Software-Defined Networking Enhanced Edge Computing: A Network-Centric Survey

This article discusses how software-defined networking and related technologies are integrated to facilitate the management and the operations of edge servers and various Internet-of-Things (IoT) devices.

By An Wang, Zili Zha, Yang Guo, and Songqing Chen, *Senior Member IEEE*

ABSTRACT | Edge computing is burgeoning along with the rapidly increasing adoption of the Internet of Things (IoT). While there are studies on various aspects of edge computing, we find there is a lack of network perspective. In this paper, we, thus, first present an overview of how software-defined networking (SDN) and related technologies are being investigated in edge computing. Our purpose is to survey the state of the art and discuss the potential (remaining) challenges for future research. For this, we survey how SDN and related technologies are integrated to facilitate the management and operations of edge servers and various IoT devices. For the former, we review how SDN has been utilized in the access network, the core network, and the wide area network (WAN) between the edge and the cloud. For the latter, we focus on how SDN is leveraged to provide unified and programmable interfaces to manage devices. Through our discussion, we suggest that the SDN-related network support for edge computing deserves more in-depth investigations. We also identify several challenges and open issues to be addressed in the future.

KEYWORDS | Edge computing; Internet of Things; software defined networking.

I. INTRODUCTION

The explosive growth of the Internet of Things (IoT) and mobile devices leads to an explosion of new applications and services, increasing the burden of what today's Internet could carry. What makes it worse is the heterogeneous platforms that support these applications and the diverse requirements of the applications from multiple perspectives, such as service quality, security and privacy, and computing and storage resources. Edge computing has been proposed to address these pressing needs as a complementary solution to cloud computing. Although a lot of research have been investigated on various aspects of edge computing [1]–[5], prior research has been mainly focused on the architecture, resource provisioning and management, programming models, new application development, and so on. The networking perspective is lacking, especially with the newly available networking support of software-defined networking (SDN) and related technologies.

Some previous efforts have investigated mobile edge computing (MEC). For example, Mao *et al.* [6] surveyed MEC, with a focus on joint radio-and-computational resource management. The challenges and opportunities of radio communication techniques were discussed for facilitating resource management. In our investigation, we, instead, focus on the general design and deployment of the state-of-the-art network management techniques (e.g., SDN) in the edge computing environment. Mach and Becvar [7] conducted another survey on the user-oriented use cases in the MEC system. Several MEC concepts were introduced to integrate cloud capabilities, e.g., small-cell cloud (SCC) [8], mobile microcloud (MMC) [9], fast-moving personal cloud, follow-me cloud

Manuscript received January 31, 2019; revised April 13, 2019; accepted May 18, 2019. Date of current version August 5, 2019. This work was supported in part by the NIST under Grant 70NANB18H272, in part by NSF under Grant CNS-1524462, and in part by the Institute for Smart, Secure and Connected Systems, Case Western Reserve University, through a grant provided by the Cleveland Foundation. *(Corresponding author: An Wang.)*

A. Wang is with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: axw474@case.edu).

Z. Zha and **S. Chen** are with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA.

Y. Guo is with the National Institute of Standards and Technology (NIST), Gaithersburg, MD 20899 USA.

Digital Object Identifier 10.1109/JPROC.2019.2924377

(FMC) [10], and CONCERT [11], as well as their network architectures. We, instead, focus on how advanced network technologies enhance edge computing. Abbas *et al.* [12] also surveyed relevant research and technological development in MEC. The authors mentioned that SDN technology can help control in MEC to be more efficient and reliable. Wang *et al.* [13] surveyed the key technologies in MEC computing and caching and provided a summary of MEC applications and use cases. They envisioned SDN and network function virtualization (NFV) as the key enablers for the concept of MEC ascribable to the flexibility and operating efficiency they provided. Tran *et al.* [14] illustrated the benefits and applicability of MEC collaboration in 5G networks by discussing three use cases. Baktir *et al.* [15] discussed the capabilities of SDN and aligned them with the technical shortcomings of edge computing implementations. The discussions were also focused on the integration of edge computing and SDN by demonstrating multiple use case scenarios. However, no systematic comparisons among different SDN-enhanced edge computing architectures are available. In contrast to these previous studies, in this paper, we focus our investigations on the integration of the SDN into edge computing and how their advantages could be utilized by edge computing. We also classify the integration techniques based on their architectures to provide comprehensive discussions and comparisons.

For this purpose, we, thus, start from the development of edge computing and SDN and NFV by discussing their background. Then, we discuss why SDN could benefit edge computing by surveying several use cases. We also classify the current research in this area into four categories based on architectural designs and implementations. Since IoT is one of the thrusts behind edge computing, we also discuss the different applications of SDN in such an environment. We expect that this paper not only gives a comprehensive overview of the current networking research in supporting edge computing but also identifies future challenges and open issues that are worth further in-depth explorations.

The remainder of this paper is organized as follows. Section II discusses the development of edge computing and SDN technologies. Section III discusses the different integration designs of SDN and edge computing. Section IV shows the use cases of IoT management and applications using SDN. Section V discusses several challenges and open questions in this area. Finally, this paper concludes in Section VI.

II. EVOLVEMENT OF EDGE COMPUTING AND SDN

In this section, we provide some background information on the edge computing and the evolvement of SDN and the relevant technology.

A. Edge Computing

Since Amazon released its Elastic Compute Cloud product in 2006 [16], cloud computing has gained tremendous success by reaping its field from various business sectors to personal end users in the past decade or so. By providing centralized (and elastic) resources and a flexible pay-as-you-go cost model, cloud computing provides services with different service models [17], including Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), to its customers with great performance–cost ratio and convenience. As a result, a significant portion of the enterprise services have been migrated onto clouds, and an increasing number of end users also rely on clouds for daily activities.

In tandem with the fast development of cloud computing services, the past decade has also witnessed another radical change over the Internet: there are more and more smart and mobile devices, such as smartphones and tablets, and various sensors and actuators becoming available and pervasive, benefiting from the advancement of wireless communication technology. While smart and mobile devices provide a full-fledged computing stack, sensors and actuators often are dedicated to data collection and communication, forming the basis of the IoT systems or various cyber–physical systems (CPSs). A lot of such applications have also been conceived and/or prototyped accordingly, such as healthcare [18], smart cities [19], auto-driving [20], and smart spaces [21]. With the ever-decreasing hardware cost and the ever-improving CPU speed/wireless communication technology, Cisco has estimated that there will be 500 billion devices to be connected to the Internet by the year 2030 [22].

Despite the ever-improving technology, smart devices and sensors are constrained by the limited on-device resources, namely, the slow CPU, the limited memory, and the short battery lifetime. This is especially true when compared with their counterparts on the popular cloud platforms. Naturally, this has motivated the ideas of utilizing the plentiful cloud resources to support applications running on smart devices and sensors. The key of this idea is to wisely offload complicated or computing-intensive applications on smart and mobile devices or sensors to clouds. These efforts led to mobile cloud computing (MCC) [5] initially. MCC nicely complements the mobility offered by the smart/mobile devices while also being able to leverage the powerful computing capability of clouds. Therefore, a lot of efforts have been made to investigate how to efficiently utilize the cloud and the smart device for the best user experience and/or system performance [23]–[27]. Clonecloud [28] is one of such early efforts that seek to utilize virtual machine (VM) techniques to create an identical running environment for mobile devices on the cloud.

However, with more and more emerging applications, MCC often suffers from unpredictable network latency, which is detrimental to the latency-sensitive mobile applications or location-constrained ones. For example, a decision for auto-driving needs to be made in milliseconds, while the communication latency to the cloud is often much larger. In addition, utilizing MCC often requires transmitting a large amount of data collected from the mobile devices and sensors to the cloud before the processing can take place. For example, the cameras on the cars need to upload the images continuously to the cloud for processing in order to know the hazards on the road. Such a data demand can easily make the communications a bottleneck. Furthermore, along with the emerging of various new applications, such as healthcare and smart home, the users' security and privacy concerns further aggravate the challenge since the data generated by the smart devices and/or sensors often carry some private or sensitive information.

To this end, a new computing paradigm, edge computing, emerged to deal with such challenges. Albeit being named differently, efforts such as Cloudlets [29], Fog Computing [4], and mobile edge computing or multiaccess edge computing (MEC) [30], share similar goals with entirely or largely overlapping principles and application scopes. The fundamental of such a computing paradigm is to deploy resources on the edge of a network, namely, edge nodes (or edge cloud or edge servers or fog nodes, cloudlets, microcloud, and so on[1]), in close proximity to the edge devices or sensors so that the capability of such edge nodes can be utilized to reduce the network latency, save bandwidth, and improve security and privacy. With the increasing adoption of the big data applications, additional services could also be offered by these edge nodes, such as data analytics.

At a high level, edge computing can be viewed as cloud services migrated from remote clouds to the nearby network edge. While edge computing comes from a different direction from the IoT applications, these days, they are often heavily intertwined. In some occasions, they are even regarded as inter-exchangeable. In this paper, we treat them differently, where edge computing is more of a computing diagram and IoT are more of applications. On the other hand, IoT applications can be broadly defined to include most of today's applications using some sensors and/or mobile devices and, thus, include smart home, smart space, smart cities, and so on. In our study, instead, we separate these applications from IoT applications as separate categories (e.g., healthcare, smart city, and autodriving) as they are important and have enough challenges to overcome.

B. SDN and NFV

The creation of the modern Internet offers universal connectivity that jump-started the digital age and tremendously improved people's daily life. The packet-based switching and distributed architecture are key design principles adopted by the Internet, which contribute to the networks' scalability, flexibility, and fault tolerance. Despite being successful, traditional IP networks become increasingly complex, hard to configure/manage, slow to incorporate new innovations, and expensive to buy equipment, operate networks, and provide services.

The root cause rests on the design of network routers/switches and overall distributed network architecture. A traditional IP router/switch consists of two layers: a data plane and a control plane. The data plane is designed to forward network packets at a very high speed, while the control plane implements the configuration and management functions that govern how forwarding plane routes the packets. Although the data plane functions locally, the control plane typically implements distributed algorithms/protocols that collectively provide certain services, e.g., distributed network routing. A traditional router becomes a complex proprietary box that is hard to configure (need to remotely log in to configure individual routers) and difficult to roll out new services (need to coax the distributed protocols to realize new services) and incurs high capital expenditure (CAPEX) and operating expense (OPEX).

To overcome these shortcomings, and also being motivated by the emerging needs of cloud computing and network infrastructures demanded by search and social media services, such as Google[2] and Facebook, SDN was proposed [31]. In its original design, SDN refers to a network architecture that decouples the forwarding plane from the control plane. As defined in OpenFlow [32], an SDN switch consists of a pipeline-based packet forwarding engine and a simple agent that communicates with a (at least conceptually) centralized SDN controller. The forwarding rules are remotely managed by the central controller that implements network management functions. Such decoupling of the control plane and data plane greatly simplifies the switch design and lowers the bar for vendors to enter the switching building business. In addition, replacing the distributed algorithms with centralized ones and implementing them in a central controller also simplifies the control plane. Multiple open-sourced controllers [33], [34] have emerged, which allows open-source community to contribute to the network innovations. Finally, the controller effectively works as an abstraction layer for the underlying network switching devices. The networking programming languages [35], [36] are developed, which run on top of the controller to provide the so-called network programmability.

SDN continues to evolve. Recently, next-generation SDN has emerged, aiming to offer operators' complete control of their networks, zero-touch configuration and management, and true network programmability. Next-generation SDN replaces the signature OpenFlow with a set of new

[1] While we generally call them edge nodes, we follow the paper's original term when we surveyed them, but they are inter-exchangeable in our paper.

[2] Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology nor to imply that the materials or equipment identified are necessarily the best available for the purpose.

