Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review

Keshav Sood, *Student Member, IEEE*, Shui Yu, *Senior Member, IEEE*, and Yong Xiang, *Senior Member, IEEE*

***Abstract*—With the emergence of Internet-of-Things (IoT), there is now growing interest to simplify wireless network controls. This is a very challenging task, comprising information acquisition, information analysis, decision-making, and action implementation on large scale IoT networks. Resulting in research to explore the integration of software-defined networking (SDN) and IoT for a simpler, easier, and strain less network control. SDN is a promising novel paradigm shift which has the capability to enable a simplified and robust programmable wireless network serving an array of physical objects and applications. This paper starts with the emergence of SDN and then highlights recent significant developments in the wireless and optical domains with the aim of integrating SDN and IoT. Challenges in SDN and IoT integration are also discussed from both security and scalability perspectives.**

***Index Terms*—Internet-of-Things (IoT), software-defined networking (SDN), SDN use case, software-defined wireless networks (SDWNs).**

I. Introduction

MOBILE carrier networks are approaching a tipping point. Emerging mega trends in the information and communication technology (ICT) domain has reached a significant level of integrating the Internet into every object in a network. With the evolution of the Internet-of-Things (IoT), mobile networks will handle an influx in big data, massive network traffic, and new types of connected devices including industrial machines, thermostats, sensors, actuators, smart cars, wearables, and smart appliances. They will share the same network with PCs, tablets, and smartphones, which are already bandwidth sensitive. Presently, there are 9 billion connected devices and the number is expected to rise to 24 billion by 2020 [1]. With such significant involvement of connected devices, carriers are already experiencing complex control on elements and overloaded networks [2], [3]. If the networks are not prepared, this flood of "IOT," where the *things* are producers of traffic, not just consumers in the network, could leave the network paralyzed [1]. Furthermore, IoT devices are getting wirelessly connected to the Internet serving diversity of applications where no single wireless standard can adequately prevail. In this case, choosing the right wireless connectivity and to form a potential control on IoT wireless device are another challenging task, while the traditional network is insufficient to meet this challenge.

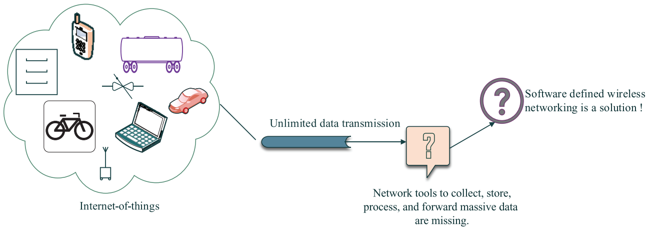

Internet-of-things
Unlimited data transmission
Network tools to collect, store, process, and forward massive data are missing.
Software defined wireless networking is a solution !

Fig. 1. IoT networks and SDN.

To address this need of the network, to support the onslaught of connected zillions of devices and to remain competitive, service providers are required to look into other alternatives such as software-defined networking (SDN) to increase their bandwidth and reinforce their networks [4]. SDN's wide acceptance from industry ensures SDNs ability to develop a tighter connection within the ecosystem of IoT that provides cyberspace to every object. We also emphasize that SDN and IoT are evolving parallel, intersecting, and perhaps dependent on each other. Fig. 1 reflects the need to simplify the network control mechanisms in IoT.

Significant benefits of integrating SDN and IoT as follows.

1) SDN has a potential to intelligently route traffic and use underutilized network resources. This will significantly enhance network's ability and therefore it will be much easier for networks to prepare for the data onslaught of IoT. This will eliminate bottlenecks to efficiently process the data generated by IoT without placing a large strain on the network, especially on Wi-Fi network.
2) SDN integration with IoT will simplify the information acquisition, information analysis, decision-making, and action implementation process.
3) The deployment of SDN in IoT will provide visibility of the network resources and management of access based on user, group, device, and application that eventually enables the ability to exchange data capacity between users and even devices.
4) Researchers are designing intelligent algorithms in SDN to build effective traffic pattern analyzer, which simplifies the tools of data collection from IoT devices. This facilitates the design of novel debugging tools. IoT networks will benefit with the integration of software-defined wireless networking (SDWN) technology to strengthen network's controlling ability.

Manuscript received March 30, 2015; revised August 27, 2015; accepted September 10, 2015. Date of publication September 28, 2015; date of current version July 27, 2016.

The authors are with the School of Information Technology, Deakin University, Melbourne, Vic. 3125, Australia (e-mail: ksood@deakin.edu.au; syu@deakin.edu.au; yxiang@deakin.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2015.2480421

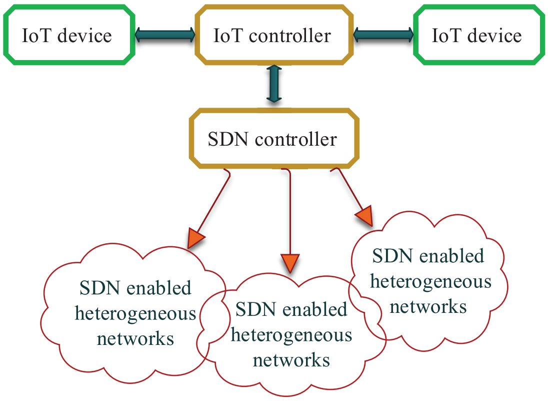

IoT device
IoT controller
IoT device
SDN controller
SDN enabled heterogeneous networks
SDN enabled heterogeneous networks
SDN enabled heterogeneous networks

Fig. 2. Typical architecture of integration of SDN and IoT, a high-level view.

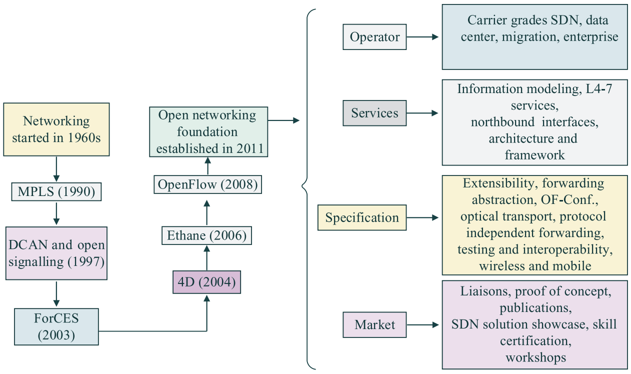

Networking started in 1960s
MPLS (1990)
DCAN and open signalling (1997)
ForCES (2003)
4D (2004)
Ethane (2006)
OpenFlow (2008)
Open networking foundation established in 2011
Operator
Services
Specification
Market
Carrier grades SDN, data center, migration, enterprise
Information modeling, L4-7 services, northbound interfaces, architecture and framework
Extensibility, forwarding abstraction, OF-Conf., optical transport, protocol independent forwarding, testing and interoperability, wireless and mobile
Liaisons, proof of concept, publications, SDN solution showcase, skill certification, workshops

Fig. 3. Early efforts of SDN and major technical communities in ONF.

5) With SDWN, IoT networks can become more agile and scalable based on demand. In Fig. 2, we provide a typical high-level view of integrated SDN and IoT architecture. SDN deployment in wireless segment is known as SDWN; therefore, the meaning of SDN and SDWN in this paper is same.

Efforts have been made to investigate SDWN in the context of infrastructure-based SDN-enabled Wi-Fi networks such as OpenRoad [5], Odin [6], OpenRadio [7], OpenRAN [8], SoftRAN [9], CellSDN [10], and SoftMoW [11]. The field of SDWN is in its infancy stage and there are still many important challenges to be addressed to control IoT network with a unified protocol. Therefore, in this paper, we aim to provide the recent developments of SDWN that can bring a lot of research opportunities in IoT.

