computer Networks Research Laboratory, Department of Computer Engineering, Bogazici University, 34342 Istanbul, Turkey (e-mail: cihat.baktir@boun.edu.tr; ersoy@boun.edu.tr).

A. Ozgovde is with the Department of Computer Engineering, Galatasaray University, 34349 Istanbul, Turkey (e-mail: aozgovde@gsu.edu.tr).

Digital Object Identifier 10.1109/COMST.2017.2717482

I. Introduction

The domain of ubiquitous computing has gone a long way and the Mark Weiser's vision of "computation being integrated into the fabric of our daily lives" is approaching to be a reality [1]. One trend is that the mainstream computational device for the casual user has become the smartphone itself. The new edge devices are not just the old desktops that have a small form factor but they have a lot more: intermittent and bursty traffic with geospatial distribution, social networks with audio visual contents, and a large set of multimodal sensors are changing the scene for the new smartphone applications.

A. The Journey to Edge Computing

Either accompanied by the smartphones or in a standalone mode, a large variety of commercially available wearable gadgets also contribute to make the ubiquitous computing vision a reality. A new edge device like a Smart Glass, Smart Watch, Smart Bracelet and even Smart Plaster is appearing in the market. The evolution of the personal devices as the computational resources over the last years is illustrated in Figure 1. These new devices continuously log data, implement various services and pump intermittent audio-visual data traffic into the network. Yet, there exists a separate category under the umbrellas of Internet of Things (IoT) and Machine-to-Machine (M2M) Communications where machines themselves participate in various services. IEEE [2], AllSeen Alliance [3], Thread Group [4], Open Interconnect Consortium [5], and many others have a large set of standards for IoT [6]. Accompanying the ongoing research activities, great effort is put into developing real applications with the help of available standards bodies and specifications.

Combining all these, we see a pervasive computing infrastructure with a network of multimodal, multi-dimensional data sources, dispersed geospatially, potentially offering a wide range of novel services. One problem, however, is about how to implement complicated services for the envisioned novel use cases via mobile, tiny and gadget-like nodes that are computationally restricted. A straightforward alternative is to enhance the computational capabilities of these edge nodes using the centralized cloud computing resources. However, despite their "unlimited" computational capacities, legacy cloud computing infrastructures cannot be the remedy for solving all problems of these edge nodes due to the inherent latency constraints of the Wide Area Network (WAN) used

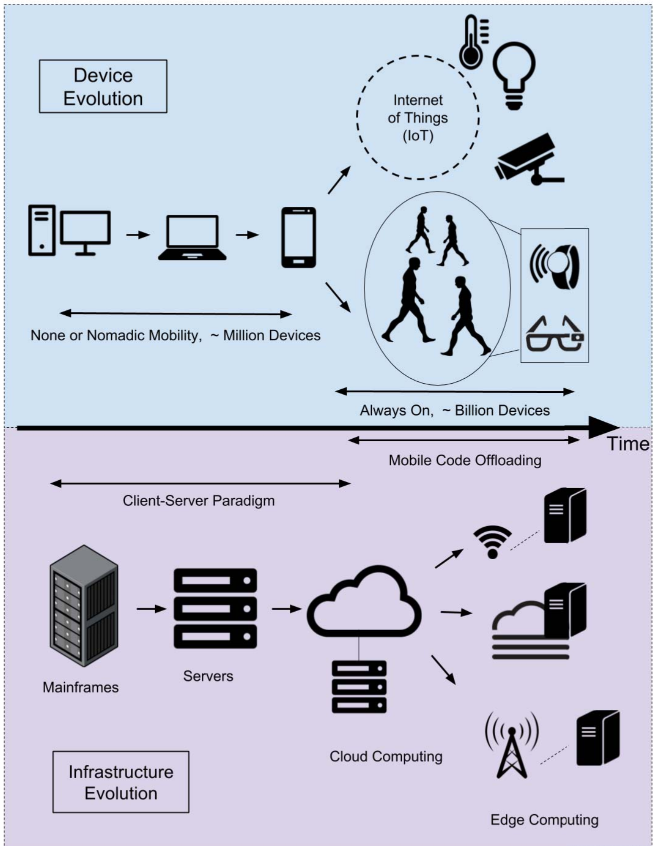


Fig. 1. The evolution of the end-user devices and servers for computing.

for accessing the cloud services [7]. Since a large variety of the new edge services have real-time requirements or interactive behavior with high Quality of Service (QoS) expectations, this drawback cannot be neglected. When the limitations of the public Internet connection are taken into consideration, another technical alternative is bringing the "Cloud" closer to these devices. Lately, we witness a relevant trend in ubiquitous computing called Edge Computing where the computational resources are being brought nearer to the end user.

Proliferation of the edge devices creates a necessity for the applications to process at least some of the data at the edge rather than carrying them to the remote data centers for minimizing not only the service delay but also the energy consumption. Recently, there have been many proposals for the operation and architectural design of the Edge Computing systems. It is no coincidence that the terms Mobile Cloud Computing (MCC) [8], Cloudlet [9], Fog Computing [10], Edge Computing [11], Mobile-Edge Computing (MEC) [12] (by March 2017, it is named as Multi-access Edge Computing [13]) and Mist Computing [14] are all hot topics in the literature. All these proposals define various practical implementations for the Edge Computing. When carefully inspected, these approaches have common grounds but differ and specialize in their targeted use cases.

B. The Technical Challenges of Edge Computing

It is important to note that augmenting the computational capacities of edge devices through Edge Computing is not a replacement for cloud, on the contrary they are complementary paradigms that need to be employed together. However, it is not a straightforward attempt to deploy extra computational resources at the edge of the network and establish this complementarity while letting them be accessible ubiquitously. There are many significant challenges in addition to the complexity of the technologies such as the mobility, limited energy and computational resources of the edge devices, heterogeneity, scalability, reliability, security and privacy issues. Some of these technical challenges are already being addressed in [8] and [15].

However, even in the case where each technical challenge is resolved, one meta-challenge is that the resulting system and its internal interactions are very complex. Some of the reasons creating this complexity are as follows:

1) Service Synchronization & Orchestration: Typical client-server style interaction assumed for the cloud services are based on two tiers. However, Edge Computing resources will also need to interact with the cloud servers. This will entail a minimum of three tier architecture with its own set of coordination and orchestration requirements. An intermediate networking layer should orchestrate the interaction between edge servers and the central cloud, and inter-Cloudlet communications. Moreover, even intra-Cloudlet operations should be handled properly with this networking layer in order to provide a smooth cooperation throughout the edge tier.

2) Seamless Service Delivery: The connectivity at the edge computing infrastructure may be intermittent with mobility. In this respect, to achieve seamless service delivery, handover mechanisms that consider multi-tenancy on the same local cloud and multiple service providers will be required.

3) Service-Centric Structure: As the focus shifts to the service itself rather than its location, traditional IP-based operations become infeasible to handle the interactions between clients and servers. In an Edge Computing facility, this problem is more emphasized as the service itself may reside on a number of local servers as well as it may partially reside on local servers and the cloud. A service-centric design that handles all the complexities involved is a necessity.

4) Soft State: Unlike its cloud counterpart which operates on hard state and permanent data, Edge Computing services may not always assume the availability of the local infrastructure. Soft state incurs much more complex scenarios and should handle the fall-back operations for the end user.

Even the partial list above indicates that there exists a multi-faceted technical challenge that is difficult to solve using the classical distributed system (client-server) paradigm. Developing, provisioning and configuring new applications will be challenging through that approach. This will create a barrier for the practical deployments of the novel Edge Computing solutions.

C. Software-Defined Networking (SDN) As an Edge Solution

In order to realize the envisioned pervasive computing scenarios, we must come up with a solution that hides all internal complications from the users, especially from the application developers and the service providers. To this end, Software Defined Networking (SDN) with its network programming capabilities stands out as a natural candidate for orchestrating the network, the services and the devices by hiding the complexities of this heterogeneous environment from the end users.

SDN is the most promising proposal so far for the programmable networks which separates the control from data plane and enables the programmable control mechanism [16]. It can easily simplify the management of the network, define network flows, increase network capability and facilitate virtualization within the network. The control mechanism that is provided by SDN can lower the complexity of the Edge Computing architectures and implementations by bringing a novel approach to the networking and utilizing the available resources in a more efficient manner. In a system that incorporates edge servers into the traditional cloud data center, the traffic originated at the edge can be dynamically routed to the tier and server that may provide the highest quality service to the user by using the available SDN mechanisms. Since the SDN paradigm concentrates the network intelligence at the central software-based controller, it will relieve the relatively simpler edge devices from executing the complex networking activities such as service discovery and orchestration.

When we gather all these features and advances together, we see that a multi-tier Edge Computing infrastructure that is managed by SDN has a significant potential for mitigating the barriers and restrictions that edge devices encounter. It has the capability to meet the QoS requirements such as performance and delay, and improve the user satisfaction [17].

There are recent surveys on cloud computing [18], [19] and MCC [8], [20]–[22]. Most of these works focus on accessing the traditional cloud services over mobile devices and do not cover this area from a broad perspective and hence, managing the novel services and orchestrating the dynamic environment are not addressed so far. In the context of Edge Computing, there are surveys for Cloudlets [23], Fog Computing [24] and Mobile-Edge Computing [25]. However, these studies remain incapable to discuss the common Edge Computing proposals together by focusing on their requirements and differences among them, and depict the general view over Edge Computing concept.

Moreover, there are surveys on SDN [26]–[30], Software Defined Wireless Networks (SDWN) [31], [32] and Network Functions Virtualization (NFV) [33], [34] which are the complementary technologies for improving the operations of edge servers. Since SDN is still a developing technology, applying SDN to cloud or Edge Computing has not been addressed in depth yet. Additionally, the discussions in the literature are still inadequate to assign the fully defined role of NFV in this envisioned architecture.

In this paper, we survey the Edge Computing and SDN technologies by supporting the possible cooperation with use case scenarios. On the other hand, the missing points that are not referenced so far for the integration of the Edge Computing and SDN are studied and extracted for designating the future direction.

As a summary, the original contributions of this paper are:

- Illustrating the potential of multi-tier Edge Computing architecture managed by SDN through discussing the possible use cases and scenarios
- Surveying the Edge Computing proposals in detail and discussing them together to outline the general concept, targeted use cases and differences among them,
- Surveying the SDN technology and presenting a set of features that Edge Computing can benefit from with supportive use cases,
- Introducing a future direction that focus on the missing points in SDN that needs to be studied in detail to make more contributions to Edge Computing.

The remainder of this paper is organized as follows. Section II introduces the concept of the cooperation of Edge Computing and SDN technologies with example use case scenarios. Section III surveys and presents the Edge Computing paradigm and its proposals in detail and discusses the common properties and key differences between these prominent technologies. Section IV presents the targeted use cases that are proliferated by the Edge Computing. Section V initially presents the SDN paradigm and OpenFlow with their features and then goes into the details of SDN by discussing its capabilities to extract how SDN can help to facilitate Edge Computing. Section VI presents the directions to enhance the functionality of SDN-Edge Computing collaboration by discussing the missing parts and immature pieces. Section VII summarizes the contributions, gives future directions and concludes the study.

II. SDN-Edge Computing Cooperation

The complexities resulted by deploying the cloud-like resources and related services at the edge of the network can be solved by a control mechanism that is able to orchestrate the distributed environment. The benefits of programmable networks align with all these requirements and the recent form of SDN has the inherent capability to mitigate the barriers that prevent Edge Computing to reach its full potential. All data flow management, service orchestration and other management tasks are accomplished by the central SDN controller that is transparent to the end-user.

Before getting into the technical details of the SDN paradigm, examining the real-life scenarios in detail helps to clarify the operations of the SDN-enabled Edge Computing systems. In this respect, different areas of use and possible scenarios are discussed from the perspective of SDN.

A model for service orchestration through SDN and its components are illustrated in Figure 2. In this setup, Cloudlets are employed as the underlying Edge Computing technology. Cloudlets in different physical locations are connected via OpenFlow-enabled switches and provide access to their users through a WLAN. These switches are controlled by an SDN controller through OpenFlow messages and other SDN management protocols such as OF-Config [35].

The controller hosts a variety of SDN northbound applications to execute the necessary functionalities and present the

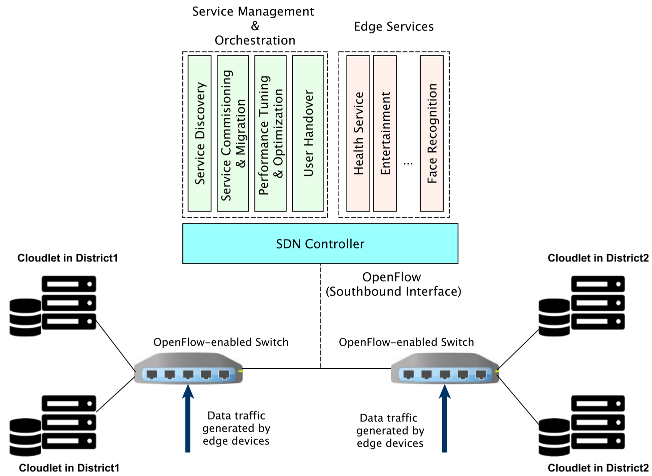


Fig. 2. Orchestrating Edge Computing facilities using SDN.

fundamental behavior for realizing the overall service management and orchestration. Another set of applications are directly related to the edge services themselves, such as health services, entertainment and video processing.

Each of these applications, whether they belong to an edge service or the orchestration functionality, communicate with the SDN controller through the northbound API and generate commands in reply to an incoming event forwarded from switches to the controller. These high-level commands are then compiled and translated into low-level OpenFlow messages by the controller to be forwarded to the switches and realize the functionality there. Possible modules in an edge service management and orchestration application are Service Discovery, Service Commissioning & Migration, Performance Tuning & Optimization and User Handover. In a practical implementation, the functionality attributed to the service management and orchestration can follow a different modularization path.

a) Service discovery: It is quite possible that end user devices are specialized for a set of functionalities and request different types of services from the network. However, the user and the application that runs on the mobile device do not have any prior knowledge of the available services on the Edge Computing system. For instance, there should be an environment where a client application may generate a service request by specifying the necessary computation power and the required storage area [36]. Then, a broker mechanism discovers the service and servers that are able to satisfy the aforementioned requirements.

SDN environment leverages the similar functionality where the Service Discovery module maintains the necessary information about the existence and location of the service content available on the Edge Computing facilities. The module contains a mapping table which links the service name with the corresponding server locations. This table is updated frequently since each edge server informs the Service Discovery application about the provided services periodically and new service registrations.

b) Service commissioning & migration: This module is responsible for commissioning a service onto a Cloudlet. Since each Cloudlet has only a finite computational capacity which is already preoccupied with a set of services, it may not be possible to just initiate a service on the most nearby Cloudlet. This module decides where to commission the service based on various performance factors such as server utilization and network conditions. Also, whenever a service has low utilization, this management module can decide to migrate the virtual machine (VM) that hosts this service to another Cloudlet on the network assuming other conditions are met. At this point, the space required for the deployment of the new service will be provided which contributes to an increase in the overall performance.

c) Performance tuning & optimization: Performance of an Edge Computing system as experienced by the users should always be at the maximum attainable level. This requirement comes from the fact that most of the end-users utilize the Edge Computing facilities due to real-time requirements or related performance criteria [37], [38]. Performance Tuning & Optimization module make use of SDN and OpenFlow capabilities to monitor the service utilization levels and flow sizes to manage the load on various edge servers, and hence the overall throughput of the system is maximized.

There are proposals that utilize the functionalities of SDN and OpenFlow to carry out the load balancing procedures among network resources [39], [40], computation resources [41], [42], or both of them [43], [44]. In fact, a load balancer should take both aspects into consideration for optimizing the overall performance of the system. The joint optimization problem can be solved through either a single northbound application that is able to assemble the loads on network and computation resources, or separate inter-operable applications.

d) User handover: Due to mobility, an edge device can leave the coverage of a Cloudlet and enter the proximity of another one. Without any handover management involved, the user starts over with service discovery and service commissioning procedures whenever the active Cloudlet is changed. This would not only cause performance disruptions as experienced by the end-user but also brings in inefficiencies for the operations of the Edge Computing infrastructure. User Handover module can use related techniques to forecast the next coverage domain and can employ various methods to supply service continuity to the user such as flow redirection or instant live migration of the relevant virtualized resource through SDN [45].

The VM migration plays an important role in the real-time access to the servers [46]. Study made by Secci *et al.* [47] proposes a mechanism which links the VM mobility with the user mobility. As the user moves and the distance between the corresponding service/server increases, the performance of the system and the user experience are deteriorated. Therefore, Secci *et al*. propose the Protocol Architecture for Cloud Access Optimization (PACAO) which determines the best location for migrating the service to compensate the user mobility. Although PACAO is originally designed for cloud datacenters, it is also applicable for the VM migration at the edge of the network.

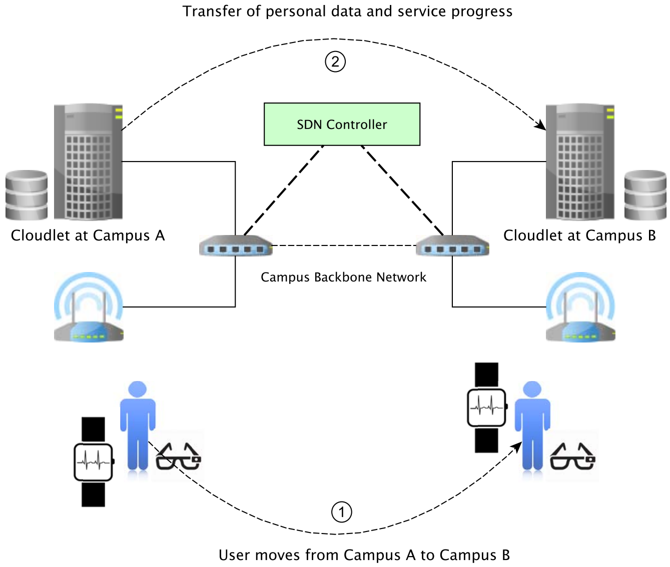


Fig. 3. Mobility and application layer handover scenario.

Apart from these, additional functionalities may be integrated into the system in order to supply further control and management procedures. As an example, firewall and other security implementations can be important constituents for this framework since multiple tenants share the same infrastructure where private data may be stored on the nodes and forwarded over the network. As the network gets larger, these applications can fall short of satisfying the expectations. Therefore, flexibility is enabled by virtualizing these network functionalities as separate northbound applications and deploying them over distinct servers. If there is any necessity for a network function or higher performance at a certain location, the application can be replicated over a newly created virtual machine instantly without the need for installing a specialized hardware for serving the corresponding functionality.

On the other hand, the SDN control mechanism should also support the mobility with the application layer handover tasks for maintaining the orchestration processes without any interruption.

Considering a campus-based scenario, students of a university with more than one campus probably have to visit each of the campuses in a day or even in a shorter period. In such an environment, the mobility and handover scenarios are inevitable. Figure 3 highlights a possible handover scenario between the campuses. After the user generates a service request at Campus A, it may visit the other campus before the Cloudlet at the former location completes the execution of the procedures. During this change in location, it may disconnect from the network and establish a new connection at the recent position.

For this case, the SDN controller can track the changes and acquire the recently assigned IP address for the user. Since the controller is aware that the user is in the same network with a Cloudlet but its latest request is forwarded to another one, it may trigger a handover at the application layer for handling the subsequent requests more effectively. When the decision is taken, the source and destination servers are informed. The former Cloudlet initializes the process of transferring the service code that is to be executed, the recent memory state and the personal database. At this step, the SDN controller has the responsibility of adjusting the data transfer. The important point to note is that the handover process should not deteriorate the performance and user experience. Therefore, the controller should determine the most feasible path between the two Cloudlets that is able to minimize the duration of data transfer. By sending *OFPMP_PORT_STATS* messages, which is defined by OpenFlow, to the switches, the controller can determine the least loaded links and paths. Lastly, the flow rules that define the path are installed on the nodes and the data transfer is progressed. If the same user requests for the same service at Campus B in the future, the overall delay is minimized because it can get the service from the nearest Cloudlet.

By hiding all the operational tasks from the user and remove the burden of locating the service from the user application, the service-centric model implemented at the edge is helpful for maximizing the user experience. Deployment of SDN-enabled switches and implementation of northbound applications with an SDN controller may enable these functionalities without modifying or reshaping the entire network infrastructure and protocol stack.

In addition to these functionalities that SDN provides for enabling novel use cases, the detailed examination of the SDN-enabled edge scenarios is discussed further in Sections V-C–V-E.

III. Technological Panorama of Edge Computing

Edge Computing is an umbrella concept that covers a range of practical schemes for its implementation. This section initially discusses MCC since it forms the basic mechanism for mobile computing and code offloading. Then, following a chronological order Cloudlets, Fog Computing and Mobile-Edge Computing (MEC) approaches are surveyed. While implementing the same principles, these approaches have differences in various aspects. These differences, which deserves further attention for understanding the exact vision of the Edge Computing, are also inspected and discussed.

