SDN based WLANs Design and Implementation: A Survey

KHALID IBRAHIM QURESHI, Dalian University of Technology, China

WLANs being the most common access technology, the use of SDN in WLANs is inevitable as SDN paradigm has been widely accepted and being implemented in the networking industry. Currently the main focus of SDN is wired-networks. OpenFlow that is the most widely accepted protocol for SDN do not fully support the wireless medium. We believe that SDN can not only solve the long existing issues for WLANs but can also enable the researchers to bring more advanced techniques for higher data rates, better security and QoE. In this survey, we have emphasized the importance of integrating SDN into WLANs, the benefits WLANs can gain by adopting SDN. More specifically we have explained the advantages as well as the implementation details of SDN for WLANs along with the most prominent recent research efforts and contributions.[1]

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: Wireless local area networks, software defined networking, load balancing

ACM Reference Format:
Khalid Ibrahim Qureshi. 2010. SDN based WLANs Design and Implementation: A Survey. *ACM Comput. Entertain.* 9, 4, Article 39 (March 2010), 24 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Wireless Local Area Networks (WLANs) is the most preferred method for connecting mobile devices to an organizations network. WLAN services are the primary source of connection to the network and are thus found in almost all the companies, universities, coffee shops, hotels and even in the streets. Furthermore, the number of WLAN users is increasing at a remarkable rate. In recent years, as the new applications requiring high bandwidth has surfaced, the overall bandwidth demand has risen drastically. According to CISCO by 2021 the data traffic will reach up to 278 Exabyte per month. WLANs contribute 40% of the data traffic currently. Also, there will be nearly 541.6 million public WiFi hotspots, a six-fold increase since 2016 [CISCO 2015; Cisco Systems 2017]. With the emergence of the Internet of Things (IoT), even more, devices will connect to the Internet. By 2020, it is predicted, 50 billion devices will connect to the Internet [Evans 2011]. Moreover, the new high-speed standards, like IEEE 802.11ad, are employing high frequencies which are absorbed by the walls thus, to provide coverage, more Access Points (APs) need to be deployed [Moura et al. 2015]. Managing this enormous traffic and such a large number of APs is a big challenge. Researchers are working to develop efficient techniques to make network management easier. Various techniques need to be opted to handle this growing need for traffic.

[1]This is an abstract footnote

Author's address: Khalid Ibrahim Qureshi, Dalian University of Technology, 104 Jamestown Rd, Williamsburg, VA, 23185, China, khalid.ibrahim@mail.dlut.edu.cn.

1.1 Need for SDN enabled WLANs

Conventional networks are highly inflexible as they have hard coupled software which is proprietary and cannot be modified. So there is no (very little) room for innovation in them. In traditional WLAN management architecture, the controllers used are proprietary, which can perform specific tasks like network-wide optimizations such as controlling transmission power, selecting the best channel for communication, mobility management, and precise localization as well as QoS and security policy implementation. But, these controllers can only manage the compatible devices typically of the specific vendor. Software-defined networking (SDN) is attracting a lot of attention recently. SDN is a network paradigm that gives hope in changing the current system limitations. It breaks down the vertical integration of underlying networking devices by taking away the control logic from the network devices such as routers and switches. Separating the control plane from the routers and switches makes them simple forwarding devices responsible for forwarding the data. All the control logic is shifted to the logically centralized controller. This, in turn, simplifies the implementation of network policy and network (re)configuration and evolution [Kreutz et al. 2015]. De-coupling Control Plane from the Data Plane provides the freedom of programmability and thus enabling the development of more efficient network applications [Chen et al. 2015]. SDN also makes the Local Area Networks (LANs) less complex which in-turns make the network management easy [Raumer et al. 2014]. SDN is mainly being investigated in wired networking domain. However, due to the characteristics of SDN, many researchers are also working to bring features of SDN architecture to wireless networks such as mobile networks and WLANs [Chen et al. 2015; Granelli et al. 2015; Riggio et al. 2015b; Zhao et al. 2014b]. This paper provides the survey of research efforts for SDN enabled WLANs (from here onwards SDWLANs).

1.2 Review of Related Survey Articles

SDN is gaining popularity and a number of good survey articles have been published covering different aspects of SDN. [Kreutz et al. 2015] presented one of the most comprehensive survey on SDN. The authors depicted SDN in layered architecture and explained each layer in detail starting with the lower layer consisting of infrastructure and connected to higher layers via southbound API. Above this layer is network hypervisors, and network operating system (NOS). Further above this is northbound API connecting these to top layers that are language based virtualized APIs. Very detailed discussion on each of the layers as well as industry trends towards SDN, future research directions, and research gaps, is carried out.

Previous surveys [Alvizu et al. 2017; Bannour et al. 2017; Bizanis and Kuipers 2016; Dargahi et al. 2017; Huang et al. 2017; Li and Xu 2017; Michel and Keller 2017; Nde and Khondoker 2016; Scott-Hayward et al. 2013; Trois et al. 2016] have covered different aspects of SDN paradigm ranging from data plane to control and application planes. [Bannour et al. 2017] has focused on distributed control plane whereas [Li and Xu 2017] discuss the load balancing research efforts with distributed controllers. [Bizanis and Kuipers 2016] have in detail discussed the applications of SDN and Network virtualization in the Internet of Things (IoT).