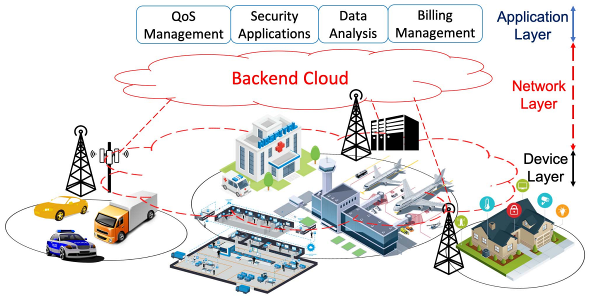


Fig. 1. ***General architecture of SDN-integrated edge computing environment.***

interfaces. SDN switch pipeline is controlled through P4Runtime [37], while gNMI/OpenConfig [38], [39] and gNOI [40] are the interfaces for SDN switch configuration and operations that were solely lacking in the original SDN design. Open-source project Stratum [41] is developing a reference implementation for white box switches supporting all next-generation SDN interfaces. The next-generation SDN avoids the vendor lock-in of today's data planes (i.e., proprietary silicon interfaces and closed software APIs), enables easy integration of SDN devices into traditional networks, and offers a migration path from traditional IP networks to fully SDN-enabled networks.

NFV is an initiative to virtualize network services, such as load balancing and firewalls, which are traditionally run on proprietary, dedicated hardware (so-called middleboxes). NFV realizes these services as software or VMs and runs them on commodity hardware. NFV is complementary to SDN, sharing the goal of accelerating innovation inside the networks by allowing automation and programmability via a shift to software-based platforms. Virtualized network functions (VNFs) can be easily managed by the SDN's central controller that forwards data packets to and from network functions.

III. SDN INTEGRATION WITH EDGE COMPUTING

A typical edge computing environment is composed of an array of connected edge servers (interchangeable with edge or fog nodes). They are usually generic virtualized equipment with three fundamental capabilities: storage, computing, and communications. To reduce the storage demand, various proactive caching mechanisms have been proposed and applied [42]. Computations are also performed at the edge servers that are transparent to users with cross-platform and cross-application supports. The glue that holds different components together in edge computing is the underlying network architecture. It allows the service providers to extend their services and functions closer to the end users. A typical architecture of such an environment is sketched in Fig. 1. As shown in Fig. 1, to support the applications and interact with various devices, the network layer needs various support.

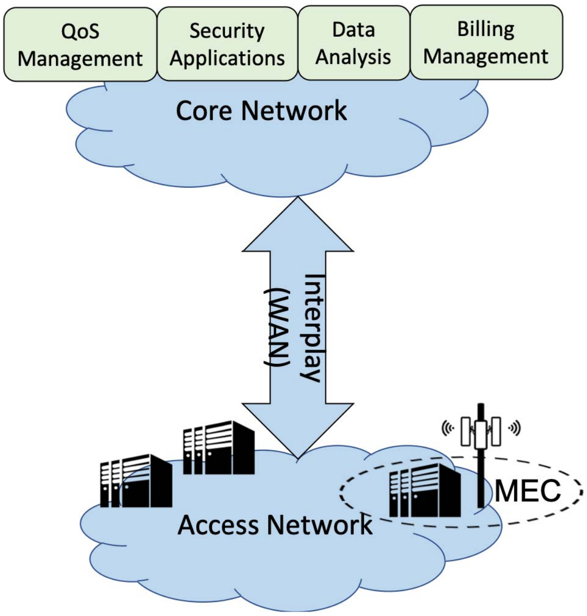


Fig. 2. ***General architecture of the network layer in edge computing.***

In this section, we focus on the integration of the state-of-art networking technologies, SDN and NFV, into the network layer of edge computing.

Fig. 2 demonstrates a detailed layout of the network layer in an edge computing environment. The access network connects the heterogeneous devices to the services deployed at the edge. MEC, among others, was proposed in response to the emerging benefits of edge computing technology that reached beyond mobile networks and into Wi-Fi and fixed access technologies. The core network refers to the data center clouds at the core, which manage resources and applications with a centric view. Wide area network (WAN) connects many different actuators, gateways, and devices sending transmissions from the edge to the cloud. As a network control paradigm, SDN should be compatible with access, WAN, and cloud technologies.

On one hand, SDN/NFV deployed at the access network could support diverse requirements and agile service creation. On the other hand, such techniques are often employed in the data centers and clouds to configure and orchestrate services on the edge servers. For time-sensitive services, such as in an Industrial IoT (IIoT) environment, software-defined WAN (SD-WAN) has also been introduced [43]. Next, we discuss some existing research efforts following these paradigms.

A. SDN in Access Network

One of the early efforts to adopt SDN in edge computing is a collaborative project between AT&T and Open Networking Lab, called Central Office Re-architectured as a Datacenter (CORD). The motivation of CORD is to lower CAPEX and OPEX for service providers to maintain

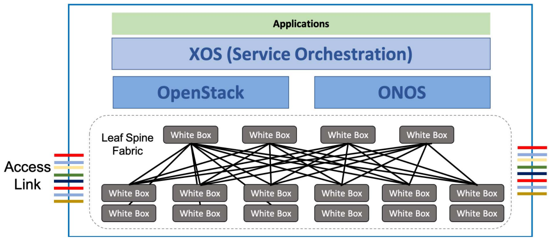


Fig. 3. ***Architecture of CORD.***

hundreds of thousands of hardware appliances. To that end, CORD was designed to build cost-effective, agile networks to enable rapid service creation and monetization. To achieve this goal, CORD merges the benefits of the programmable control plane of SDN and the virtualized data plane required by NFV. CORD is designed to be a general platform. There are different configurations targeting different deployment scenarios, including mobile users (M-CORD), enterprise users (E-CORD), residential users (R-CORD), and analytics for CORD (A-CORD). The common building blocks of the CORD architecture are illustrated in Fig. 3.

CORD leverages several open-source projects for software implementations, such as OpenStack [44], ONOS [45], and XOS [46]. OpenStack is utilized to help organizations to offer cloud-computing services running on standard hardware. It is a modular architecture with various components for system management. It is also responsible for creating and provisioning VMs and virtual networks (VNs). ONOS is the network operating system that controls the underlying white-box switching fabric. XOS is a model-based platform for assembling, controlling, and composing services. ONOS hosts a collection of control applications by interacting with XOS. Also, it interconnects VMs by implementing VNs and managing traffic flows through the switching fabric. To transform traditional Central Office into CORD, the first step is to virtualize the existing hardware devices, including optical line termination (OLT) [47], customer premises equipment (CPE) [48], and Broadband Network Gateways (BNGs) [49]. These devices combined are applied to manage user subscriptions and to provide a number of essential functions to users, such as Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT), firewall, and parental control.

To virtualize OLT, the underlying devices are equipped with MAC chip that is controlled by the remote control program, called virtual OLT (vOLT), via OpenFlow. It is also responsible for authenticating the users. The virtualized CPE, called virtual Subscriber Gateway (vSG), moves their functions to the Central Office by running in a VM on commodity servers. Finally, virtulized BNG, called virtual Router (vRouter), is implemented as an ONOS-based control program that manages flows on behalf of subscribers. The second step of the transformation process is to orchestrate the software programs running on the control plane as described earlier. To make the control functions scalable, XOS provides a service abstraction layer to represent dependence relationships among a set of services for service assembly. Each service corresponds to a VNF in the NFV architecture.

Currently, CORD has officially released the stable version 5.0, including R-CORD, E-CORD, and M-CORD service profiles. The CORD community has grown quickly and attracted a diverse range of collaborators, including major service providers, such as Comcast, AT&T, China Unicom, Turk Telekom, and NTT Communications. These collaborators contribute significantly to the transformation of CORD from the proof-of-concept stage to field trial stage. AT&T initiated their field trial with CORD in early 2016 by deploying the open hardware, i.e., the initial gigabit passive optical network (GPON) OLT, and vBNG to see how it would fit in their business model. It is still out in the field and serves their business today. NTT Communications also showed their use cases with CORD integration and the evaluation plan at the Okinawa Open Laboratory.

The most distinctive feature of CORD is its integration with the SDN technologies. Such integration benefits the service providers from transforming the traditional Fiber-to-the-Home (FTTH)-as-a-service to the SaaS platform, which is more scalable and cost-efficient. Following the trend in edge computing, CORD also attracts a series of academic explorations in this direction. Recently, Moyano *et al.* [50] proposed a novel network management model that integrates the software-defined access network (SDOAN) and edge computing principles. This unified framework allows users to manage the SDN-based residential gateway (RGW) and to control the access network resources by defining customized service function chain (SFC) in charge of providing a differentiated traffic treatment with QoS.

For this purpose, SDOAN provides a customizable virtual slice of access network for each residential network. The user requirements are submitted to the SDOAN management system and a number of parameters can be adjusted to configure the virtual slice. Similar to CORD, an NFV infrastructure is co-located with the OLT at the Central Office. On the cloud side, each virtual network slice is managed by an instance of virtualized Management and Networking Domain (vMANDO), running VNFs of the residential network. To provide differentiated networking services, the user is able to define different traffic classes and the corresponding SFC for each network segment, identified by a unique ID. This ID is also used to define a virtual path from the residential network to the distribution network, and the user could define a QoS level for this path (e.g., to watch a specific TV show). The network component of vMANDO has different VNFs; some of them are fixed, such as Router + NAT and Classification & Shaping, while others are customizable on demand, such as content filter and firewall. This project built a testbed based on the proof-of-concept scenario by extending OpenStack with the components of SDOAN. This project is still under development. However, the initial prototype, which operates different network segments, has been validated.

In addition to the wired access network, Petrov *et al.* [51] recently proposed to build a softwarized 5G access network utilizing the NFV techniques. The motivation of this project is to provide end-to-end reliability for the mission-critical traffic. One particular scenario they focus on is to provide reliable and high-rate data transmission for vehicles in motion, such as an ambulance vehicle that is transporting a patient to the hospital, while the paramedics and the doctors in the hospital are jointly assessing the patient's condition. In this case, three types of access points are available: 1) cellular mmWave network; 2) cellular microwave network (LTE); and 3) non-3GPP microwave network (Wi-Fi).

Due to the mobility nature of the vehicles, they may need to switch among different access technologies. Thus, some critical network functions are required to be deployed, including session management function (SMF), access and mobility management function (AMF), and policy control function (PCF). SMF manages the establishment, modification, and release of the sessions. AMF controls the access decisions as well as handling mobility-related issues. PCF provides the policy rules to the relevant control plane functions. In the proposed framework, all these functions are functioning as VNFs. Hence, a virtualized SDN controller should be implemented and designed to enforce the policies. Their evaluation results revealed that supporting mission-critical traffic in 5G systems is still costly even in the idealistic cases. This also brings considerable degradation to the service of other user sessions. However, the flexible network configuration and management can mitigate some of the negative effects for other users.

B. SDN-Enabled Core Network

The frameworks mentioned earlier are deployed at the edge to manage the access network. These solutions are designed to handle the QoS-related issues. However, they lack the ability to address the scalability and connectivity issues that are critical for some IoT applications, such as Vehicular Ad Hoc Networks (VANETs), and the application of them will be discussed in detail in Section IV.

Besides, SDN has also been proposed to be employed to manage ubiquitous IoT devices. One such application has been proposed by Sun and Ansari [52] in 2016, called EdgeIoT. One of the concerns they seek to address is user privacy in such an environment. In their scenario, all the IoT devices are connected to some wireless access points that receive services from edge servers in their vicinity. To preserve user privacy, each user is associated with a proxy VM that is considered as the user's private VM. However, the IoT devices may roam away to other access points. Thus, the proxy VM migration is necessary to minimize the traffic going to the core network.