This paper is organized as follows. The early efforts of SDN is presented in Section II and SDWN opportunities in IoT are described in detail in Section III. In Section IV, the SDWN open-research challenges are elaborated mainly from the prospectives of security and scalability. In Section V, we conclude this paper.

II. SDN: Early Efforts

Immigrant Paul Baran, a researcher working at Rand Corporation US in the 1960s proposed to transmit the voice signals of phone in the form of packet data that could travel autonomously through the network [12]. To further increase the packet forwarding intelligence for different reasons such as developing fine-grained traffic forwarding decision to save bandwidth and increase network performance, policy-based routing (PBR) methods were proposed [13]. At that stage, a new term *flow* was generated to describe particular set of traffic between two end points that receive the same forwarding treatment. PBR defines a set of criteria (commonly known as *match-action* criteria in SDN) that determines whether an incoming packet corresponds to a particular flow or not. This was a centralized approach of programming forward rules that has provided further under piping for SDN technology. In this regard, we can incorporate PBR at the ground level of SDN. Fig. 3 illustrates the early efforts of SDN and major open networking foundation (ONF)'s technical communities (operator, services, specifications, and market) responsible for various tasks.

From its birth, SDN has been based on the notion of constructing forwarding tables-defining actions to take on flow rather than having forwarding tables merely map destination address to output port [14]. Over time networking functions moved from software to hardware such as application-specific integrated circuits (AISCs), field programmable gate array (FPGA), and ternary content addressable memory (TCAM). As time passed, networking of devices has become increasingly complex. This is due in part to the existing independent and autonomous design of devices that make it necessary for so much intelligence being placed inside each device. This made the functionality in some ways very simple but made the device more complicated because of the difficult handshake and tradeoffs between handling packets in hardware versus software [10]. With time, researchers attempted to move control off the device placed into centralized controller that is having a full network view and the ability to make optimal forwarding and routing decisions [14]. Control software means the intelligence that determines optimal paths and responds to outages and new networking demands. Forwarding responsibilities implemented in hardware tables, filtering based on access control lists (ACLs), and traffic prioritization are enforced locally on device remain on the device [13]. The forwarding table on hardware device is available to be programmed by external software controller. Above the controller, the network application runs, implementing higher level functions, involving to make decisions to best manage the traffic and network.

There is a steady progression of solutions and ideas around advancing networking technology prior to OpenFlow. The early efforts include multiprotocol label switching (MPLS) (1990) to separate control software, establishing semi-static forwarding paths for flows in traditional routers, devolved control of ATM network (DCAN) to separate control and forwarding plane in ATM switches (1997) and open signaling (1997) began with ATM switches [13]. Forward and control element separation (ForCES) (2003), 4D named after four plane decisions (2004), and ethane (2006) are all known as precursors of SDN [13]. Although all these solutions adequately and automatically reconfigure the edge network, the static and manually configured core of the network remains the same [15]. The long-awaited solution of this problem is now available in the form of OpenFlow (2008) [16].

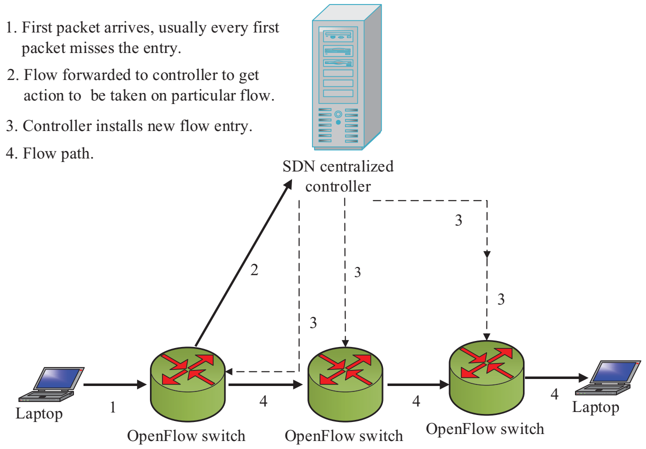

1. First packet arrives, usually every first packet misses the entry.
2. Flow forwarded to controller to get action to be taken on particular flow.
3. Controller installs new flow entry.
4. Flow path.
SDN centralized controller
1
2
3
3
3
3
4
4
4
Laptop
Laptop
OpenFlow switch
OpenFlow switch
OpenFlow switch

Fig. 4. SDN OpenFlow switching.

A. OpenFlow and ONF

OpenFlow was developed by researchers to begin experiments and innovate with new protocols every day. The specifications of OpenFlow encourage vendors to implement and enable OpenFlow in switches and other products. OpenFlow protocol delineates to use it between the controller and switch, hence a unified control protocol comes into existence. The basic switching operation of OpenFlow, as shown in Fig. 4, is a switch that evaluates every incoming flow independently, finds a matching flow against it, and performs the associated action. If no match is found, the switch forwards packet to controller for getting instructions on how to deal with packet. The SDN controller populates the switch with flow table entries. Typically, controller updates switches with new flow entries as new flow patterns are received. Wild card rules are also accepted. This technology is known as *SDN*.

Before 2011, the OpenFlow standards and protocol versions had been designed by Stanford University. ONF [17], a new autonomous body was formed in 2011 by companies including Google, Facebook, Cisco system, and Microsoft to design the standards for OpenFlow. Various research groups are working toward designing SDN OpenFlow specification, configuration, management protocols, and so on. The extensibility working group focuses on SDN deployment in wireless (IEEE 802. XX) and telecom sector. It is important to note that ONF has introduced 64 OpenFlow products to market, and more than 20 members have demonstrated interoperability of OpenFlow standards till date. In Section II-B, we will shed some light into various open source tools simulating the ideas of SDN.

B. Test Bed and Troubleshooting Tools

Table I highlights the most common SDN simulation tools used by researchers nowadays. But till 2010, choosing an appropriate prototyping tool to simulate ideas had been a challenge because real test bed was expensive and out of reach from most scientists.

Lantz *et al.* designed "Mininet" to support collaborative network research [18], which is a flexible, deployable, interactive, scalable, realistic, and at the most shareable software simulation tool. Virtual machines (VMs) can be created to test desired network behavior with OpenFlow. This is a python-based open-source tool. At the same time, it is very critical in SDN to detect and debug the failures in large-scale network. Scott *et al.* argued that concise and specific policies and more sophisticated tools are required for the testing of SDN large-scale network [19]. Because SDN software stacks itself in a complex-distributed system, the working of SDN is in a challenging, synchronous, and failure prone environment. They developed a tool called W3 to troubleshoot bugs in SDN control software. W3 stands for "What network problem exist?" "Where the problem arises first in the software?" and "When the triggering event happened?" Corresponding checking and simulation-based causal interference are the two approaches designed in W3. This tool successfully tested on various SDN platforms but still needs many rounds of improvement to fully reap its benefits in large-scale enterprise domain. Reitblatt *et al.* [20] developed a declarative language "FatTire" for developers to express fault-tolerance requirements and provide compiler that targets SDN fast-failover mechanism. Further, fs-sdn has been recently designed by Gupta *et al.* [21] that offers simulation at large scale in comparison to Mininet.

TABLE I
SDN Simulation Tools

Tool/reference	Methodolgy
Mininet [18]	Using lightweight OS containers to emulating hosts and switches in a network
W3 [19]	A gdb-like debugger for OpenFlow based networks
FatTire [20]	Provides complier target fault tolerance requirement
fs-sdn [21]	Enabling direct use of OpenFlow controller components

III. SDN Opportunities in IoT

In this section, only key research efforts of SDWN over the period between 2008 and 2014 are discussed.