[Granelli et al. 2015; Haque and Abu-Ghazaleh 2016; Khan et al. 2015] are more related to our work. [Khan et al. 2015] presented a short survey on SDN in heterogeneous networks, [Granelli et al. 2015] discussed in detail the benefits of virtualization in the wireless domain. [Haque and Abu-Ghazaleh 2016] has a more detailed survey on wireless SDN with the main focus on four wireless network domains namely Cellular Networks, Mesh Networks, Sensor Networks, and Home Networks.

Although many good and comprehensive surveys have been published on different aspects of SDN, [Soo et al. 2018] and [Dezfouli et al. 2018] are the most relevant to our topic of discussion. In [Dezfouli et al. 2018] the authors have discussed SDWLANs from broadly two perspectives: (i) SDWLANs architecture and (ii) Association Control and Channel Assignment mechanisms. The authors have discussed what research efforts have been made until now and the gaps and future research directions. In [Soo et al. 2018] authors have covered the load-balancing in 802.11 networks with the research efforts made over the recent years with and without SDN.

Our survey article is different from the other articles as we are more focused on attracting researchers' attention towards most needed SDWLANs directions. We aim to do so by summarizing the testbed designs as well as simulation environments that have been designed and used by the researchers for SDWLANs. Furthermore, we have also discussed the most persistent unsolved issues in legacy WLANs with their proposed SDN based solutions.

1.3 Contributions of this Survey

SDN has been mainly explored in wired networks and data center networks. However, some research efforts have started to investigate SDN in wireless domain. Our main aim in writing this article is to highlight the research efforts in SDWLANs thus attract the researchers' effort in this direction. This paper presents a comprehensive survey of the research and development in the field of SDWALNs. More specifically the contributions of this article are as follows:

- It identifies the major benefits of SDN based networks with reference to the WLANs
- The main focus of this article is to highlight up-to-date research efforts made in SDWLANs.
- The concept of Virtual Access Points.
- Deployment strategies
- Gaining control of the MAC and PHY-layers using OpenFlow

1.4 Article Structure

Section 2 will present a short overview of SDN. In section 3 we will have a look at the possible benefits of SDNs for WLANs and discuss un-solved problems in legacy WLANs. Section 4 gives details for implementing SDN enabled WLANs for research purposes including simulation/emulation environments, and real-world test-beds design along with components. In sections 5 we will discuss the most significant research work. In Section 6 we are going to discuss some potential research directions and research gaps. Section 7 will conclude this article.

2 OVERVIEW OF SDN BUILDING BLOCKS

Today's legacy networks implement a large number of protocols running on numerous devices (such as switches and routers) in order to implement a variety of network services. To manage this large number of devices highly skilled personnel are required to carefully configure and manage the network. Furthermore, every device has integrated control logic. For a small network with a few network devices, this is good and easy to handle. With the extension of network more devices are added, and the problem arises. With hundreds of routers and switches and thousands of hosts, configuring and managing the network becomes a hectic job. Since every network device has its control logic built-in, in order to deploy a new global policy, each device needs to be configured separately. Moreover, making some changes in the network or testing some innovative policies is not easy. The vertical integration of control and data planes leaves no room for innovation and limits the flexibility of the network.

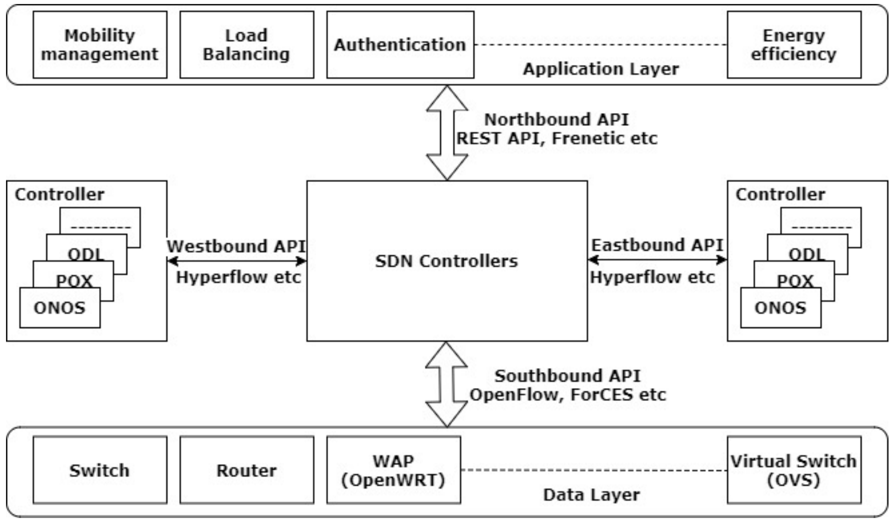

Fig. 1. Basic Architecture of SDN

The creation of SDN is an effort by the researchers to tackle the problems mentioned above. SDN was created by researchers at Stanford University and UC Berkeley [Sezer et al. 2013]. SDN architecture breaks down the tight coupling between the control plane and data plane. The control logic is removed from the underlying devices making them merely the forwarding devices. Network intelligence is logically centralized [Shin et al. 2012]. Network operators can write programs to configure and manage the network and can simply implement a global policy.