A. Edge Computing Paradigm

During the last decade, cloud computing drew considerable attention especially in the context of enterprise IT infrastructure as it offers lower cost solutions for fluctuating and unforeseen computational demand [48]. However, recently we see a much wider range of devices than the mainstream servers and desktop computers whose functionality the cloud computing aims to augment. On one hand, the wide adoption of wearable technology, on the other hand, strong influence of IoT as an enabler for ambient intelligence and smart environments change the scenery for the spectrum of computational devices that are used at the edge.

As a result of the trend towards wearable devices, smart home and IoT, we can see that there are several proposals by different groups in order to overcome the related issues

TABLE I
DIFFERENCES BETWEEN CLOUD AND EDGE COMPUTING

Requirements/Features	Cloud Computing	Edge Computing
Latency	High	Low
Network Access Type	Mostly WAN	LAN(WLAN)
Server Location	Anywhere within the network	At the edge
Mobility Support	Low	High
Distribution	Centralized	Distributed
Task/Application Needs	Higher computation power	Lower latency
User Device	Computers, mobile devices (limited)	Mobile-smart-wearable devices
Management	Service Provider	Local Business
Number of Servers	High	Low
State	Soft and hard state	Soft state

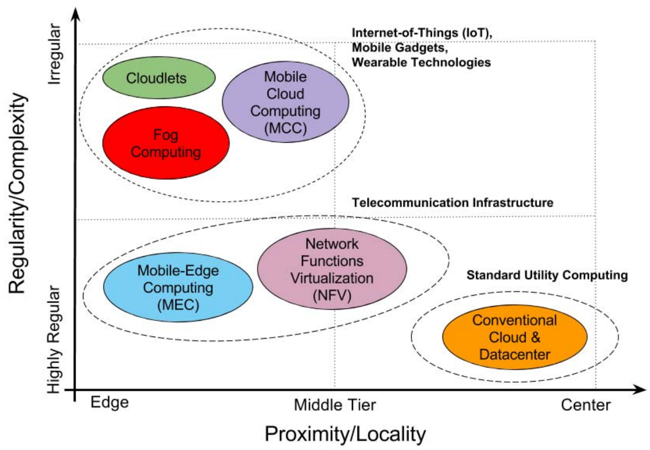


Fig. 4. Comparison among the Edge Computing spectrum including cloud computing.

of these technologies. These proposals are covered under the umbrella of a new paradigm: Edge Computing.

Although intended for different parts of the overall network, Edge Computing and cloud computing are interrelated. Analyzing the features of various Edge Computing proposals [49], [50], a comparison between cloud and Edge Computing technologies is depicted in Table I.

Figure 4 compares the various approaches for providing the necessary computation power to the users at the edge. Conventional datacenter design is highly regular with identical servers and networking hardware aligned in a grid-like fashion. MEC, although being at the edge of the network, still depicts a more rigid and well-defined structure than other Edge Computing proposals. The main reason for this behavior of MEC is due to its envisioned existence in a telecommunications infrastructure which is inherently regulated. Functionality served over MEC will not be some individual services locally available to the edge users but instead will be highly controlled and orchestrated in accordance with the overall state of the 5G network.

When compared with MEC, Cloudlets and Fog Computing solutions have less stringent constraints in terms of hardware and application execution model. A Cloudlet hardware can be a micro-sized server in a coffee-shop where WLAN is available for users to carry out code offloading. Fog servers can be co-located on a networking device to handle the IoT traffic at the edge. In that respect, Cloudlet and Fog have much wider design spaces allowing irregularity.

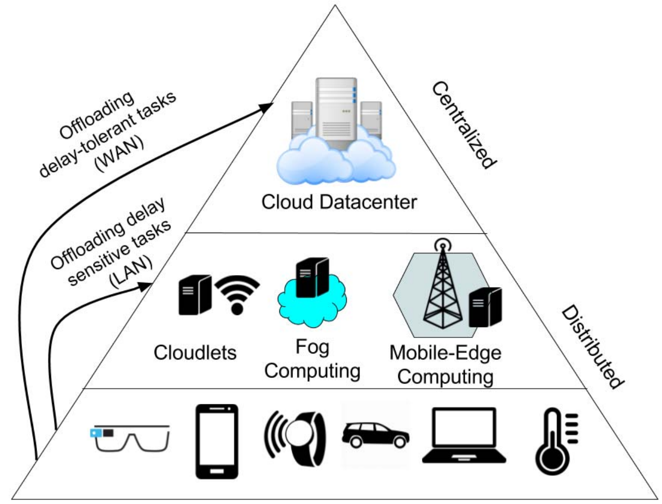


Fig. 5. The possible cooperation of edge technologies and cloud computing.

The utilization of various Edge Computing proposals and traditional cloud servers is illustrated in Figure 5. The smart devices, vehicles and IoT-related appliances can offload tasks to the edge servers which are reachable at one hop, through different access technologies. At the same time, a subset of requests can be directly forwarded to the traditional cloud datacenters through WAN or the edge servers may operate as an intermediate computation layer for pre-processing the offloaded tasks.

B. The Need for Edge Computing

The previous subsection introduces the Edge Computing paradigm by discussing how traditional cloud servers fall short of satisfying the requirements of the novel use cases. The centralized form of remote computing resources is not compatible with enormous traffic originated from geographically distributed edge devices. Hence, pushing the servers to the edge of the network is not an unnecessary trend, in fact it becomes inevitable. The importance and necessity of Edge Computing comes from a set of factors that drive this evolution. These factors need to be analyzed separately from the viewpoint of end-users and operators.

The causes that lead to the emergence of Edge Computing are discussed in detail in order to form a basis for the edge server proposals.

1) Real-Time QoS & Delay Sensitiveness: Although these edge devices are as powerful as they have never been so far, most of them still lack enough capacity for accomplishing real-time use cases with the pre-defined QoS requirements. Cloud computing is acknowledged as a rescuer technology for limited-capacity devices and machines by providing a pool of computation and storage infrastructure. However, the wearable devices and IoT are designed for use cases that are delay sensitive. Since most of these devices demand high QoS requirements because of the mobility, interactive environment and real-time requirements, legacy cloud servers cannot be the sole solution because of the delay resulting while accessing them through the WAN.

The rise of IoT and its domains, such as healthcare, necessitates high QoS requirements [51]. Proliferating the real-time interaction requires external computation resources at the edge of the network. For instance, a healthcare data generated by the body worn sensors needs to be processed immediately in case of an emergency. As another example, the data generated by the cameras integrated into autonomous vehicles needs to be processed in real-time to determine the necessary driving action in an instant [52]. Due to the limited Internet bandwidth and WAN delay, remote cloud servers deteriorate the user experience. By deploying servers closer to the devices that require real-time interaction, the overall latency can be decreased through high LAN bandwidth and decreased number of hops [37].

A Cloudlet-based offloading scheme for IoT devices proposed by Shukla and Munir [53] shows that utilization of Cloudlets for processing the IoT data provides an improvement in decreasing the latency compared to the cloud servers.

2) Battery Lifetime: Energy consumption is one of the most important parameters when mobile devices are considered [54]. Although smart phone's processing capabilities are improving steadily, their battery lifetime is not improving at the desired rate.

One of the main objectives of task offloading is decreasing the energy consumption. The results of the related studies show that offloading reduces the total energy consumption [55]–[57]. Offloading can be achieved through two different approaches: (1) cloud servers and (2) edge servers. Although the offloading operations inherently decrease the energy consumption, utilization of edge servers helps to decrease it further. Ha *et al.* [58] analyzes energy consumption rates for applications such as face recognition and augmented reality. It is stated that offloading tasks to the edge servers results in lower energy consumption when it is compared to the cloud servers. Unsurprisingly, executing these applications on the device itself leads to the highest energy consumption among all methodologies.

As an example case, ME-VoLTE [59] is a video telephony system that is based on Mobile-Edge Computing (MEC) servers. The main objective of this system is reducing the energy consumption by offloading the encoding operations to a nearby MEC server. According to the evaluations, the total energy consumption of a user equipment is decreased by 13%.

The energy conservation becomes more critical in the case of wearable devices and IoT gadgets. Since most of the IoT devices have lower energy capacity than the smart phones, offloading the task from an IoT device to an edge server is an energy efficient operation [52]. Therefore, it can be concluded that Edge Computing is necessary for conserving the battery lifetime while executing the latency-intolerant applications.

3) Regulating Core Network Traffic: The limited bandwidth of the core network makes it vulnerable to the congestion. In 2015, approximately 97 million wearable devices generated 15 pentabytes of traffic per month and this rate is expected to increase [60]. As a result, operators face the difficulties in managing the cumulative data traffic with varying sizes and characteristics.

In the traditional approach, the traffic generated by the edge devices flows through the core network to access cloud servers. If this traffic is kept at the edge, the burden on the core network is relieved and the utilization of the bandwidth is optimized [61]. This shift in networking prevents the consumption of the limited bandwidth of the core network by billions of devices at the edge. Therefore, the traffic that core network is responsible for becomes manageable in size and the operations are simplified.

Not only the network operators, but also the cloud service providers face the same challenge. For instance, if the data generated by IoT sensors such as video cameras is processed at the edge of the network, the demand for computational resources at the cloud datacenters gets lower [62]. Hence, Edge Computing solves the problem of congestion within both the core network and datacenters, and the traffic can be regulated with less effort.

4) Scalability: The number of end user devices is expected to reach to trillions in a few years and this evolution creates a significant scalability problem [63]. In order to support the dynamic demands that may change in an instant, the cloud can be scaled accordingly [64]. However, sending tremendous volumes of data to cloud servers create a congestion within the datacenters [65]. The changing characteristics and tremendous amount of the data traffic generated by IoT and wearable devices make the operators' duties more difficult. With these rates, the centralized structure of cloud computing fall short of providing a scalable environment for the data and applications.

Instead, distributing the services and applications and replicating them in the form of virtual machines (VMs) over edge servers create an opportunity of enhancing the scalability [66]. If an edge server becomes congested and fail to satisfy the incoming requests, the corresponding service can be replicated over another edge server in the vicinity and let the further requests to be handled there. On the other hand, preprocessing data at the edge of the network and forwarding smaller size traffic to the cloud servers relieve the burden of scalability on the cloud [67].

C. Mobile Cloud Computing

Mobile Cloud Computing (MCC) has several definitions in the literature but none of them provide a clear and complete characterization of the terminology involved. In order to have a better understanding of MCC, legacy cloud computing needs to be revisited.

Cloud computing is a service model where computing services are delivered over a network on demand independently from device type and location [68]. It facilitates offloading the tasks ubiquitously from a local computer to a cloud server. Not long after the emergence of cloud computing, due to proliferation of the mobile devices, mobile computing terminology has emerged to offer not only the mentioned benefits to mobile users but also specific services.

TABLE II
SUMMARY OF MCC TYPES

Research	Cloud-based MCC	Ad-hoc MCC	Cloudlet-based MCC
[8], [70], [15]	✓	✗	✗
[20], [71], [72]	✓	✓	✗
[69], [73], [74]	✓	✗	✓
[55]	✓	✓	✓
[75]	✗	✓	✗

MCC is an extension of traditional cloud that covers the offloading process for the mobile devices by combining mobile computing, mobile Internet and cloud computing into a joint system [69]. Because of the limited resources, offloading the tasks to the cloud datacenters empowers the operations of the mobile devices.

If we go back to the definition and technical organization of MCC, we see that it appears in different forms. In [20], the mobile cloud is defined in two different ways:

- Infrastructure based
- Ad-hoc mobile cloud

According to this classification, the infrastructure based model specifies the hardware infrastructure of cloud datacenters serving to mobile users. On the other hand, ad-hoc mobile cloud means that a group of not far located mobile devices act as a cloud and provide access to the Internet or cloud services for other mobile devices. Although the early definitions of the infrastructure based MCC gather around the traditional cloud servers, the "cloud" in the definitions of MCC does not have be a powerful server at a remote location [76].

The important point in MCC is providing a variety of services for mobile devices with powerful computational and storage resources. This objective can be achieved through bringing the resources of cloud servers to the proximity of the end-user devices, in addition to the traditional approach that covers the cloud datacenters [55]. This advance offers fast network connection in LAN, cloud-like resources at the edge, and battery saving for the devices by alternating the ad-hoc communication. All these contributions made by edge servers enhance the functionality and the performance of MCC environments. In the context of utilizing edge servers for code offloading, Cloudlet is considered as one of the major enabler technologies for MCC [77]. When compared to initial proposals for MCC, Cloudlet enables the use cases that cannot be achieved so far with the cloud-based or mobile ad-hoc-based solutions because each of them addresses only a subset of the requirements.

The research on MCC and the technical infrastructures that are utilized for task offloading are summarized in Table II. Since our envisioned multi-tier Edge Computing architecture that covers cloud and edge servers does not rely on ad-hoc computing, the reference definition and offloading processes are based on the joint cloud/Cloudlet-based MCC environment.

After a discussion on the definition of MCC, it is important to focus on its technical details. The main motivations of MCC are [78]:

- Extending the battery life-time
- Diverse application services
- Overcoming resource shortage issues in mobile devices
- Virtualization

TABLE III
FEATURES OF MCC AND COMPARISON WITH TRADITIONAL CLOUD

Feature	MCC	Cloud Computing
Conserving Energy	✓	✗
Mobility Management	✓	✗
Decreasing the Execution Time of a Task	✓	✓
Enhancing Storage Capacity	✓	✓
Reliability	✓	✓
Scalability	✓	✓
Easy and On-demand Access	✓	✓
Multi-tenancy	✓	✓

The key idea of MCC is offloading a task from a mobile device to traditional cloud servers or Cloudlets. In either way, MCC becomes a tool to mitigate the computing and storage limitations of the mobile device [57]. In addition to the computation and storage capacities, mobile devices are also restricted in terms of energy capacity. Thus, MCC aims conserve energy resources and eliminate the factors that limit the mobility such as being constantly connected to electricity [79].

On the other hand, there are benefits inherited from legacy cloud computing such as multi-tenancy, ease of integration, dynamic provisioning and scalability [8]. The most advantageous aspects of MCC are summarized in Table III.

D. Cloudlet

The convergence of MCC and traditional cloud computing introduced an architecture called Cloudlets [80]. One of the initial proposals [9] defines the Cloudlet as a computational resource accessible by mobile users in their physical vicinity for making use of the services provided. It is a virtualized architecture that is designed to empower mobile devices for keeping up with the recent technological capabilities such as speech recognition, natural language processing, machine learning and augmented reality.

At the time when Cloudlet was initially proposed, smart phones were the common devices of the end-users for executing the desired daily life tasks. Although smart phones get much more powerful recent improvements, alternative mobile devices showed up such as wearable devices and IoT gadgets which are more specialized but still weak in terms of computation power. As stated by Satyanarayanan *et al.* [81], these devices are still being improved but they will always be weaker in terms of resources when they are compared with servers. The tasks that are meant to be executed by today's mobile devices not only demand high computation and memory capacity but also require low latency. These tasks can be offloaded to the Cloudlet instead of a remote cloud server because although they are not as powerful as cloud data centers, they can provide the computation power enough for meeting the requirements with a low response time. This is the exact reason why Cloudlets are sometimes referred as "smaller cloud". They are located as an intermediate layer between the traditional cloud infrastructure and mobile devices. This presence produces an opportunity for Cloudlets to operate collaboratively with each other and with the cloud tier for more centralized services.

Cloudlets are considered as an enabling technology of MCC and it is proposed for overcoming the problems caused by accessing to the cloud data centers such as latency and cost. Since a Cloudlet is accessible at one-hop and can operate without Internet connection, it promises several benefits for MCC over cloud-based solutions such as eliminating WAN latency, higher bandwidth, offline availability, cost effectiveness, and minimum management effort [82].

In addition to infrastructure based MCC, Cloudlets are also proposed to overcome the problems of ad-hoc MCC when there is no Internet connection to access cloud servers [83]. Accessing to a Cloudlet through a LAN or WLAN saves energy because they do not need to form an ad-hoc network to offload task to each other when there is not Internet connection. Moreover, deploying Cloudlets to replace ad-hoc mode provides better mobility and connectivity options to the devices [84].

The Cloudlet proposal also attract the interest of industry and the investment in this area is increasing [62]. The Open Edge Computing [85] initiative is formed in 2015 by the telecom operators in partnership with the academia. The main objective of this community is leveraging the Cloudlets. In this direction, there is an effort on the practical implementations by creating a testbed environment for the deployment of Cloudlet applications.

Although Cloudlets bring essential advantages for the users, there are important problems need to be tackled in order to reveal the exact potential of this approach. The planning phase of the Cloudlet network requires an intense effort so that the location of the server deployments, orchestration of the resources and assignment of the Cloudlets to the users can be optimized for providing a better service [86], [87]. Moreover, deployment of Cloudlets is still not widespread, most of the servers are personal and there is not any standardization so far. This challenging aspect brings up security and privacy issues as well as maintenance problems [82].

E. Fog Computing

According to Cisco [10], Fog Computing is compatible with the real-time requirements of IoT devices by bringing the infrastructure closer to the edge of the network. However, just like the case of Cloudlets, Fog Computing should not be considered to be totally independent of the cloud because coherent interaction is a necessary condition especially for data management. It is important to state that both cloud and Fog Computing provide services to end-users but there are key differences between them such as mobility supported by Fog Computing [88].

The promising growth of Fog Computing attracts the industry and OpenFog Consortium [89] was formed in November 2015 in order to solve the challenges which are encountered in

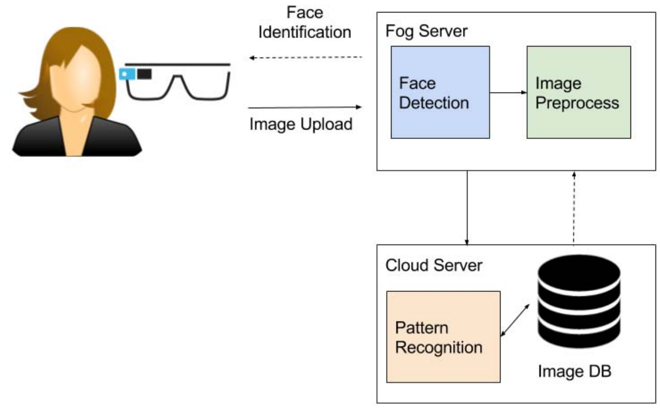


Fig. 6. Face identification with fog and cloud servers.

the real applications of IoT and Tactile Internet such as bandwidth and latency. The main objective of this consortium is forming a public-private Fog ecosystem that pave the road for building testbeds and improving the fog technology. OpenFog Consortium also publishes use case scenario documents in order to pave the road for the practical implementations of Fog Computing. One of the use case scenarios is related to the concept of using drone for enhancing the coverage in which Fog Computing is utilized for providing the high-speed computation and storage to enhance the control of the drone traffic [90]. This huge traffic can be a video stream and data generated by IoT gadgets that requires instant process.

As a result of all these efforts, there are commercially available products for Fog technology such as IOx and LocalGrid [65]. These advances indicate that the promised features are not just facilitated in theory but also in practice.

Fog Computing addresses the services and applications which are deployed in a distributed manner in contrast to the centralized cloud architecture [91]. Augmented reality and similar applications need to access the computational resources with the minimum latency and require real-time interaction [49]. As smart devices are getting more popular in daily life, cloud computing alone cannot satisfy the intrinsic requirements of these use cases and handle the mobility. The delay caused by accessing the WAN and the centralized structure, which hinders the ability to track the mobility of the end-users, make cloud-based solutions infeasible for IoT. In addition to the mobility support, Fog can explore user application demands and provide localized services accordingly [92].

Some of the use cases may require higher computational power that cannot be granted by the fog servers. Besides, there are many applications that require the localization of Fog and globalization of cloud such as the Smart Grid. Thus, it is important to regulate the interaction between cloud and Fog computing architectures to take advantage of both ends. One of the example use cases that utilize both fog and cloud servers is the face identification [93]. A face identification service is composed of three non-identical sub-services:

- Face detection
- Image preprocessing & feature extraction
- Pattern recognition

In addition to these service components, there should be a database system that stores the set of face images belonging to a corresponding user. For carrying out the face identification service with the highest performance, the sub-services and database can be distributed over fog and cloud servers.

Figure 6 shows an example case of sub-service deployments for providing the face identification service. When the service request by the user arrives to a fog server, with image as an input, the server executes the face detection and image preprocessing sub-services. Since feature extraction requires more computation power than the initial steps, the last task is offloaded by the fog server to the cloud server, where the image database resides. After the cloud server accomplishes the face identification, the response is forwarded to the user via the fog server.