Since VM migrations are determined by the locations of the devices, the proxy VM should be aware of the mobility of the registered devices. On the other hand, most existing IoT devices do not support location update protocol involved in the LTE network. To address this issue, the authors proposed to use the location of the user's mobile phone or other wearable devices as a gateway to report the location of the IoT devices in the environment [52]. Emulation experiments were conducted with data traces of more than 13k users and extracted mobility in one day in Heilongjiang Province of China. The results demonstrate that such a framework could substantially reduce the traffic load in the core network and the end-to-end delay between the IoT devices. For service migration, Rosário *et al.* [53] studied operational impacts and benefits associated with service migration from the cloud to multitier fog computing for video distribution utilizing the SDN technologies. In their design, the SDN components enable service migration to deliver videos with adequate QoE for mobile users.

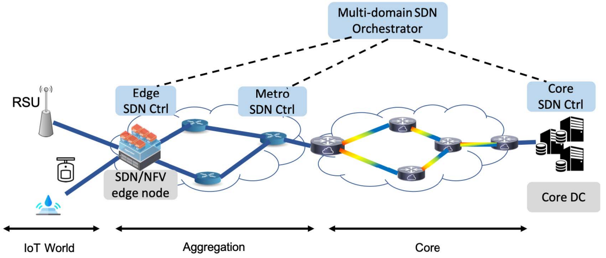


Fig. 4. ***Architecture of end-to-end SDN-enabled orchestration (source: [56]).***

SDN deployed at the core network is also often utilized to support mobility for the device. Bi *et al.* [54] designed signaling operations to provide seamless and transparent mobility support to mobile users. In the proposed framework, the SDN switches and controllers are deployed in the network layer to install mobility logic for handling mobility-related signaling when mobile users change their attached edge servers. The SDN controller performs algorithms to find the optimal path with minimum latency. Ouyang *et al.* [55] proposed an approximation algorithm based on the Markov approximation to find an optimal solution for service latency, given unpredictable user mobility and cost budget constraints. The SDN paradigm was introduced to enforce service placement in MEC.

Finally, SDN frameworks provide programmable interfaces for service orchestrations. Vilalta *et al.* presented an SDN/NFV-enabled edge node for IoT services by means of the E2E SDN orchestration of integrated cloud/fog and network resources [56]. The architecture of the proposed design is shown in Fig. 4. In the architecture, a multidomain SDN orchestrator is responsible for provisioning E2E network services. Edge SDN controllers are also introduced to control the OpenFlow-enabled switches. The evaluations were conducted on the IoT World Testbed [57]. Lombardo *et al.* [58] designed and implemented an SDN-based framework aimed at deploying NFV

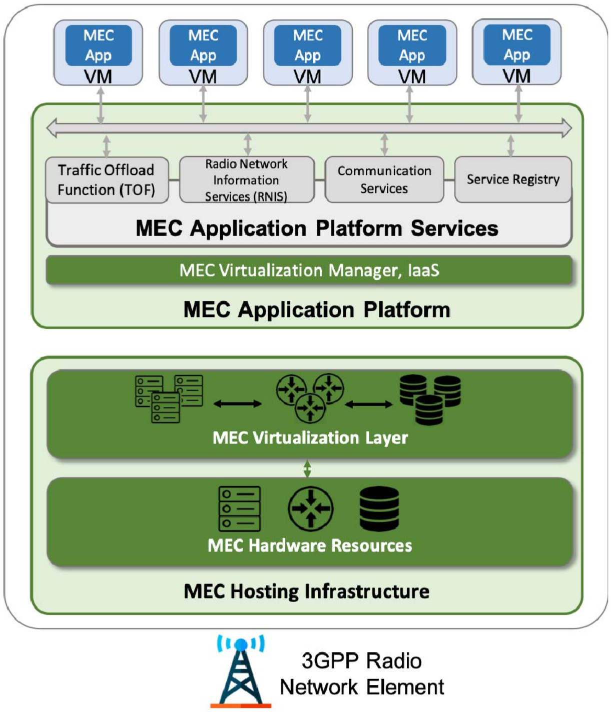

MEC App VM
MEC App VM
MEC App VM
MEC App VM
MEC App VM
Traffic Offload Function (TOF)
Radio Network Information Services (RNIS)
Communication Services
Service Registry
MEC Application Platform Services
MEC Virtualization Manager, IaaS
MEC Application Platform
MEC Virtualization Layer
MEC Hardware Resources
MEC Hosting Infrastructure
3GPP Radio Network Element

Fig. 5. ***General architecture of MEC server.***

at the edge. Within the core network, an orchestration component was introduced that is built on top of an SDN controller platform, POX [59]. Similar ideas have been proposed by Zhu and Huang [60] and Baidya *et al.* [61].

Overall, the proposed frameworks that integrate SDN into MEC servers share one common limitation. Most of the proposed systems present a tight coupling of the SDN control platforms and the virtualization platform of the MEC servers. Therefore, they become application-specific MEC servers and lack the flexibility to adapt to other scenarios.

C. SDN-Assisted Multiaccess/Mobile Edge Computing

An important effort in standardizing edge computing in the mobile network was initiated by The European Telecommunications Standards Institute (ETSI) in 2015 [62]. The framework that provides an IT service environment and cloud-computing capabilities at the edge is called mobile edge computing or multiaccess edge computing (MEC). It is deployed within the radio access network (RAN) and in close proximity to mobile subscribers. Conceptually, MEC specifies one form of the edge computing architectures and is dedicated to serve mobile devices. The key element of MEC is the MEC IT application server that is integrated at the RAN element. The MEC server provides computing resources, storage capacity, connectivity, and access to user traffic and radio and network information. According to a recent technical white paper of ETSI [30], the architectural blueprint of the MEC server is shown in Fig. 5.

As illustrated in Fig. 5, an MEC server has three layers, including the application layer, the application platform layer, and the hosting infrastructure. MEC is based on a virtualized platform that should also host VNFs so that the network operators could benefit as much as possible from their investment by reusing the infrastructures. The MEC hosting infrastructure provides connectivity to the radio network element [Evolved Node B (eNB) or radio network controller (RNC)] and/or the network. The MEC application platform provides the capabilities for hosting applications and consists of the application's virtualization manager and application platform services. Specifically, the virtualization manager provides IaaS facilities. The MEC application-platform services provide a set of middleware services as shown in Fig. 5.

For the management of the MEC server, there are three corresponding management systems, i.e., application management system, MEC application platform management system, and MEC hosting infrastructure management system. These systems provide interfaces for network operators to manage the MEC application platform as well as the life cycle and operability of the applications and services that are hosted on the MEC platform. MEC has many market drivers that enable MEC to support a wide variety of use cases, such as e-Health, connected vehicles, industry automation, augmented reality, gaming, and IoT services. ETSI encourages the proof-of-concept implementations of MEC to demonstrate the viability of MEC in various scenarios. Enormous research efforts have been invested following this direction. From the architecture perspective, these works could be further categorized into two groups, i.e., integrating SDN techniques into the virtualization manager and virtualization layer of the MEC servers and extending the management of MEC with the SDN techniques.

1) SDN-Enabled MEC Server: Salman *et al.* [63] proposed to build an architecture that employs the SDN paradigm while extending the MEC concept, called SD-MEC. The purpose of this project is to address the heterogeneity of the IoT devices, the privacy and security concerns, and the scalability of the network. In the proposed framework, the major components are the software-defined gateways (SD-Gateways). These gateways provide interoperability between different communication protocols and also ensure communication between different heterogeneous network islands. A collection of network functions, including routing access control, en-/de-capsulation of packets, NAT, and QoS, are also performed by these gateways. On the other hand, the southbound interfaces of the gateways are very specific for each type of network access techniques. The orchestration of these SD-Gateways is managed by a centralized SDN controller located in the cloud. Thus, the SD-Gateway combines the functionality of the MEC virtualization manager and the MEC virtualization layer to provide virtualized functions for the IoT devices.

Similarly, Liu *et al.* [64] also proposed an SDN-assisted MEC system to demonstrate its viability in the vehicular network. In this proposed framework, MEC servers are utilized to support delay-constrained response to client requests and facilitate deployment of new latency-sensitive services for service providers. While the programmability and scalability of SDN are leveraged to orchestrate different network entities, the SDN controller is deployed on the MEC server as a software component, managing the state of the client's virtual resources and the topology changes as the vehicle moves. For this purpose, the SDN controller needs to acquire position, direction, velocity, and network connectivity in real time. In the data plane, the underlying network resources, such as road side units (RSUs), vehicles, BS transceivers, and ethernet interfaces, are abstracted as SDN switches. Therefore, all these equipment and devices are OpenFlow compatible, and they are managed by the SDN controller via the OpenFlow protocol. The SDN components are also employed to provide virtualization functions and management functions in this architecture.

An emerging important application of SDN and MEC is the next-generation tactical networks that require high-speed data transmission to support necessary services. One such effort invested by Phemius *et al.* [65] aims to build an architecture to guarantee the QoS of tactical applications and maximize the usage of the radio links since they are valuable resources in tactical networks. The proposed architecture is an MEC server integrated with the SDN support. The underlying hardware resources in this project are software-defined radio (SDR) devices that are directly managed by an SDR controller. There is another SDN layer on top of the SDR layer. It consists of a virtual switch and an SDN controller. The functionality of this layer is to transparently steer the traffic through the applications and the radio interfaces by enforcing the strategy decided by the MEC application management system. Then, the application platform runs above the SDN layer and obtains statistics information collected by the SDN controller.

The MEC application management system receives updates from the SDR controller, the SDN controller, as well as the application layer. Then, it decides whether a radio parameter should be modified or not, or switching to another interface at the virtual switch via the SDN controller, or modifying the application parameters or the service chain to adapt to the underlying conditions. They demonstrate the viability of their system by testing RTT for different application scenarios. The evaluations show promising results in satisfying the QoS requirement for these applications as well as in efficient usage of the wireless resources.

2) SDN-Enhanced MEC Management: Since the MEC servers are usually deployed to manage hundreds of applications to millions of users, another paradigm is to utilize the SDN techniques to enhance the MEC servers with scalability and cost-efficiency. Jararweh *et al.* [66] proposed a software-defined framework to enable efficient MCC services by integrating different software-defined system (SDSys) components with the MEC system. These components include SDN, software-defined compute (SDCompute), software-defined storage (SDStorage), and software-defined security (SDSec). The main purpose of the system is to handle the global and local client requests in a smooth way. Thus, it follows a hierarchical design with a global controller layer and a local controller layer.

The global controller contains controller units for each component mentioned earlier. For the networking controller unit, it manages all the network components by configuring the switch flow tables. The local controller is responsible for covering, checking, and controlling its own domain and communicates with the global controller when needed. Such layering designs not only reduce the overhead of the global controller, avoiding the single point of failure problem, but also make the entire system more scalable, facilitating the process of expansion of the network.

Another integration effort has been invested by Huo *et al.* [67] to support energy-efficient information retrieval. In this paper, the authors proposed an integrated framework that can orchestrate different resources to meet the requirements of the next-generation green wireless networks. In the proposed architecture, the data plane contains three types of wireless access methods as examples: cellular networks, WLANs, and WiMAX networks. Each network node is equipped with caching and computing capabilities. The MEC servers are also considered as network nodes with computing and caching capabilities.

These networks are managed by a central controller to enforce different forwarding strategies. The central controller will collect information from the heterogeneous wireless devices and then define how packets should be handled based on the topology information and network applications deployed. The framework is evaluated using simulations. The results show that the proposed framework can decrease latency and save energy by jointly considering all the three resources.