A. SDN in Wireless Networks

Researchers in [16] developed OpenFlow protocol to run experiments in a uniform way in real-time running network, aiming to develop a new switch feature that can potentially extend programmability into campus network. Further, Yap *et al.* [22] have proved the conceptual idea of OpenFlow by implementing a test bed with OpenFlow at Stanford University Campus that allows heterogeneous network experiments to be concurrently conducted in production environment so that multiple networks will act as a single network. In [5], OpenRoad is embedded into mobile/cellular networks with a vision to attract virtual service providers to independently allow seamless handover between different wireless technologies. Also, Yap *et al.* [5] extended SDN to Wi-Max services and described the brief idea of SDN and virtualization in carrier domain. A major breakthrough in SDN came into existence when Jain *et al.* [23] from Google presented a design, implementation, and evaluation of the B4 project, a private wide area network (WAN)-connecting Google's data centers across globe. Jain *et al.* [23] described the 3 years of B4 experience in production domain, lessons learnt, and significant future research areas in their tutorial paper.

Suresh *et al.* developed Odin to introduce programmability in wireless local area network (WLAN) networks [6]. With Odin, the association of client with access point (AP) can be made by SDN controller. Odin introduces light weight APs (LVAPs) that runs over controller to make decision of association and disassociation of client with AP while ensuring seamless handover. The limitations of this work are as follows: 1) the association and disassociation processes do not count the load on APs and 2) Odin runs over single controller; hence, it will be difficult to prioritize the multiple applications running over single controller. In [24], Odin's work was extended with the proposition of a novel load-aware hand-off scheduling algorithm running over SDN-centralized controller for SDN-based WLAN networks. In [25], the impact of OpenFlow SDN over wireless networks was evaluated, particularly in terms of end-to-end delay, throughput, and jitter.

Costanzo *et al.* [26] analyzed SDN in IEEE 802.15.4 networks (low rate-wireless personal area networks, LR-WPANs). They elaborated the opportunities of SDN and requirements to implement SDN with different scenarios in this network domain, i.e., IoT domain. Authors argued that requirements-alike SDWN must support duty cycle, provides flexible definitions of rules, and effectively tracks the node mobility in network data aggregation where sensor nodes transmit and receive tremendous amount of data in IoT sector. Dely *et al.* [27] showed an SDN-based system architecture in WLAN tested for fast handover to improve streaming video. The controller and streaming server are connected to OpenFlow virtual switch (OVS) which is further connected to APs. Controller updates forwarding table of OVS to route traffic from streaming server to the station to connect several APs simultaneously. At this point, SDN appears as a viable alternative integrated architecture that facilitates the possibilities of creating novel IoT SDN integrated services, architectures, data-driven protocols, and more efficient applications to cover the actual requirements of programmable network.

In [28], a similar approach to [27], i.e., media independent handover mechanism using OpenFlow, was investigated. In this work, the mobile device initiates the signaling process to handover to another service point. The mobile device sends signals to SDN central controller which then sends information about the best available controller and the best available service point, back to mobile device for association and handover. From this work, it seems that novel load-aware algorithms can be proposed that may sense the load on wireless sensors. This may facilitates rerouting of the traffic to less-loaded node in coverage zone. A complete IoT network can be divided into zones and controlled by logically centralized controller. At this point, in order to maintain service quality, it is essential to analyze the performance of distributed SDN-enabled IoT networks. Such an effort, to deploy distributed architectures, was evaluated in [29]. Authors significantly analyzed quality of service (QoS) for streaming applications. Extensive simulations were done to prove that distributed architectures have much more advantages than centralized SDN. We emphasize that the path inflation factor must be investigated before deploying SDN IoT-distributed topology to avoid latency.

Rukert *et al.* [30] pointed out that current broadband network architecture uses tunneling concept to subscribe traffic through a single aggregation point irrespective to different types of services. This implies huge bandwidth requirements and high end-to-end latency. A proof-of-concept approach was discussed to show the feasibility of the proposed novel SDN-based flexible traffic management architecture. Feng *et al.* [31] proposed a price-based joint allocation model to fairly allocate the bandwidth and flow table space. In its analysis, the maximum mean forwarding rate and minimum mean delay time are calculated. The flow volume at each port is calculated and according to that volume of flow, bandwidth is allocated to flow. Authors proved that "the mean delay of SDN network is minimum if the mean delay of each OpenFlow switch is minimum" [31].

These research efforts are also highlighted in Table II. All these efforts provide opportunities to design novel architectures in IoT control by SDN. The controller senses the user application type and allocates the node or switch according to bandwidth requirement. The SDN controller has full network wide view and usage of each element/node, and thus it can easily and efficiently distribute the bulk of data to different nodes. SDN also provides an opportunity to virtualize the IoT networks, as demonstrated by Lee *et al.* [32]. They designed meSDN architecture to demonstrate WLAN virtualization with SDN.

B. SDN-Enabled Hybrid Architecture

SDN can also bring new opportunities in IoT by designing hybrid network architecture. It has the potential to control circuit and packet switching by a unified control protocol. Gudla *et al.* presented a unified control architecture with OpenFlow to dynamically control packet and circuit switching networks [33]. Latency and link-up time is reduced in comparison to traditional networking methodologies. The OpenFlow switch functionality was implemented in NetFPGA. An independent testing of OpenFlow in circuit and packet switching was also conducted. Cerroni *et al.* [34] designed the OpenFlow-enabled hybrid network architecture. They extended the OpenFlow table-matching rule entry by introducing two additional tuples, i.e., channel and transport class for wavelength/timeslot and guaranteed circuit/packet switching, respectively. Channegowda *et al.* [35] have demonstrated the improved path set-up times and control stability with SDN in optical transport technologies. From these research efforts, one can infer that novel hybrid architecture in IoT will have lesser effect on convergence time that is affected by the limited information of the node to recalculate the route.

C. Other Relevant Research Work of SDWN

Bansal *et al.* [17] presented a key conceptual novel design for programmable wireless data plane, OpenRadio, to provide modular and declarative programming interface within wireless