2.1 Building Blocks - Planes

Figure 1 shows a typical architecture of SDN. In this section we are going to give the brief description of the components of SDN and their working.

2.1.1 Data Plane (DP). The physical or virtual network devices including routers, switches, and middleboxes constitute the SDN data plane. In contrast to conventional network devices, which have the control plane and data plane implemented in their firmware, these devices have no control logic and only forward the packets under the instructions from the control plane. So the network operators do not need to modify the configuration of every single device in the network to implement a global policy rather, they just need to instruct the controller which in turn will modify the configuration of each device accordingly.

2.1.2 Control Plane (SDN Controller). The SDN controller is logically a centralized (physically, may or may not be centralized) entity having all the control logic and is responsible to set-up the flows and paths for the data within the network and to tear down or modify these paths when required. The controller-specific interfaces to communicate with the data plane called southbound APIs. By using the southbound API, controllers can obtain the information about underlying devices such as capacity and requirements, and in accordance with the global policy it can set up flow rules. Controllers gather the information of physical network resources and using northbound API bind this information to the application layer. More than one controller can be used in a network in order to mitigate the single point of failure or to scale the network. Two or more controllers can communicate via east and westbound APIs. These APIs are also required to enable the controllers from different domains to communicate.

2.1.3 Application Plane (Network Applications). Since the control and data planes are decoupled the network operators can design applications that can be used to monitor, manage and configure the entire network. The deployment of new policies in the existing network is very simple with these applications. Furthermore, the researchers can experiment with the new ideas in the existing network by designing an application without affecting the normal operations of the network.

2.2 Building Blocks - Interfaces

There are basically two main classes of Interfaces, Southbound Interface (SBI), that is between Control and Data Plane, and Northbound Interface (NBI), which is between Application and Control Plane.

2.2.1 Southbound Interfaces (SBIs). SBIs allow the network elements to exchange control and state information with the control plane. They are responsible to provide the programmatic control of forwarding operations, device resources, statistics and event notifications. Controller is capable to communicate with network elements via this interface. It is necessary for all the devices to have a standard open southbound interface so as to allow interoperability among devices from different vendors. OpenFlow [Mckeown et al. 2008] promoted by Open Networking Foundation (ONF) is the first and currently most accepted open southbound interface. While OVSDB [Pfaff and Davie 2013], ForCES [A. Doria et al. 2010], protocol oblivious forwarding (POF) [Song 2013] are some less popular open SBIs.

2.2.2 Northbound Interfaces (NBIs). Northbound Interfaces provides communication between controller and the applications. NBIs helps the application plane to get abstract view of the network and collect statistics information and enforce the business policy. The NBIs allows automation, innovation and management of SDN network. RESTful JSON APIs is a most common NBI. No standard definition is yet available. ONF is working on definition of standard NBIs.

2.3 OpenFlow (OF)

To practically implement the SDN controller two requirements must be met. Having a common logical structure for all of the networking devices is the first requirement. The implementation of this structure may be different for the devices from different vendors but the controller should see a uniform logical switch function. Second, there must be a secure standard protocol for communication between switch and the networking devices.

These two requirements are fulfilled by the OpenFlow protocol, which act as both the secure standard communication protocol between the controller and the networking devices as well as uniform logical switch function specification for the networking devices. OpenFlow is designed and defined by the Open Networking Foundation (ONF) which is consortium of service providers, vendors and software providers in networking industry. ONF purpose is to promote SDN.

Since 2009 ONF has released 5 versions of OpenFlow switch specifications the details of which can be found at [ONF [n. d.]].

3 POSSIBLE BENEFITS OF SDN FOR WLANS

SDN has revolutionized the networking. With its open programmable interfaces researchers are able to implement and test the ideas that were not possible before. In this section first we will discuss general advantages of SDN over legacy networks and then we are going to highlight the features of SDN that WLANs can benefit from.

3.1 Advantages of SDN

A number of advantages can be gained by implementing SDN into Wireless Networks a few of which we are going to discuss here.

3.1.1 Enhanced Configuration. For network management, the most crucial task is the configuration of network devices. Since the networks are heterogeneous with devices from different vendors, the configuration becomes a tedious task and requires a high level of expertise. Since configuration process is manual, it is prone to human error that is a simple mistake in router configuration, etc. SDN tends to solve this issue by introducing centralized control of all the devices through a single control interface. All of the network devices e.g. switches, routers, wireless access points, firewalls, etc are possible to be controlled from a single point, automatically.

3.1.2 Improved Performance. Maximum utilization of invested network infrastructure is desired in network operations. However, since a single network is comprised of different technologies thus optimizing the performance is not an easy task. So currently, a part of the network or network services is optimized to enhance user experience. This technique may not optimally improve the overall system performance as it is based on local information. SDN, however, introduces the global view of the network with the central controller keeping track of all the resources of the network thus enabling means to optimize the performance of the whole network. New SDN based solutions of these problems like end-to-end congestion control [Hertiana et al. 2017], load-balancing [Homma and Shinomiya 2017; Sang et al. 2017] etc are promising.