F. Mobile-Edge Computing (MEC)

Facilitating cloud-like resources at the edge of the network is a hot topic also for the telecommunication sector. In 2014, ETSI (European Telecommunications Standards Institute) and contributing organizations began to develop a technology called Mobile-Edge Computing (MEC) which is well aligned with the vision put forward by Cloudlets [12], [94]. It brings the mobile operators, service providers, mobile subscribers and over-the-top (OTT) players together, and aims to provide a sustainable business model for them [25], [95].

MEC aims to provide cloud server capabilities within the Radio Access Network (RAN) in the vicinity of mobile subscribers. As other Edge Computing approaches aim to achieve, MEC leverages accelerated services, contents and applications by increasing responsiveness at the edge. The main motivation behind this technology is the growth of mobile traffic especially by the smart phones. According to the definition of ETSI, MEC is able to reduce latency and provide location-awareness which are the essential requirements for the operations of mobile devices.

In order to concentrate on the standardization processes more, an Industry Specification Group (ISG) within ETSI is organized. In January 2017, this group published a standard (GS MEC-IEG 006) that describes the performance metrics that is possible to be improved by deploying a service on a MEC environment [96]. These performance metrics are categorized under two different headings: (1) functional metrics and (2) non-functional metrics. Functional ones are composed of latency, energy efficiency, throughput, goodput, packet loss, jitter, and QoS. On the other hand, the non-functional set includes metrics like service availability, reliability, service load and number of API requests. The next generation of mobile networks needs to be designed to meet the strict requirements such as higher bandwidths, lower latencies and increased mobility. In order to satisfy these, not only radio-access technology needs to be improved, but also the core network should be capable of serving billions of devices [97]. For instance, it is noted that 5G systems should provide end-to-end latency lower than 10ms, even 1ms for some specific cases [98], and MEC is a potential solution to lower the latency to meet the requirements. Besides, deploying MEC servers at the edge dissolves issues related to congestion at the backbone because the traffic generated by the mobile users is processed locally [99].

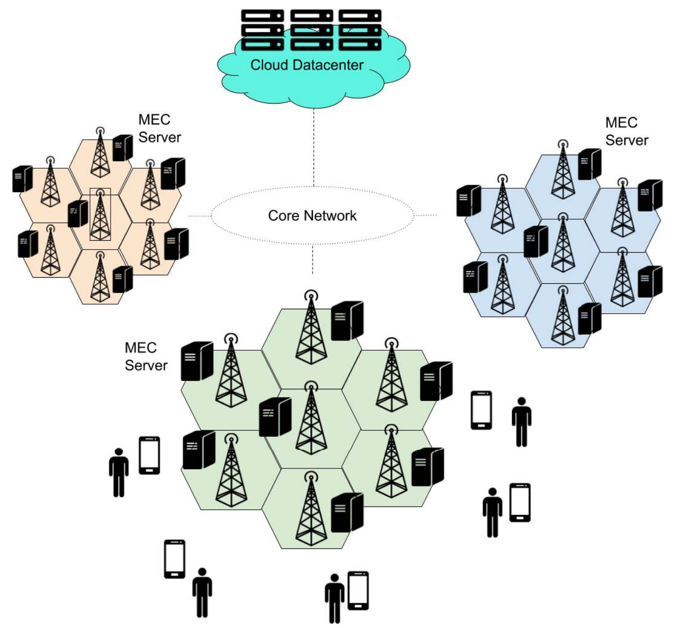


Fig. 7. MEC infrastructure and cooperation with the central cloud.

While Cloudlets or Fog nodes are mostly managed by individuals and can be deployed at any appropriate location, MEC servers are owned by mobile operators and need to be located near the base stations in order to provide an access to the mobile network users over the RAN [100]. This helps operators to increase the service quality through the effective mobility management in addition to the benefits of utilizing cloud-like resources at the edge [101].

In order to prove the applicability of MEC on the novel use cases, different organizations offer implementations based on the ETSI ISG MEC Proof of Concept (PoC) Framework [102]. As an example case, a PoC team composed of three different companies demonstrates that MEC is a feasible technology for "Edge Video Orchestration", where the users are able to access live video streams through an orchestration application installed on a MEC server [103]. On the other hand, another PoC team composed of six different companies shows that MEC is a suitable technology for advanced service delivery and service function chaining [104].

It is important to state that MEC servers can be deployed at LTE base stations (eNodeBs), 3G Radio Network Controllers (RNC), or multi-technology sites which consists of 3G and LTE. A remarkable point is that the recent form of MEC is not accessible through WLAN by mobile users as ETSI proposes MEC servers to be deployed only at base stations.

As an illustration of MEC servers that are located near to the base stations, Figure 7 presents an environment where the users can offload tasks through their smart phones. By processing the cellular data and offloaded tasks, the edge mitigates the congestion within the core network. On the other hand, MEC servers are able to access cloud datacenters in case of a need for higher computation power or storage capacity.

TABLE IV
A BREAKDOWN OF THE FEATURE SET OF THE EDGE TECHNOLOGY SPECTRUM

Common Set of Properties	Differentiated Areas
• Providing Computational Resources at the Edge • Use Cases • End-user Devices • Performance Metrics • Wireless Access • Distributed Servers • Context-awareness	• Ownership • Coverage • Access Technology • Server Density • Cost/CAPEX • Flow Characteristics • Number of Active Users

Although there are promising benefits and intense effort for standardization of MEC, there are still challenges that may be directive for the further efforts. Since there are multitude of third-party partners such as application developers, content providers and network device vendors, the complexity of the services and management of such a large scale environment becomes challenging [105]. On the other hand, providing cloud-resources at the edge brings the challenges of traditional cloud systems such as security and privacy.

Another notable issue is that when a MEC platform become overloaded, the system may fail and it affects the quality of the provided service to the mobile subscribers [106]. From the viewpoint of the operators, this may become an even more important problem because the downtime of MEC servers can result in huge costs.

It is observed that the example use cases and scenarios that are presented by ETSI ISG are the explicit indicators of a need for a computation power at the edge of the network [107]. The direction of the sector and academic studies intersect at a point and it is inevitable to deploy high-computation power at the edge which needs to cooperate with traditional cloud computing.

G. Comparison Among Edge Computing Proposals

Although the main objective of Edge Computing paradigm is deploying computational resources at the edge of the network, the specific implementations of this concept differ at various aspects. The most important similarities and differences among edge server types are summarized in Table IV. Since the similar features are explicitly introduced while the edge proposals are presented, this subsection discusses the key differences in depth.

The purpose of MEC is exploiting traditional cloud capacity at the edge for executing the mobile network operations and processing the subscribers' offloaded tasks. MEC helps to minimize the communication latency and energy consumption by providing accessibility within the range of RAN instead of the core WAN [25]. It also improves the service quality since real-time location information can be analyzed instantly at the edge. However, Cloudlets and Fog servers are in service mostly for executing the user offloaded tasks.

One of the key aspects that affect the edge server operations are the service providers. The MEC technology is maintained by the mobile network operators and naturally the

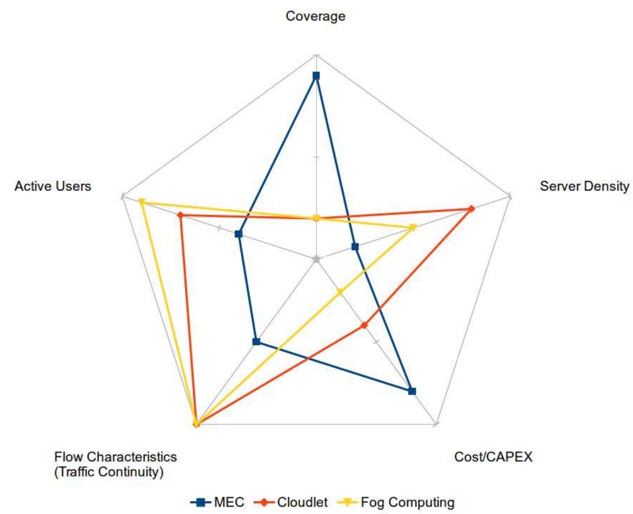


Fig. 8. A comparative visual summary among MEC, Cloudlet and Fog Computing.

intended group is the subscribers. On the other hand, any casual user such as a public place owner can deploy a Cloudlet or Fog server within a private environment [77]. However, since Fog Computing is provided by Cisco, it is expected that the providers of Fog Computing will consists of a specific subset rather than a normal user. Because of the major differences with respect to the implementation and deployment of the mentioned proposals, the security mechanisms and underlying protocols will also vary [77].

The general overview of the differences between MEC, Cloudlet and Fog Computing proposals are illustrated in Figure 8. The figure focuses on five individual features and presents how they behave for each proposal. These are the aspects that proposals differ from each other explicitly. Apart from these, dissimilarities can be observed in terms of other properties but they are relatively minor details. Therefore, we focus on the aspects that are grouped under coverage, active users, flow characteristics, cost/CAPEX and server density.

Each of the edges in Figure 8 represents one of the features that edge server types exhibit differences. The core of the radar graph represents the minimum value for all of the categories while an edge stands for the maximum achievable value for a corresponding feature. For example, it can be deduced from the radar graph that deploying MEC servers at the edge has a higher cost than the Cloudlets where Fog Computing has the lowest cost.

1) Coverage: The locations of the server deployments differ as a natural result of being provided by distinct operators. The specifications of MEC indicates that a MEC server needs to be co-located with the cellular network base station which means that the possible locations of MEC servers are already specified. Fog servers are mostly independent but they can be also deployed in the ISP infrastructures such as gateways and routers. The initial proposal of the Cloudlet indicates that they need to be distributed over a wide range of area to provide high accessibility. In other words, there is not any specific operator or vendor for Cloudlets. This property directly affects the area of coverage provided by the servers. Among all, MEC has the most coverage since a MEC server is accessible through a 3G/LTE base station. However, Fog servers and Cloudlets are accessible through a wireless access point (AP) whose coverage is much less than a macrocell, a microcell and even a picocell.

Although most of the Cloudlet research focuses on using Wi-Fi as the main access technology, it is applicable to other wireless communication infrastructures [94]. Agarwal *et al.* [119] propose that small cell networks such as femtocells and picocells are feasible alternatives for Cloudlet deployments. On the other hand, it is stated that Cloudlet antenna can support WiMAX, which can be used for inter-Cloudlet communication, in addition to Wi-Fi which is the main technology for short range communication with the Cloudlets [116]. However, as a result of the recent advances in cellular networks, operators focus on MEC as the main edge server technology instead of Cloudlets. On the other hand, since WiMAX is not widespread as desired, studies on Cloudlets mostly focus on and demonstrations are usually performed over Wi-Fi networks.

Similar to the Cloudlet proposal, Fog Computing is access technology independent and it may be implemented using Wi-Fi, 3G, LTE and even Bluetooth [10], [120]. However, new generation of cellular networks are more commonly referred to MEC instead of any other Edge Computing proposal and the cellular network operators have already begun to present real-life demonstrations with MEC servers. In other words, MEC is the de-facto edge server technology for the cellular networks. On the other hand, because of its master/slave design and very short range, the technical structure of Bluetooth is not preferred for accessing edge servers. Therefore, like Cloudlets, Fog seems to focus more on Wi-Fi for practical studies and their coverage is not able to reach to the level of MEC in the short-term.

2) Active Users: It is observed that most of the Fog Computing studies address the IoT and vehicle-to-vehicle (V2V) communication use cases. Therefore, it is expected that the number of active users and devices in Fog Computing are higher than the Cloudlet, which also targets IoT devices but does not cover the vehicular communications. On the other hand, since MEC only serves to its providers' subscribers, its target users form a smaller set.

3) Flow Characteristics: The diverse characteristics of the served users and devices concurrently affect the type of the traffic generated at the edge. While the devices using Fog servers or Cloudlet generate continuous data through sensors, the traffic destined to MEC servers are mostly intermittent and less continuous because of the pricing policy of cellular networks and targeted use cases, although it addresses some video stream scenarios [121].

4) Cost/CAPEX: Cost of the deployment is another factor that demonstrates the diversities. Deploying MEC servers at the base station results in higher CAPEX since installing the base stations are expensive and incorporating a server results in higher costs than deploying a single server, like Cloudlet. On the other hand, Fog servers can be adapted from an available networking device such as a wireless access point and a router [92]. For this reason, the cost of obtaining a Fog node is the least among edge proposals.

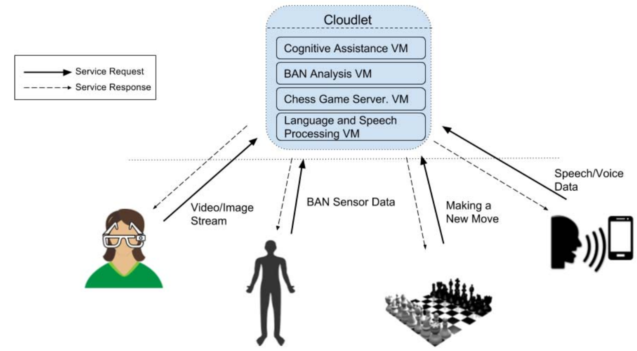


Fig. 9. Cloudlet-based use case scenarios.

5) Server Density: A Cloudlet can be deployed at any appropriate public location, such as a shopping mall, and accessible through a WLAN. Consequently, the density of Cloudlet deployment is higher than the other proposals. On the other hand, a MEC server can be deployed only at base stations which means that this restriction decreases its relative node density. The density of fog servers is between the other two as they cannot be deployed everywhere Cloudlet can reside but it does not have rigid restrictions like MEC server installations.

IV. Edge Computing Use Cases

Specific Edge Computing technologies such as Fog Computing and MEC are proposed by different technical communities with different agendas in mind. However, the underlying principles are common for all of them and there are no deep technical barriers for practical implementation of an Edge Computing use case under a certain edge technology. On the contrary, certain areas are targeted by more than one Edge Computing technology such as augmented reality.

As summarized in Table V, compatibility of the use cases with the edge server types varies. Some of the main factors that affect this kind of relationship are the vendors, server locations and target users.

The Cloudlet proposal aims to decrease the latency for the scenarios that require real-time interaction such as augmented reality. Besides, there are other novel use cases that are embraced by Cloudlets and they are strongly emphasized by the related studies to give the insight of the necessity for Cloudlets. A Cloudlet environment that gathers some of these use case scenarios together are presented in Figure 9.

Throughout the survey, it is observed that Fog can also satisfy the requirements of recently popular cases [88] which are depicted in Figure 10.

The Group Specification (GS MEC-IEG 004) [107] by ETSI ISG introduces a set of scenarios and use cases for presenting the potential of MEC to improve QoS for the end users. It is mentioned that in the near future, MEC will enable novel applications and services for various domains. These examples include intelligent video acceleration, video stream analysis, augmented reality, intensive computation, connected vehicles

TABLE V
EDGE COMPUTING PROPOSALS AND TARGET USE CASES

Use Cases	Cloudlet	Fog Computing	Mobile-Edge Computing
Body Area Networks (BAN) & Healthcare [108], [109]	Highly Compatible	Highly Compatible	Less Compatible
Augmented Reality [110], [24], [107]	Highly Compatible	Highly Compatible	Highly Compatible
Face Recognition [93]	Highly Compatible	Highly Compatible	Less Compatible
Language, Video & Speech Processing [111], [112], [107]	Highly Compatible	Highly Compatible	Highly Compatible
Video Streaming & Analysis [92], [24], [107]	Highly Compatible	Highly Compatible	Highly Compatible
Connected Vehicles [113], [107]	Less Compatible	Highly Compatible	Highly Compatible
Intensive Computation [81], [10], [107]	Highly Compatible	Highly Compatible	Highly Compatible
Smart Grid [63], [113], [10]	Less Compatible	Highly Compatible	Less Compatible
IoT & Wireless Sensor Networks (WSN) [10], [37], [107], [114], [115], [116]	Highly Compatible	Highly Compatible	Highly Compatible
Military & Hostile Environments [117], [118]	Highly Compatible	Highly Compatible	Less Compatible

Fig. 10. A heterogeneous environment with Fog servers.

and IoT gateway. In order to prove the functionalities of edge servers in real-life implementations, this section focuses on the background of the use cases that are commonly referred and how edge servers can help to satisfy their key requirements. Below given subsections discuss the use cases given in the Edge Computing literature. In the discussion, name of the specific Edge Computing technology is preserved in order to be consistent with the original cited research. However, this does not indicate an exclusivity relation between the use case and a certain Edge Computing technology.

A. Cognitive Assistance

By accomplishing the tasks with a low response time, Cloudlets enable a new type of real-time cognitive assistance applications which run on wearable devices such as smart glasses [94]. As a practical demonstration of this improvement, a framework based on Cloudlets is proposed by Ha *et al.* [110]. This framework serves the functions of sensor data analysis to the end-users. In addition to the control and user guidance mechanisms, the server-side comprises of multitude of VMs where each one is responsible for non-identical sub-services that constitute the major role of the framework such as face recognition and OCR (Optical Character Recognition).

B. Body Area Networks (BANs)

The ultimate goal for a BAN is monitoring the collected data in a reliable manner with low delay [108]. Nodes that comprise BAN generate a large amount of data, and require powerful computational resources and a large storage area. Besides, the generated data is life-critical and needs to be monitored as soon as possible. When all these requirements are assembled, it seems that Cloudlets are a very good fit for analyzing the collected sensor data with low response time and storing them for further analysis.

Healthcare applications and BANs are also focus areas of Fog Computing. In the case of ECG (electrocardiography), sensors usually upload huge amount of data periodically. In order to provide real-time interaction, fog servers collect the data for carrying out the tasks such as data filtering and data aggregation [109]. This helps for taking an action immediately if ECG sensors collect urgent data that requires instant processing.

C. Military and Hostile Environments

The sensors that are deployed in a hostile environment such as a military field are vulnerable to attacks [118]. In the case of an intrusion detection system, the fog servers is able to provide low-latency services to sensors within the environment and detect the malicious nodes by analyzing the gathered sensor data. Aside from providing minimized response time with Fog Computing, Cloudlets are also beneficial for a military environment with functionalities such as reducing DoS (Denial of Service) vulnerability in military operations by locating it in a single wireless hop [117].

D. Language and Speech Processing

Multilingual processing applications, which take voice as input for language translation, need a continuous Internet connection and large resource pool [111]. Cloudlets are able to supply the necessary resources and mitigate the load on the mobile devices with limited capacity. Also, the application based on a Cloudlet infrastructure eliminate the WAN latency and requirement of continuous Internet connection.

One of the real-life scenarios is collecting the speech data from patients with Parkinson's Disease (PD) with an Android smartwatch and transferring it to a fog server to be processed [112]. The fog server processes the collected speech data and transfer it to the cloud servers for further analysis and storage.

E. Smart Grid

Fog computing may be an enabling technology for smart grid concept where energy load balancing applications can run on edge servers that consider alternative energy resources according to demand and availability [113]. Smart grid application is a well-directed application for a multi-tier environment because it needs both Fog and Cloud architectures [10]. Global coverage is provided by cloud where the data is stored for months and years for business intelligence analysis. Fog collectors may process the data generated or send a portion of the data to the higher tiers in order to carry out the visualization or real-time reporting. In addition to Fog Computing, MEC servers are also able to serve the similar functionalities for the smart grid applications [122].

F. IoT and Wireless Sensor Networks

Sensors within a network mostly have limited capacity for storage and computation which means that they remain incapable of functioning except sensing and relaying data [10]. Thanks to its beneficial features, Fog Computing may fulfill the deficiencies in the storage and computation related tasks, and be an essential part of a suitable environment for WSN. Besides, by deploying heterogeneous Fog nodes, various types of data can be collected and the interoperability problem is solved [114]. Different IoT or Wireless Sensor Network (WSN) environments can be controlled and managed by a single point through Fog nodes and this feature allows to leverage the potential of novel services and use cases [91].

CitySee [115] is an environment monitoring system and it is one of the largest WSN environments. The data collected by the sensors are transferred to the sink node which then forwards it to the Cloudlet after preprocessing. The aim of CitySee is providing services to the users, and hence Cloudlets are utilized within this architecture for creating a set of APIs for customers to access and use the sensed data.

As similar to the WSNs, the nature of IoT results in connected gadgets and devices in various forms. These devices are capable of generating huge traffic in different characteristics, and hence there is a need for aggregating all the information and provide an environment where heterogeneous devices can operate coherently. A MEC server may act as a gateway to provide services such as decision logic, database logging, remote provisioning and access control to the devices [107].

On the other hand, Cloudlets are also compatible with the IoT scenarios since they create an environment where IoT gadgets interact with the services deployed at the Cloudlets [37]. Through bringing these services closer to the edge, the interactions can occur in real-time which is necessary for the IoT operations.