A similar strategy that considers networking, caching, and computing techniques, in a systematic way was proposed by He *et al.* [68] in the scenario of smart city applications. In this paper, the authors proposed an integrated framework that can dynamically orchestrate these resources to improve the performance of applications for smart cities. In the proposed architecture, the SDN controller is used to manage the virtualized network resources equipped with caching and computing capacities. The resource allocations should be optimized to improve QoS for content delivery services, e.g., tourism services.

The resource allocation strategy is formulated as a joint optimization problem. A big data deep reinforcement learning model is proposed to resolve this problem. The input of the model includes collected status from each BS, MEC server, and content cache from each virtualized network. Then, all the information is assembled into a system state. An optimal policy for arranging which resources for a certain user is obtained as feedback. For the implementation, they implement the model with TensorFlow [69]. Through simulated evaluations, the results demonstrate that the optimization improves the total utility in different scenarios.

D. SDN in Edge-Cloud Interplay

Essentially, edge computing is designed to provide the same services as the cloud with reduced transmission latency and faster speed. While the edge nodes provide localization, the core cloud provides centralization, which is often required by applications. Thus, interplay and cooperation between the edge and the core are inevitable. Meanwhile, such interactions have some conflicting requirements, e.g., low-latency and power consumption, energy-efficiency and bandwidth, and security and resource sharing. Many efforts have been putting forth from both industries and academia to deal with these challenges.

In 2015, Deng *et al.* [70] developed a systematic framework to investigate the power consumption-delay tradeoff issue in the cloud-edge computing paradigm. They formulate the workload allocation problem and decompose it into three sub-problems that could be solved independently. The simulated results highlight the need to optimize the edge–cloud interactions. Following this line of direction, Borylo *et al.* [71] also proposed to build an SDN supported energy-aware interplay between edge and cloud. Meanwhile, the framework also needs to ensure the latency of video streaming delivery services.

To formulate this problem, they first categorize the data center nodes into two classes, green DCs and brown DCs, based on the energy source they utilize. They also assume that edge nodes appear at every edge of the network. There exists a central network controller that has full knowledge about the network topology and its state, including existing lightpaths (a path between two nodes in fiber optical networks) and the dc locations. The dynamic resource provisioning algorithms operate on a lightpath request (LR) basis.

LRs have two types: 1) a unicast LR and 2) an anycast LR (from a source node to one from the set of possible destinations). In their work, the unicast LRs are used to handle background traffic, and the anycast LRs could handle both fog node requests (FRs) and cloud node requests (CRs). Based on the different request types [i.e., Processing, Storage, and Software as a Service (PaaS, StaaS, and SaaS)], different strategies are applied. The simulation results demonstrate that the overall energy consumption of DCs can be reduced without compromising the network performance.

More recently, Kaur *et al.* [72] proposed an SDN-based edge-cloud interplay system to maintain QoS for various application in the IIoT environment without causing network congestion. In the proposed architecture, the edge-cloud interplay relies on the middleware that is SDN compatible in the networks. For this purpose, OpenFlow switches need to be deployed in the data plane, and a central controller is utilized to manage and schedule the traffic flows in the WAN. There are three phases when scheduling flows.

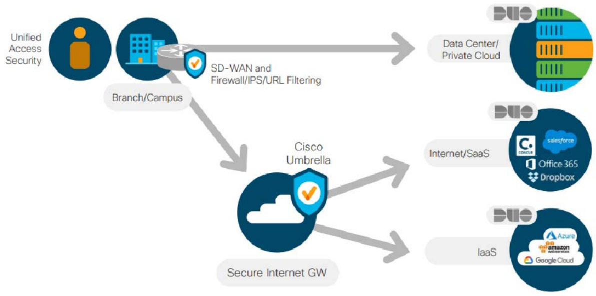


Fig. 6. ***Architecture of SD-WAN security stack (source: [74]).***

First, edge nodes classify flows into two categories: batch processing and stream processing, depending on which factor matters more, bandwidth or latency. In the second phase, the corresponding control logic will be selected based on the classification results. For both categories, there are optimization strategies to minimize the energy consumption while preserving the corresponding QoS requirements with two sets of QoS parameters. Finally, the selected logic will be executed on the SDN controller to achieve energy-driven flow scheduling and routing. The proposed schemes are evaluated with MATLAB to demonstrate their efficacy. The results show that the schemes could improve energy utilization while restraining service level agreement (SLA) violations.

Cisco, as the largest manufacture in the routing and switching market, has easy access to deploying SD-WAN functionality in their appliances. In August 2017, Cisco completed its acquisition of Viptela, a new SD-WAN solution that now includes the companies Intelligent WAN (iWAN) product [43]. Cisco SD-WAN based on Viptela has been envisioned to be the preferred solution for large, complex deployments.

According to Cisco, Viptela provides a compelling SD-WAN solution with advanced routing, segmentation, and security capabilities for interconnecting complex enterprise networks. From a recent blog of Cisco, they will deploy an SD-WAN security stack with capabilities that solve critical edge security challenges [73]. Unlike the traditional way of security mechanisms deployed at the edge sending all the traffic back to the data center for inspections, this new SD-WAN security stack provides a complete shield operating at the edge, in the branch router, with centralized control for both network and security management. The motivation to deploy security stack in SD-WAN is to balance the security need and user experience for enterprises. An overview architecture of SD-WAN integrated with security solutions is shown in Fig. 6. In Fig. 6, Cisco Umbrella provides web-layer

Table 1 Summary of Projects Integrated SDN With Edge Computing

Project	SDN Integration Architecture	Functions							Application Scenario
		QoS	Security & Privacy	Visibility	Reliability	Energy Efficiency	Resource Utilization	Scalability	
CORD	Edge	N/A							Residential/Enterprise/Mobile Users
SDOAN	Edge	✓	✓	✓					Residential Network
Softwarized 5G Network	Edge	✓			✓				VANET
Software Defined VANET	Core				✓	✓			MANET/VANET
FSDN	Core	✓					✓		VANET
EdgeIoT	Core		✓		✓				Smart City
SD-MEC	MEC	✓	✓					✓	IoT Devices
SDN-enabled VANET	MEC	✓						✓	VANET
SDN-enabled Tactical Network	MEC	✓					✓		Tactical Network
SD-MEC	MEC	✓			✓			✓	Mobile Users
SDN Green Wireless Network	MEC	✓				✓			Mobile Users/VANET
SDN enabled MEC for Smart cities	MEC	✓					✓		Smart City
Energy-aware Fog-Cloud Interplay	Edge-Cloud	✓				✓			Enterprise Network
Edge Computing in IIoT	Edge-Cloud	✓				✓			Industrial IoT Devices
Cisco SD-WAN	Edge-Cloud	✓		✓					Enterprise Network

security by blocking traffic based on DNS requests. The combined SD-WAN security stack and Cisco Umbrella protects enterprise traffic in various business models.

There are four key traffic profiles that will expose threat surface for attackers: 1) sensitive data need to be protected at rest and in transit—in the branch and in the cloud; 2) existing open-network ports with direct internet connectivity; 3) providing direct access to cloud resources and SaaS applications bypassing existing centralized security solutions; and 4) enabling guest access to local Wi-Fi from personal devices. For the first profile, Cisco SD-WAN security adds an embedded application-aware firewall in the branch router that learns and enforces which applications can access sensitive data types. Then, SD-WAN routes sensitive traffic through a secure VPN to the applications in the enterprise data center. For direct Internet access, the SD-WAN provides a combination of embedded security functions. Then, the SD-WAN fabric intelligently routes traffic to and from branches based on the SecOps policies.

To address the security threats with direct cloud access, SD-WAN security leverages the DNS security layer, together with intrusion detection, to prevent the most aggressive Denial of Service, phishing, malware, and the ransomware attacks. Finally, organizations should prevent guests from accidentally, or maliciously, downloading malware that could infect the branch network. For this purpose, SD-WAN security stack provides web filtering, intrusion detection, and prevention capabilities to prevent internet infections from guest devices spreading through the network. Employing these security functions with SD-WAN makes it more flexible and scalable to accommodate new security schemes and functions while preserving user experience.

To summarize these research and development efforts, Table 1 lists their architecture, functionality, and application scenarios.

IV. INTEGRATION OF SDN AND IoT

The recent advances of the IoT, which incorporates a large number of edge devices with heterogeneous characteristics (e.g., device category, functionality, manufacturer, and communication protocol), demand new network architectures to accommodate a variety of new challenges, such as the complexity in the management of heterogeneous devices, protocol, and network resources, the explosion of generated data, as well as security and privacy. In Section III, we have discussed that from the network perspective, SDN and NFV have emerged as promising technologies to provide the scalability, versatility, and security that are essential for the IoT services. On the other hand, there has been a clear trend of integrating SDN and IoT. In the following, we survey the recent research efforts that focus on leveraging SDN to address the various challenges. In particular, we investigate how SDN could be employed to achieve efficient and effective device control and network management, resource allocation, and security/privacy guarantee.

A. SDN for IoT Device Control and Network Management

1) Using SDN to Manage WSNs: The wireless sensor network (WSN) has been a key enabler of IoT. There have been a series of works that leverage SDN for the WSN management. The study [75] proposed SDN-WISE, a scalable SDN solution for managing WSN. SDN-WISE supports duty cycle and data aggregation to reduce the amount of information exchanged between the sensor nodes and the SDN network controller. Moreover, it makes sensor nodes programmable as finite state machines, such that operations unsupported by stateless solutions could be executed. Bera *et al.* [76] proposed a software-defined WSN (Soft-WSN) architecture to support application-aware service provisioning. Soft-WSN relies on a specialized SDN controller that focuses on both device management and topology management to meet run-time application-specific requirements of IoT while enhancing flexibility and simplicity of WSN management. The work [77] proposed an approach of leveraging SDN programmability for smart management in WSNs. The proposed framework consists of a base station (BS) as the controller node and several sensor nodes. The controller node communicates routing decisions to the sensor nodes that contain flow tables configured by the SDN controller.

Network updates remain to be an important issue for device management and network control in SDN-IoT networks. In [78], a new batch-level update mechanism (BLLC) is developed to achieve safe and consistent network update with low resource consumption. It first bundles the update rules into the batch-level control packets and then performs network update in the reverse direction of new flows (i.e., from the destination to source). In addition to the batching of control commands, it offloads the transmission of control packets to the IoT nodes, which can greatly reduce the resource consumption of network update.

2) Software-Defined Vehicular Networks: Internet of Vehicles (IoV) has emerged as an essential component of the IoT landscape. In a vehicular network, multiple vehicles with specialized onboard hardware are interconnected through a series of communication technologies (e.g., LTE, 5G, and WIFI) to form vehicle-to-vehicle (V2V) communication. Through the V2V connections, real-time traffic status information could be expeditiously disseminated to the adjacent vehicles to avoid congestion. Moreover, dedicated infrastructures, such as cellular BSs and roadside units, could provide data services (e.g., mobile Internet access) to the vehicles through the vehicle-to-infrastructure (V2I) connections. However, the heterogeneity of communication protocols, the diverse QoS, and the scalability requirement pose challenges for the development of practical vehicular networks. As a result, SDN has arisen as a promising approach for vehicular network management and control.

Due to the increase of VANET applications, such as unbalanced flow traffic among multipath topology and inefficient network utilization, flexible vehicular architectures are key requirements. In 2014, Ku *et al.* [79] proposed to build an SDN-based VANET framework to address the service deployment issues. In this design, each RSU is equipped with a local agent that is controlled by the remote SDN controller. The central controller enforces policies by inserting rules into RSU. Due to the awareness of the central controller, several existing services, such as path selection, frequency/channel selection, and power selection, could be further improved. For example, traffic rerouting becomes faster to reduce congestion with the help of the SDN controller. Also, the SDN controller has the ability to reserve a certain communication channel for emergency traffic. However, the proposed SDN controller does not address the specific management challenges of the provided services in the VANET environment. The authors implemented the architecture using the NS-3 simulator to testify the feasibility of applying SDN to VANET. The evaluation results demonstrated SDN as a very promising technique to enhance VANET. Nonetheless, due to the lack of testbed, it is unclear what are the operational challenges to deploy the proposed framework.