TABLE II
OVERVIEW OF KEY SDWN RESEARCH BETWEEN 2008 AND 2014

Year	Project theme/reference	Overview	Technological domain wireless/telecom: W, optical: O, hybrid: H
2008	Openflow: enabling innovation in campus networks [16]	Proposed a new feature in the switch with OpenFlow protocol	H
2009	The stanford OpenRoad deployment [22]	First test bed that allows heterogeneous network experiment to be concurrently conducted in production environment so that multiple networks will act as a single network	W
2010	OpenRoad: empowering research in mobile networks [5]	Extended SDN in mobile networks	W
2010	Experimental demonstration of OpenFlow control of packet and circuit switching [33]	Proposed unified control architecture in circuit and packet switching	H
2010	A network in a laptop [18]	Designed simulation software Mininet for SDN application testing	H
2011	OPF [17]	A new body to design and standardise SDN protocols	–
2012	W3, a troubleshooting tool in SDN [19]	A troubleshooting tool to detect bugs in SDN networks	W
2012	OpenRadio: a programmable wireless data plane [7]	Base stations can be remotely programmable to enable operators and vendors to upgrade and optimise the network more easily	W
2012	Towards programmable enterprise WLANs with ODIN [6]	Introduce light weight access points (LVAP) that runs over controller to make decision of association and disassociation of client with AP while ensuring seamless handover	W
2012	SDWNs: unbridling SDNs [26]	Elaborate the opportunities of SDN and requirements to implement SDN with different scenarios in IEEE 802.15.4 network (LR-WPANs)	W
2012	Towards software defined cellular networks [36]	Proposed extensions to controller platform, switches, and base stations	W
2012	Towards software defined MB networking [37]	A design of a software defined MB networking framework capable of supporting future scenario	W
2012	On scalability of software defined networks [44]	Deconstructed the scalability concerns in SDN networking and argued that they are not unique to SDN	H
2012	A SDN approach for handover management with real-time videos in WLANs [27]	Demonstrating OpenFlow for improving streaming video	W
2012	Empowering SDWN through media independent handover management [28]	Media independent handover mechanism was demonstrated using OpenFlow	W
2013	Design and test of a software defined hybrid network architecture [34]	Extension of the OpenFlow table matching rule entry by introducing two more tuples, i.e., channel and transport class for wavelength/timeslot and guaranteed circuit/packet switching, respectively	W
2013	SDN optical network technology and infrastructure [35]	Demonstrated improved path set up times and control stability when SDN is implemented directly in optical transport technologies	O
2013	SDN networks for telecom operators: architecture and applications [38]	The aggregate and core layer of wireless backhaul can also be virtualized depending on the SDN controller's ability	W
2013	OpenRAN [8]	Designed open and flexible network management system where user is free to join the strongest network nearby	W
2013	SoftRAN [9]	Abstracts all base stations in a local geographical area as a virtual big base station comprised controller and radio appliances	W
2013	Applying SDN to telecom domain [39]	Enables data migration from native layer to overlay layers and thus requires encapsulation and de-encapsulation	W
2013	FatTire [20]	A declarative language for developers to express fault tolerance requirements and provide complier that targets SDN fast-failover mechanism	H
2014	Modelling and evaluation of SDN scalability [51]	Mathematical tools for evaluating the scalability of SDN network	W, O, and wired
2014	Traffic management in broadband access networks [30]	A proof-of-concept approach that can show the feasibility of proposed novel SDN-based flexible traffic management architecture	W
2014	Fair network resources allocation and scheduling [31]	The flow volume at each port is calculated for bandwidth calculation	W
2014	Distributed QoS architecture for multimedia streaming with SDN [29]	Proved that distributed architectures have much more advantage than centralized SDN	W
2014	Extension of ODIN [24]	Load aware hand off scheduling algorithm for SDN-based WLAN network is proposed	W
2014	Performance analysis of SDN over wireless networks [25]	Evaluated the impact of OpenFlow SDN over wireless networks particularly in terms of end-to-end delay, throughput, and jitter	W
2014	SDN mobile extension [32]	An meSDN architecture that demonstrates that SDN control on mobile client enables WLAN virtualization and application aware QoS improves power efficiency	W
2014	Application aware data plane processing [50]	Implemented apps in the OVS switches for stateful switch actions	W, O, and wired

stack. Network operators are always in need to dynamically adjust spectrum and power at base stations according to traffic requirements. According to the proposed design, base stations can be remotely programmable to enable operators and vendors to upgrade and optimize the network more easily. Operators are able to define and set a protocol based on matching subsets of traffic streams and then can specify actions on them. Bansal *et al.* [17] defended the feasibility of this idea by arguing that physical layer (PHY) and media access control (MAC) layers are shared across different protocols (different versions of long term evaluation (LTE), 4 G, 3 G, and Wi-Fi). In [36], it is argued that today's cellular networks do not have fine-grained control over routing and altering traffic to direct destination. Middle boxes (MBs, i.e., network appliances deployed by different vendors) are also problematic causing serious performance issues in carrier domain. Li *et al.* [36] proposed extensions to controller platform, switches, and base stations. They have made some changes in carrier domain to implement SDN architecture in SDWN. The argument made in [37] is that current MBs (such as firewall, load balancer, and intrusion prevention) are clumsy and unsuitable to handle future networks. A software-defined MBs networking framework capable of supporting future scenarios was proposed in [37].

Currently, we are forced to associate with network even if the network performs quite poorly and even if we have other more appropriate networks around. This is a closed and non-flexible approach. In [8], a conceptual overview was presented for the design of open and flexible network management system known as open random access network (OpenRAN). The term Open indicates that users are free to join the strongest network nearby. This term here does not indicate the open from OpenFlow, rather the idea is built on OpenFlow SDN technology. Further, Gudipati *et al.* [9] proposed SoftRAN which was the extension of OpenRoad. Authors in this paper have provided a wireless programming interface in random access network (RAN) network called SoftRAN that abstracts all base stations in a local geographical area as a virtual big base station comprised controller and radio appliances.

SDN research in telecom sector is also gaining momentum. Wang *et al.* [38] pointed out that the aggregation and core layer of wireless backhaul can be virtualized depending on the SDN controller's ability, whereas Hampel *et al.* [39] argue to enable data migration from native layer to overlay layers, thus requiring encapsulation and de-encapsulation.

Researchers also argue that conventional network management algorithms (especially multimedia traffic management algorithms) are inappropriate for security-aware multimedia applications [41]. For a security-critical multimedia service architecture in the IoT context, there are various other challenges including traffic classification and analysis for various multimedia applications streaming over IoT, developing novel architectures for media-aware traffic security, designing and evaluating the proposed security-critical traffic management scheme, etc. To alleviate these challenges, the authors of [42] have recently provided effective general media-aware security architecture that jointly considers the security services and multimedia traffic characteristics in the IoT context.

Fig. 5 shows the web search popularity, as measured by the Google search trends during the last ten years for the terms IoT, wireless sensor networks, and ubiquitous computing. This evidence clearly supports our vision and inspires the need of SDN and IoT integration. This trend motivates researchers to deploy novel and simplified network control mechanism in IoT. Recent research presented above inspires scientists to innovate novel SDN-enabled architectures in IoT domain and eventually beats the challenges caused by the rapid growth of *things* in wireless cyberspace.

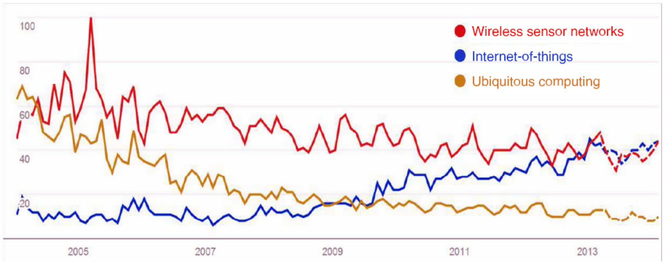


Fig. 5. Google search trends since 2004 for terms IoT, wireless sensor networks, and ubiquitous computing [1].

D. Testing Novel OpenFlow-Enabled IoT Products and Design

The rapid growth of devices and online services for information transaction over the network consolidated the concept of IoT. The rigidity of traditional architecture is inefficient in this concept, suggesting rethinking new ways to use the infrastructure and technology. SDN provides an alternative to the current problems of traditional networks. It gives potential opportunities to allow administrator to have a global view of the network, as well as to control the network according to the need of individual organization and its users. Organizations are looking for deployment-ready solutions or novel IoT products running with IPv6 for their current networking environment. They are also looking for future compatibility built on the OpenFlow specification. ONF is a user-driven organization dedicated to the promotion and adoption of SDN through open standard development. The wireless and mobile project in ONF collects use cases and determines architectural and protocol requirements. The aim of the project is to extend ONF-based technologies to carrier networks such as backhaul network, and cellular evolved packet core (EPC). Further, they aim to provide a unified access and management across enterprise wireless and fixed networks including IoT domain. The ONF OpenFlow conformance certification is the highest level of assurance available in the market by ONF today to validate product conformance. For this, ONF has approved six laboratories till 2014 to test the novel design applications and products, etc. [17]. This highly encourages researchers to design and test the novel SDN-enabled IoT products and ideas on live and real platform.