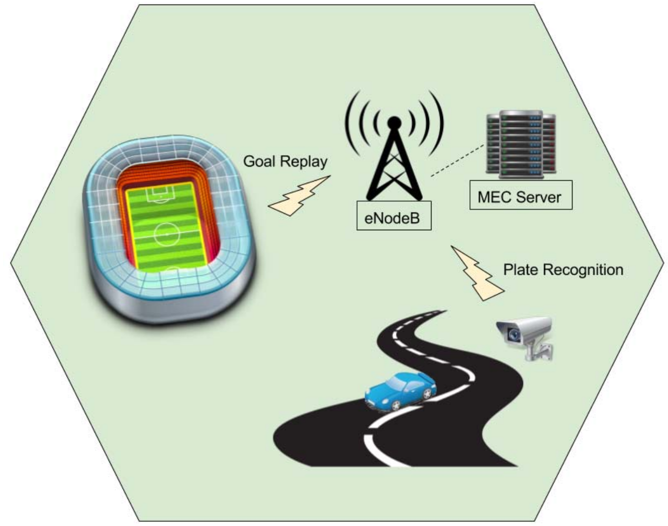


Fig. 11. Video streaming and analysis with MEC technology.

G. Video Streaming and Analysis

Video streaming is one of the applications that requires low latency and jitter. Therefore, providing this service at the edge of the network helps to achieve these requirements by eliminating the congestion within WAN. As an example scenario, a fog server that is deployed in a bus can provide the service of video streaming to the passengers through Wi-Fi [92].

The analysis of a video that is recorded by a camera requires also real-time interaction. Fog servers possess the necessary computational and storage capacity to store the video streams and analyze them for object recognition and tracking [24]. In order to decrease the cost of deploying multitude of cameras for the cases such as vehicle plate recognition, the data analysis tasks need to be assigned to a server rather than the camera itself. Deployment of MEC servers close to the base stations provides flexibility and mitigate the congestion caused by the video stream data to be transferred to the cloud datacenters for further analysis [107].

Gabriel [81] is a cognitive assistant application that runs on a Cloudlet. The video recorded by a wearable glass is forwarded to the Cloudlet after discovering it in the vicinity. Then, the video is processed there for providing a guidance for the user. As an optional case, the video can be transferred to the cloud servers for centralized tasks such as error reporting. By offloading the video analysis task to the Cloudlet, it is expected to improve the user experience by decreasing the latency under a certain threshold.

A typical environment that is covered by a MEC server that is located at a base station, eNodeB in the case of 4G/LTE, is shown in Figure 11. The audience at a stadium may request the replay of a goal as a video stream, which is made available by the provider instantly. To support thousands of requests and mitigate the burden on MEC servers, the operator may initiate additional VMs in an instant. On the other hand, a road-side security camera may request for a video analysis service from the MEC server in order to recognize the plate of the vehicle in a shorter time.

H. Augmented Reality

The trend towards smart phones and smart glasses increases the popularity of augmented reality applications [24] which require a level of computation power that is not present within the device itself. At the initial step, cloud computing may provide the demanded computational power. However, these scenarios are sensitive to the delay which may decrease the user satisfaction dramatically. Thus, an augmented reality system that is enabled by Fog Computing results in an increased throughput and decreased delay.

Zao *et al.* [123] proposed a multi-tier fog/cloud architecture to leverage the augmented brain computer interface. One of the benefits that Fog Computing promises, real-time interaction, is demonstrated by playing a multi-player and on-line brain computer interface game.

In order to provide a real-time interaction for a visitor to a sports event or a historical monument, the output of the end user's camera can be forwarded to the MEC server in the vicinity [107]. The main advantage for this methodology is that information the mobile user seeks for are highly localized so a centralized mechanism cannot provide a feasible solution for this case. Since augmented reality requires low latency and context-aware data, MEC servers can satisfy the user's expectations.

I. Connected Vehicles and Smart Traffic Lights

There are important constraints in vehicular communication that directly affect the performance of the overall architecture such as high speeds, unreliable wireless connection and highly dynamic topology [124]. Within these limits, it is quite challenging to achieve low delay and satisfy the user experience. By deploying computational resources and keeping the communication among vehicles at the edge via fog technology may improve the performance and relax the limitations of the inherent constraints.

It is possible to measure the distance and speed of the vehicles in the range of smart traffic lights [113]. Smart lights can communicate with each other and prevent the possible accidents through the utilization of Fog technology [10]. Besides, a distributed video camera system may recognize an ambulance and act accordingly for changing the traffic lights to open a way. For urgent operations, Fog orchestration layer can execute the necessary tasks instantly such as coordinating the actions for the communication between smart lights. On the other hand, the data collected from smart lights can be analyzed further at the cloud-tier for long-term optimizations.

In addition to support connected vehicles with Fog technology, the concept becomes popular with the proposal of LTE (Long Term Evolution) which provides long distance connectivity. For improving the safety by generating a notification to the user about road hazards and traffic congestion, an instantaneous interaction with the traveling driver is required at the roadside. MEC servers that are deployed at the LTE base station (eNodeB) can be used for storing the data and provide road status messages to the travelers by cooperating with the roadside sensors [107]. A MEC server collects information from the cars or gathers data through sensors monitoring the road. The data collected from the cars, such as acceleration, can be distributed to the other vehicles in the vicinity. On the other hand, the drivers that are in the area of coverage of a server can be informed in the case of a road hazard, a traffic accident or traffic jam. These types of communications and information sharing require minimum latency but the server can continuously communicate with the cloud datacenter through EPC (Evolved Packet Core), which is the core network of LTE, for data storage and further data analysis.

Relaying the messages between vehicles over LTE eliminate the necessity of building a Dedicated Short Range Communication (DSRC) [125] network that allows vehicle-to-vehicle communication for short distances. Hence, deploying MEC servers near to the available base stations for connected vehicles prevents investing in completely new protocols and network infrastructure that may be costly in terms of CAPEX/OPEX.

Beside of the advantages that MEC provides for leveraging the instant communication between vehicles, there are some security concerns including malicious or fake data reported by vehicles, user's privacy, trust issues, and certain attacks at application and infrastructure levels [126], [127]. Therefore, MEC developments should intrinsically enable security and privacy for protecting the applications and user data at the edge server and provide an authentication system for the vehicles [25], [77], [99].

V. Employing SDN Capabilities to Facilitate Edge Computing

It is envisioned that SDN has the capability to offer remedy for a large set of challenges that are encountered within the traditional networking approach. Similarly, when the intrinsic properties of SDN are considered, a fruitful cooperation with the Edge Computing framework can be foreseen. This section initially gives a brief introduction to the SDN concepts and OpenFlow. Then, eight benefit areas provided by SDN are topic-wise identified and discussed in the context of Edge Computing. To demonstrate the collaboration of SDN and Edge Computing in real world scenarios three practical cases, namely Multi-tier Edge Computing Architecture, Service-Centric Access to Edge and NFV are explored in Sections V-C–V-E. In order to further identify trends that cover SDN and Edge Computing cooperation latest research are surveyed in Section V-F.

A. Software-Defined Networking Concepts and OpenFlow

SDN is a recent paradigm proposed for using the limited network resources optimally and enabling flexible network management by separating the control layer from the data layer [30]. Since the main logic is extracted from the forwarding nodes that are no longer capable of taking decisions on their own, it is concentrated on the software-based controller which is capable of having a general view of the underlying network [128]–[131]. Routing and forwarding behavior of the network elements can be inquired and modified upon customized policies in the SDN controller. These operations are achieved through modifying and populating the flow tables

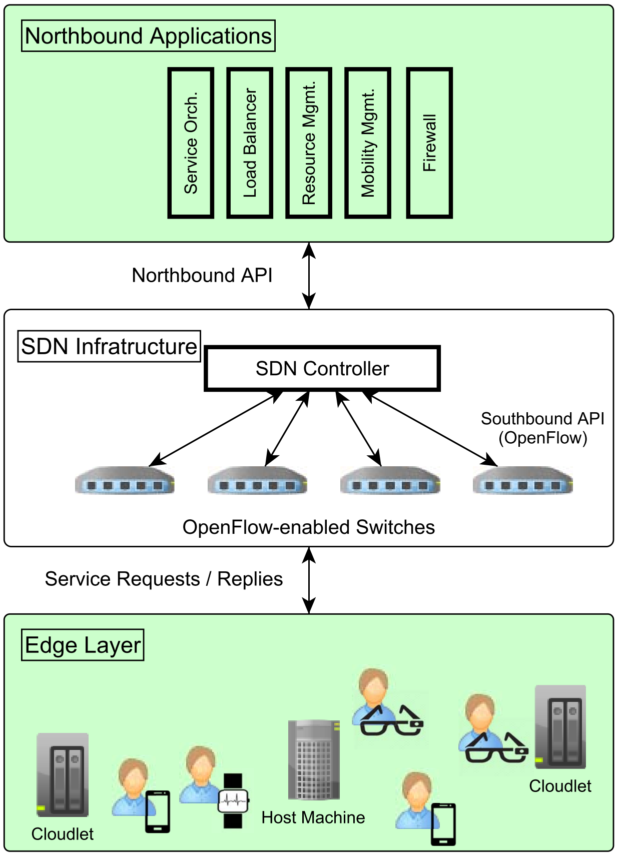


Fig. 12. A view on SDN architecture for Edge Computing.

of the forwarding nodes with the definitions of match/action flow rules [132]. When a packet appears at an ingress port, the switch initially checks its flow table to find any matching rule. If found, it applies the action that is defined by the matched flow rule. In case of an unmatched packet, the switch buffers this packet and forwards an encapsulated replication of it to the controller for figuring out the most appropriate action [133]. The SDN controller then decides on an action to be applied on this packet, and installs the necessary flow rule on the corresponding switch for applying the same action on the similar packets in the future.

All operations of SDN and the flexible communication between the controller and the switches are carried out with the OpenFlow protocol [133], [134] which is currently developed by Open Networking Foundation (ONF) [27]. OpenFlow represents the main functionalities of SDN such as managing the flow tables on the forwarding nodes, populating them, defining flow rules, gathering statistics and many other managerial operations [132].

The layers of the envisioned SDN architecture for orchestrating multi-tier edge system is depicted in Figure 12:

- Edge layer
- SDN infrastructure
- Northbound applications.

At the bottom layer, there are servers configured to handle the user requests and various edge devices that seek for services. At the intermediate level, the traditional SDN infrastructure can be seen which is composed of OpenFlow-enabled switches and the SDN controller. Although the SDN considers them as separate planes, it forms the typical SDN infrastructure where the responsibilities of both the controller and forwarding nodes are clearly defined. The novelty of this adapted SDN architecture is specified by the top layer, which consists of customized and virtualized northbound applications that define the behavior of the control mechanism. The main responsibilities and functionalities of these applications are service management and orchestration, optimal resource allocation, mobility management and other operational tasks. Applications that define the network behavior can communicate with the controller through the northbound interface API that is implemented by the controller itself [130]. The high-level commands generated by the northbound applications are forwarded to the controller via the northbound API which is not standardized yet [131]. Then, the controller transforms these commands into low-level OpenFlow messages to be sent to the data plane [135]. If the control plane is distributed and several controllers are deployed over geographically distant servers, then there is a need for mechanism to provide communication between the controllers for synchronization through east/west interfaces. Like the northbound interface, there is not a standard for the east/west interface yet.

As stated by McKeown *et al.* [134], OpenFlow was firstly used for a campus network which is an inspiration to support campus-based scenarios for Edge Computing. Besides controlling the switches, OpenFlow also enables the analysis of the network traffic and reactive behavior. The controller can collect statistics such as the number of received packets or transmitted bytes by the switches through OpenFlow request messages, and balance the load accordingly or form the flow characteristics. All these properties can be applied to the OpenFlow-enabled switches without being dependent on device types so that a heterogeneous network composed of forwarding devices by different vendors can be controlled by SDN [137]. In addition to the hardware switches, software-based switches gain popularity. The most common software switch is OpenvSwitch designed for flexible and high-performance networking in virtual environments [138]. Its popularity gains significant attention and eventually the core modules of OpenvSwitch is embedded into the Linux kernel [139].

Another important challenge that SDN addresses is virtualization [16]. In some cases, both concepts are used interchangeably. However, it should be stated that SDN is independent from virtualization but complementary to it [140]. The Network Functions Virtualization (NFV) is the technology which separates the network functions such as firewalls or DNS from the hardware and hand them over to a software-based application [16]. Due to similar underlying principles, NFV and SDN technologies can be integrated into each other to mitigate some challenges by transitioning the network management from hardware to software. The further discussion on the integration of NFV and SDN is presented in Section V-E.

The utilization of SDN in optical networking is referred as Software-Defined Optical Network (SDON) which enables joint dynamic provisioning and optimization of multiple layers through the global view that the SDN controller holds [141]. It

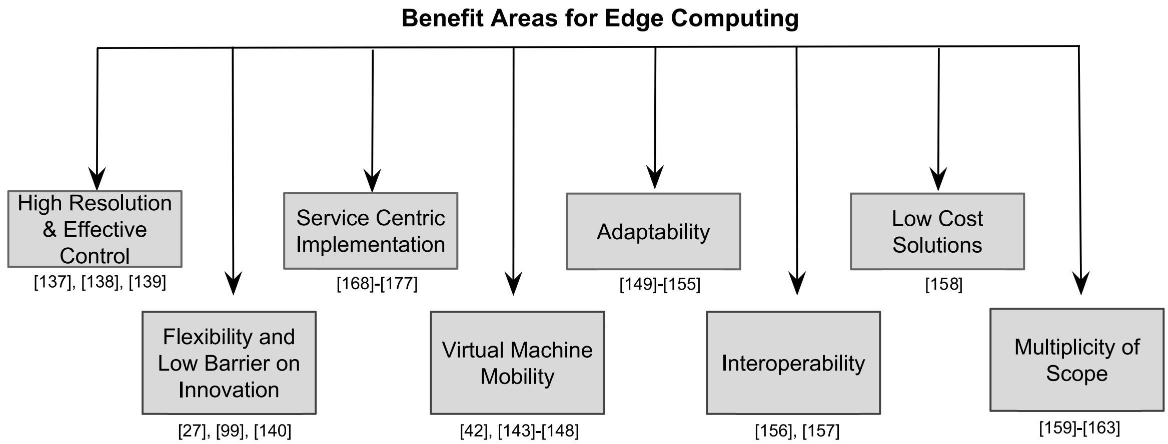


Fig. 13. SDN properties and benefit areas for Edge Computing.

enhances OpenFlow to support the forwarding planes that are not capable of packet-switching. So far, the benefits of SDN were limited to wired environments [140]. Adapting SDN to wireless networks results in a new form referred as Software-Defined Wireless Networking (SDWN).

B. How Can SDN Help Edge Computing: The Benefit Areas

The characteristics and capabilities of SDN that are discussed by the previous subsection lay down the basis for the potentially successful interaction between SDN and Edge Computing. When technical properties of SDN and the way they are employed in the literature are examined, certain benefit areas stand out. As illustrated in Figure 13, eight benefit areas are identified within the scope of this work: (i) High Resolution & Effective Control, (ii) Flexibility and Low Barrier on Innovation, (iii) Service-Centric Implementation, (iv) Virtual Machine Mobility, (v) Adaptability, (vi) Interoperability, (vii) Low Cost Solutions, and (viii) Multiplicity of Scope. Each of these areas, and the way that SDN and Edge Computing interact are discussed further in the remainder of this section.

1) High Resolution & Effective Control: Centralized view enables SDN to dictate user defined policies and describe dynamic behavior through programmatic access [142]. Specialized northbound applications can generate commands to cope up with the dynamic behavior of the network. Besides the reactive behavior, proactive control can also be based on network statistics collected at the control layer. This feature of SDN renders operating the network in near optimal conditions possible. Real-time optimization based on cross-layer information is possible through centralized access, real-time statistics and programmatic access.

The Edge Computing infrastructure is likely to include many servers [143]. For instance, there can be physically many Cloudlets located in a campus environment or a public spot such as shopping malls and airports. A combined computational and link load balancing will be necessary to be able to serve users by considering the QoS requirements [144]. As an example, a temporary large group of users may coincide in a certain location and result in extraordinarily high utilization of the computational capacity of the nearest Cloudlet. This would inevitably cause an increased service time despite the high bandwidth and redundant Cloudlet hardware available on the premises. This is in contradiction with the nature of Edge Computing which owes its existence to high access delays caused by WAN. As a remedy, a northbound application can provide the central view by monitoring not only link but also server loads through periodic collection of statistics via OpenFlow messages. This application can direct and shape traffic towards the less-loaded elements of the computational pool.

A non-negligible portion of Edge Computing scenarios are based on mobile and small form factor devices taking computational services residing in the vicinity. The capabilities and characteristics of these devices are quite varied. An end user, while getting a continuous service from an Edge Computing facility, can be handed over from one Cloudlet to another as a result of mobility or an orchestration/management activity including load balancing. To enable fluent operation of this considerably dynamic environment, high resolution and effective control offered by SDN is an essential advantage over the traditional network application paradigm based on socket or RPC/RMI (Remote Procedure Call / Remote Method Invocation) programming.

An example scenario is depicted in Figure 14. A mobile user sends a service request to one of the Cloudlets in the vicinity. Before the request is accomplished by the server, the user is authenticated to another network by changing the location. The controller can track this movement with its ability to discover the topology and get the necessary information about the user's new location, such as its recently assigned IP address. This allows the service result to be reached to the user by adding new flow rules to the switches on the path. During this entire process, the user is not aware of the operations occurring within the network, and the user experience is not interrupted.

2) Flexibility and Low Barrier Over Innovation: The traditional infrastructure of the network is restricting for innovations because there is a small area for innovation when the hardware has the responsibility of both control and forwarding layers [136]. By decoupling the control and forwarding layer,

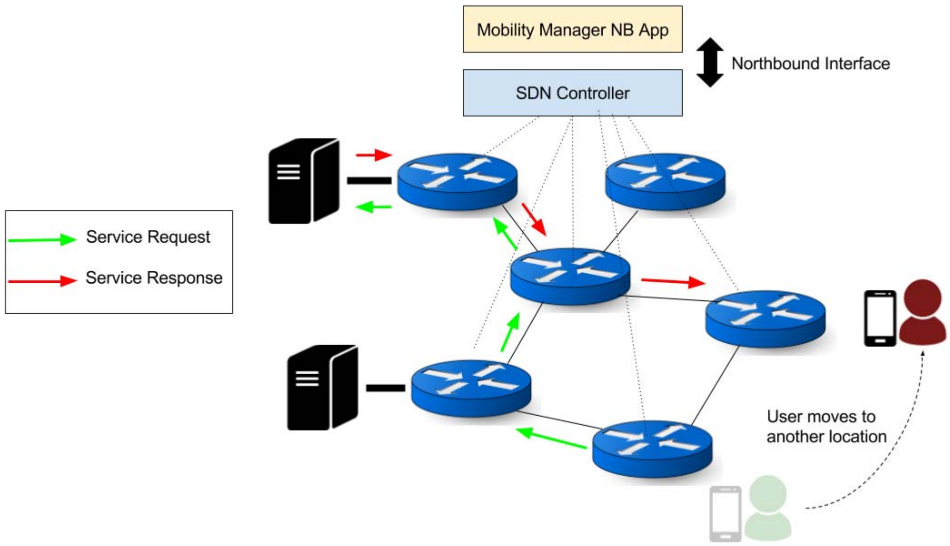


Fig. 14. SDN for handling handover scenarios.

SDN provides flexible programmable interface that enables innovation. As the integration of cloud and edge servers demands high flexibility because of the increasing number of devices, SDN can treat network as a flexible software [145]. With the help of centralized controller and user-implemented northbound applications, the large scale environment can be managed in every level of orchestration.

In traditional networking, the emergence of a new protocol introduces the necessity of a new hardware or redesigning/updating the switching-chip, which results in high costs and causes vendor-dependency. For instance, VXLAN (Virtual Extensible LAN) is a recently popular protocol that is mostly used in the cloud environments and enterprise data centers [146]. As predicted, upgrading the network infrastructure of the whole cloud site is not feasible. Instead, software-based switches can be programmed to implement this technology, in fact OpenvSwitch (OVS) has already this functionality. On the other hand, recent versions of OpenFlow has a support for VXLAN which means that the controller can also be implemented accordingly to define the operations of this protocol within the network and operate with OVS in harmony.

3) Service-Centric Implementation: Considering a wide range of services and their distribution over various Edge Computing nodes, enabling the end-users to request a service by just identifying "what" instead of specifying "where" would be an enormous advantage. However, as constrained by the host-centric traditional network design, it requires an extra effort on both edge devices and the servers to achieve this.

SDN allows service-centric as opposed to host centric solutions to be devised. It is not a coincidence that the Future Internet paradigm embraces both SDN and Information-centric Networking (ICN) [147], [148]. The dynamic environment at the edge, geographic distribution of the services and highly mobile end users are the issues addressable by a service-centric scheme. The role of SDN in this area and the necessary conditions to enable a service-centric model is discussed thoroughly in Section V-D.

4) Virtual Machine Mobility: Virtual machine (VM) migration is a technique that is normally employed in datacenters for effective operation in terms of energy consumption and load distribution [149]. Within the context of Edge Computing, ability to migrate VMs over the edge infrastructure whenever needed provides fine control and optimization possibilities over the whole system. VM migration can be triggered by the user movement, energy conservation, reducing the traffic load or service replacement [150].