Following this work, Truong *et al.* [80] proposed another SDN-based architecture to support VANET with edge computing, called FSDN. The authors argue that real-time processing is required by many time-sensitive services, such as safety services. Thus, edge nodes are essential to VANET. However, the increasing number of nodes complicates the network management. The adoption of SDN techniques could resolve this challenge by taking into account various heterogeneous characteristics, such as physical medium, mobility, topology, and capability. On the other hand, it might generate more traffic in the core network for service orchestration and creation on the edge servers. The proposed solution could not address this issue. In addition, the proposed framework was not sufficiently evaluated using simulations or prototyping.

In the proposed architecture, the RSU nodes are OpenFlow compatible, and they are controlled by a local SDN RSU controller (RSUC). The local RSUC stores local road information and handles emergency service requests. Similar to RSUC, each cellular BS also provides local intelligence. Both RSUC and BS are under the control of an SDN controller that is located between the cloud and the edge nodes. The RSUCs and BS share their resources with the cloud through the SDN controller for controlling vehicles. The SDN controller mainly functions as the edge nodes orchestration and resource manager. The benefits of this design are twofold. First of all, the SDN controller could optimize the service configurations for particular vehicles. Second, the service provider becomes aware of the mobility of vehicles. Thus, resources could be provisioned to nearby vehicles when the target vehicle is out of reach of RSUs.

Jiacheng *et al.* [81] proposed a generic software-defined IoV (SD-IoV) architecture that integrates SDN with IoV and discussed the challenges and potential solutions for realizing SD-IoV. SD-IoV consists of layered architecture of functions. The logical SDN controllers are responsible for normal network management tasks and some advanced functions, such as data processing and learning. They can be physically placed into the cloud or in the local proximity, which allows distributing the various functions among the controllers. Moreover, the controllers manage the layered data plane devices (i.e., SDN switches, wireless access infrastructures, and vehicles) through a wired or wireless control path. To enable vehicular packet transmission control, SD-IoV proposed and compared three possible wireless control path implementation schemes. Furthermore, this paper discussed the open challenges for SD-IoV, such as scalable resource sharing, control functionality placement, and security/privacy.

Besides, a series of works focused on a specific aspect (e.g., resource sharing, network management, and security) of SD-IoV. Peng *et al.* [82] proposed a novel architecture for flexible resource management and balanced resource utilization. As shown in Fig. 7, the architecture integrates SDN with MEC to achieve optimal bandwidth and computing/storage resource allocation. The SDN control modules in the cloud/MEC servers enable the interworking of multiple wireless access works to handle the sheer data volume. The computing and storing capabilities deployed at the MEC make it possible

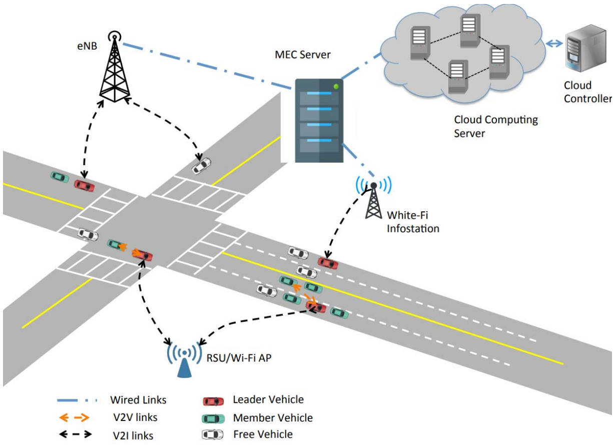


Fig. 7. *SDN-enabled MEC architecture for autonomous vehicles.*

to meet the heterogeneous QoS requirements (e.g., short response delay is essential for safe and cooperative driving). Under the new architecture, various resource management schemes are formulated as an optimization problem to enhance computing/storing and bandwidth resource utilization. Each MEC controller is featured with SDN and NFV control modules that allocate bandwidth resources of different access technologies among BSs to enhance bandwidth utilization. A case study is conducted to demonstrate the effectiveness of the proposed resource management schemes.

Chen *et al.* [83] focused on the vehicular connection management in SD-IoV. They developed a centralized approach to allocate dedicated communication resources and underlying vehicular nodes to fulfill the need for each service. It formulates the dynamic vehicular connection as an overlay vehicular network creation (OVNC) problem and designs a novel utility function that incorporates the network resource utilization efficiency, service QoS, and concurrent request as the objective of the optimization problem. Finally, a graph-based genetic algorithm and a heuristic algorithm are developed to solve the OVNC problem. The effectiveness of the proposed approach for vehicular connection management is evaluated through simulations.

Quan *et al.* [84] investigated vehicle-assisted data offloading in SD-IoV to alleviate the pressure on the core networks for transmitting the ever-increasing vehicular data. It proposed a software-defined collaborative offloading approach for heterogeneous vehicular networks. A centralized offloading controller is employed to globally disseminate offloading policies to the vehicular nodes. It consolidates two major functionalities: hybrid awareness path collaboration (HPC) and the graph-based source collaboration (GSC). HPC solves the path collaboration problem based on a path quality metric that jointly considers the path bandwidth, packet loss rate, and round-trip time. GSC selects the offloading sources as the minimum vertex cover set, such that all the nodes in the network could be connected.

A safe and intelligent transportation system necessitates efficient distribution of bandwidth-intensive content, such as real-time traffic status and multimedia data. Cao *et al.* [85] proposed a type-based content distribution (TBCD) approach for data-intensive content distribution in vehicular networks. TBCD performs segment-by-segment transmission by following the publisher–subscriber model. The server publishes the content once, and the SDN logic agents on switches determine the copies of content and the paths to the RSUs according to subscriber location and the number of subscription requests. Then, the SDN agents on RSUs keep track of the subscribers of specific content and employ control flooding to broadcast the contents to the subscribers.

3) Using SDN in Smart Environment: The emergence of IoT has enabled a variety of applications, such as smart city/home, intelligent transportation, and smart healthcare. To support the diverse IoT applications, it is a crucial challenge to achieve efficient management of the underlying physical infrastructure. Due to the flexibility and programmability of SDN in network flow control, integrating SDN with IoT serves as a potential avenue for resolving this challenge in various smart environments, for instance, smart cities [68], [86]–[89], smart homes [90], [91], and smart health [92]. Recently, a number of SDN-enabled IoT network architectures have been proposed. The general architecture of an SDN-enabled IoT network typically consists of three layers: the application layer, the controller layer, and the infrastructure layer [92], [93]. Nonetheless, a single centralized controller with limited processing capacity is insufficient to provide the scalability and reliability in large-scale IoT networks. It is therefore a natural choice to deploy multiple controllers to support the vast amount of IoT applications and devices [88], [89], as shown in Fig. 8.

Wu *et al.* [88] presented UbiFlow, a framework for mobility management in the urban-scale software-defined IoT (SD-IoT) multinetworks. The system architecture for UbiFlow is illustrated in Fig. 8. UbiFlow coordinates multiple distributed controllers to execute various tasks, such as mobility management, flow scheduling, and handover optimization. Using multiple controllers, an urban-scale SDN could be divided into different geographic partitions so that IoT flows could be controlled in a distributed way. Moreover, to preserve network consistency and scalability, Ubiflow incorporates a distributed hashing-based overlay structure. Given network status analysis and flow requests as inputs, the controller employs an optimal assignment algorithm to match the most suitable access points to the IoT devices. By analyzing flow characteristics variation, UbiFlow develops a load balancing scheme to allocate flow requests among distributed controllers. More specifically, network and device information of the IoT multinetwork is

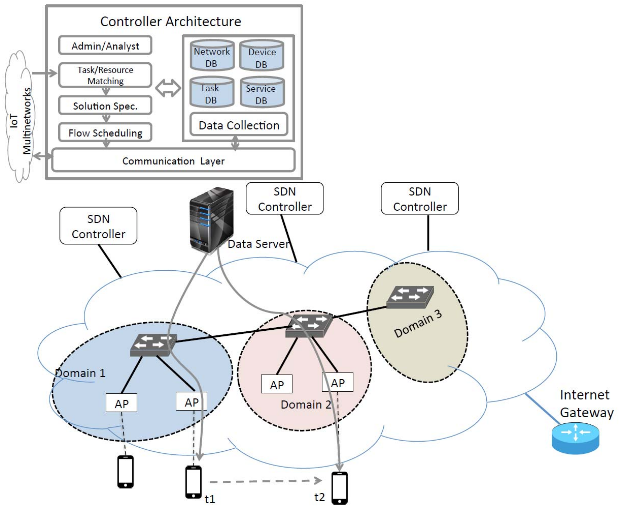


Fig. 8. ***System architecture of UbiFlow.***

collected by the data collection module that is utilized by the layered components in the controller in order to schedule flows based on both flow requirements and network characteristics and constraints.

The domain partitioning problem in SD-IoT networks is explored in depth in [89]. A partitioning algorithm for SD-IoT network (PASIN) is proposed to partition the SDN data plane based on flow paths of urban sensing applications. It aims to optimize the load of requests to the controllers with the constraint of total switch-to-controller delays in each domain. Extensive simulations are conducted to demonstrate the effectiveness of the proposed scheme.

Liu *et al.* [93] proposed an SD-IoT architecture for smart urban sensing (as shown in Fig. 9), decoupling high-level applications from the physical infrastructure, including sensor platforms, forwarding devices, and servers. It provides well-defined service APIs in terms of data acquisition, transmission, and processing, through which each application (e.g., smart transportation, air pollution monitoring, and noise-level monitoring) could customize their own service requirements. This paper also presents some open problems, such as mobility management, conflict resolution, and optimization for the sensor platform and QoS enable traffic scheduling, followed with their potential solutions. The architecture of SD-IoT enables flexible control and management of the physical infrastructure and facilitates the development of urban sensing applications.

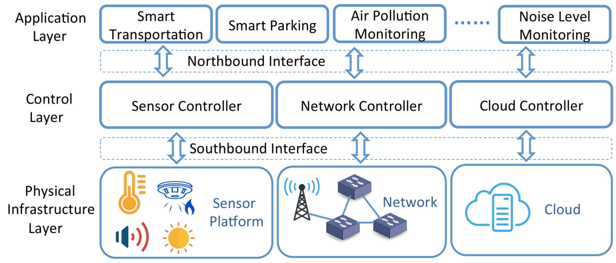


Fig. 9. ***Architecture of SD-IoT.***

Network heterogeneity and ultra-densified deployment of BSs and access points pose great challenges for future smart cities, including load balancing, handover, and interference issues. To cope with these issues, a converged cell-less communication architecture is proposed in [86]. Specifically, a mobile device does not associate with any BS/AP before data transmission, whereas a centralized SDN controller is employed to dynamically adjust the BS/APs that perform the transmission based on the requirement of the mobile terminal and wireless channel status. Simulation results demonstrate that the converged scheme improves the coverage probability and energy saving at both BSs and mobile terminals.

With the growing number and extensive heterogeneity of smart devices in smart homes, current management platforms fall short in providing the convenience and flexibility to the users. To address the challenges, Xu *et al.* proposed a software-defined smart home platform (SDSH), by using SDN's features of centralization, optimization, and virtualization [90]. The SDSH platform consists of three layers: a smart hardware layer, a controller layer, and an external service layer, where the controller layer could be deployed either in physical hardware at the user's home or in abstract equipment in the cloud. To achieve the intelligent and adaptive control and management of the smart home devices, the control layer shields the hardware details, perceives user demands, and manages system resources and task scheduling in a centralized manner. Nonetheless, a real-world deployment of such a management platform still faces several challenges. For example, the restricted battery capacity and home obstacles pose challenges for the communication between the devices and the controller. Besides, security mechanisms are needed to protect the privacy of users' data.