IV. Open-Research Challenges

A. Security of SDN and IoT Network

The reason that security is always a major issue in cyberspace is that measures are often considered only *after* launching a new technology. The IoT network, i.e., a network of physical objects is more sensitive to security, and contains embedded system to

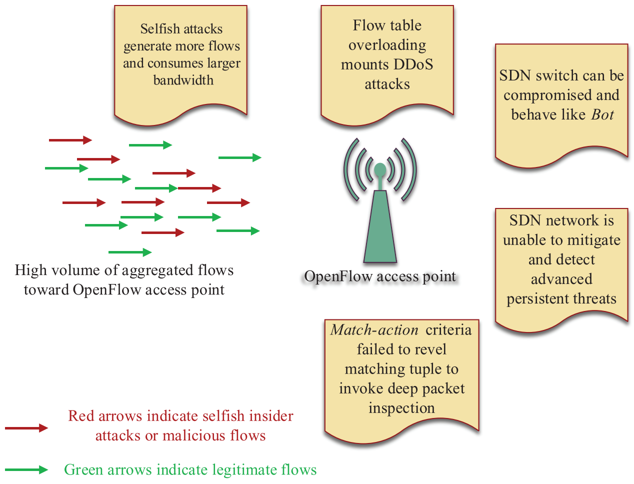

Selfish attacks generate more flows and consumes larger bandwidth
Flow table overloading mounts DDoS attacks
SDN switch can be compromised and behave like *Bot*
High volume of aggregated flows toward OpenFlow access point
OpenFlow access point
SDN network is unable to mitigate and detect advanced persistent threats
Match-action criteria failed to revel matching tuple to invoke deep packet inspection
Red arrows indicate selfish insider attacks or malicious flows
Green arrows indicate legitimate flows

Fig. 6. Security concerns in SDN network.

communicate machine to machine. In IoT, there are potential risks to network such as advanced encryption standard (AES) public/private key exchange methods, protecting attached transmission control protocol (TCP)/Internet protocol (IP) networks from intrusion through your device, and protecting preshared keys from reverse engineering through an microcontroller unit (MCU) debugger.[1] Fig. 6 represents basic security concerns in SDN domain.

SDN and IoT integration will doubtlessly simplify the network control using common protocol in every technological domain, but SDN also poses some risks. For example, in SDN, logically centralized controller controls the switch which evaluates every incoming packet based on match-action criteria, as shown in Fig. 4. In order to temporarily store match-action rules to take decision on incoming packet or flow, each switch possesses a flow table. Unfortunately, these flow tables are implemented using expensive and power-hungry TCAM. As a result, the flow table size is limited. Usually, the flow table size cannot scale beyond few hundreds entries or rules. Therefore, SDN switch can only handle limited number of flows per time [40]. This limited flow table size is a potential weakness of SDN and is vulnerable to attackers. High volume of traffic can very easily consume the table capacity and thus the switch is overloaded. Continuing high volume of traffic (flow) may make switch disabled or knock it down. As a result, the later arriving packets may all be dropped, failing to be forwarded. Therefore, attackers can easily knock down the switch and thus disable the network services by mounting distributed denial of service (DDoS) attack.

To hamper network security to extreme level, the OpenFlow switch can be compromised and can serve as a *bot* (a compromised host used to perform malicious task). In real world, attackers can use compromised hosts, such as a botnet, to start DDoS attack. If the switch is compromised by botmaster, then effective mechanisms viable in network to detect the advanced persistent threats (APTs) are missing in the current SDN technology. In this case, an SDN switch may become a bot and serves as a bridge to disrupt the network. Similar things can happen in IoT network which is large in scale and composed of millions of devices. Furthermore, enhancing the controller's intelligence software may increase controller vulnerability to hackers and attack surfaces. If attackers have access to the controller, they can damage every aspect of the network and eventually knock down the whole network.

TABLE III
DECONSTRUCTING SDN SECURITY ISSUES

Security challenges
To prevent an authenticated application from being hacked or authenticate an application's access to the control plane
How to mitigate the DDoS threat to prevent server, controller, or switch to overload?
To test novel designs in order to evaluate clients not nodes, and evaluate payload not packet
SDN security advantages
SDN allows for the decoupling of the control plane from the data plane. If hackers were to reach the data plane in an SDN ecosystem, they would be unable to use the data because the controls would no longer be embedded
The distributed protocols, which are more resilient and harder to attack because they are not concentrated
SDN is capable of automatically quarantining an endpoint or network that has been infected with malware
Security application in SDN [43]
DDoS mitigation applications (defencef low, defence pro, APSolute vision), aiming to mitigate L2, L3, and L7 attacks
Security enhanced-floodlight (providing integration of a security mediation kernel into the BigSwitch floodlight OpenFlow controller) to avoid controller hacking. Additionally it provides role-based authorization and strong security constraints enforcement
SDN security actuator , a middle-ware abstraction service that provides flexibility to integrate legacy INFOSEC security products and technology into an OpenFlow network stack. It also enables security services to communicate high level threat response directives, which are further translated into stateful OpenFlow flow rule insertions to be sent to SE-floodlight
OF-BotHunter, a sample OpenFlow security application that provides interfaces with OpenFlow network stack via the SDN security actuator. worm propagation, an application to detect malware for mobile devices in SDN Stack
BlueCat DNS director, an application by Hewlett-Packard to deliver network-driven enforcement of DNS policies which allow security infrastructures to gain complete visibility and control with IP applications
Ecode evolve, an SDN orchestrator to facilitate dynamic service provisioning with built-in QoS and DDoS mitigation
F5 BIG DDoS umbrella, a network application that allows network customers to DNS and SSL DDoS protection at the network edge that is closer to the attacker
GuardiCore defense suite for software-defined data centers, detecting and mitigating advanced persistent threats, malware propagation, and insider attacks *KEMP* adaptive load balancer application, providing end-to-end visibility of network path for optimal routing of applications *Real status hyperglance*, providing simplified context aware hybrid cloud and SDN management

SDN security risks come out because of the absence of integration with existing security technologies and SDN's inability to poke around every packet. Therefore, it is essential that a packet has to undergo a deep inspection for risk assessment before routing to certain levels.

SDN security requires to support the authentication and authorization classes of the network administrators at every plane, but it may prevent the access to flow management policies. Thus, SDN must construct novel security mechanism, different from the traditional ones. Although it is still at an early stage in the context of the IoT and SDN, it is clear that change is afoot. In SDN, research is gaining momentum to secure both the control and data plane [15]. Table III highlights the research progress that deconstructs various SDN security issues.

We argue that SDN integrity with IoT will simplify the information process, information acquisition, information analysis, decision-making, and action implementation. This will ease the IoT network management. Secure flow automation needs

[1] [Online]. Available: http://www.link-labs.com/internet-of-things-security challenges

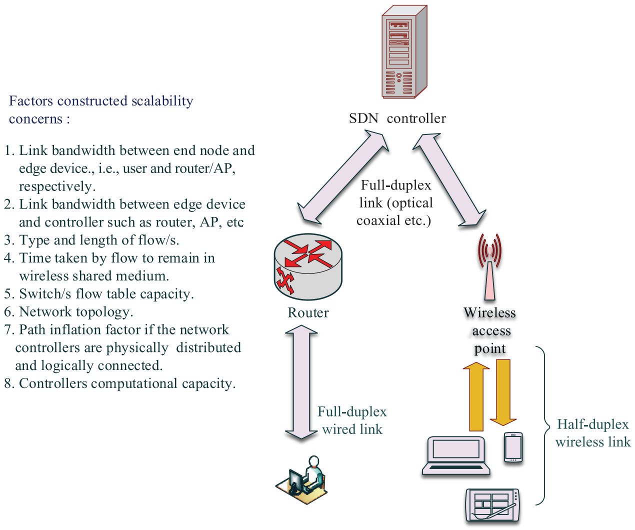

Factors constructed scalability concerns :
1. Link bandwidth between end node and edge device., i.e., user and router/AP, respectively.
2. Link bandwidth between edge device and controller such as router, AP, etc
3. Type and length of flow/s.
4. Time taken by flow to remain in wireless shared medium.
5. Switch/s flow table capacity.
6. Network topology.
7. Path inflation factor if the network controllers are physically distributed and logically connected.
8. Controllers computational capacity.
SDN controller
Full-duplex link (optical coaxial etc.)
Router
Wireless access point
Full-duplex wired link
Half-duplex wireless link

Fig. 7. Factors constructed scalability concerns in SDN network.

central management of data forwarding which is very complex without SDN. As the IoT network diameter grows, much more simplified and secure mechanisms are required. We also argue that as organizations depend more on machine-generated data for real-time business processes, it is essential to ensure the trust of data. Strong detection mechanisms to detect rogue devices trying to interact with the IoT infrastructure both from device and network vectors must be deployed. As the IoT connects many devices together, it provides many decentralized entry points for malwares. Cheap devices or things in physically compromised locales are fragile to tampering, whereas SDN's centralized approach is significantly better than traditional networks. Moreover, SDN has full potential to globally view the traffic patterns, mobility of nodes, and change in traffic volumes. Therefore, security policies can be easily implemented in SDN-enabled IoT networks.