3.1.3 Encourage Innovation. One of the most prominent challenge today's network face is that there is no room for innovation. The reason behind that is the use of proprietary network hardware and software that do not provide any open interface for experimentation. This proprietary equipment cannot be modified to experiment with the new innovative ideas. Furthermore, the experimentation is done on either simulators or simplified test-beds. The results of these experiments do not provide enough confidence for the industry to adopt these new ideas thus most of even good ideas do not get enough attention from industry and therefore go in vain. SDN tends to change that.

3.1.4 Efficient Network Monitoring. As SDN controller is a logically centralized having the global view of the entire network and it keeps track of all the network devices and resources, it can continuously monitor the network and with the change in traffic patterns, it can adapt to these changes automatically. Monitoring capability can enhance the performance of the network. Via effective monitoring suspicious traffic that flows in the network can be detected and in order to prevent the network from further damage effective countermeasures can be taken thus improving the network security. To design and implement countermeasure application is rather easy when using SDN [Suh et al. 2014; Yu et al. 2013].

3.2 SD-WLAN Benefits

3.2.1 Efficient Network Management. Wireless networks have become very dense with a large amount of data flowing through them. This huge amount of traffic needs to be carefully managed in order to ensure the QoS without increasing cost and complexity. Another major concern is interference between the neighboring APs. The centralized controller and global network view in SDN, smart management techniques can easily be applied to solve these problems. The controller can adjust the transmission channel of each AP in such a way to best reduce the interference from neighboring APs. Furthermore, if not properly managed the traffic load on certain APs tends to increase their maximum capacity. Traffic engineering techniques can easily be implemented since the controller has the global view of the network, and status of all the devices. Thus the controller can distribute the load equally and efficiently amongst the network devices making more efficient use of the network resources [Costanzo et al. 2012]. Mahout [Curtis et al. 2011a], DevoFlow [Curtis et al. 2011b], Hyperflow [Tootoonchian and Ganjali 2010] are some load balancing schemes implemented using SDN based networks.

3.2.2 Managing wired and wireless networks in unison. The WiFi-enabled devices are increasing consistently across the enterprise, public and residential networks, which introduced a big question for network operators that is how to support and manage these networks? Furthermore, wired and wireless networks are being used in unison. Thus managing the network from a service provider to the users' premises is a huge challenge. In particular, integrating management of wired and wireless network seamlessly is a challenge. Furthermore, as the commodity off-the-shelf hardware is used for WiFi systems achieving a centralized management is not possible. Moreover, there is no such solution for legacy networks that can support multi-vendor and multi-technology devices. SDN tends to change this by providing open interfaces that can support any SDN based equipment from any vendor and also can support multiple network services at a single managing point.

3.2.3 Unplanned and uncoordinated deployment. TodayâĂŹs home WiFi networks mostly suffer from poor performance because there is no coordination between the neighboring networks and deployment is not well-planned. All neighboring parties normally deploy their own dedicated wireless LAN infrastructure. As a result, these neighboring APs, as well as public APs, cause interference with one another and results in unnecessary transmission delays which ultimately reduces network capacity. In order to improve the user experience the typical features of EWLANs such as mobility management, load balancing, transmission power, channel selection, can be introduced into the home WLANs. However, todayâĂŹs off-the-shelf commodity WiFi APs used in home networks are unable to provide infrastructure controlled hand-offs and has no coordination, thus optimized network operation is impossible. However, with the use of SDN these simple APs can be centrally managed and reap the benefits of EWLANs.

3.2.4 Heterogeneous devices. As the trend of Bring-Your-Own-Device (BOYD) [BellTechlogix [n. d.]] is getting popular, a diverse set of devices are becoming part of today's network. This causes a diverse set of problems including Network Overload, Security risks, etc. This introduces the challenge of providing Network Access Control (NAC) and keeps it abstracted from the internal working of the IEEE 802.11 MAC layer while keeping the 802.11 protocol complexities hidden from the network operators. Programmability and global view of the network in SDN make it possible to implement policy based NAC efficiently.

3.2.5 Efficient Resource management. Differentiating traffic based on the application-aware manner is not possible in legacy networks, i.e., at the last hop, applications requirements are ignored. Due to network neutrality policies such differentiation may not be possible over the Internet but it is legally allowed and desirable to differentiate the requirements based on application at the last hop in home networks. SDN based solutions can be designed to differentiate traffic on demand of the applications. Thus fine-grained traffic control and application-specific transmission control can be achieved to provide fine-grained resource management, but requires programmability at WiFi datapath and integrated control of wired and wireless networks.

3.2.6 Traffic Prioritizing. Users cannot define traffic prioritizing in legacy networks. There is no such interface that allows the user to define the priorities. Neither do the applications can report and/or specify their requirements to the Wireless AP. So the users are unable to control the application awareness in their home networks.

3.2.7 Flexible control. TodayâĂŹs enterprise solutions are coalesced, expensive, and vertically integrated thus only offer a limited set of solutions to WLAN-specific challenges. For example, in dense environment such as auditoriums or lecture halls, the performance of WiFi networks can be degraded due to the high collision rate. Solutions such as de-duplication filter for WiFi frames at a centralized (logically) location in the network can increase the transmission probability thus enhancing overall network performance. However, such solutions are specific to certain use case and mostly are proprietary thus provide no room for flexibility in network architecture. Moreover, commodity off-the-shelf WiFi hardware is not considered by such solutions. Thus, by using SDN, not only the performance of the network can be greatly improved but also the cheaper cost effective solutions can be designed.