Aside from its application in smart cities and smart homes, SDN is also employed in health surveillance systems. Hu *et al.* [92] presented a general software-defined healthcare networking architecture for intelligent health surveillance based on the healthcare IoT (HealthIoT). A centralized controller manages the shared infrastructure and provides APIs for health surveillance and intelligent healthcare applications. The integration of SDN into health monitoring systems facilitates elastic control and management of the shared infrastructure.

B. SDN for IoT Resource Management

The coexistence of multiple heterogeneous device and network resources in the wide-area deployments of IoT subnetworks creates opportunities for a wide range of applications with varying service requirements to execute concurrently. However, it also poses new challenges, such as efficient and shared provisioning of network and sensor resources among applications. Resource provisioning and management are critical for ensuring end-to-end QoS [94].

In [95], an SDN-based architecture is proposed to dynamically achieve application-specific quality levels in heterogeneous wireless networking scenarios. The proposed architecture extends the Multinetwork INformation Architecture (MINA) middleware with a layered IoT SDN controller. The controller introduces commands to differentiate flow scheduling over task-level heterogeneous ad hoc paths and employs network calculus and genetic algorithms to optimize the usage of available IoT network. Various network and device information for the IoT Multinetworks is gathered by a data collection component and stored into the corresponding databases. This information is subsequently utilized by the layered controller components. By introducing multiple levels of abstractions, the IoT controller could flexibly leverage the heterogeneous multinetworks' resources.

Considering the inflexibility of deploying new features in legacy mobile networks, Pentikousis *et al.* [96] proposed MobiFlow, a software-defined mobile network architecture that fosters innovations inside the mobile network. An existing mobile broadband ecosystem consists of diverse hardware with standardized interfaces, which requires indispensable hardware or API modifications to support new technologies, such as cloud computing and content distribution networks. In response, MobiFlow leverages SDN to enhance the programmability and flexibility of future carriers without mandatory interface changes. It introduces a blueprint for flow-based forwarding in the mobile network. To decouple mobile network control from the data plane elements, it incorporates a MobileFlow controller (MFC) and a MobileFlow forwarding engine (MFFE). MFC is logically centralized and provides APIs for various functions, such as network resource management and topology discovery. MFFE performs the data plane operations programmed by MFC.

C. SDN for IoT Security and Privacy

Despite the benefits of IoT, widespread concerns have been raised about the security and privacy associated with the vulnerable IoT devices [97]–[101]. A multitude of SDN-based solutions has been proposed to enhance IoT security [91], [102], [103]. One focus of the existing research is on the design and implementation of new SDN-based security architectures for IoT.

One example is IoT SENTINEL [102], a security system for mitigating the security and privacy risks posed by insecure IoT devices. IoT SENTINEL is capable of automatically identifying the types of IoT devices and applying SDN-enabled mitigation measures to those vulnerable devices. As shown in Fig. 10, it consists of an SDN-based security gateway and a cloud-enabled IoT security service. Colored lines in Fig. 10 represent the data flow from the security gateway to the security service for device identification and information returned by the security service, respectively. Specifically, a device-specific isolation level with additional information is returned and stored in the local cache at the security gateway, which is further utilized to generate enforcement rules. The security gateway is an edge-based vantage point that interconnects the local IoT network to the Internet. It serves as the core traffic control and monitoring component. Specifically, it monitors and profiles the behaviors of the IoT devices and extracts network-based device fingerprints from the initial network sessions when the device joins the network. The device fingerprints are then sent to the centralized IoT security service that employs supervised machine learning to identify the device category and performs vulnerability assessment based on pre-established threat intelligence. For potentially vulnerable devices, the SDN controller within the security gateway deploys various mitigation measures via the security gateway to minimize the risk of harm, such as traffic filtering, device isolation, and user notification. However, the IoT SENTINEL has some limitations. First, its centralized design (in both fingerprint-oriented traffic monitoring and machine-learning-based device identification) is not scalable in large-scale IoT networks. Second, device fingerprinting requires that the new device falls into the existing categories, which is not feasible due to the heterogeneity of IoT devices. Therefore, IoT SENTINEL is merely applicable to small home and office networks.

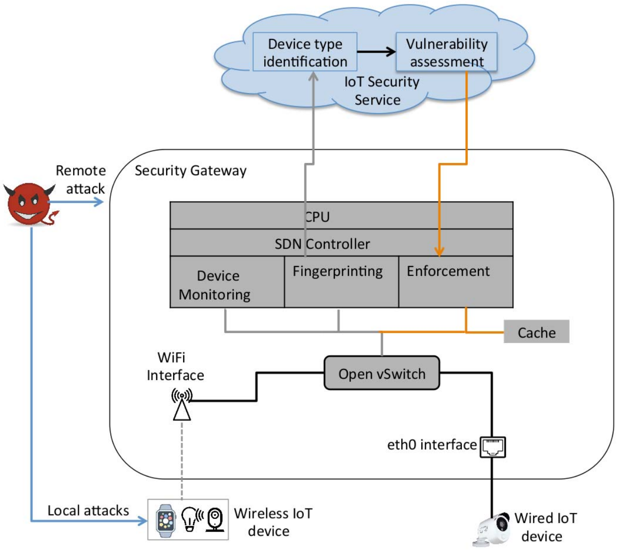


Fig. 10. ***IoT SENTINEL system architecture.***

Sahoo *et al.* [104] proposed a secure architecture for the SDN-based ad hoc IoT network. The proposed security mechanism does not rely on global traffic monitoring. Instead, it leverages the SDN controllers to enforce the authentication of IoT devices. Specifically, the SDN controller implements and initiates the authentication logic whenever an ad hoc network device joins. When a secure connection is initially established between the switch and the controller, all the switch ports are blocked, and the controller will start the authentication process with the device. Only after the device is successfully authenticated, the controller would install suitable flow rules into the switches for the forwarding of the device traffic. The proposed architecture has several limitations. First, the protected network needs to incorporate specialized nodes that integrate legacy interfaces, SDN controller, and programmable data plane, which is unrealistic in a practical IoT environment. Second, the architecture is too generic. There is no discussion about the applicable authentication mechanism. Moreover, the proposed architecture is not scalable in realistic IoT network that could be largely distributed. In distributed settings with multiple domains, appropriate synchronization mechanisms should be devised to guarantee the consistency of cross-domain security rules.

A secure IoT architecture for smart cities was proposed in [105] that aims to ensure data security and privacy. The proposed architecture consists of four basic components: black networks, trusted SDN controller or trusted third party (TTP), key management system, and unified registry. Specifically, a black network guarantees data security and privacy through encryption. To mitigate cyberattacks initiated from IoT nodes, strict authentication is also enforced for the heterogeneous devices. One limitation of the black network is that it encrypts the network layer headers, which raises challenges for routing IoT flows. To address the routing challenge, the trusted SDN controllers control and coordinate the communication flows among the IoT nodes and the remaining networking infrastructure. It maintains a global view of the IoT network topology and generates routing rules for the black network packets. The key management system is hierarchical and distributed. It is responsible for the generation, distribution, and revocation of the shared keys used in secure communication. The unified registry accommodates the heterogeneity of devices/protocols involved in realizing smart city. It serves as a unified repository that manages the identity and attributes of IoT devices.

Inspired by existing network security and access control techniques, Flauzac *et al.* [106] proposed an SDN-based security architecture that secures both wired and wireless network infrastructures. The architecture is also extended to include ad hoc networks and IoT devices. Specifically, the entire network is separated into multiple distributed SDN domains, and each domain is managed by one or multiple controllers. Within each domain, a specialized root controller, called the border controller, is introduced to communicate with the border controller in the other domains. In this way, the routing functions and control rules are distributed on each border controller. Since the security policies and management strategy are domain-specific, a Grid of Security concept was employed to resolve the issues associated with security policy heterogeneity.

V. DISCUSSION

The fast development of edge computing and IoT applications brings a lot of new opportunities and motivates many active research projects and products in SDN, of which some representative ones have been discussed before. This leads to a mix of various setups and platform configurations, as depicted in Fig. 11. As we can see, the edge computing platforms usually consist of five layers: device layer, edge cloud layer, WAN (network) layer, back-end cloud layer, and application layer.

Such mixtures introduce complexities and lead some challenges that future networking research, in particular, SDN and NFV, needs to address. Not exclusively, we discuss some of these challenges in each layer, particularly focusing on how they impact the network layer.

A. Heterogeneity

One of the ultimate goals of edge computing is to accommodate many devices and to provide various services. As a result, the heterogeneity could only increase along time. Such heterogeneity includes not only the devices and sensors used for different applications (e.g., auto-driving uses a different set of sensors and actuators than those used for smart and connected health) but also the way how these devices communicate with each other and with their service providers, e.g., using Bluetooth, Zigbee, Wi-Fi, cellular direct interfaces, and running HTTP/TCP or UDP protocols. While the OpenFlow is the de facto standard to communicate between the SDN controller and the switch, there is no such standard for the communication from the devices to the access points. With the proliferation and the varieties of the sensors and edge devices, new standards are needed to be in place to enable smooth communication with the network layer.

B. Interoperability

The existing literature shows that a lot of edge computing innovations, platforms, and architectures are application driven, that is, they are designed or built for a specific type of applications, such as auto-driving with RSUs or smart homes with smart gateways. In the future, applications may be composed on such existing services, which will require the interoperability and coordination among different edge platforms and edge computing nodes. Moreover, different edge computing nodes may belong to different organizations. In addition, interoperability and coordination are essential to address the mobility of different devices. Although SDN can quickly facilitate the deployment of different nodes, how SDN can help with the coordination and interoperability of different edge computing nodes remains to be addressed. Ultimately, we envision that specialized and generic edge computing platforms would co-exist.

C. Mobility/Connectivity

In the application layers, some IoT devices naturally demand mobility support from the edge computing nodes. For example, with auto-driving, when the vehicle is traveling, the vehicle is interacting with the RSUs for traffic light

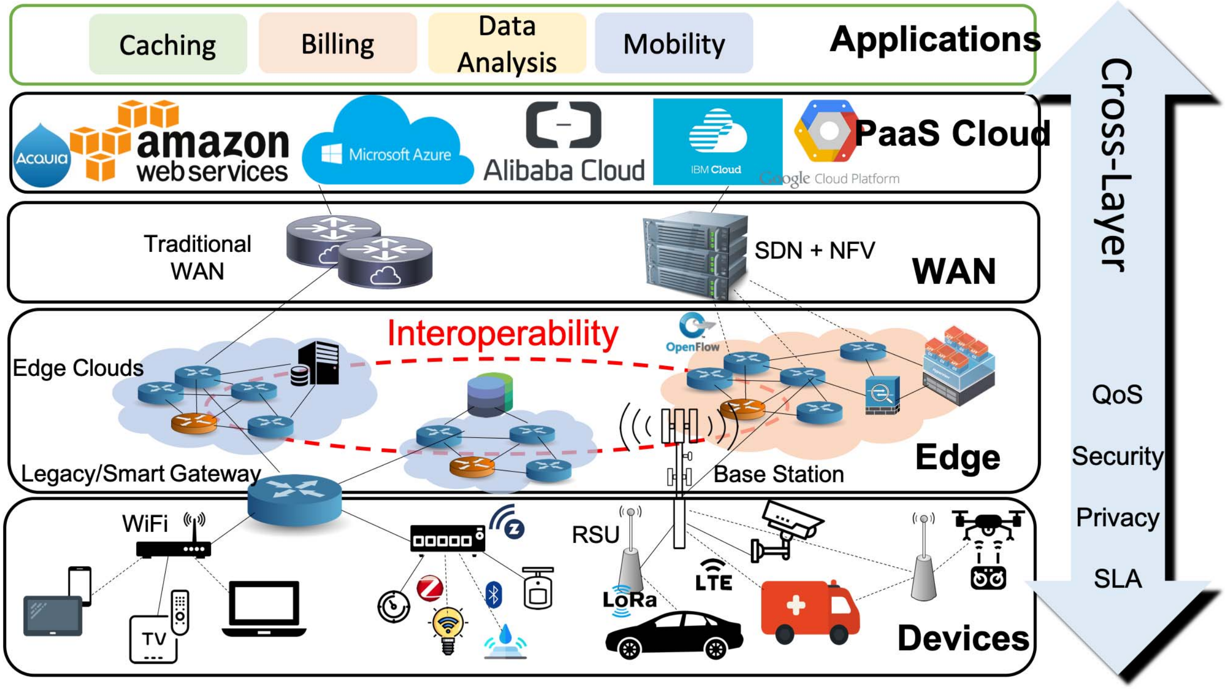


Fig. 11. ***Architecture of edge computing platforms.***

information and other road condition information. Existing studies often focus on the smooth handover, with different prediction strategies, to make sure the shortest connection break. From the information or data perspective, such communications may be intermittent or sporadic depending on weather or other external factors. Given such considerations, how to properly arrange the information delivery (e.g., caching and CDN) with the SDN support so that critical information will not get lost deserves more in-depth study.