To conclude, based on SDN's potential to classify/slice traffic and to attain a full global view on it, we emphasize that SDN integration with IoT can bring novel and simplified ways to deal with such critical issues.

B. Scalability of SDN and IoT Networks

Yeganeh *et al.* [44] deconstructed the scalability concerns in SDN networking and argued that they are not unique to SDN, and can be overcome by deploying multiple controllers and switches. But based on our literature review, we highlight various factors that construct scalability concerns, shown in Fig. 7. The reasons for these concerns are described as follows.

1) As shown in Fig. 4, flow handling may generate additional network overhead because any new flow entry is treated like an alien (missed entry in flow table), and packets are then forwarded to controller for designing new flow entry. This forces the controller to install new rules into the flow table to process the flows. The controller takes additional time, because of its limited processing capability, to generate new flow entry and then populates that to switch. This whole process may add extra latency in end-to-end flow transaction.

2) We argue that the bandwidth between the switch and controller is an additional significant resource which cannot be ignored. It is observed that scientists consider the link bandwidth, the computational power or flow handling capacity of controller, and the switch's capability to handle the flow as the three main resources in SDN networking. We emphasize that the scalability concerns are also dependent on the time consumed by flow length propagated in air that further depends on the type of traffic. Mobile traffic (mostly user datagram protocol (UDP)) is inelastic and insensitive to bandwidth such as browsing *youtube* videos, online gaming; if the flow size is large then it will affect the other legitimate users to join the network. Larger the flow length, more is the time taken by the particular flow in air that eventually increases the collision rate, delay, and jitter in WLAN networks. Therefore, it is necessary to taken into account the type and length of flow while constructing scalability concerns.

IoT networks are more sensitive to bandwidth and thus the scalability concern must be taken care more strongly in this territory with SDN. While analyzing the scalability of IoT network, it is important to consider the network topology, average service rate of controller, average arrival rate of initiation requests, the path inflation factor that depends on the distance of the distributed controllers, channel capacity, and flow size.

Another research challenge for SDN and IoT under scalability theme is the optimum controller placement concern that influences every aspect of the decoupled plane. For example, high propagation delay in WLAN networks limits availability and convergence time. This has practical implications on software design, affecting whether controllers can respond to events in real time or must push forwarding actions to forwarding elements in advance. This can create another issue of controller placement in the designed network topology and the number of controllers for processing flows. Random placement for a small k-value in the k-median problem, a clustering analysis algorithm, will result in an average latency between $1.4\times$ and $1.7\times$ greater than that of the optimal placement [46]. A reliability-aware controller placement problem has been proposed in [47]. Heller *et al.* [46] proposed a latency-aware controller placement problem with the objective to provide an initial analysis for further study of the formulation of fundamental design problems.

We have observed that in SDN the research is continuously growing to overcome scalability issues. Table IV highlights the recent research to enhance scalability of SDN oriented Internet architectures. We argue that without SDN it will be difficult for IoT network to effectively process the real-time data because the problem lies in the nature of the IoT itself. It connects remote nodes and provides a data stream between nodes and decentralized management systems. The amount and type of big data differ than other sets of data comes from social media. Some features of the IoT data can be summarized as follows.

1) The IoT data tend to arrive as a steady stream and at a constant pace, although it could arrive in batches like test logs that can be processed and passed on straight away.

2) The data come in very large quantities and accumulate very fast.

TABLE IV
RECENT RESEARCH DIRECTIONS TO ENHANCE SDN SCALABILITY

Simplifying abstraction
Beehive [44]: aiming to build a programming abstraction to enable the SDN platform to automatically infer how applications maintain their state and how they depend on one another. This simplifies the implementation process of distributed applications
Controller design [15]
Distributed controllers such as flat structure multiple controllers, e.g., ONIX Recursive controller design, e.g., Xbar Hierarchical controller design, e.g., Kandoo
Distributed architectures [15]
DIFANE: providing scalable solution keeping all traffic in the data plane DevoFlow: decreases number of interactions between switch and controller DISCO: providing the intercommunication between E2E network services
Research on future internet oriented SDN architecture [45]
ALICANTE: a media ecosystem deployment through ubiquitous application aware network environment with the aim of providing flexible access to multimedia services

TABLE V
INDUSTRY INITIATIVES TO INTEGRATE SDN AND IoT

SDN, IoT *apps*
Intel is creating an SDN environment to run IoT applications, developing open source code for packet processing and for aspects of the orchestrator to allow it to understand server capabilities
Meru network manager for managing, configuring and monitoring a Meru wireless LAN. This will manage connection of mobile device on-boarding and guest access. *Meru spectrum manager* for identifying sources of wireless interference. *Meru services assurance manager*, for performing predictive ”health checks” on network applications
In May 2015, Huawei unveiled the world’s first SDN-ased IoT solution named *AgileIoT*. This consists of Agile IoT getaways, operating system, and an Agile controller
IoT challenges that SDN can solve
SDN automation meets IoT
SDN can develop a scalable distributed system to manage the flow of events (data flows from zillion of IoT devices)
SDN can effectively provide frictionless integration of new IoT components (designed and deployed by multiple vendors) into the distributed systems and the various data flows in a scalable manner. What primitives a network need to support the variety of protocols in IoT? and the solution is OpenFlow

3) The real value of data can sometimes only be uncovered using effective analytics.
4) The data are rarely used for production purposes.
5) It can be deleted very quickly, unless there is a need for compliance reasons.

From the above concerns, we observe that the traditional storage architecture, processing, and management software will treat IoT big data in the same manner as they treat other unstructured data. Therefore, we conclude that although SDN has some research issues about scalability, SDN architecture, which is different and significantly better than traditional architecture, can effectively handle IoT big data streams.

C. Deep Packet Inspection

Another limitation of OpenFlow is that the deep packet inspection (DPI) is unfortunately not supported in standard OpenFlow [49] because currently defined match fields to evaluate packet are limited to the packet header only.

The common usage of DPIs includes lawful intercept, targeted advertising, and copyright enforcement. In security, it is sometimes essential to thoroughly investigate the data part of the packet as it passes an inspection point. Searching for protocol noncompliance or defined criteria to decide whether the packet may pass or if it needs to be routed to a different destination, is critical and essential in security. Also, to distinguish the suspicious flows for the purpose of collecting statistical information needs DPI. Therefore, more advanced firewalls may need to sometimes examine and act on fields that are not available to OpenFlow match and action. In order to set up the flow rule to invoke DPI for security reasons, one question to be answered is, “which matching tuple will detect the packet that might need special treatment?” Besides, whether the SDN edge switch has enough processing power on board (for DPI) rather than sending flows to controller for risk assessment is another challenge. These concerns are not unique in IoT networks. At this moment, we are not aware about any research outcome addressing this issue besides [50].