SDN, with the centralized view of the system, is an ideal scheme for managing the VM migration process. SDN is also advantageous for this task due to its capabilities for reducing the down time during migration, preventing congestion, improving throughput and determining the location for the best performance [151]. In the case of a cloud datacenter managed by OpenStack [152], the SDN controller can monitor the network, discover a possible congestion and mitigate it through installing new flow rules after a live VM migration [153]. There are several proposals that employ SDN based approaches for VM migration in datacenters [45], [154], [155].

Although most of the proposals consider the VM migration at the cloud site, the same operations can equally be applied to the edge servers. The utilization of the edge servers and the traffic destined to them can be monitored by a particular northbound application. A VM can be migrated closer to the user or to a powerful server, depending on the requirement. Since VM migration causes considerable traffic within the network, the SDN control mechanism needs to manage the data transfer carefully. When the migration decision is taken, a close coordination among all relevant nodes would be possible with SDN. Integrating SDN with a Cloudlet management system, such as OpenStack++ [156] which is an extension of OpenStack, can simplify the VM migration across Cloudlets.

VM migration also provides energy-wise benefits to an Edge Computing system. If the number of requests arriving to the VMs of an edge server is low enough, the SDN control plane can decide that the VMs to be migrated to another site and the former one can be shut down to minimize the energy consumption.

5) Adaptability – Plug & Play Behavior: The number of wearable gadgets, smart phones and devices under the umbrella of IoT are increasing with enormous rates. New devices that are connected need to adapt to the network as soon as possible. Besides the end user devices, there is a necessity for deploying additional computational and networking resources in order to cope up with the increased traffic.

One of the key benefits that SDN provides is that it enables the plug & play operation through integrating a newly deployed service or device without any manual configuration [157], [158]. With the default configurations, OpenFlow-supported switches do not possess any protocol to detect the topology. In the SDN structure, controller's task is to identify and implement the certain functionalities for topology discovery. OpenFlow Discovery Protocol (OFDP) [159] exploits Link Layer Discovery Protocol (LLDP) and modifies it slightly to enable the similar functionality for the SDN-based environments. Therefore, any new device connected to the network can be detected by the SDN controller and the routing application can modify the rules instantly. There are several proposals that enhance the performance of OFDP, such as OFDPv2 [160], by reducing the overhead that is caused by the control messages relayed to the controller. Soft-WSN is an example architecture that leverages the real-time configuration of the sensor devices and networks in an IoT environment [161]. This architecture introduces a module for topology management that maintains the topology of the underlying network at any time. Therefore, it is able to detect every single node within the network, and hence the controller may act accordingly. Through this module which checks the topology frequently, a recently integrated sensor is directly added to the topology view of the controller without any additional action.

On the other hand, plug & play functionality is also applicable to the scenario of the connected vehicles. In order to react to the topology changes and maintain the distribution of the vehicles, sdnMAC [162] is proposed. Whenever a vehicle enters to the range of the roadside unit, the controller can acquire the necessary information about it such as its direction and speed.

Adaptability is also an important aspect of NFV. Through an adaptable solution that combines SDN and NFV capabilities, such as OpenSCaaS [163], service providers can accelerate the application of service chaining without any manual configuration.

6) Interoperability: As the interest in IoT increases, there are and will be many players and vendors around. In order to support interoperability between the devices belonging to different vendors and mitigate the complexity caused by the heterogeneity of Edge Computing, there should be a vendor-independent environment. As a result of the immense work on the standardization by ONF, SDN leads to a network environment which eliminates dependency on vendors [164]. Since SDN is able to manage the heterogeneous environments, distinct Wireless Sensor Network (WSN) and Body Area Network (BAN) setups with different types of sensors can operate in a single environment without any complications.

In order to cover networks with multiple transmission technologies, interoperability of the forwarding devices is necessary. Consider a scenario where the end-user requests for a computation-intensive and latency-tolerant task. There are two alternatives for realizing the desired objective: (1) pre-processing at the edge server and executing the remaining computation-intensive steps at the cloud servers (2) forwarding the request directly to the cloud site. As observed, the data generated by the end-user passes through at least two different transport technologies. In other words, interfaces of the forwarding devices differ among tiers. Therefore, a unified control mechanism that is able to manage different types of underlying technologies and devices from different vendors is mandatory. The cooperation of SDN and SDON ensures the harmony between distinct transport technologies and administrative domains. It can be clearly said that OpenFlow has the intrinsic capability to enable interoperability among network elements at the data plane. Although the GMPLS control plane is also a candidate for managing the diverse network infrastructures, it does not serve well to dynamic changes within the network which eventually causes some performance issues [165].

In addition to OpenFlow, there is an ongoing study for standardizing the northbound interface so that the interoperability of the northbound applications from different vendors can also be accomplished.

7) Lower Cost Solutions: Keeping pace with the increased number of mobile devices requires a large number of network nodes to be installed at the edge. In addition to the edge servers, network functions that are provided by each middlebox are also essential for managing and operating the immense mobile traffic generated at the edge. Traditional network design introduces hardware based solutions for managing the network and executing network functions which are expensive and difficult maintain. The smooth collaboration of NFV and SDN does not only improve the service orchestration, but also eliminates the requirement for updating the forwarding devices and integrating new protocols. These operations within the traditional network infrastructure, where the control plane is also hardware-based, result in high costs for the service providers.

Since SDN requires less complex hardware to operate, it lowers the costs compared with the traditional network infrastructure. Any solution can be provided as an implemented application that can dynamically program the network lowers the CAPEX and OPEX because they can be updated just by software patches and upgrades instead of installing a new hardware. Thanks to NFV, middlebox functionalities are shifted from hardware-based to the software-based which enables a virtualization opportunity to decrease the cost [166].

8) Multiplicity of Scope: As cloud computing becomes mainstream, the applications and use cases become more diverse and specialized. In addition to the smart device use cases, constructing Edge-cloud architecture based on SDN widen the scope and range of the context. Managing such a large-scale environment through SDN can provide higher performance operation of datacenter and edge server networks. SDN can be used for managing huge datacenters which mitigates the complexity from the distinctly located servers such as Google's SDN-based B4 [167].

5G PPP (The 5G Infrastructure Public Private Partnership) states that 5G will be based on software and it combines the MEC, SDN and NFV for improving the performance and flexibility [168]. Although it is not yet fully defined, the 5G is expected to include SDN and NFV as an enabler to provide an infrastructure for billions of new devices with less predictable traffic patterns [169]. Within the context of MEC, network services are virtualized and deployed as Virtual Network Functions (VNF) [170]. The role of SDN in such an architecture is leveraging the connectivity between the VNFs via service chaining and improving the control over them. In addition to boosting the functionality of VNFs, the intrinsic properties of SDN bring additional benefits to the MEC architecture such as the northbound interface for third-party applications and handling the traffic forwarding over cloud RAN [171] to support MEC services. Hence, building MEC on SDN with the support of NFV simplifies the management of this network and provides services to a variety of devices.

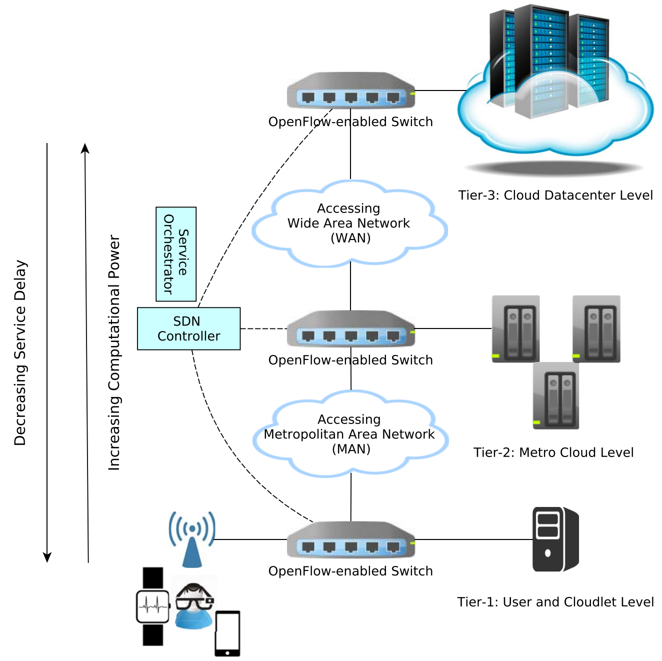


Fig. 15. Cooperation of the SDN mechanism in the multi-tier Edge Computing architecture.

C. Multi-Tier Edge Computing Architecture

Meeting the needs and QoS requirements of various tasks that are demanded by mobile devices cannot be achieved by solely employing cloud servers or Cloudlets. In order to cover and serve wide range of envisioned services, both cloud servers and Edge Computing facilities need to be incorporated into a joint architecture. Within the scope and coverage of this architecture, it would be beneficial to position an additional tier comprised of medium scale server hardware to be accessed via a Metropolitan Area Network (MAN). This supplementary layer brings flexibility while optimizing the tradeoff between the increasing delay and computational power. GigaSight [80] is a framework that is proposed to exploit Cloudlets at the edge of the network in order to prevent the congestion that may occur within MAN due to high-load traffic destined to the cloud. Using this approach as a base, enhancing the modularity of the system by embedding an intermediate level may serve as a catalyzer for achieving better performance.

Figure 15 depicts a multi-tier Edge Computing architecture that is capable of providing the necessary computation power for latency-tolerant scenarios at the higher tiers and minimum delay for delay sensitive applications at the lower tiers. Multitude of services and a wide range of requirements can be supported with this framework where each tier is formed considering characteristic features such as coverage area and resource capacities.

Determining the optimal tier at the time of a service request by considering the real-time conditions of the network and requirements of the applications can be carried out through the programmable control mechanism offered by the SDN. Without the flexibility of SDN, the dynamic changes within the network may not be noticed, static decisions are taken and scarce resources are not used efficiently. Hence, the system will not be scalable enough. The SDN controller can periodically retrieve meaningful information from the network such as the topology [172], and load on the links or servers [173]. This information is aggregated on the controller to have a centralized view of the overall system so that even instantaneous changes due to the mobility at the edge, geographically distributed services and dynamic behavior of the network can be captured and the control layer can act correspondingly.

As aforementioned, mobility is an essential feature which may change the state of the network at any time. Thus, the multi-tier structure should also support mobility and include a kind of state handover mechanism between edge servers. As an example to the concept of application state handover, one can imagine a face recognition based service. It is quite likely that the user specific data such as the database of face images is already fetched to the Cloudlet serving at the user's current location. If the user changes his or her location that may trigger a connection establishment with another network, the same parts of the database will be used by the new Cloudlet to serve. Since the SDN-based orchestrator is already aware of the network layer handover [174], it can also cooperate with the application for fetching the corresponding part of the database proactively in order to improve the QoS.

D. Service-Centric Access to Edge

The integration of Edge Computing into the existing infrastructure leads to the proliferation and diversification of services offered to the end-users. Since these services, which are not feasible to be realized so far, are presented in a distributed manner by the edge servers, the volume of the data traffic generated at the edge increases at a high rate. In order to keep the service quality at the highest level anywhere at the edge, a service instance should be provided by the geographically distributed servers as replications. GRECO [175] is a

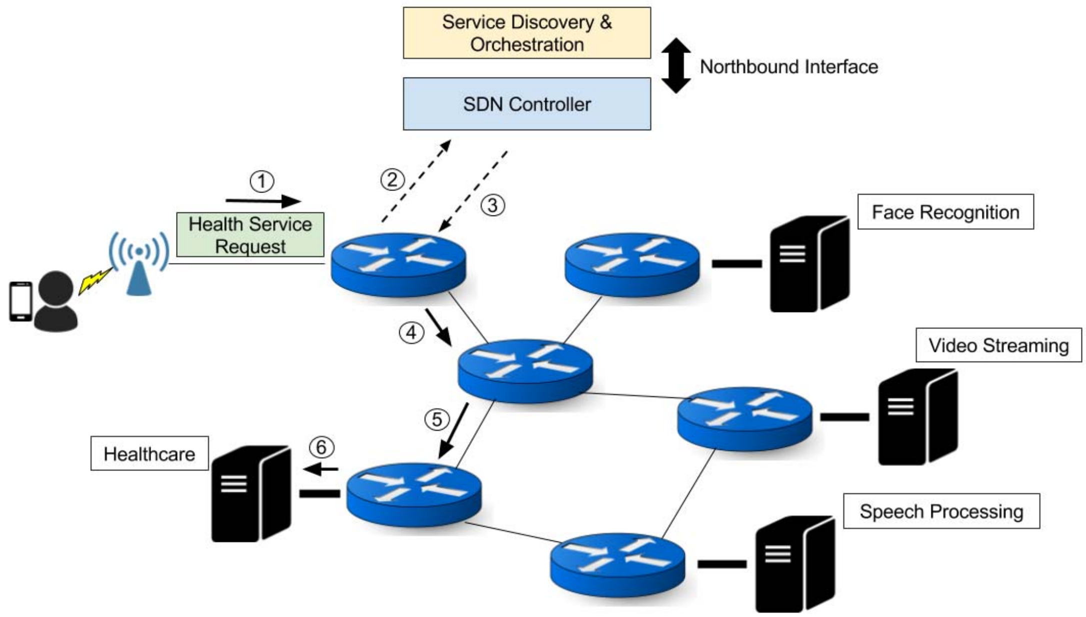


Fig. 16. Service-centric access through SDN.

distributed genetic algorithm that solves the optimal placement problem for the service-oriented applications within an architecture composed of cloud datacenters and edge servers. The service requests generated by a variety of mobile gadgets and service mobility creates a dynamic and heterogeneous environment at the edge of the network. Since the loads on the servers and the locations of the services are changing at any time, low service delay demanded by the end-users cannot be achieved with the host-centric approach. In the case of a traditional client-server interaction, the user application needs to be aware of the server that provides a particular service and resolve the location of the server through its IP address.

When all these requirements and conditions are gathered together, it is seen that leaving the complex operations of service discovery to the responsibility of the end-user devices greatly degrades QoS in this dynamic environment. As the content becomes more important than the physical location (IP address), the legacy network protocol stack becomes infeasible to keep pace with the changing requirements. For handling all the complexities by hiding them from the end-users, the traditional location-oriented network needs to transform into a service-centric architecture. However, putting the "what" in the core of the processes instead of "where" is not an easy transformation for the network operations. Relevantly, there is an ongoing trend for the Information-Centric Networking (ICN), Content-Centric Networking (CCN) and Service-Centric Networking (SCN) which focus on transforming the process of the traditional network by shifting the concentration from hosts to the contents, information or services [176]. The current trend shows that Future Internet is envisioned as service-centric [177]. In the context of service-centric design, the user and the application can request a service without any knowledge about the location of the server [178].

A reliable, secure and scalable service orchestration and large-scale network management are the key problems. SDN can provide fast service orchestration because the separated control plane is a programmable layer with the controllers and the applications [28]. All the services requested by wearable devices and IoT forming a heterogeneous environment can be orchestrated by SDN [179]. Not only the control mechanism but also the programmability of the forwarding devices come into prominence in order to orchestrate a large number of end-user devices that offload tasks to the resource pool [130].

The key functionality of SDN that leverage this model is the ability to have a general view of the network. Whenever a server is deployed with a service instance, it may inform the SDN controller for storing this information or controller itself may periodically inquire about the service-server matchings. In other words, the SDN controller is able to track the locations of every single service instance under its coverage. There are proposals that enable the functionality of ICN/CCN through exploiting the flexibility and valuable functionalities provided by SDN [180]–[184].

Assume that a user with a wearable gadget in a university campus requests a health service from the edge servers. This service may consist of analyzing the heartbeat stream for detecting medical urgency [185]. Figure 16 shows the sequential operations within an SDN-enabled service-centric environment. When the end-user requests the healthcare service by just identifying the name of the service, instead of specifying an IP address, the OpenFlow-enabled switch consults to the controller and the northbound application for determining the destination. After the control layer locates the servers that provide this service, it chooses the one that is the least loaded and installs the flow rules that create the path towards it. Then, the request is forwarded to the destination and the service response is sent to the end-user by following the reverse path.

Over the years, the rate of over-the-top (OTT) contents such as audio, video, messaging and VoIP traffic has increased and they limit the available bandwidth and similar resources for other applications [186]. In order to mitigate the burden from the core network and optimize the resource allocation, these contents and services may be provided by the edge servers. The operators in the case of MEC can reshape the network infrastructure and reveal the capabilities to OTT players and application developers to enable the innovation through flexible service and content deployment [187]. It is possible to maintain the current structure and organization for enabling OTT players to deploy their services at the edge and serve to end-users. However, the complexity at the edge and multiplicity of the service replications need to be managed carefully by the operators and Internet service providers (ISPs).

Although it is mentioned that the edge servers are owned personally or by the mobile network operators, ISPs will also be able to deploy edge servers within the local loop or at the end offices with the development of a new business model. However, operations between the user equipment (UE) and the Broadband Remote Access Servers (BRASs) are handled at Layer 2 and enabling the service-oriented access within this environment requires to use Layer 3 protocols. This infrastructural complexity prohibits to design and install the service-centric model, and utilize the edge servers with an optimized resource allocation. In order to overcome at least a portion of these intricacies, it is necessary for ISPs to make huge investments at the edge for redesigning the infrastructure and enabling the operations handled at Layer 3. In fact, with SDN technology, it is possible to avoid such large investments because SDN brings a cross-layer solution for the network traffic flows generated by UE. Since OpenFlow-enabled switches are not dedicated to any protocol or layer, and OpenFlow brings a flexible match/action scheme that handles Layer2-Layer4 fields, the SDN infrastructure may enable the essential operations without too much effort and cost. With the flexible and transparent solutions provided by SDN, a viable and innovative business model that covers the edge servers deployed and owned by ISPs may become feasible.

When all the mentioned challenges and requirements are kept in sight, it is seen that SDN is the best candidate for removing the barriers that prevent the coherent transformation into the service-oriented design. In the long run, it is believed that this design replaces the traditional network and to proliferate this process, SDN can be a remedy.

E. Network Function Virtualization (NFV) & Service Function Chaining

Network Function Virtualization (NFV) is another key technology that supports service orchestration [188]. In traditional networks, functions of a corresponding service to be provided by the operators are actually delivered as proprietary devices [33]. These functions should be connected to each other by following an order, and the data traffic should flow through the chain for carrying out the whole service [189]. The collection of these operations is defined as Service Function Chaining (SFC) by IETF [190].

As an example of SFC, a network service provided by the operator is decomposed into three components: (1) firewall, (2) Deep Packet Inspector (DPI) and (3) load balancer. In order to accomplish function chaining, any network packet generated by a client should initially flow through the firewall, then it should be forwarded to DPI. Lastly, the load balancer should process the packet and forward it to the destination server.

However, the current infrastructure of the network cannot easily tackle the strict requirements of network services and SFC. The major factors that complicate the operations are the varying characteristics and size of the data traffic, heterogeneous environment and dependence on the physical hardware. Bringing in a new service at a certain region or enhancing the performance of a service may require the deployment of a customized middlebox which is a costly and vendor-dependent operation [191].

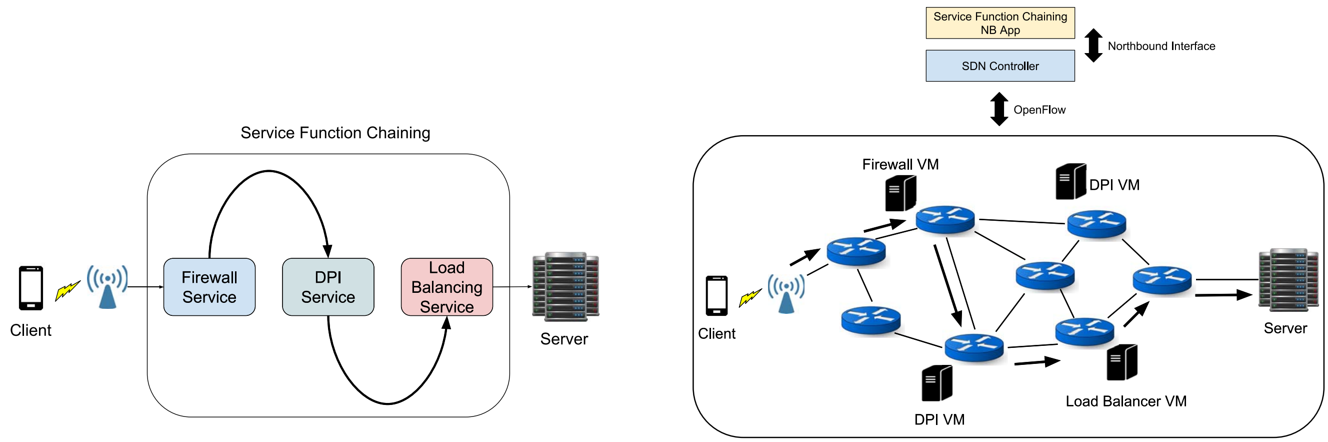
Although utilization of NFV improves the flexibility and enables the network services to be easily deployed [192], there is a need for an innovative orchestration mechanism that carry out the operations of SFC within an environment where functions are geographically distributed over the network. With NFV, any service function can be deployed over multitude of locations. This opportunity brings in the performance enhancement but as the scale gets larger, the orchestration of the functions and services gets more complex.