There are some cross-layer applications that require joint efforts from each layer, such as QoS and security. However, network layer plays a significant role to enforce policies of certain requirements.

D. QoS and SLAs

Similar to cloud computing, there are different QoS requirements from the application perspectives to the edge computing servers. For example, virtual reality is both delay-sensitive and bandwidth-intensive, while monitoring the heartbeat is only delay-sensitive. On the other hand, healthcare traffic deserves a higher priority than entertainment traffic. Given the multitenant environment, how SDN can provision the available network resources to support different levels of QoS so as to meet the SLAs will be a critical challenge before they can be practically adopted. Virtualization (e.g., NFV) can help, but we envision that the proper interactions between the NFV and SDN are a key.

E. Security and Privacy

While delay and bandwidth are two key motivations for edge computing, an equally important one is the security and privacy by which the local data with sensitive information do not have to be uploaded to the cloud. However, when the edge servers may belong to different organizations, and multiple applications may co-exist on an edge server, security, and privacy concerns again become prominent. Furthermore, with SDN networking with different edge devices, it greatly increases the attack surface and it becomes harder to defend against attacks. On the other hand, SDN also makes it more agile and scalable to deploy security functions at the edge. How SDN can help properly separate the traffic from different devices, applications, and users and provide sufficient security and privacy protection yet maximally utilizing the available communication channel deserves more research. The monitoring capabilities need to be improved for more inclusion and flexibility.

F. Testbed

Last but not least, a lot of existing studies mainly focus on investigating and designing new architectures to enable SDN-based networking support for edge computing, where the SDN controller can manage a set of sensors/actuators directly or via SDN capable switches, or the communications to the SDN-enabled data centers or back-end clouds. To validate such proposals, experiments were often conducted via simulations or small-scale prototypes.

To facilitate the in-depth research on the SDN and NFV support in edge computing and the deployment of edge computing platforms, proper testbeds are in imperative need so that future innovations can be quickly evaluated and "benchmarked" on such standard testbeds. Such testbeds can also establish a common ground, where different approaches and innovations could be validated and compared. Living Edge Lab [107] and EdgeNet [108] are examples of such efforts but still at the initial stage.

VI. CONCLUSION

In this paper, we have surveyed some recent and representative work in designing and implementing edge computing framework equipped with the most advanced network technologies, e.g., SDN, and how they have been applied to different application scenarios. Our investigation focuses on the network layer (including both the access network and WAN) of the framework. Without loss of generality, we consider two types of "edge" devices in our explorations: mobile or stationary devices that are equipped with computing and storage capabilities, e.g., wearable devices and smart gateways; and servers and storage units deployed at the edge to provide services for other IoT devices. Considering different situations, SDN and NFV could be deployed in either access network, or core network, or the WAN between the edge and the core.

For the deployment of SDN in the access network, most existing frameworks are proposed to address the QoS or network reliability issues for mobile devices or vehicles. For the core deployment of SDN, it is usually utilized to coordinate traffic or services for different purposes, e.g., to reduce the amount of traffic being sent to the cloud, to improve the resource utilization at the edge, or to save operation energy. SD-WAN is usually deployed in a smaller scale and controllable environment, such as industry, airport, or enterprise. It could serve different application requirements. For example, Cisco SD-WAN is proposed to bring security functions closer to the access points to prevent security threats from spreading through the network. A special type of edge servers is exclusively studied, i.e., MEC server, since it has been standardized by ETSI and believed to be a key technology and architectural concept to enable the evolution of 5G. For this specific edge node, SDN could be utilized to replace the existing virtualization layer or the management systems of the server so that SDN could facilitate the agile creation and deployment of services at the edge. Furthermore, the management systems will be simplified, leveraging the programmability of SDN. There are also many applications of SDN supported MEC edge frameworks, such as its employment in the tactical networks to improve the utilization of valuable MEC servers in the tactical network. Besides the network level services, SDN also provides a unified framework to manage various IoT devices.

To deal with the heterogeneity issues of the devices, SDN could either virtualize physical devices or provide customized services for different devices. SDN could be utilized to achieve effective and efficient IoT device control and network management, resource allocation and provisioning, and security/privacy guarantees. The programmability and flexibility of SDN could enable smart management of traditional IoT environment (e.g., WSN) and more heterogeneous smart environments (e.g., smart vehicular network and smart city). Moreover, by abstracting the underlying network resources and providing centralized intelligence, SDN serves as a promising paradigm to realize application and QoS-aware resource provisioning and management. However, the centralized nature of SDN may cause scalability issues in real-world IoT setup. A distributed and hierarchical SDN controller paradigm could be a potential solution. The immense heterogeneity, sheer scale, and prevalent device vulnerabilities of IoT also raise security and privacy challenges. In this scenario, various security architectures could be built based on SDN. For example, SDN could provide intelligent access control of IoT devices through authentication, secure transmission of IoT data through encryption, as well as adaptive enforcement of security policies.

However, some challenges need further investigations for the marriage of edge computing and SDN/NFV technologies. For example, SDN is designed to be a network layer architecture. To accommodate and interact with different IoT and edge devices, it needs to support various physical and data link layer protocols, e.g., Zigbee. Furthermore, due to the different communication channels, the network functions deployed in SDN need to be carefully designed to work in such a hybrid environment. With the development of auto-driving and unmanned aerial vehicles (UAVs), mobility and network connectivity of these vehicles become imperative to be addressed. Especially for the long-distance movements, interoperability of different edge clouds belonging to different organizations has not been considered yet. Some critical service requirements, such as QoS and security, become more challenging to enforce since they are cross-layer tasks that require a joint effort. In these cases, a hierarchical SDN design might be necessary. Through our investigation, we find that edge computing facilitated by SDN/NFV is still in its infancy stage. It is a promising area full of opportunities and challenges. ■

Acknowledgments

The authors would like to appreciate the constructive comments from the reviewers.

REFERENCES

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

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[3] Q. Zhang, Z. Yu, W. Shi, and H. Zhong, "Demo abstract: EVAPs: Edge video analysis for public safety," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 121–122.

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[5] 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, Dec. 2013.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[7] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[8] F. Lobillo *et al.*, "An architecture for mobile computation offloading on cloud-enabled LTE small cells," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2014, pp. 1–6.

[9] S. Wang *et al.*, "Mobile micro-cloud: Application classification, mapping, and deployment," in *Proc. Annu. Fall Meeting ITA (AMITA)*, 2013, pp. 1–7.

[10] K. Wang, M. Shen, J. Cho, A. Banerjee, J. Van der Merwe, and K. Webb, "MobiScud: A fast moving personal cloud in the mobile network," in *Proc. 5th Workshop Things Cellular, Oper., Appl. Challenges*, 2015, pp. 19–24.

[11] J. Liu, T. Zhao, S. Zhou, Y. Cheng, and Z. Niu, "CONCERT: A cloud-based architecture for next-generation cellular systems," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 14–22, Dec. 2014.

[12] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, 2018.

[13] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[14] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," 2016, *arXiv:1612.03184*. [Online]. Available: https://arxiv.org/abs/1612.03184

[15] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017.

[16] *Amazon EC2*. [Online]. Available: https://aws.amazon.com/ec2/

[17] *SAAS, PaaS, & IaaS: Cloud Computing Service Models|Doublehorn*. [Online]. Available: https://doublehorn.com/saas-paas-and-iaas-understanding/

[18] A. M. Rahmani *et al.*, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[19] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[20] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 26–32, Sep. 2018.

[21] F. Cicirelli *et al.*, "Edge computing and social Internet of Things for large-scale smart environments development," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2557–2571, Aug. 2018.

[22] *Internet of Things at a Glance*. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/se/internetof-things/at-a-glance-c45-731471.pdf

[23] S. Misra, S. Das, M. Khatua, and M. S. Obaidat, "QoS-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 181–193, Apr. 2014.

[24] M. H. Zarei, M. A. Shirsavar, and N. Yazdani, "A QoS-aware task allocation model for mobile cloud computing," in *Proc. 2nd Int. Conf. Web Res. (ICWR)*, 2016, pp. 43–47.

[25] A. Karamoozian, A. Hafid, M. Boushaba, and M. Afzali, "Qos-aware resource allocation for mobile media services in cloud environment," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2016, pp. 732–737.

[26] M. Akter, F. T. Zohra, and A. K. Das, "Q-MAC: QoS and mobility aware optimal resource allocation for dynamic application offloading in mobile cloud computing," in *Proc. Int. Conf. Elect., Comput. Commun. Eng. (ECCE)*, 2017, pp. 803–808.

[27] A. A. Laghari, H. He, A. Khan, N. Kumar, and R. Kharel, "Quality of experience framework for cloud computing (QoC)," *IEEE Access*, vol. 6, pp. 64876–64890, 2018.

[28] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, 2011, pp. 301–314.

[29] M. Satyanarayanan, V. 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. 2009.

[30] M. Patel *et al.*, "Mobile edge computing introductory technical white paper," ETSI, Sophia Antipolis, France, White Paper, Sep. 2014.

[31] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "ETHANE: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.

[32] *OpenFlow-Switch-v1.5.1.PDF*. [Online]. Available: https://www.opennetworking.org/wpcontent/uploads/2014/10/openflow-switch-v1.5.1.pdf

[33] *ONOS—A New Carrier-Grade SDN Network Operating System Designed for High Availability, Performance, Scale-Out*. [Online]. Available: https://onosproject.org/

[34] *Opendaylight*. [Online]. Available: https://www.opendaylight.org/

[35] N. Foster *et al.*, "Frenetic: A network programming language," *ACM SIGPLAN Notices*, vol. 46, no. 9, pp. 279–291, 2011.

[36] C. Monsanto, N. Foster, R. Harrison, and D. Walker, "A compiler and run-time system for network programming languages," *ACM SIGPLAN Notices*, vol. 47, no. 1, pp. 217–230, Jan. 2012.