3.2.8 Improve Quality of Service. The major features of SDN like flexibility, global view of the network, central control, and programming abstraction favors the high QoS for the users. Implementation of SDN in IEEE 802.11 networks can help to attain better connectivity as it has the capability to select the best wireless AP for the users that give better QoE. The network-wide view can help to mitigate interference in neighboring WiFi devices [Chaudet and Haddad 2013]. Overall performance and security of WLANs can be greatly enhanced by the flexibility offered by SDN. Different users can be allocated a different level of privileges by configuring Rules in the network. For instance, restricting the low-level employee from accessing the other applications running on the network is simple.

3.3 Legacy WLANs and their problems

Due to characteristics like high data throughput and wide coverage, the IEEE 802.11 based wireless communication been adopted globally over the past decades [Society 2010]. WLANs are deployed as a last hop wireless extension to the existing LAN infrastructure so that clients can connect and roam freely within the coverage area of the network.

WLANs can be broadly divided into three categories based namely: Enterprise WLAN (EWLAN), home WLANs and public Wi-Fi networks. Our major focus in this paper is EWLANs and home WLANs. These two are quite different from each other as EWLANs has a centralized controller whereas, the home WLANs lack this entity and the APs are standalone in the sense of control.

The centralized controller in EWLANs can control various parameters of AP including channel selection, power adjustment etc. The controller also has the capability to monitor the entire network as it has the global view of the network thus it can attain the status of all the APs in the network and also the wireless channel information. In order to meet the ever-growing need for network performance, modern EWLANs require network services which include mobility management, AP load-balancing, transmission power adjustment, and interference mitigation.

Home networks generally lack a centralized controller. Home networks are deployed in an uncoordinated manner without any cooperation between APs that have overlapping service areas. Due to lack of coordination, these APs suffer mostly from interference from the neighboring APs. As a result, the overall QoE for the end user is reduced.

The traditional WLANs have a number of problems that need to be solved. In this section, we are going to discuss some of the problems that traditional WLANs have not yet been able to solve.

3.3.1 Mobility Management. The networks are becoming more complex, EWLANs requires a novel architecture to fulfill the growing needs of the network. One of the key issues in EWLANs is the Mobility Management. To guarantee the Quality of Service (QoS) seamless handoff is an important requirement as the real-time multimedia services, like video streaming, are being used and any delay in hand-off can result in degradation of QoE. In legacy WLAN hand-off algorithms (CAPWAP [Stanley et al. 2009]), the RSSI value is used as the hysteresis margin. Using only one parameter is highly inefficient in providing seamless mobility and can cause imbalanced traffic load on wireless APs, which significantly reduce the user experience. Many vendors have designed commercial solutions to solve these mobility issues. Unfortunately, these existing solutions [Cisco 2017; Fortinet 2017] are expensive, proprietary, and lack programmability. These solutions are not designed to handle low-cost off-the-shelf AP hardware which is mostly used in EWLANs.

3.3.2 Load Balancing. In legacy WLANs the user association to an AP is based on the highest RSSI as measured by the user [Ercetin 2008]. However, this technique may result in an unbalanced load on each AP since many users may tend to associate with a particular AP due to its high RSSI which may result in the degradation of system performance. Load balancing techniques are required in order to cope with this issue so that load can be appropriately distributed amongst all the APs and thus increase the overall network performance. The proprietary solutions tend to solve this load-balancing issue, but they are vendor specific and are unable to manage the low-cost and multi-vendor APs commonly deployed in the providers' network.

3.3.3 Energy Efficiency. [Jardosh et al. 2009] argued that WLANs can go in low load conditions making most of its APs to go idle. In case of low load conditions, it is desirable to put some of the APs to sleep and serve the coverage area with a lesser number of APs for low load time. This will significantly reduce the energy consumption and, thus, operating cost. The selection of APs to be put in sleep mode and which APs are to serve the users is a critical task and shall be done with much care so it may not deteriorate the overall QoE for users.

3.3.4 Interference Mitigation. In urban areas home networks, the APs are deployed in un-coordinated fashion and it is common for large number of APs to be within transmission range of each other. These APs suffer from interference with the neighboring APs and thus causes congestion and deterioration of QoS. Moreover, these APs also face interference from non-WiFi devices such as microwave ovens etc.

4 EXPERIMENT SETUP FOR SD-WLAN BASED RESEARCH

Implementing SDN on WLANs is a challenge as until now OpenFlow is mainly focusing on wired network and thus do not provide control over the PHY and MAC layer of Wi-Fi devices. In this section, we will discuss the techniques and methods that enables researchers to experiment with SD-WLANs. First, we are going to discuss the simulators/emulators that have been used by researchers for experimenting and later we will in detail discuss the design of real-world testbeds, components needed to design a testbed. Then at the end of this section we will discuss most significant research work that has been conducted in this direction.