When we consider the features and benefits of SDN, it can be concluded that its operations can be utilized to leverage NFV by serving as the orchestrator. SDN can automate the service chaining through manipulating the flow rules at the forwarding nodes and provisioning the network connectivity [193], [194]. This complementary relationship between these two promising technologies also applies for the reverse direction. Since the SDN controller is implemented and deployed as a service, it can benefit from NFV in terms of reliability and elasticity [33].

With the smooth integration of NFV and SDN, the network services and their functions can be implemented as software and deployed at the edge of the network as virtualized implementations [195]. Therefore, this collaboration makes service orchestration more effective through the installation of a required network function at any location instantly when there is a necessity.

When we reconsider the aforementioned example case of SFC, these functions are implemented as software and deployed over servers by employing NFV. Building this environment on top of Edge Computing enhances the performance of SFC because deploying the functions closer to the end users decreases the latency and eliminates long-haul transmission of data traffic for carrying out the whole service [196].

Figure 17a illustrates an NFV environment where the firewall, the DPI and the load balancer are the functions of a network service that form a chain. Any network packet should flow through the sequential functions before arriving to the corresponding server. On the other hand, Figure 17b presents a more detailed environment where the same network functions are deployed over edge servers and the network is managed with an SDN controller. When a client application sends a network packet to a server, it is initially forwarded to the edge



(a) General overview of SFC

(b) SDN for enabling SFC between virtualized network functions

Fig. 17. The concept of service function chaining and the role of SDN.

server which provides the firewall function. This operation is accomplished by the SDN controller with the help of function discovery. As it is able to track the locations of functions, it can install the corresponding flow rules at the switches to form a chain that connects the sequential functions. After it passes through the firewall, it is forwarded to the DPI and the load balancer respectively, with the defined flow rules. After chaining operations are completed, the data packet is routed to the destined server.

The SDN controller can also behave in a reactive manner instead of installing pre-defined flow rules. For example, if firewall function is replicated over another edge server, SDN controller can acquire this information and dynamically modify the chain rules for the traffic at that geographic location. With the help of SFC northbound application, it instantly calculates the shortest path or the minimum latency path for completing the chain and draw a new route between the service functions. The main contribution of SDN in this architecture is forming such a chain that minimizes the latency by optimizing the route between functions. On the other hand, deployment of edge servers helps to decrease the latency further by keeping the application of network functions at the edge of the network.

There are proposals that aim to achieve the service chaining by incorporating the available features of SDN into NFV [197], [198]. As the data traffic generated by IoT gadgets increases at tremendous levels, the research in this area also focuses on combining the technologies of SDN, NFV, IoT and Edge Computing [199]. As a natural consequence of these advances in SDN and NFV, the requirement of installing new hardware for keeping pace with the increased network traffic is eliminated.

Beside of applying the SDN on NFV for practical deployments, there is an ongoing effort for the standardization of NFV [210]. IETF RFC 7665 [211] is published in 2015 and describes the architecture for creating and maintaining SFC operations. On the other hand, ETSI also focuses on the standardization of NFV and there are more than 50 group reports and specifications published by ETSI NFV ISG in the areas such as security, use cases and orchestration [212].

F. Ongoing Research on SDN and Edge Computing Cooperation

Edge Computing paradigm, or its ancestor cloud computing, is not directly related with the SDN implementations. However, when the requirements of edge servers and properties of SDN are paired up, it is observed that SDN is capable of filling the voids through the flexibility of programmable networks. There are already available tools for cloud computing management such as OpenStack [152] which has the supportive modules for incorporating the SDN control layer.

In order to depict the future direction and current status of the SDN and edge server technologies, related studies should be examined by focusing on the requirements of edge servers and functionalities of SDN. The proposals for SDN-Edge Computing integration are surveyed and summarized, and their key characteristics are presented in Table VI.

SAVI [213] is a multi-tier computing testbed that extends the traditional clouds in order to provide the required computational power at the edge and improve the QoS [214]. However, managing such a diverse network of devices spanning a wide area while hiding all the complexities throughout the network and handling the user mobility are clearly challenging issues. In order to deal with the complex objectives simultaneously, a control mechanism that embodies the scope of the whole network is a mandatory. SAVI provides an API for supporting SDN applications to manage the cloud and network resources.

Open Application Delivery Network (OpenADN) allows Application Service Providers (ASPs) to express and enforce traffic management policies and application delivery constraints because most applications recently need to serve global users [200]. Using OpenADN-aware data plane entities, Internet Service Providers (ISPs) can offer application delivery services to ASPs. In order to achieve these, OpenFlow, SDN, session splicing, cross-layer communication and application flow labels are main innovations. OpenADN allows an ASP's controller to communicate with ISP's controller to provide policies and states.

Ku *et al.* [201] propose a SDN based mobile cloud architecture, focusing on ad hoc networks. They also present the requirements for building such architecture. The main

TABLE VI
ONGOING RESEARCH IN SDN-EDGE COMPUTING COOPERATION

Research	Server Type	Scope	The Role of SDN
OpenADN [200]	Cloud	Traffic management, application delivery constraints	Supporting distinct control plane applications
Ku et al. [201]	Ad-hoc	Mobile data traffic	Frequency allocation, wireless network virtualization, managing the traffic
MOCA [202]	Cloud	What and when to offload from mobile devices	Data plane redirection mechanism through flow rule installations
HomeCloud [203]	Edge Servers	Application delivery at the edge	Configuring and managing VNFs at the edge
Xu et al. [204]	Fog Servers	Service orchestration for IoT	Implementing the broker functionality at the switches
Salahuddin et al. [205]	Road Side Units	Internet of Vehicles	Reconfiguring the services within the network, data forwarding
Amraoui and Sethom [206]	Cloudlet	Pervasive healthcare service for wireless sensors	Detecting the topology, device discovery, retrieving sensor status
Monfared et al. [207]	Cloud & Edge Servers	Heterogeneous access networks	Enabling programmable radio, virtualizing resources
UbiFlow [208]	Edge Server	Heterogeneous IoT networks	Mobility management, device configuration, flow control
FSDN [209]	Fog Servers	Vehicular Ad-Hoc Networks (VANET)	Improving the connectivity, enhancing scalability, supporting mobility
WiCloud [105]	MEC	Cloud functionality at the edge with OpenStack	Enabling location-awareness, managing inter-MEC communications

motivation of this study is gaining the advantage of wireless SDN in order to manage and control the growth of mobile data traffic. In this architecture, wireless SDN nodes act as both switches and hosts. These nodes communicate with the SDN controller via a long range broadband wireless network such as LTE and they communicate with each other via a high bandwidth connection such as a WLAN. Each wireless SDN node has its own local SDN controller, which is seen as a backup controller, and different interfaces. These interfaces may have static frequency channels or their frequency channels may be changed by the global controller according to the need of the applications and traffic types. The global SDN controller has knowledge about the interfaces and their frequencies. By this, a wireless network virtualization can be achieved through slices and a specific frequency can be reserved in case of an emergency for prioritization. They simulated the mobility and handover mechanisms of their architecture using NS3.

Offloading is considered as one of the most attractive research areas of mobile computing, especially MCC. Deciding on what to offload and how to offload are the main issues of MCC which may lead to an increase in the energy consumption unnecessarily. In order to overcome this problem, some of the studies focus on offloading by combining it with SDN. MOCA [202] is a lightweight Mobile Cloud Offloading Architecture that enables offloading the data to in-network cloud computing platforms. The main contribution of this study is adopting SDN into this architecture. The duty of SDN within this architecture is using the data plane redirection mechanisms to help traffic to reach the in-network cloud offloading platform that is mentioned. By utilizing this architecture, offloading can be done in a more efficient way.

HomeCloud [203] is a framework that combines NFV and SDN paradigms for enabling an efficient application delivery and orchestration in the servers that are deployed at the edge. Within this framework, NFV is utilized for local computation and storage through the virtualized functionalities at the edge, which are managed and orchestrated by SDN. This framework targets the scenarios including applications such as smart home, IoT and augmented reality. Similarly, SDN and NFV cooperation is utilized as the orchestrators for 5G networks that combines Fog and cloud nodes to handle the mobile devices and massive data generated by them [215].

A Fog computing structure based on SDN is proposed by Xu *et al.* [204] which provides the services required for the data generated by IoT devices. This framework uses MQTT (Message Queuing Telemetry Transport) protocol for data transfer between IoT devices and the remote servers. They modified the edge switch and incorporate the functionality of broker node, which is then called as Fog node. In addition to this, in order to make Fog node behave as a computing node, they integrated SDN controller within the OpenvSwitch.

In [205], a novel roadside unit (RSU) is proposed as a backbone of Internet of Vehicles (IoV) system which is based on traditional IoT. This study proposes to utilize SDN for managing the RSU network in order to keep up with the dynamic nature of the network and efficient service procurement. The vehicles are able to communicate with the RSU cloud and the communication within this cloud is managed by an OpenFlow controller.

The architecture proposed by Amraoui and Sethom [206] aims to provide an environment where heterogeneous wireless sensors for pervasive healthcare can operate in a compatible manner. Through SDN, this framework can enable fast access for data collection and analysis through the Cloudlet nodes. The topology information, device discovery and sensor status are controlled by the SDN layer within the framework. On top of that, a Cloudlet orchestrator is implemented which provides an API for the Cloudlet provider in order to control the underlying system.

Monfared *et al.* [207] designed a two-tier cloud architecture that is composed of cloud data centers and edge devices for bringing the nodes closer to the user side. For the control and management system of this architecture, Software-defined Infrastructure is highlighted which brings flexibility on the architecture with heterogeneous devices.

With the dramatic increase in the number of IoT devices and increase in the traffic between these devices and remote servers, the management of IoT devices became an important challenge [157]. This study suggests that SDN and OpenFlow provide a flexible environment for home networking system.

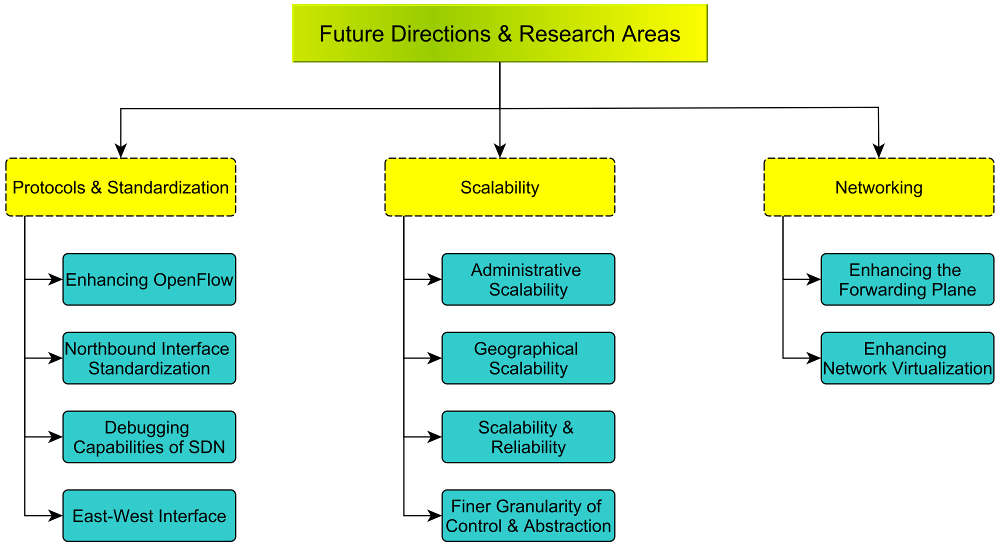


Fig. 18. Research directions for enhancing SDN for edge computing.

UbiFlow [208] is a system that combines SDN and IoT to provide ubiquitous flow control and manage mobility in the network. This system exploits the distributed control mechanism of SDN controllers in different geographic locations to maintain the scalability. The reason for utilizing SDN in such system is keeping up with the changing demands and traffic volumes provided by SDN in a flexible manner.

FSDN [209] is a Vehicular Ad-hoc Network (VANET) which combines Fog Computing and SDN technologies because it is seen that VANETs suffer from poor connectivity and less scalability for lots of years. The flexibility, programmability and centralized control mechanism of SDN makes it an inevitable component of a VANET architecture. SDN takes the heterogeneous features such as mobility, physical medium and topology into consideration and integrated into the system to resolve related problems. SDN controller, SDN wireless nodes (data plane elements), SDN road-side-unit (Fog device), SDN road-side-unit controller (Fog devices under the orchestration of the controller) and cellular base station are the SDN components of FSDN architecture.

WiCloud [105] is a MEC platform that is based on NFV and SDN. This platform uses OpenStack for providing cloud functionality at the edge. The SDN controller manages the communication between MEC servers which form a data center at the edge. The main objective of this platform is to provide proximity and location awareness that are promised by MEC.

VI. Directions to Enhance SDN for Further Contribution to Edge Computing

Cloud and Edge Computing interaction seems as an exciting architecture with the involvement of SDN as their driving force. The previous section presents the advances in SDN and its intrinsic properties that are highly possible to leverage the hidden potential of the Edge Computing paradigm. Since both Edge Computing and SDN have not reached maximum level of their potential yet, the architecture that is composed of these exciting concepts may not meet the expectations and may not achieve the desired performance. Neither SDN nor its current de facto standardization in the form of OpenFlow is mature enough to handle all possible use cases and managerial operations that are pointed out. This immaturity comes from several factors but the most important ones are:

- Requirements are changing fast, even the relatively more standard domains like IaaS (infrastructure as a service) based cloud is evolving.
- Requirements are getting sophisticated.
- Integrating SDN into existing networks is not easily achieved by just deploying SDN-enabled networking devices [216].
- Investments in hardware is expensive, therefore SDN-enabled hardware cannot get to the required level in terms of capability and deployment [217].

SDN scenarios based on IoT and Edge Computing surpass the initial vision of SDN and OpenFlow which revolved around campus and telecommunication networks. In this section, we endeavor to present and discuss what is technically lacking in the current SDN landscape to realize the novel scenarios of Edge Computing and propose a direction towards which possible solutions may be implemented. These discussions are summarized and presented in Figure 18.

A. Enhancing Network Virtualization

Network virtualization is an essential functionality that leverages abstraction and sharing of the underlying infrastructure which eventually improves the resource management and the overall utilization. Network virtualization, NFV and SDN are recently considered as inseparable terms and technologies because of their inter-beneficial operations. Their cooperation takes an important part in realizing the novel use cases that are enabled by the Edge Computing. As stated by Feamster *et al.* [218], SDN proves its success and offers technologies in support of the specific use case of network virtualization. Although available SDN hypervisors for network virtualization are feasible for abstraction and isolation, it is observed that there is a lack of complete performance evaluation [219]. Reliability, abstraction, performance, scalability and security of these platforms need to be studied in detail for creating a complete solution for virtualizing the network infrastructure by exploiting the capabilities of SDN.

On the other hand, SDN can support the network virtualization in wireless systems such as LTE [220]. However, virtualization in wireless and mobile networks is more challenging and complicated [31]. Since wireless communication plays an important role in Edge Computing for accessing servers, virtualization of wireless networks should also be given importance. Since the initial design of SDN and OpenFlow covers the wired networks, more work needs to be done so that the benefits of SDN in virtualization can be applied to the wireless networks as well.

B. Northbound Interface Standardization

Although OpenFlow comes to mind firstly when it comes to SDN, the northbound interface also has the same level of responsibility and importance in controlling the underlying network infrastructure. The role of the northbound interface in SDONs is still raw for the time being and research in this area is more challenging since there are multitude of interfaces that should be managed in physical and transport layers [221]. An efficient communication between the network applications and the controller requires a simple vendor-independent interface.

Instead of generally applicable ones, special API's for particular purposes will emerge that can cover the most common controllers. At 2014, ONF has formed a working group that carries out intensive studies in order to standardize the northbound interface [222]. It is believed that this standardization process will lead to a concrete acceleration in the SDN innovation because the northbound interface is the essential point for carrying out the control operations as complete. Also, standardizing this API leads to a variety of new applications which improves the control over the underlying network infrastructure.

C. Enhancing OpenFlow

The latest version of OpenFlow (v1.5.1) offers a variety of matching fields from Layer2 to Layer4 with the *OXM_MATCH* property. It is discussed that the network is on the way of being transformed from host-centric to service-centric networking because of the increasing number of services. Although this evolution in networking seems as the fundamental solution for hiding the complex operations from the end-users, there is still a long way to go. On one hand, it requires some large scale modifications in the traditional networking operations and protocols, on the other hand, SDN and OpenFlow also need some advances and improvements in order to support the service-centric environment properly.

OpenFlow assumes that forwarding nodes have a fixed behavior and there is a pre-defined set of supported protocols. At the time being, there are more than 40 fields that can be used for matching and it is increasing. This progress actually prohibits the flexibility that is promised by SDN, and service-centricness even require more flexible match/action schemes. Therefore, improvements in this area should preserve the flexibility aimed by SDN originally. It is expected that new matching fields will be implemented in the next versions of OpenFlow for supporting protocol independent behaviors.

As a complementary solution, P4 (Programming Protocol-Independent Packet Processors) language [223] is proposed as a way to enable the programmability of the switch behaviors, which is assumed as fixed by OpenFlow. The emergence of P4 changes the way forwarding nodes process the packets. By doing so, the modular specification and protocol independence are achieved through a module implemented with P4.

P4 have the potential to leverage service-centricness with the ability of defining headers that switches can recognize. By incorporating P4 into the system and adding some modifications to OpenFlow, the switches and the network becomes fully flexible and programmable. Hence, the flexible matching mechanism without modifying the existing TCP/IP protocol stack will enable the service-centric networks. However, the mechanisms that enable P4 and OpenFlow to work together are not studied in detail yet.

D. Finer Granularity of Control & Abstraction

The initial SDN proposal covers a campus network that is managed by a single controller. As a result of the large expansion of the covered area and the reliability problems, logically centralized but physically distributed control mechanisms are proposed. In such a large scale network, abstraction and finer granularity of control plays an important role [224].

Moreover, one of the most important challenges for SDN is generally ignored in practice: Gradually transforming a traditional network to an SDN-enabled one [225]. There is a need for the incremental deployment of SDN which leads to a hybrid network that is composed of both legacy networking devices and SDN-enabled devices. In order to make this hybrid environment feasible by extending SDN benefits, abstraction requires additional consideration [226].

Still today, most of the SDN-enabled switches are not pure OpenFlow devices. The vast majority of them consist of hybrid switches, which are just traditional networking devices with the ability of communicating through OpenFlow. The supported version of OpenFlow may differ and this may cause additional problems for the networks, especially for the ones that are composed of multiple domains. Because of this limitation, the current circumstances create a barrier for the SDN integration to provide a finer granularity of control. They need to support the common OpenFlow version and hybrid switches should not cause vendor-related problems [227].

E. Debugging Capabilities of SDN

Shifting from hardware-based behavior to a software-based one generates a natural outcome which is related to the software errors [99]. Both SDN controllers and northbound applications are software so it is inevitable to encounter with errors. As the network scale gets larger for Edge Computing, the software for realizing different functionalities become more complex and prone to errors. Installing northbound applications, distributing the control mechanism and increasing the control level for the cooperation of cloud and Edge Computing infrastructure require additional implementations for the software. The bugs that stay hidden within the code of the controller or the switch may cause an incompatibility between the forwarding and control planes [228]. For all these situations, debugging capabilities of SDN needs to be improved for further implementations.

F. East-West Interface & Geographical Scalability

The SDN is mostly commercially viable inside a datacenter for IaaS perspective and in a campus environment. However, new scenarios dictate novel traffic characteristics and these traffic flows may need to be forwarded through multiple domains. As a result of distinct domains that may belong to different service providers, using multiple controllers where at least one controller is responsible for a single domain becomes a necessity. The controller of each domain is responsible for its own domain but there should be an inter-domain communication in order to provide reliable data transfer [229]. This type of communication between adjacent controllers is provided by the east-west interface. Since the applications of SDN do not highly depend on the east-west communication, the research on this area is still at its beginning. The secure, reliable and efficient communication needs to be provided for controllers.

In addition to all these requirements, with the inclusion of SDON in the system, the east-west interface gains more importance, since the communication between the controllers that manage different type of infrastructures are transmitted through multiple transport technologies. In the case of SDON, standardization of this interface may improve the operations to a greater level since an optical network may cover a large area consisting multiple domains [221]. The network infrastructure of the multi-tier architecture should be able to meet the requirements of the communications between large number of domains and support various network devices and technologies. SDON are not only used for the cloud tier, but also for the communication between metro-scale datacenters [230]. Thus, the tiers reside above the edge tier are possibly enabled through the optical networks version of SDN.