[37] *P4 Runtime*. [Online]. Available: https://p4.org/p4-runtime/

[38] *OpenConfig—Home*. [Online]. Available: http://www.openconfig.net/

[39] *GNMI Protocol*. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/iosxml/ios/prog/configuration/169/b_169_programmability_cg/gnmi_protocol.pdf

[40] *GitHub—Openconfig/Gnoi: gRPC Network Operations Interface (gNOI) Defines a set of gRPC-Based Microservices for Executing Operational Commands on Network Devices*. [Online]. Available: https://github.com/openconfig/gnoi

[41] *Stratum—Open Networking Foundation*. [Online]. Available: https://www.opennetworking.org/stratum/

[42] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," 2014, *arXiv:1405.5974*. [Online]. Available: https://arxiv.org/abs/1405.5974

[43] *What is Cisco SD-WAN Approach?—SDxCentral.com*. [Online]. Available: https://www.sdxcentral.com/sd-wan/definitions/cisco-sd-wan/

[44] 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.

[45] P. Berde *et al.*, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

[46] L. Peterson *et al.*, "XoS: An extensible c loud operating system," in *Proc. 2nd Int. Workshop Softw.-Defined Ecosyst.*, 2015, pp. 23–30.

[47] Wikipedia. *Optical Line Termination*. [Online]. Available: https://en.wikipedia.org/wiki/Optical_line_termination

[48] *Customer-Premises Equipment*. [Online]. Available: https://en.wikipedia.org/wiki/Customer-premises_equipment

[49] *Broadband Network Gateway*. [Online]. Available: https://it.wikipedia.org/wiki/Broadband_Network_Gateway

[50] R. F. Moyano, D. Fernández, L. Bellido, N. Merayo, J. C. Aguado, and I. de Miguel, "NFV-based QoS provision for software defined optical access and residential networks," in *Proc. IEEE/ACM 25th Int. Symp. Qual. Service (IWQoS)*, Jun. 2017, pp. 1–5.

[51] V. Petrov *et al.*, "Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 485–501, Mar. 2018.

[52] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[53] D. Rosário, M. Schimuneck, J. Camargo, J. Nobre, C. Both, J. Rochol, and M. Gerla, "Service migration from cloud to multi-tier fog nodes for multimedia dissemination with QoE support," *Sensors*, vol. 18, no. 2, p. 329, 2018.

[54] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility support for fog computing: An SDN approach," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 53–59, May 2018.

[55] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[56] R. Vilalta, A. Mayoral, D. Pubill, R. Casellas, R. Martínez, J. Serra, C. Verikoukis, and R. Muñoz, "End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node," in *Proc. Opt. Fiber Commun. Conf. Exhibit. (OFC)*, 2016, pp. 1–3.

[57] J. Serra, D. Pubill, A. Antonopoulos, and C. Verikoukis, "Smart HVAC control in IoT: Energy consumption minimization with user comfort constraints," *Sci. World J.*, vol. 2014, Jun. 2014, Art. no. 161874.

[58] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene, "An open framework to enable NetFATE (network functions at the edge)," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–6.

[59] S. Kaur, J. Singh, and N. S. Ghumman, "Network programmability using POX controller," in *Proc. IEEE Int. Conf. Commun., Comput. Syst. (ICCCS)*, 2014, pp. 134–138.

[60] H. Zhu and C. Huang, "IoT-B&B: Edge-based NFV for IoT devices with CPE crowdsourcing," *Wireless Commun. Mobile Comput.*, vol. 2018, Dec. 2018, Art. no. 3027269.

[61] S. Baidya, Y. Chen, and M. Levorato, "eBPF-based content and computation-aware communication for real-time edge computing," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 865–870.

[62] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[63] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 603–608.

[64] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge

computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.

[65] K. Phemius, J. Seddar, M. Bouet, H. Khalifé, and V. Conan, "Bringing SDN to the edge of tactical networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 1047–1052.

[66] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, and E. Benkhelifa, "SDMEC: Software defined system for mobile edge computing," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 88–93.

[67] R. Huo *et al.*, "Software defined networking, caching, and computing for green wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 185–193, Nov. 2016.

[68] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[69] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16. 2016, pp. 265–283.

[70] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3909–3914.

[71] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk, "Energy-aware fog and cloud interplay supported by wide area software defined networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[72] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani,, "Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[73] *Securing the Cloud Edge With SD-WAN*. [Online]. Available: https://blogs.cisco.com/enterprise/securing-the-cloud-edge-with-sd-wan

[74] *Cisco SD-WAN Security*. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1115-business-services-ckn.pdf

[75] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 513–521.

[76] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

[77] A. D. Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *Proc. 27th Biennial Symp. Commun. (QBSC)*, Jun. 2014, pp. 71–75.

[78] W. Ren, Y. Sun, H. Luo, and M. Guizani, "BLLC: A batch-level update mechanism with low cost for SDN-IoT networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1210–1222, Feb. 2018.

[79] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *Proc. 13th Annu. Medit. Hoc Netw. Workshop (MED-HOC-NET)*, Jun. 2014, pp. 103–110.

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

[81] J. Chen, H. Zhou, N. Zhang, P. Yang, L. Gui, and X. Shen, "Software defined Internet of vehicles: Architecture, challenges and solutions," *J. Commun. Inf. Netw.*, vol. 1, no. 1, pp. 14–26, Jun. 2016.

[82] H. Peng, Q. Ye, and X. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," 2018, *arXiv:1809.08966*. [Online]. Available: https://arxiv.org/abs/1809.08966

[83] J. Chen *et al.*, "Service-oriented dynamic connection management for software-defined Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2826–2837, Oct. 2017.

[84] W. Quan, K. Wang, Y. Liu, N. Cheng, H. Zhang, and X. S. Shen, "Software-defined collaborative offloading for heterogeneous vehicular networks," *Wireless Commun. Mobile Comput.*, vol. 2018, Apr. 2018, Art. no. 3810350.

[85] Y. Cao, J. Guo, and Y. Wu, "SDN enabled content distribution in vehicular networks," in *Proc. 4th Ed. Int. Conf. Innov. Comput. Technol. (INTECH)*, 2014, pp. 164–169.

[86] T. Han, X. Ge, L. Wang, K. S. Kwak, Y. Han, and X. Liu, "5G converged cell-less communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 44–50, Mar. 2017.

[87] M. Condoluci, F. Sardis, and T. Mahmoodi, "Softwarization and virtualization in 5G networks for smart cities," in *Internet of Things. IoT Infrastructures*. Rome, Italy: Springer, 2015, pp. 179–186.

[88] 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)*, Apr./May 2015, pp. 208–216.

[89] C. Song, J. Wu, X. Chen, L. Shi, and M. Liu, "Towards the partitioning problem in software-defined IoT networks for urban sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–9.

[90] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao, "Toward software defined smart home," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 116–122, May 2016.

[91] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home IoT devices," in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2015, pp. 163–167.

[92] L. Hu *et al.*, "Software defined healthcare networks," *IEEE Wireless Commun. Mag.*, vol. 22, no. 6, pp. 67–75, Jun. 2015.

[93] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined Internet of Things for smart urban sensing," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 55–63, Sep. 2015.

[94] A. El-Mougy, M. Ibnkahla, G. Hattab, and W. Ejaz, "Reconfigurable wireless networks," *Proc. IEEE*, vol. 103, no. 7, pp. 1125–1158, Jul. 2015.

[95] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the Internet-of-Things," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[96] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 44–53, Jul. 2013.

[97] M. Antonakakis *et al.*, "Understanding the Mirai botnet," in *Proc. 26th USENIX Secur. Symp. Secur.*, 2017, pp. 1093–1110.

[98] *An Elaborate Hack Shows How Much Damage IoT Bugs Can Do*. [Online]. Available: https://www.wired.com/story/elaborate-hack-shows-damage-iot-bugs-can-do/

[99] *Hackers Can Now Hitch a Ride on Car Computers*. [Online]. Available: https://www.latimes.com/business/autos/la-fi-hy-car-hacking-20150914-story.html

[100] *Hacked Terminals Capable of Causing Pacemaker Deaths*. [Online]. Available: https://www.itnews.com.au/news/hacked-terminals-capable-of-causing-pacemaker-mass-murder-319508

[101] *9 Baby Monitors Wide Open to Hacks That Expose Users Most Private Moments*. [Online]. Available: https://arstechnica.com/information-technology/2015/09/9-baby-monitors-wide-open-to-hacks-that-expose-users-most-private-moments/

[102] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 2177–2184.

[103] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the age of machine learning and software-defined networking," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4829–4842, Dec. 2018.

[104] K. S. Sahoo, B. Sahoo, and A. Panda, "A secured SDN framework for IoT," in *Proc. Int. Conf. Man Mach. Interfacing (MAMI)*, 2015, pp. 1–4.

[105] S. Chakrabarty and D. W. Engels, "A secure IoT architecture for smart cities," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2016, pp. 812–813.

[106] O. Flauzac, C. González, A. Hachani, and F. Nolot, "Sdn based architecture for IoT and improvement of the security," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2015, pp. 688–693.

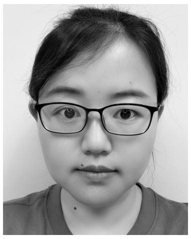
[107] *Open Edge Computing*. [Online]. Available: http://openedgecomputing.org/

[108] *Edge-Net.Org*. [Online]. Available: http://edge-net.org/

ABOUT THE AUTHORS

An Wang received the B.S. degree in computer science from Jilin University, Changchun, China, in 2012, and the Ph.D. degree in computer science from George Mason University, Fairfax, VA, USA, in 2018.

She is currently an Assistant Professor with the Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, OH, USA. Her current research interests include the areas of security for networked systems and network virtualization, focusing on software-defined networking (SDN) and cloud systems, and large-scale network attacks. She is also interested in the security and privacy issues in the Internet-of-Things (IoT) environment.

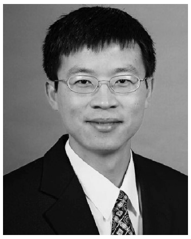


Zili Zha received the B.S. degree from the University of Science and Technology of China, Hefei, China, and the M.S. degree from the College of William and Mary, Williamsburg, VA, USA. She is currently working toward the Ph.D. degree with the Computer Science Department, George Mason University, Fairfax, VA, USA.

Her current research interests include data center traffic measurement, network programmability, and security.

Yang Guo received the B.S. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from the University of Massachusetts at Amherst, Amherst, MA, USA, in 2000.

He was with Bell Labs Research, Crawford Hill, NJ, USA, from 2010 to 2015, working on the monitoring and security of software-defined networking and cloud orchestration. From 2005 to 2010, he was a Principal Scientist with Technicolor Corporate Research (formerly Thomson), Princeton, NJ, USA, working on the Internet-wide content distribution and consumption. He is currently a Computer Scientist with the Advanced Networking Technologies Division, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. He has published many research papers in several renowned technical journals and conferences. He holds a number of U.S. patents. His current research interests include distributed systems and networking, with a focus on software-defined networking (SDN), cybersecurity, and AI and machine learning.



Songqing Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA.

He is currently a Professor of computer science with George Mason University, Fairfax, VA, USA. His current research interests include design, analysis, and implementation of algorithms and experimental systems in the distributed and networking environment, particularly in the areas of the Internet-content delivery systems, the Internet measurement and modeling, mobile and cloud computing, network and system security, and distributed systems.

Dr. Chen is a Senior Member of the Association for Computing Machinery (ACM). He was a recipient of the U.S. NSF CAREER Award and the AFOSR YIP Award. He serves as the Chair of the IEEE Technical Committee on the Internet (TCI). He also serves on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, IEEE Internet Computing, *ACM Transactions on Internet Technology*. He also serves in various capacities in conference organization committees, such as the General Co-Chair of the ACM/IEEE Symposium on Edge Computing (SEC) 2019 and the Finance Chair of the ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT) 2019.