4.1 Simulators/Emulators

To develop and test new research ideas it is desired to test them with simulations since implementing an idea with real-world test-beds requires more time and may require more resources. If the simulation results are satisfactory enough, the same work can be implemented and tested on actual test-beds to further investigate the research outcomes. However the selection of simulator is very critical. It is necessary to ensure if the selected simulator is suitable for current experiment, that is would it be able to generate the results that are comparable to the real test-beds. There are a number of simulation tools that are being most widely used for SDN based network simulations and testing. We are going to discuss the most significant ones here.

4.1.1 Mininet. Mininet [Lantz et al. 2010] is a BSD licensed open-source simulator/emulator that is most widely used in SDN based network simulations. With the help of mininet an, inexpensive and simple network test-bed for SDN based wired networks can be created. It is based on Linux container, a very light feature that provides virtualized network interfaces, routing and ARP tables to individual processes. This network emulator is great to be used in wired SDN environments as the same code can be run on a real network. The major disadvantage of mininet is it does not support modeling of the wireless channel, and hand-off process cannot be modeled in this tool. Furthermore, the mininet runs on a single client machine and cannot exceed the available CPU or bandwidth, so it suffers from resource limitations greatly as tested in [Keti and Askar 2015]. It can be used for small-scale network simulations as tested and reported by [De Oliveira et al. 2014].

4.1.2 Mininet-WiFi. Mininet-WiFi [Fontes et al. 2015] is the modified version of mininet with the added features to simulate the SD-WLANs while all the features of mininet are included. The developers have added the virtualized mobile stations and access points. Furthermore, it can simulate the mobile client. This simulator has same limitations as of mininet with respect to scalability and fidelity of wireless channel.

4.1.3 NS-3. NS-3 (network simulator 3) [ns3 2017] is a well documented and highly modularized network simulator licensed under GPLv2 open-source. It can be easily modified and includes the tools for wireless channel and mobility modeling, but it lacks the scan mechanism yet which makes the layer-2 handoff impossible to be simulated in ns-3. Moreover, the project of ns-3 supporting OpenFlow simulates the operation of OF-switch by compiling a module written in C++. Same is the case with the OF-Controller. A real OpenFlow controller cannot be linked with a ns-3 program so ns-3 has to implement its own OF-controller written in C++, the behavior of which may differ from the real OF-Controller.

The integration of Mininet with NS-3 to emulate the SD-WLAN was explored and tested in [Pakzad et al. 2017]. The test results indicate high degree of accuracy.

4.1.4 EstiNet. EstiNet [Estinet-Technologies 2013] is a commercial simulator/emulator designed for OpenFlow networks. It is a well-developed tool with GUI support through which users can easily setup a network by drag-and-drop. It also includes wireless channel modeling and OpenFlow switch support. EstiNet out-performs the mininet, however, this tool is proprietary and even after purchasing this software the source code is not entirely modifiable.

Comparison of simulators features is given in Table 1.

Table 1. Comparison of Simulators

	Mininet	**Mininet-WiFi**	**ns-3**	**Estinet**	**Opennet**
OpenFlow Specification	1.3.1	1.3.1	1.3	1.3.1	1.3.1
Simulation Mode	No	No	Yes	Yes	Yes
Emulation Mode	Yes	Yes	Yes	Yes	Yes
Real Controllers Support	Yes	Yes	No	Yes	Yes
Reliability	Results may differ under different load conditions	Results may differ under different load conditions	High	High	High
Scalability	Limited by system resources	Limited by system resources	High	High	High
Wireless Support	No	Yes	Yes	Yes	Yes
Open-Source	Yes	Yes	Yes	Partially (needs to purchase)	Yes

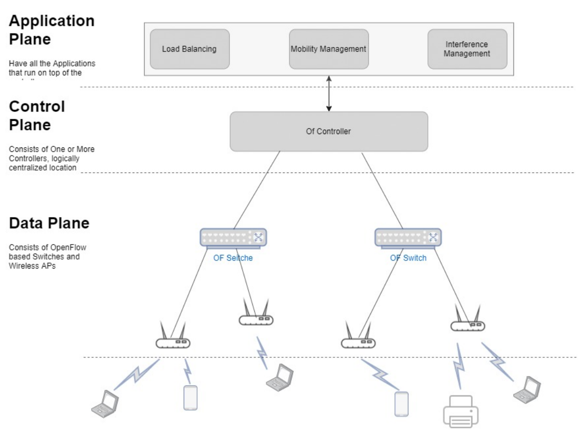

Fig. 2. SD-WLAN Architecture

4.2 Designing Real world Test-Beds

SD-WLAN based research efforts cannot be accurately studied and evaluated based on simulations. Furthermore, designing a test-bed for SD-WLANs is comparatively easy and provides with more accurate results. Additionally designing these test-beds is also cheap as it would require the same server to run the controller and designed applications, and data-plane can be designed on cheap off-the-shelf APs. Hence, many researchers prefer to build their own testing environment based on real SDN hardware switches. We are going to discuss the components required to make a test-bed for SD-WLANs.

4.3 Components of Test-bed for SD-WLANs

4.3.1 Control Plane for SD-WLANs. Since the wide acceptance of SDN in the research community, a number of SDN controllers have been designed. Most of these are open source. Many vendors, such as cisco and juniper, are also making consumer level controllers (basically derived from other open source controllers). No specific separate controllers are needed for SD-WLANs. The basic controllers discussed in this section works with SD-WLANs. We are going to discuss few basic controllers which, to the best of our knowledge, are most widely being used by the researchers in SDN. Table 2 summarizes the basic features of these controllers.