D. Packet Drop at AP

Dynamic change in user and SDN applications to action the flow can take time to set up rules to route traffic. For example, in security check application, the SDN manager forwards such packets to the controller. The controller contacts the malicious sites and only after verification sends flow entries back to edge device. If this process takes too much time, the further packets will be dropped [40] by edge node because of limited flow handling table capacity. The path inflation may add additional latency in wireless segment [51]. The edge network device runs the risk of being overloaded with flow entries [40]. Minor latency because of network overhead in wired network will introduce high latency in wireless time division multiple access (TDMA) scheduling. To address this problem, the following mechanisms can be studied thoroughly. 1) Immediately offload the traffic to different AP or LTE might be a better option in this situation [52]. 2) Because of already limited capacity, AP will not support multiple flow tables. Thus, dividing the flow table stack in different groups and state them as prior hierarchy gives less space to the multiple flow table stack. This ensures that only applications having higher priority, e.g., voice over IP (VoIP) and multimedia applications, can stay longer in multiple flow table groups of AP. So far, we have not been aware of any significant outcome on this topic from the SDN research community.

From our literature review, we emphasize that SDN can minimize the data center investment. Further, the programmable network enables IoT devices to talk each other without much hardware investment. Adequate uptime of the services, with the integration of SDN and IoT, can be ensured. We argue that the reliability can be guaranteed by service level agreements (SLAs) that providers have to meet irrespective of the technology they use in their networks. Table V illustrates SDN and IoT apps[2] developed by Intel and Meru networks[3] in this direction.

[2] [Online]. Available: http://searchsdn.techtarget.com/feature/SDN-to-support-Internet-of-Things-devices

[3] [Online]. Available: http://www.crn.com/news/networking/300074651/meru-networks-launches-sdn-app-store-management-platform.htm

To uplift the SDN integration to a higher and at broader level, *Web-of-Things* (WoT) seems a novel opportunity for SDWN researchers. In WoT, using normal web application designing tools, the real-world objects or things can be represented as resources that can be accessible via web technologies. This will minimize the need of waiting for other new components in networks, installing new infrastructure, or to redesigning the way we build our applications [53].

V. Conclusion

In this paper, we have presented the current key research efforts on SDWN. We emphasize that integration of SDN in IoT network can potentially bring exciting opportunities. We also highlighted that the traditional network tools to collect, store, process, and forward massive data are inefficient to meet critical future IoT network needs, whereas SDN can significantly simplify the network control and management needs. Furthermore, we described critical security and scalability issues of SDN network that are also common in IoT network. We conclude that SDN technology is gaining much attention from researchers from both industry and academia. Significant growth over coming years has been observed from industry such as Google and Juniper. The next generation of technology is almost ready to reap the benefits of controlling networks with a unified control protocol almost in every technological domain. Recent industry initiatives to integrate SDN and IoT technology are also presented.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] W. H. Chin, Z. Fan, and R. J. Haines, “Emerging technologies and research challenges for 5G wireless networks,” *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 106–112, Apr. 2014.

[3] P. M. Julia and A. F. Skarmeta. *Extending the Internet of Things to IPv6 With Software Defined Networking*, White Paper, 2014 [Online]. Available: http://www.euchina-fire.eu/wp-content/uploads/2014/06/SKARMETA-A.-Extending-the-Internet-of-Things-to-IPv6-with-Software-Defined-Networking.pdf

[4] L. Faughnan, “Software defined networking,” [Online]. Available: http://www.techcentral.ie/software-defined-networking. Accessed: Mar. 28, 2015.

[5] K. K. Yap *et al.*, “OpenRoad: Empowering research in mobile networks,” *Newslett. ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125–126, 2010.

[6] L. Suresh, J. S. Zander, R. Merz, A. Feldmann, and T. Vazao, “Towards programmable enterprise WLANs with odin,” in *Proc. 1st ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN’12)*, 2012, pp. 115–120.

[7] M. Bansal, J. Mehlman, S. Katti, and P. Levis, “OpenRadio: A programmable wireless dataplane,” in *Proc. 1st ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN’12)*, Helsinki, Finland, 2012, pp. 109–114.

[8] M. Yang, Y. Li, D. Jin, S. Ma, and L. Zeng, “OpenRAN: A software-defined RAN architecture via virtualization,” in *Proc. ACM SIGCOMM Conf.*, 2013, pp. 549–550.

[9] A. Gudipati, D. Perry, L. E. Li, and S. Katti, “SoftRAN: Software defined radio access network,” in *Proc. ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN’13)*, 2013, pp. 25–30.

[10] L. E. Li, Z. M. Mao, and J. Rexford, “CellSDN: Software defined cellular networks,” Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep., 2012 [Online]. Available: http://ftp://ftp.cs.princeton.edu/techreports/2012/922.pdf

[11] M. Moradi, L. E. Li, and Z. M. Mao, “SoftMoW: A dynamic and scalable software defined architecture for cellular WANs,” in *Proc. 3rd Workshop Hot Top. Softw. Defined Netw. (HotSDN)*, 2014, pp. 201–202.

[12] P. Goransson and C. Black, “Introduction,” in *Software Defined Networking: A Comprehensive Approach*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2014, pp. 1–20.

[13] P. Goransson and C. Black, “The genesis of SDN,” in *Software Defined Networking: A Comprehensive Approach*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2014, pp. 37–57.

[14] P. Goransson and C. Black, “How SDN works,” in *Software Defined Networking: A Comprehensive Approach*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2014, pp. 59–79.

[15] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, “A survey of software defined networking: Past, present, and future of programmable networks,” *IEEE Commun. Surv. Tuts.*, vol. 16, no. 3, pp. 1617–1634, Aug. 2014.

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

[17] “OpenFlow conformance certification labs for product testing.” [Online]. Available: https://www.opennetworking.org/certification/product#labs. Accessed: Mar. 15, 2015.

[18] B. Lantz, B. Heller, and N. Mckeown, “A network in a laptop: Rapid prototyping for software-defined network,” in *Proc. 9th ACM SIGCOMM Workshop Hot Top. Netw. (HotNets)*, Monterey, CA, USA, 2010, pp. 1–6.

[19] C. Scott, A. Wundsam, K. Zarifis, and S. Shenker, “What, where, and when, software fault localization for SDN,” EECS Dept., Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2012-178, 2012.

[20] M. Rietblatt, M. Canini, A. Guha, and N. Foster, “FatTire: Declarative fault tolerance for software-defined networks,” in *Proc. ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN13)*, 2013, pp. 109–114.

[21] M. Gupta, J. Sommers, and P. Barfors, “Fast, accurate simulation for SDN prototyping,” *in Proc. ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN’13)*, 2013, pp. 21–36.

[22] K. K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazimian, and N. Mckeown, “The stanford OpenRoad deployment,” *in Proc. IEEE 4th Int. Workshop Exp. Eval. Charact. (WiNTECH’09)*, Beijing, China, 2009, pp. 59–66.

[23] S. Jain *et al.*, “B4, experience with a globally-deployed software defined WAN,” *Newslett. ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[24] A. K. Rangisetti, H. B. Baldaniya, P. B. Kumar, and B. R. Tamma, “Load-aware hand-offs in software defined wireless LANs,” *in Proc. IEEE 10th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2014, pp. 685–690.

[25] G. Araniti, J. Cosmas, A. Lera, A. Molinaro, R. Morabito, and A. Orsino, “OpenFlow over wireless networks: Performance analysis,” *in Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2014, pp. 1–5.

[26] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, “Software defined wireless networks: Unbridling SDNs,” *in Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, Oct. 2012, pp. 1–6.

[27] P. Dely *et al.*, “A software defined networking approach for handover management with real-time video in WLANs,” *J. Mod. Transport.*, vol. 21, no. 1, pp. 58–65, Mar. 2013.