Not only OpenFlow-enabled networks, but also non-SDN infrastructures should be taken into consideration where multi-tier architecture operates. While the eastbound interface connects an SDN domain to a non-SDN domain, the westbound interface serves for providing the necessary communication for multiple OpenFlow-enabled domains. It would be not a trivial idea to consider the presence of non-SDN domains in the network backbones. Thus, combining all these requirements show a clear direction that it is essential to provide a standard communication between the adjacent domain controllers through the east-west interface.

When talking about geographical scalability, it is also necessary to consider the metro-scale datacenters. Although there are studies focus on this area [231], [232], server placement in the Metropolitan area need to be studied in detail and analyzed further. This challenging aspect explicitly affect the operations of SDN since locations of the controllers, assuming that a distributed control mechanism is necessary for this scale, need to be determined accordingly.

G. Administrative Scalability

In novel architectures such as the multi-tier edge system, administrative domains are more flexible and they may overlap in the case of a virtualization. In the case of Edge Computing, a complicated business interaction and service models may emerge. SDN should be flexible enough to embrace them. At the same time, a secure environment needs to be maintained for the increasing number of administrative domains. Considering a scenario [233], a mobile user may be migrated between distinct administrative domains. The devices at the border of these domains may fail to discover other devices which are outside of their region, even though they are close to each other. Therefore, a synchronization between Cloudlets at the borders and an efficient discovery mechanism are mandatory.

H. Scalability & Reliability

Scalability could be mentioned as an issue of SDN because one central controller may not be sufficient for controlling all of the forwarding elements [27]. A report published by ONF in 2016 [234] states that scaling OpenFlow to millions of flows in a large environment is still a challenging aspect that needs to be addressed. Since the whole operation of the SDN depends on the controller, a logically centralized but physically distributed controller mechanism is efficient to utilize in large-scale networks in order to lower the load on the controllers and leverage a resilient control layer [235].

ONOS (Open Network Operating System) [236] project partially addresses this problem by providing a cluster-based controller mechanism. Besides, there are several proposals that utilize distinct methodologies for achieving the distributed controller architecture and improving the performance of it [237]–[240]. Although there is a lot of effort and work in this area, there is still not a complete proposal that consider each of the important aspects for enabling a fully operational and high performance distributed control mechanism. As a result of this, scalability and reliability are currently two important future directions that need to be elaborated in detail.

I. Enhancing the Forwarding Plane

As SDN is being improved, the control plane becomes more resilient, flexible and scalable with the new implementations of OpenFlow. On the other hand, as the network becomes complex and traffic generated at the edge multiplies, the switching capacities of the network elements become the bottleneck for the network performance. The processing power of software-based controllers can be improved by parallel processing or installing the controller on a more powerful server but the same cannot be applied for the OpenFlow-enabled switches. The current data plane elements are in lack of enough CPU power for the control operations since they are not originally designed for handling a large number of OpenFlow messages [241].

As discussed, additional fields are required for OpenFlow in the near future to fully-support service-centric networks. In addition to these modifications, it is important to pay sufficient attention that all these fields consume the TCAM type memory on the network devices. Similar to the switching capabilities, memory capacities are also need to be improved in the near future to support the extensions of OpenFlow and other needs such as preventing the overload on the switch in the case of an attack [216]. In the long term, it is expected that high performance white box switches, such as Pica8 [242], will appear more in the market and replace the current hybrid switches.

VII. Conclusion

Edge Computing is an umbrella term covering the latest trend of bringing the computational resources to the proximity of the end devices. It is driven by the widespread adoption of IoT, and a large spectrum of small form factor gadgets and mobile devices. These end devices need to be computation-wise complemented and legacy cloud computing is not the best fit and further alternatives need to be sought. In this work, we provide a survey on the literature of the constituent technologies of the Edge Computing paradigm, namely Cloudlets, Fog Computing and MEC. With this effort, we aim to provide a clear big picture for the Edge Computing domain and also underline the technical nuances between the approaches available.

Although very promising and tries to fulfill a technical gap, Edge Computing still needs to resolve various technical challenges to become thoroughly pervasive. In our work, we have categorized and discussed these challenges to provide a deeper insight for the complexities involved in the practical implementations of the Edge Computing. The remedy we proposed for overcoming the technical barriers involved is to employ the network programmability approach provided by SDN. SDN, like Edge Computing, is a hot technological trend, however, its full potential is far from being realized and it is currently evolving. In this paper, we endeavored to take a novel approach and technically align the capabilities of the SDN with the shortcomings of Edge Computing. To achieve this goal with the necessary technical depth and sophistication, we first give an account on the motivation and technical background on SDN and Edge Computing cooperation by giving real life use cases. Later, we refined a set of "benefit areas" where interplay between the aforementioned technologies can render efficient and feasible designs for bringing low cost computational solutions into the proximity of the modern day edge devices.

Most notable of these benefit areas are: High resolution and effective control over the resulting Edge Computing facility, flexibility, enabling innovation, ease of implementation for service-centric approach, adaptability, interoperability with the OpenFlow protocol itself and portfolio of commercial SDN products, lower cost solutions, VM mobility at the edge and multiplicity of scope. We surveyed the current literature and presented the few examples that currently exist on the cooperation of SDN and Edge Computing.

As another and final important contribution of this paper, we laid down the directions for SDN standardization so that it can further be improved in its interaction with Edge Computing. We presented a detailed account of the possible room for improvement of the SDN functionality in all axes with the Edge Computing cooperation. We believe that SDN has the true potential of significantly improving many computational and networking scenarios of the future and it is of utmost importance to align SDN evolvement with actual service requirements.

References

[1] M. Weiser, "The computer for the 21st century," *Sci. Amer. Special Issue Commun. Comput. Netw.*, vol. 3.3, pp. 3–11, Sep. 1991.

[2] *IEEE Standards Association, Internet of Things Related Standards*. Accessed on Aug. 2016. [Online]. Available: http://standards.ieee.org/innovate/iot/stds.html

[3] S. Husain, A. Prasad, A. Kunz, A. Papageorgiou, and J. Song, "Recent trends in IoT/M2M related standards," *Tsp*, vol. 4, no. S6m, p. S6n, 2014.

[4] *Thread Group, Thread*. Accessed on Jan. 2017. [Online]. Available: http://www.threadgroup.org/

[5] *Open Interconnect Consortium, Open Interconnect*. Accessed on Jan. 2017. [Online]. Available: http://openinterconnect.org/

[6] R. Stackowiak, A. Licht, V. Mantha, and L. Nagode, "Internet of Things standards," in *Big Data and the Internet of Things*. New York, NY, USA: Apress, 2015, pp. 185–190.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, 2013.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM 1st Edition MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, 2012, pp. 13–16.

[11] H. H. Pang and K.-L. Tan, "Authenticating query results in edge computing," in *Proc. IEEE 20th Int. Conf. Data Eng.*, Boston, MA, USA, 2004, pp. 560–571.

[12] European Telecommunications Standards Institute. *Mobile-Edge Computing (MEC) Terminology*. Accessed on May 2017. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/01.01.01_60/gs_MEC001v010101p.pdf

[13] European Telecommunications Standards Institute. *Multi-Access Edge Computing*. Accessed on May 2017. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[14] J. S. Preden *et al.*, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, Jul. 2015.

[15] X. Xiang, C. Lin, and X. Chen, "EcoPlan: Energy-efficient downlink and uplink data transmission in mobile cloud computing," *Wireless Netw.*, vol. 21, no. 2, pp. 453–466, 2015.

[16] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Comput. Netw.*, vol. 72, pp. 74–98, Oct. 2014.

[17] V. R. Tadinada, "Software defined networking: Redefining the future of Internet in IoT and cloud era," in *Proc. IEEE Int. Conf. Future Internet Things Cloud (FiCloud)*, Barcelona, Spain, 2014, pp. 296–301.

[18] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," *ACM Comput. Surveys*, vol. 47, no. 1, p. 7, 2014.

[19] S. Bera, S. Misra, and J. J. P. C. Rodrigues, "Cloud computing applications for smart grid: A survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1477–1494, May 2015.

[20] A. ur Rehman Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart., 2014.

[21] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 133–143, 2014.

[22] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[23] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A survey of cloudlet based mobile computing," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Shanghai, China, 2015, pp. 268–275.

[24] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Mobidata*, Hangzhou, China, 2015, pp. 37–42.

[25] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th IEEE Int. Conf. Intell. Syst. Control (ISCO)*, Coimbatore, India, 2016, pp. 1–8.

[26] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[27] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Quart., 2014.

[28] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[29] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 4th Quart., 2014.

[30] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Comput. Netw.*, vol. 81, pp. 79–95, Apr. 2015.

[31] M. Yang *et al.*, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 4–18, 2015.

[32] N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey," *ACM Comput. Surveys*, vol. 47, no. 2, p. 27, 2015.

[33] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[34] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[35] *Open Networking Foundation, OF-Config*. Accessed on Oct. 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1-1-1.pdf

[36] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 4, pp. 373–392, Dec. 2012.

[37] R. Want, B. N. Schilit, and S. Jenson, "Enabling the Internet of Things," *Computer*, vol. 48, no. 1, pp. 28–35, Jan. 2015.

[38] B. Yang, W. K. Chai, G. Pavlou, and K. V. Katsaros, "Seamless support of low latency mobile applications with NFV-enabled mobile edge-cloud," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 136–141.

[39] M. F. Ramdhani, S. N. Hertiana, and B. Dirgantara, "Multipath routing with load balancing and admission control in software-defined networking (SDN)," in *Proc. 4th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Bandung, Indonesia, 2016, pp. 1–6.

[40] J. Li, X. Chang, Y. Ren, Z. Zhang, and G. Wang, "An effective path load balancing mechanism based on SDN," in *Proc. IEEE 13th Int. Conf. Trust Security Privacy Comput. Commun. (TrustCom)*, Beijing, China, 2014, pp. 527–533.

[41] R. Wang, D. Butnariu, and J. Rexford, "Openflow-based server load balancing gone wild," in *Proc. Hot-ICE*, vol. 11. Boston, MA, USA, 2011, p. 12.

[42] Z. Guo *et al.*, "Improving the performance of load balancing in software-defined networks through load variance-based synchronization," *Comput. Netw.*, vol. 68, pp. 95–109, Aug. 2014.

[43] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-serve: Load-balancing Web traffic using OpenFlow," *ACM Sigcomm Demo*, vol. 4, no. 5, p. 6, 2009.

[44] S. Sathyanarayana and M. Moh, "Joint route-server load balancing in software defined networks using ant colony optimization," in *Proc. Int. Conf. High Perform. Comput. Simulat. (HPCS)*, Innsbruck, Austria, 2016, pp. 156–163.

[45] J. Liu, Y. Li, and D. Jin, "SDN-based live VM migration across datacenters," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 583–584, 2014.

[46] P. Raad, S. Secci, C.-D. Phung, and P. Gallard, "PACAO: A protocol architecture for cloud access optimization in distributed data center fabrics," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–9.

[47] S. Secci, P. Raad, and P. Gallard, "Linking virtual machine mobility to user mobility," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 927–940, Dec. 2016.

[48] P. Pocatilu, F. Alecu, and M. Vetrici, "Measuring the efficiency of cloud computing for e-learning systems," *WSEAS Trans. Comput.*, vol. 9, no. 1, pp. 42–51, 2010.

[49] M. Firdhous, O. Ghazali, and S. Hassan, "Fog computing: Will it be the future of cloud computing?" in *Proc. 3rd Int. Conf. Inf. Appl.*, Kuala Terengganu, Malaysia, 2014, pp. 8–15.

[50] A. Bahtovski and M. Gusev, "Cloudlet challenges," *Procedia Eng.*, vol. 69, pp. 704–711, Mar. 2014.

[51] S. Nunna and K. Ganesan, "Mobile edge computing," in *Health 4.0: How Virtualization and Big Data Are Revolutionizing Healthcare*. Cham, Switzerland: Springer, 2017, pp. 187–203.

[52] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[53] R. M. Shukla and A. Munir, "A computation offloading scheme leveraging parameter tuning for real-time IoT devices," in *Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst. (iNIS)*, Gwalior, India, 2016, pp. 208–209.

[54] M. Ahmadi, N. Khanezaei, S. Manavi, F. F. Moghaddam, and T. Khodadadi, "A comparative study of time management and energy consumption in mobile cloud computing," in *Proc. IEEE 5th Control Syst. Graduate Res. Colloquium (ICSGRC)*, Shah Alam, Malaysia, 2014, pp. 199–203.

[55] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "A context sensitive offloading scheme for mobile cloud computing service," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, New York, NY, USA, 2015, pp. 869–876.

[56] M. Nir, A. Matrawy, and M. St.-Hilaire, "An energy optimizing scheduler for mobile cloud computing environments," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2014, pp. 404–409.

[57] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 2014, pp. 2239–2244.

[58] K. Ha *et al.*, "The impact of mobile multimedia applications on data center consolidation," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, Redwood City, CA, USA, 2013, pp. 166–176.

[59] M. T. Beck, S. Feld, A. Fichtner, C. Linnhoff-Popien, and T. Schimper, "ME-VoLTE: Network functions for energy-efficient video transcoding at the mobile edge," in *Proc. 18th Int. Conf. Intell. Next Gener. Netw. (ICIN)*, Paris, France, 2015, pp. 38–44.

[60] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020 White Paper*. Accessed on Mar. 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[61] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. 6th Int. Conf. Adv. Future Internet*, 2014.

[62] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[63] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[64] D. Thomas, "Cloud computing-benefits and challenges!" *J. Object Technol.*, vol. 8, no. 3, pp. 37–41, 2009.

[65] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principals, architectures, and applications," *arXiv preprint arXiv:1601.02752*, 2016.

[66] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

[67] M. Femminella, M. Pergolesi, and G. Reali, "Performance evaluation of edge cloud computing system for big data applications," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 170–175.

[68] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—The business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, 2011.

[69] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.

[70] R. K. Lomotey and R. Deters, "Architectural designs from mobile cloud computing to ubiquitous cloud computing—Survey," in *Proc. IEEE World Congr. Services (SERVICES)*, Anchorage, AK, USA, 2014, pp. 418–425.

[71] Y.-S. Chang, C.-T. Fan, W.-T. Lo, W.-C. Hung, and S.-M. Yuan, "Mobile cloud-based depression diagnosis using an ontology and a Bayesian network," *Future Gener. Comput. Syst.*, vols. 43–44, pp. 87–98, Feb. 2015.

[72] M. Chen, Y. Hao, Y. Li, C.-F. Lai, and D. Wu, "On the computation offloading at ad hoc cloudlet: Architecture and service modes," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 18–24, Jun. 2015.

[73] H. Raei and N. Yazdani, "Performability analysis of cloudlet in mobile cloud computing," *Inf. Sci.*, vols. 388–389, pp. 99–117, May 2017.

[74] Y. Jararweh, L. Tawalbeh, F. Ababneh, and F. Dosari, "Resource efficient mobile computing using cloudlet infrastructure," in *Proc. IEEE 9th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dalian, China, 2013, pp. 373–377.

[75] N. Fernando, S. W. Loke, and W. Rahayu, "Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing," in *Proc. 4th IEEE Int. Conf. Utility Cloud Comput. (UCC)*, 2011, pp. 281–286.

[76] Y. Wang, I.-R. Chen, and D.-C. Wang, "A survey of mobile cloud computing applications: Perspectives and challenges," *Wireless Pers. Commun.*, vol. 80, no. 4, pp. 1607–1623, 2015.

[77] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *arXiv preprint arXiv:1602.00484*, 2016.

[78] M. Rahman, J. Gao, and W.-T. Tsai, "Energy saving in mobile cloud computing*," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, Redwood City, CA, USA, 2013, pp. 285–291.

[79] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 369–392, 1st Quart., 2014.

[80] M. Satyanarayanan *et al.*, "Edge analytics in the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 24–31, Apr./Jun. 2015.

[81] M. Satyanarayanan *et al.*, "Cloudlets: At the leading edge of mobile-cloud convergence," in *Proc. 6th Int. Conf. Mobile Comput. Appl. Services (MobiCASE)*, Austin, TX, USA, 2014, pp. 1–9.

[82] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, "Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges," *J. Netw. Comput. Appl.*, vol. 62, pp. 18–40, Feb. 2016.

[83] B. Li, Y. Pei, H. Wu, and B. Shen, "Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds," *J. Supercomput.*, vol. 71, no. 8, pp. 3009–3036, 2015.

[84] X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, "Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds," *Wireless Netw.*, pp. 1–10, 2016.

[85] *Open Edge Computing*. Accessed on Mar. 2017. [Online]. Available: http://openedgecomputing.org/about-oec.html

[86] A. Ceselli, M. Premoli, and S. Secci, "Cloudlet network design optimization," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, Toulouse, France, 2015, pp. 1–9.

[87] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.

[88] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Warsaw, Poland, 2014, pp. 1–8.

[89] *OpenFog Consortium*. Accessed on Jan. 2017. [Online]. Available: https://www.openfogconsortium.org/about-us/#introduction

[90] *OpenFog Consortium—Out of the Fog: Use Case Scenarios*. Accessed on Mar. 2017. [Online]. Available: https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Transportation-Drone-Delivery-Use-Case.pdf

[91] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. Int. Conf. Future Internet Things Cloud (FiCloud)*, Barcelona, Spain, 2014, pp. 464–470.

[92] T. H. Luan *et al.*, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[93] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing-based face identification and resolution scheme in Internet of Things," *IEEE Trans. Ind. Informat.*, to be published.

[94] M. Satyanarayanan *et al.*, "An open ecosystem for mobile-cloud convergence," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 63–70, Mar. 2015.

[95] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile Internet: Vision, recent progress, and open challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 138–145, May 2016.

[96] European Telecommunications Standards Institute Industry Specifications Group. *Mobile-Edge Computing—MEC Metrics Best Practice and Guidelines*. Accessed on Mar. 2017. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf

[97] E. Cau *et al.*, "Efficient exploitation of mobile edge computing for virtualized 5G in EPC architectures," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput. Services Eng. (MobileCloud)*, Oxford, U.K., 2016, pp. 100–109.

[98] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[99] S. Nunna *et al.*, "Enabling real-time context-aware collaboration through 5G and mobile edge computing," in *Proc. 12th Int. Conf. Inf. Technol. New Gener. (ITNG)*, Las Vegas, NV, USA, 2015, pp. 601–605.

[100] M. Sapienza *et al.*, "Solving critical events through mobile edge computing: An approach for smart cities," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, St. Louis, MO, USA, 2016, pp. 1–5.

[101] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *Commun. Surveys Tuts.*, to be published.

[102] *European Telecommunications Standards Institute Industry Specifications Group, MEC Proofs of Concept*. Accessed on Feb. 2017. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing/mec-poc

[103] *European Telecommunications Standards Institute, PoC 2 Edge Video Orchestration and Video Clip Replay*. Accessed on Feb. 2017. [Online]. Available: http://mecwiki.etsi.org/index.php?title=PoC_2_Edge_Video_Orchestration_and_Video_Clip_Replay_via_MEC

[104] *European Telecommunications Standards Institute, PoC 7 Multi-Service MEC Platform for Advanced Service Delivery*. Accessed on Feb. 2017. [Online]. Available: http://mecwiki.etsi.org/index.php?title=PoC_7_Multi-Service_MEC_Platform_for_Advanced_Service_Delivery

[105] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: Progress and challenges," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput. Services Eng. (MobileCloud)*, Oxford, U.K., 2016, pp. 83–84.

[106] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Gener. Comput. Syst.*, vol. 70, pp. 138–147, May 2017.

[107] *European Telecommunications Standards Institute Industry Specifications Group, Mobile-Edge Computing—Service Scenarios*. Accessed on Jan. 2017. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf

[108] M. Quwaider and Y. Jararweh, "Cloudlet-based for big data collection in body area networks," in *Proc. 8th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, London, U.K., 2013, pp. 137–141.

[109] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *Proc. 2nd Int. Symp. Future Inf. Commun. Technol. Ubiquitous HealthCare (Ubi HealthTech)*, Beijing, China, 2015, pp. 1–5.

[110] K. Ha *et al.*, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2014, pp. 68–81. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594383

[111] V. S. Achanta, N. T. Sureshbabu, V. Thomas, M. L. Sahitya, and S. Rao, "Cloudlet-based multi-lingual dictionaries," in *Proc. 3rd Int. Conf. Services Emerg. Markets (ICSEM)*, Mysore, India, 2012, pp. 30–36.

[112] H. Dubey *et al.*, "Fog data: Enhancing telehealth big data through fog computing," in *Proc. ASE BigData SocialInformat.*, 2015, Art. no. 14.