Table 2. Controller Features

References	Controller	API Language Support	OpenFlow version	Open Source	GUI	REST API	Platform Support
[Gude et al. 2008]	NOX/POX	Python	v1.0	Yes	No	No	Linux, Mac, Windows
[Linux Foundation Project 2017]	OpenDayLight	Java	v1.0	Yes	Yes	Yes	Linux, Mac, Windows
[Project Floodlight [n. d.]]	Floodlight	Java	v1.0	Yes	Web GUI	Yes	Linux
[Ryu [n. d.]]	Ryu	Python	v1.3	Yes	Yes	Yes	Linux
[Tre 2017]	Trema	C, Ruby	v1.0	Yes	No	No	Linux

Note: This is a table footnote.

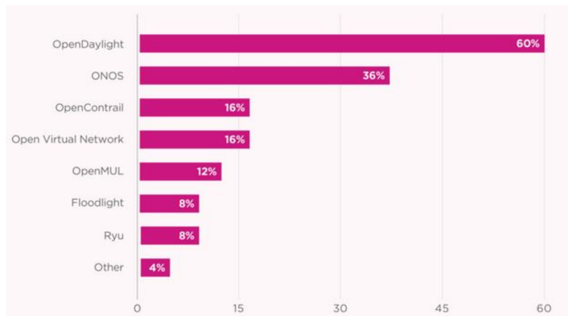

Fig. 3. Open Source SDN Controllers deployment solutions [SDxCentral 2017]

NOX/POX Controller. NOX [Gude et al. 2008], originally developed by Nicira Networks (now owned by VMware) in 2009, was the first SDN controller. It provides the framework for the development of high-level language based APIs to control and manage the networks. NOX controller was dedicated to the research community as an open source in order for the researchers to experiment with the OpenFlow and SDN-based networks, further extended and supported by researchers at Stanford and UC Berkley. It supports C++ and Python for application network applications development. It uses OpenFlow to communicate with the underlying network devices. Many early research works like Hyperflow [Tootoonchian and Ganjali 2010] used NOX controller. NOX controller was complicated to deploy and working in two different languages was not necessary. Thus POX controller [McCauley [n. d.]] was proposed as a replacement to NOX controller, distributed under GPL. POX controller is derived from the NOX with its support for python APIs, and for the southbound API, it supports OpenFlow to communicate with the data plane. It has multiple advantages over NOX controller including reusable sample components for topology discovery and path selection etc, can run anywhere using pypy for fast deployment.

OpenDaylight Controller. OpenDaylight (ODL) is a modular, open source controller platform that focuses on network programmability. ODL is developed by the collaboration of multiple vendors and user organizations. More than 50 organizations are part of project ODL, and over 1000 developers are working on it. ODL is evolving quickly, and the code is embedded and integrated into more than 50 vendors products. Further details about ODL and its implementation details can be found at [Linux Foundation Project 2017].

Floodlight Controller. It is java based, open source SDN controller licensed under Apache. Including engineers from Big Switch, a large number of developers are supporting it. Many of the recent research works adopt Floodlight controller as it is very simple to implement and is designed to support scalability. Open Networking Foundation manages this open networking controller. Many experimental research works and test beds are designed with this controller for example Odin [Suresh et al. 2012], OpenSDWN [Schulz-Zander et al. 2015], COAP [Patro and Banerjee 2014] etc. Further implementation details fo Floodlight Controller can be found at [Project Floodlight [n. d.]].

ONOS. Open Network Operating System (ONOS) is java based, open source SDN controller licensed under Apache 2.0. ONOS can run on cloud environment thus providing logically centralized while physically distributed servers with the benefit of avoiding the single point of failure due to server fault and supports live update of hardware or software without any interruption to the traffic. The ONOS APIs are also written in Java and can be bundled with ONOS in a single JVM. [Lee et al. 2016], [Yang et al. 2017] research work is based on ONOS controller. Further details can be found at [(ON.Lab) 2014].

Ryu Controller. Ryu controller is written entirely in Python. It is an open source component-based framework for SDN with all the code freely available under Apache license 2.0. This controller is continuously being tested with OpenFlow Switches by the developers. It provides well-defined API for software components thus making it easy for researchers to create and test network management and control applications Implementation details, source code, etc can be found on [Ryu [n. d.]].

Trema. Trema [Tre 2017] is an OpenFlow controller framework that can be used to create an OpenFlow controller in ruby and C languages. This framework is designed to provide extensibility, includes many libraries and functionalities that are required to interact with the OpenFlow devices.

4.3.2 Data Plane for SD-WLANs. In IEEE 802.11 standard the association decision depends on the clients. The client selects the best AP based on RSSI and initiates the association request. The infrastructure has no control over these decisions made by the client. This underlying problem faced while designing an efficient mobility management and load balancing algorithm is this lack of association control. So to gain the control over clients association with the APs, we need to make changes at the client machine which is not feasible practically.