[28] C. Guimaraes, D. Corujo, R. L. Aguiar, F. Silva, and P. Frosi, “Empowering software defined wireless networks through media independent handover management,” *in Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 2204–2209.

[29] H. E. Egilmez and A. M. Tekalp, “Distributed QoS architectures for multimedia streaming over software defined networks,” *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1597–1609, Oct. 2014.

[30] J. Rukert, R. Bifulco, M. Rizwan-Ul-Haq, H. J. Kolbe, and D. Hausheer, “Flexible traffic management in broadband access networks using software defined networking,” *in Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.

[31] T. Feng, J. Bi, and K. Wang, “Joint allocation and scheduling of network resource for multiple control applications in SDN,” *in Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–7.

[32] J. Lee, *et al.*, “meSDN: Mobile extension of SDN,” in *Proc. 5th Int. Workshop Mobile Cloud Comput. Serv. (MCS’14)*, 2014, pp. 7–14.

[33] V. Gudla *et al.*, “Experimental demonstration of OpenFlow control of packet and circuit switches,” *in Proc. Opt. Fiber Commun. (OFC)/Nat. Fiber Opt. Eng. Conf. (OFC/NFOEC)*, Mar. 2010, pp. 1–3.

[34] W. Cerroni, G. Leli, and C. Raffaelli, "Design and test of a software defined hybrid network architecture," in *Proc. 1st ACM Ed. Workshop High Perform. Netw. (HPPN'13)*, 2013, pp. 1–8.

[35] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software defined optical networks technology and infrastructure: Enabling software-defined optical network operations," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A274–A282, Jun. 2013.

[36] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, Oct. 2012, pp. 7–12.

[37] A. Gember, P. Prabhu, Z. Ghadiyali, and A. Akella, "Toward software-defined middle box networking," in *Proc. 11th ACM SIGCOMM Workshop Hot Top. Netw. (HotNets)*, 2012, pp. 7–12.

[38] J. Q. Wang, H. Fu, and C. Cao, "Software defined networking for telecom operators: Architecture and applications," in *Proc. 8th Int. Conf. Commun. Netw. China (CHINACOM)*, 2014, pp. 828–833.

[39] G. Hampel, M. Steiner, and T. Bu, "Applying software-defined networking to the telecom domain," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 14–19, 2013, pp. 133–138.

[40] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Comput. Netw.*, vol. 71, pp. 1–30, 2014.

[41] L. Zhou and H. C. Chao, "Multimedia traffic security architecture for the Internet of Things," *IEEE Netw.*, vol. 25, no. 3, pp. 35–40, May/Jun. 2011.

[42] L. Zhou, R. Q. Hu, Y. Qian, and H. H. Chen, "Energy-spectrum efficiency tradeoff for video streaming over mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 5, pp. 981–991, May 2013.

[43] Y. N. Hu, W. D. Wang, X. Y. Gong, X. R. Que, and S. D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ. Posts Telecommun.*, vol. 19, pp. 92–97, Oct. 2012 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S100588851160438X

[44] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, Feb. 2013.

[45] FP7 ICT Project, "MediA ecosystem deployment through ubiquitous content-aware network environments," ALICANTE, No. 248652 [Online]. Available: http://cordis.europa.eu/project/rcn/94029_en.html, 2011.

[46] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Top. Softw. Defined Netw.*, 2012, pp. 7–12.

[47] Y. N. Hu, W. D. Wang, X. Y. Gong, X. R. Que, and S. D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ. Posts Telecommun.*, vol. 19, pp. 92–97, Oct. 2012.

[48] S. H. Yeganeh and Y. Ganjali, "Beehive: Towards a simple abstraction for scalable software defined networking," *in Proc. 13th ACM Workshop Hot Top. Netw. (HotNets'14)*, 2014, pp. 1–13.

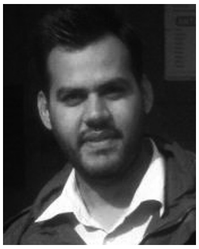
[49] G. Finnie, "The role of DPI in SDN work," *QOSMOS Technology*, White Paper, Dec. 2012, pp. 1–14.

[50] H. Mekky, F. Hao, S. Mukherjee, Z. L. Zhnag, and T. V. Lakshman, "Application-aware data plane processing in SDN," in *Proc. 3rd ACM SIGCOMM Workshop Hot Top. Softw. Defined Netw. (HotSDN'14)*, 2014, pp. 13–18.

[51] J. Hu, C. Lin, X. Li, and J. Huang, "Scalability of control planes for software defined networks: Modeling and evaluation," in *Proc. IEEE 22nd Int. Symp. Qual. Serv. (IWQoS)*, May 2014, pp. 147–152.

[52] M. Manic *et al.*, "Next generation emergency communication systems via software defined networks," in *Proc. 3rd GENI Res. Educ. Exp. Workshop (GREE)*, Mar. 2014, pp. 1–8.

[53] D. Raggett, "The web of things: Challenges and opportunities," *IEEE Comput. Mag.*, vol. 48, no. 5, pp. 26–32, May 2015.



Keshav Sood (GSM'15) received the B.Tech. degree in electronics engineering (with distinction) and M.Tech. degree in optical fiber engineering from Punjab Technical University, Jalandhar, India, in 2007 and 2012, respectively, and is currently working toward the Ph.D. degree in information technology (IT) at Deakin University, Melbourne, Vic., Australia.

He was a Trainee with the Terminal Ballistic Research Laboratory, Chandigarh, India. His research interests include network security and flow management in SDN.

Mr. Sood is a Professional Engineer, as recognized by Engineers Australia, Barton, A.C.T., Australia. He served as a TPC member of various IEEE conferences include IEEE INFOCOM, IEEE Bigdata service, and IEEE ITNAC.

Shui Yu (M'05–SM'12) received the Associate degree in mathematics, B.Eng degree in electronic engineering, and M.Eng degree in computer science from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 1993, 1993, and 1999, respectively, and the Ph.D. degree in computer science from Deakin University, Melbourne, Vic., Australia, in 2004.

He is currently a Senior Lecturer with the School of Information Technology, Deakin University. He is a Member of the Deakin University Academic Board (2015–2016). He has authored two monographs and edited 1 book, more than 100 technical papers. He initiated the research field of networking for big data in 2014. His h-index is 21. His research interest includes big data, networking theory, cyber security, and mathematical modeling.

Dr. Yu is a member of the AAAS, the Vice Chair of the Technical Subcommittee on Big Data Processing, Analytics, and Networking of the IEEE Communication Society, and a member of the IEEE Standard Committee of Big Data. He has authored papers of the top journals and top conferences including IEEE TPDS, IEEE TC, IEEE TIFS, IEEE TMC, IEEE TKDE, IEEE TETC, and IEEE INFOCOM. He is currently a member of the Editorial Boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE ACCESS, and a number of other international journals. He has served more than 50 international conferences as a member of the Organizing Committee, such as Publication Chair for IEEE Globecom 2015 and IEEE INFOCOM 2016, TPC Co-Chair for IEEE ATNAC 2014, IEEE BigDataService 2015, and IEEE ITNAC 2015.

Yong Xiang (SM'12) received the Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Melbourne, Vic., Australia, in 2003.

He is a Professor and a Director of the Artificial Intelligence and Image Processing Research Cluster, School of Information Technology, Deakin University, Melbourne, Vic., Australia. He has authored more than 110 refereed journal and conference papers. His research interests include signal and system estimation, information and network security, multimedia (speech/image/video) processing, and wireless sensor networks.

Dr. Xiang is an Associate Editor of IEEE SIGNAL PROCESSING LETTERS and IEEE ACCESS. He has served as a Program Chair, TPC Chair, Symposium Chair, and Session Chair for a number of international conferences.