[113] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. Aust. Telecommun. Netw. Appl. Conf. (ATNAC)*, Southbank, VIC, Australia, 2014, pp. 117–122.

[114] M. Aazam and E.-N. Huh, "Fog computing: The cloud-IOT/IOE middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May/Jun. 2016.

[115] Y. Liu *et al.*, "Citysee: Not only a wireless sensor network," *IEEE Netw.*, vol. 27, no. 5, pp. 42–47, Sep./Oct. 2013.

[116] M. Quwaider and Y. Jararweh, "Cloudlet-based efficient data collection in wireless body area networks," *Simulat. Model. Pract. Theory*, vol. 50, pp. 57–71, Jan. 2015.

[117] M. Satyanarayanan *et al.*, "The role of cloudlets in hostile environments," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 40–49, Oct./Dec. 2013.

[118] Q. Yaseen, F. AlBalas, Y. Jararweh, and M. Al-Ayyoub, "A fog computing based system for selective forwarding detection in mobile wireless sensor networks," in *Proc. IEEE Int. Workshops Found. Appl. Self* Syst.*, Augsburg, Germany, 2016, pp. 256–262.

[119] S. Agarwal, M. Philipose, and P. Bahl, "Vision: The case for cellular small cells for cloudlets," in *Proc. 5th Int. Workshop Mobile Cloud Comput. Services*, 2014, pp. 1–5.

[120] L. Gao, T. H. Luan, B. Liu, W. Zhou, and S. Yu, "Fog computing and its applications in 5G," in *5G Mobile Communications*. Cham, Switzerland: Springer, 2017, pp. 571–593.

[121] P. Du and A. Nakao, "Application specific mobile edge computing through network softwarization," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 130–135.

[122] Y. Jararweh *et al.*, "The future of mobile cloud computing: Integrating cloudlets and mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, Thessaloniki, Greece, 2016, pp. 1–5.

[123] J. K. Zao *et al.*, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. Int. Conf. Intell. Environ. (IE)*, Shanghai, China, 2014, pp. 374–377.

[124] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, 4th Quart., 2011.

[125] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1507–1516, Oct. 2007.

[126] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: A survey and future perspectives," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 98–104, Feb. 2016.

[127] M. Kaur, J. Martin, and H. Hu, "Comprehensive view of security practices in vehicular networks," in *Proc. Int. Conf. Connected Veh. Expo (ICCVE)*, Seattle, WA, USA, 2016, pp. 19–26.

[128] J. Teixeira *et al.*, "Datacenter in a box: Test your SDN cloud-datacenter controller at home," in *Proc. 2nd Eur. Workshop Softw. Defined Netw. (EWSDN)*, Berlin, Germany, 2013, pp. 99–104.

[129] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang, "Meridian: An SDN platform for cloud network services," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 120–127, Feb. 2013.

[130] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: A survey," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 24–31, Nov. 2013.

[131] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, "SDN based mobile networks: Concepts and benefits," *Wireless Pers. Commun.*, vol. 78, no. 3, pp. 1629–1644, 2014.

[132] *Open Networking Foundation, OpenFlow*. Accessed on Jan. 2017. [Online]. Available: https://www.opennetworking.org/sdn-resources/openflow

[133] M. S. Malik, M. Montanari, J. H. Huh, R. B. Bobba, and R. H. Campbell, "Towards SDN enabled network control delegation in clouds," in *Proc. 43rd Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, Budapest, Hungary, 2013, pp. 1–6.

[134] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[135] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "Rules placement problem in OpenFlow networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1273–1286, 2nd Quart., 2016.

[136] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 493–512, 1st Quart., 2014.

[137] K. Bakshi, "Considerations for software defined networking (SDN): Approaches and use cases," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2013, pp. 1–9.

[138] B. Pfaff *et al.*, "The design and implementation of open vSwitch," in *Proc. NSDI*, Oakland, CA, USA, 2015, pp. 117–130.

[139] J. Claassen, R. Koning, and P. Grosso, "Linux containers networking: Performance and scalability of kernel modules," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Istanbul, Turkey, 2016, pp. 713–717.

[140] C. J. Bernardos *et al.*, "An architecture for software defined wireless networking," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 52–61, Jun. 2014.

[141] P. Bhaumik *et al.*, "Software-defined optical networks (SDONs): A survey," *Photon. Netw. Commun.*, vol. 28, no. 1, pp. 4–18, 2014.

[142] R. Durner, A. Blenk, and W. Kellerer, "Performance study of dynamic QoS management for OpenFlow-enabled SDN switches," in *Proc. IEEE 23rd Int. Symp. Qual. Service (IWQoS)*, Portland, OR, USA, 2015, pp. 177–182.

[143] D. G. Roy, D. De, A. Mukherjee, and R. Buyya, "Application-aware cloudlet selection for computation offloading in multi-cloudlet environment," *J. Supercomput.*, vol. 73, no. 4, pp. 1672–1690, 2017.

[144] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., 2015, pp. 1–6.

[145] J. Bailey and S. Stuart, "Faucet: Deploying SDN in the enterprise," *Queue*, vol. 14, no. 5, pp. 54–68, 2016.

[146] M. Mahalingam *et al.*, "Virtual extensible local area network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3 networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 7348, 2014.

[147] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future Internet," *Comput. Netw.*, vol. 75, pp. 453–471, Dec. 2014.

[148] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[149] D. Amendola, N. Cordeschi, and E. Baccarelli, "Bandwidth management VMs live migration in wireless fog computing for 5G networks," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 21–26.

[150] W.-C. Lin, C.-H. Liao, K.-T. Kuo, and C. H.-P. Wen, "Flow-and-VM migration for optimizing throughput and energy in SDN-based cloud datacenter," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, vol. 1. Bristol, U.K., 2013, pp. 206–211.

[151] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 918–953, 2nd Quart., 2016.

[152] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "OpenStack: Toward an open-source solution for cloud computing," *Int. J. Comput. Appl.*, vol. 55, no. 3, pp. 38–42, 2012.

[153] H.-T. Chang and S.-Y. Wang, "Using SDN technology to mitigate congestion in the openstack data center network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 401–406.

[154] C. H. Benet, K. A. Noghani, and A. J. Kassler, "Minimizing live VM migration downtime using Openflow based resiliency mechanisms," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 27–32.

[155] H. Wang, Y. Li, Y. Zhang, and D. Jin, "Virtual machine migration planning in software-defined networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, 2015, pp. 487–495.

[156] K. Ha and M. Satyanarayanan, "OpenStack++ for cloudlet deployment," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-15-123, 2015.

[157] Á. L. V. Caraguay, A. B. Peral, L. I. B. López, and L. J. G. Villalba, "SDN: Evolution and opportunities in the development IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, May 2014, Art. no. 735142.

[158] N. Amaya *et al.*, "Software defined networking (SDN) over space division multiplexing (SDM) optical networks: Features, benefits and experimental demonstration," *Opt. Exp.*, vol. 22, no. 3, pp. 3638–3647, 2014.

[159] *Geni*. Accessed on Jan. 2017. [Online]. Available: http://groups.geni.net/geni/wiki/OpenFlowDiscoveryProtocol/

[160] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in OpenFlow-based software defined networks," *Comput. Commun.*, vol. 77, pp. 52–61, Mar. 2016.

[161] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, to be published.

[162] G. Luo, S. Jia, Z. Liu, K. Zhu, and L. Zhang, "sdnMAC: A software defined networking based MAC protocol in VANETs," in *Proc. IEEE/ACM 24th Int. Symp. Qual. Service (IWQoS)*, Beijing, China, 2016, pp. 1–2.

[163] W. Ding, W. Qi, J. Wang, and B. Chen, "OpenSCaaS: An open service chain as a service platform toward the integration of SDN and NFV," *IEEE Netw.*, vol. 29, no. 3, pp. 30–35, May/Jun. 2015.

[164] *Open Networking Foundation, SDN Definition*. Accessed on Jan. 2017. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition

[165] S. Gringeri, N. Bitar, and T. J. Xia, "Extending software defined network principles to include optical transport," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 32–40, Mar. 2013.

[166] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN transform service provider opex?" *IEEE Netw.*, vol. 29, no. 3, pp. 60–67, May/Jun. 2015.

[167] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[168] *5G Infrastructure Public Private Partnership—5G Vision: The Next Generation of Communication Networks and Services*. Accessed on Feb. 2017. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[169] A. Hakiri and P. Berthou, "Leveraging SDN for the 5G networks: Trends, prospects and challenges," *arXiv preprint arXiv:1506.02876*, 2015.

[170] S. Peng *et al.*, "QoE-oriented mobile edge service management leveraging SDN and NFV," *Mobile Inf. Syst.*, vol. 2017, Jan. 2017, Art. no. 3961689.

[171] A. Checko *et al.*, "Cloud RAN for mobile networks—A technology overview," *Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.

[172] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art," *Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 303–324, 1st Quart., 2017.

[173] *Open Networking Foundation, OpenFlow Switch Specification Version 1.5.1*. Accessed on Jan. 2017. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf

[174] S. Kukliński, Y. Li, and K. T. Dinh, "Handover management in SDN-based mobile networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, USA, 2014, pp. 194–200.

[175] R. Mennes, B. Spinnewyn, S. Latré, and J. F. Botero, "GRECO: A distributed genetic algorithm for reliable application placement in hybrid clouds," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Pisa, Italy, 2016, pp. 14–20.

[176] S. Gao, Y. Zeng, H. Luo, and H. Zhang, "Scalable control plane for intra-domain communication in software defined information centric networking," *Future Gener. Comput. Syst.*, vol. 56, pp. 110–120, Mar. 2016.

[177] T. Braun, A. Mauthe, and V. Siris, "Service-centric networking extensions," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, Coimbra, Portugal, 2013, pp. 583–590.

[178] A. Sathiaseelan, L. Wang, A. Aucinas, G. Tyson, and J. Crowcroft, "Scandex: Service centric networking for challenged decentralised networks," in *Proc. Workshop Do Yourself Netw. Interdiscipl. Approach*, Florence, Italy, 2015, pp. 15–20.

[179] R. Munoz *et al.*, "Network virtualization, control plane and service orchestration of the ICT STRAUSS project," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Bologna, Italy, 2014, pp. 1–5.

[180] A. El Mougy, "On the integration of software-defined and information-centric networking paradigms," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Abu Dhabi, UAE, 2015, pp. 105–110.

[181] J. Wang *et al.*, "A minimum cost cache management framework for information-centric networks with network coding," *Comput. Netw.*, vol. 110, pp. 1–17, Dec. 2016.

[182] S. Charpinel, C. A. S. Santos, A. B. Vieira, R. Villaca, and M. Martinello, "SDCCN: A novel software defined content-centric networking approach," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Crans-Montana, Switzerland, 2016, pp. 87–94.

[183] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, "Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed," *Comput. Netw.*, vol. 57, no. 16, pp. 3207–3221, 2013.

[184] N. B. Melazzi *et al.*, "An OpenFlow-based testbed for information centric networking," in *Proc. Future Netw. Mobile Summit (FutureNetw)*, Berlin, Germany, 2012, pp. 1–9.

[185] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, 2010.

[186] I. Bueno, J. I. Aznar, E. Escalona, J. Ferrer, and J. A. Garcia-Espin, "An OpenNaaS based SDN framework for dynamic QoS control," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Trento, Italy, 2013, pp. 1–7.

[187] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper, vol. 11, 2015.

[188] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, "Software defined networking: Meeting carrier grade requirements," in *Proc. 18th IEEE Workshop Local Metropolitan Area Netw. (LANMAN)*, Chapel Hill, NC, USA, 2011, pp. 1–6.

[189] S. Sahhaf *et al.*, "Network service chaining with optimized network function embedding supporting service decompositions," *Comput. Netw.*, vol. 93, pp. 492–505, Dec. 2015.

[190] P. Quinn and T. Nadeau, "Problem statement for service function chaining," Internet Eng. Task Force, Fremont, CA, USA, RFC 7498, 2015.

[191] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[192] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: Integrating software defined networking and network function virtualization," *IEEE Netw.*, vol. 29, no. 3, pp. 36–41, May/Jun. 2015.

[193] I. ETSI, "Network functions virtualisation–network operator perspectives on industry progress," Updated White Paper, 2013.

[194] *Huawei Observation to NFV*. Accessed on Feb. 2017. [Online]. Available: http://www.huawei.com/ilink/en/download/HW_399662

[195] G. Cheng, H. Chen, H. Hu, Z. Wang, and J. Lan, "Enabling network function combination via service chain instantiation," *Comput. Netw.*, vol. 92, pp. 396–407, Dec. 2015.

[196] F. B. Jemaa, G. Pujolle, and M. Pariente, "Cloudlet-and NFV-based carrier Wi-Fi architecture for a wider range of services," *Ann. Telecommun.*, vol. 71, nos. 11–12, pp. 617–624, 2016.

[197] P. B. Pawar and K. Kataoka, "Segmented proactive flow rule injection for service chaining using SDN," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Seoul, South Korea, 2016, pp. 38–42.

[198] A.-V. Vu and Y. Kim, "An implementation of hierarchical service function chaining using opendaylight platform," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Seoul, South Korea, 2016, pp. 411–416.

[199] R. Vilalta *et al.*, "End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node," in *Proc. Opt. Fiber Commun. Conf. Exhibit. (OFC)*, Anaheim, CA, USA, 2016, pp. 1–3.

[200] S. Paul and R. Jain, "OpenADN: Mobile apps on global clouds using OpenFlow and software defined networking," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Anaheim, CA, USA, 2012, pp. 719–723.

[201] I. Ku, Y. Lu, and M. Gerla, "Software-defined mobile cloud: Architecture, services and use cases," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Nicosia, Cyprus, 2014, pp. 1–6.

[202] A. Banerjee *et al.*, "MOCA: A lightweight mobile cloud offloading architecture," in *Proc. 8th ACM Int. Workshop Mobility Evol. Internet Archit.*, Miami, FL, USA, 2013, pp. 11–16.

[203] J. Pan, L. Ma, R. Ravindran, and P. TalebiFard, "HomeCloud: An edge cloud framework and testbed for new application delivery," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, Thessaloniki, Greece, 2016, pp. 1–6.

[204] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, 2016, pp. 1–6.

[205] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the Internet of Vehicles," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 133–144, Apr. 2015.

[206] A. El Amraoui and K. Sethom, "Cloudlet softwarization for pervasive healthcare," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, 2016, pp. 628–632.

[207] S. Monfared, H. Bannazadeh, and A. Leon-Garcia, "Software defined wireless access for a two-tier cloud system," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2015, pp. 566–571.

[208] D. Wu, D. I. Arkhipov, E. Asmare, Z. Qin, and J. A. McCann, "UbiFlow: Mobility management in urban-scale software defined IoT," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 208–216.

[209] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2015, pp. 1202–1207.

[210] K. Gray and T. D. Nadeau, *Network Function Virtualization*. Cambridge, MA, USA: Morgan Kaufmann, 2016.

[211] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," Internet Eng. Task Force, Fremont, CA, USA, RFC 7665, 2015.

[212] *European Telecommunications Standards Institute Industry Specifications Group, Mobile-Edge Computing—Network Functions Virtualisation*. Accessed on Feb. 2017. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv

[213] J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "SAVI testbed: Control and management of converged virtual ICT resources," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ghent, Belgium, 2013, pp. 664–667.

[214] M. Faraji, J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "Identity access management for multi-tier cloud infrastructures," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–9.

[215] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz, "Experimental demonstration of distributed multi-tenant cloud/fog and heterogeneous SDN/NFV orchestration for 5G services," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Athens, Greece, 2016, pp. 52–56.

[216] S. Sezer *et al.*, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[217] S. Ortiz, "Software-defined networking: On the verge of a breakthrough?" *IEEE Comput.*, vol. 46, no. 7, pp. 10–12, Jul. 2013.

[218] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, 2014.

[219] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[220] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[221] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, 4th Quart., 2016.

[222] *Open Networking Foundation, Northbound Interfaces*. Accessed on Feb. 2017. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf

[223] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.

[224] M. Kwak, J. Suh, and T. Kwon, "FRACTAL: A framework for recursive abstraction of SDN control-plane for large-scale production networks," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–5.

[225] H. Xu *et al.*, "Incremental deployment and throughput maximization routing for a hybrid SDN," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1861–1875, Jun. 2017.

[226] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann, "Panopticon: Reaping the benefits of incremental SDN deployment in enterprise networks," in *Proc. USENIX ATC*, Philadelphia, PA, USA, 2014, pp. 333–345.

[227] M. Kuzniar, P. Peresini, M. Canini, D. Venzano, and D. Kostic, "A soft way for OpenFlow switch interoperability testing," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol.*, Nice, France, 2012, pp. 265–276.

[228] M. Dai, G. Cheng, and Y. Wang, "Detecting network topology and packet trajectory with SDN-enabled FPGA platform," in *Proc. 11th Int. Conf. Future Internet Technol.*, Nanjing, China, 2016, pp. 7–13.

[229] H. Zhou, C. Wu, Q. Cheng, and Q. Liu, "SDN-LIRU: A lossless and seamless method for SDN inter-domain route updates," *IEEE/ACM Trans. Netw.*, to be published.

[230] P. Samadi, K. Wen, J. Xu, and K. Bergman, "Software-defined optical network for metro-scale geographically distributed data centers," *Opt. Exp.*, vol. 24, no. 11, pp. 12310–12320, 2016.

[231] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1002–1026, 2nd Quart., 2016.

[232] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, to be published.

[233] J. Michel and C. Julien, "A cloudlet-based proximal discovery service for machine-to-machine applications," in *Proc. Int. Conf. Mobile Comput. Appl. Services*. Paris, France, 2013, pp. 215–232.

[234] *Open Networking Foundation—Special Report: OpenFlow and SDN—State of the Union*. Accessed on Feb. 2017. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/special-reports/Special-Report-OpenFlow-and-SDN-State-of-the-Union-B.pdf

[235] F. J. Ros and P. M. Ruiz, "Five nines of southbound reliability in software-defined networks," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, Chicago, IL, USA, 2014, pp. 31–36.

[236] *Open Network Operating System*. Accessed on Jan. 2017. [Online]. Available: http://onosproject.org/

[237] S. Schmid and J. Suomela, "Exploiting locality in distributed SDN control," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Hong Kong, 2013, pp. 121–126.

[238] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 7–12, 2013.

[239] M. Canini, P. Kuznetsov, D. Levin, and S. Schmid, "A distributed and robust SDN control plane for transactional network updates," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 190–198.

[240] M. Aslan and A. Matrawy, "Adaptive consistency for distributed SDN controllers," in *Proc. IEEE 17th Int. Telecommun. Netw. Strategy Plan. Symp. Netw.*, 2016, pp. 150–157.

[241] M. Kobayashi *et al.*, "Maturing of OpenFlow and software-defined networking through deployments," *Comput. Netw.*, vol. 61, pp. 151–175, Mar. 2014.

[242] *Pica8—White Box SDN*. Accessed on Feb. 2017. [Online]. Available: http://www.pica8.com/what-we-do/

Ahmet Cihat Baktir (S'15) received the B.S. degree in computer science and engineering from Sabanci University, Istanbul, in 2012 and the M.A. degree in management information systems from Bogazici University, Istanbul, in 2014, where he is currently pursuing the Ph.D. degree in computer engineering. His research interests include software-defined networking, cloud computing, Edge Computing, and service-centric networks.

Atay Ozgovde (M'95) received the B.S. and M.S. degrees from Bogazici University, Istanbul, in 1995 and 1998, respectively, and the Ph.D. degree from the NETLAB Research Group, Bogazici University in 2009. He was with Nortel Networks as an Research and Development Engineer in various telecommunications projects from 1998 to 2001. In 2002, he was a Research Assistant with the Computer Engineering Department, Bogazici University. He was a Post-Doctoral Researcher with TETAM Research Center, where he was with the WiSE-Ambient Intelligence Group. He is currently an Assistant Professor with the Computer Engineering Department, Galatasaray University. His research interests include wireless sensor networks, embedded systems, distributed systems, pervasive computing, SDN, and mobile cloud computing.

Cem Ersoy received the Ph.D. degree from Polytechnic University, New York in 1992. He became a Professor of Computer Engineering and currently the Department Head with Bogazici University. He was a Research and Development Engineer with NETAS A.S. from 1984 to 1986. His research interests include wireless/cellular/ad-hoc/sensor networks, activity recognition and ambient intelligence for pervasive health applications, green 5G and beyond networks, Mobile Cloud/Edge/Fog computing, software defined networking, and infrastructureless communications for disaster management. He is also the Vice Director of the Telecommunications and Informatics Technologies Research Center, TETAM which funds the largest Ph.D. student body in Turkey in that field. He is a member of IFIP and the Chairman of the IEEE Communications Society Turkish Chapter.