Figure 4 shows a conceptual architecture of OF enabled AP. The concept Virtual Access Points (VAP) has overcome this problem and has leveraged the association control to the infrastructure. This concept is mainly used, but not limited to, in designing mobility management and load-balancing algorithms. To design a VAP an open programmable data-plane switch with wireless capabilities is required.

We can design an SDN based Wi-Fi AP by installing software on host operating system. The host OS can run on standard general purpose PC or laptop. Some researchers also use mini-pc or Raspberry Pi [ras 2017]. Moreover, these host OSes can be installed on cheap off-the-shelf APs to replace the built-in firmware. Pantou [pan [n. d.]] and OpenFlowClick [Mundada et al. 2009] are some popular examples of software switches. Pantou is based on OpenWRT [ope [n. d.]] which is Linux based distribution for embedded devices. Pantou turns a commercial switch into OpenFlow based switch. Open vSwitch [Pfaff et al. 2009, 2015] also known as OVS is an alternative to Pantou and is considered to be a vital part of SDN network. It can support OpenFlow protocol and thus provide an interface between OF-controller and the Data Plane. It provides an interface for fine-grained control of the forwarding which is used to control QoS, tunneling, and filtering rules etc. Since OVS provides virtual interfaces (VIF) and bind them to physical interfaces (PIF), it enables the migration of configuration state from one VIF to another. Implementation details of OVS can be found at [ovs [n. d.]].

Table 3. Experimental Setup of SD-WLANs

Project	OpenFlow Compatibility	Experimental Environment	OF-Controller	Wireless AP design	Target Problems	Abstraction
Odin [Schulz-Zander et al. 2014b; Suresh et al. 2012]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router on Atheros Card	Mobility Manager, Load Balancing, Automatic Channel Selection	LVAPs
CloudMAC [Vestin et al. 2012; Vestin and Kassler 2015]	OpenFlow	KAUMesh Testbed [Dely and Kassler 2009]	NOX	OpenWRT, VMs with hostapd	Seamless AP switch-off	VAPs
Load-Aware Hand-offs [Rangisetti et al. 2014]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router	Mobility Manager, Load Balancing	LVAPs
COAP [Patro and Banerjee 2014]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router	WiFi Channel Configuration, Context Aware AP Management, Airtime Management	COAP APs
SWAN [Lei et al. 2014]	OpenFlow	Real Testbed	NOX		Seamless Handover, Load Balancing	SAP
AeroFlux [Schulz-Zander et al. 2014a; ?]	OpenFlow	Simulation	Floodlight		Live Video Streaming, Selective IDS	LVAPs
BeHop [Yiakoumis et al. 2014]	OpenFlow	Real Testbed	POX	OVS, Click Modular Router, OpenWRT	Performance Evaluation of Dense WiFi networks	VAPs
SDWN testbed [Luengo et al. 2015]	OpenFlow	Real Testbed	OpenDaylight	OVS, OpenWRT	Testbed design and Evaluation	SDN AP
Ethanol [Moura et al. 2015]	OpenFlow	Real Testbed	POX	Pantou, OpenWRT	Load Aware Client Association, QoS, ARP Overhead	SDN AP
OpenSDWN [Schulz-Zander et al. 2015]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router, OpenWRT	NFV, Virtual Middle Boxes	LVAPs
Programming Abstractions [Riggio et al. 2015a]	OpenFlow	Real Testbed	SD-RAN	OVS, Click Modular Router, OpenWRT	Energy Aware Mobility Management, Uplink Monitoring, Interference Aware Channel Assignment	WTP (LVAPs)
Stiti [Stiti et al. 2015]	OpenFlow	Real Testbed	OpenDaylight	MNetBOX	Seamless Mobility	Virtual AP
Han [Han et al. 2016]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router, OpenWRT	Adaptive Load-Balancing	Virtual AP
Shin [Shin et al. 2016]	OpenFlow	Real Testbed	ONOS	Raspbian OS, OVS, hostapd	Quality of Service	Virtual AP
ISD-WiFi [Tu et al. 2016]	Openflow	Simulation (mininet-wifi)	Ryu	Simulated APs	Mobility Management, Interference Management	Intelligent Center
VALI [Senapathi and Marina 2016]	OpenFlow	Real Testbed	Floodlight	OVS, Click Modular Router, OpenWRT	Public WLAN QoE improvement	Extended LVAP
Yang et al. [Yang et al. 2017]	OpenFlow	Real Testbed	ONOS	OVS, Reapberry Pi	Mobility Management	Mobility Control Module
RFlow+ [Jang et al. 2017]	OpenFlow	Real Testbed	OpenDaylight	OVS, OpenWRT	Network Management	
Gilani et al. [Gilani et al. 2017]	OpenFlow	Real Testbed	Floodlight	OpenWRT	Mobility Management, Load Balancing	Logical APs
[Sequeira et al. 2017]	OpenFlow	Real Testbed	Floodlight	OVS, OpenWRT	Mobility Management	LVAPs
[Luengo et al. 2017]	OpenFlow	Emulation/Real Testbed	OpenDaylight	OVS, OpenWRT	Load Balancing	User Management System
[Carmo et al. 2017]	OpenFlow	Real Testbed	Ryu	OVS, OpenWRT	Resource Slicing	VAP

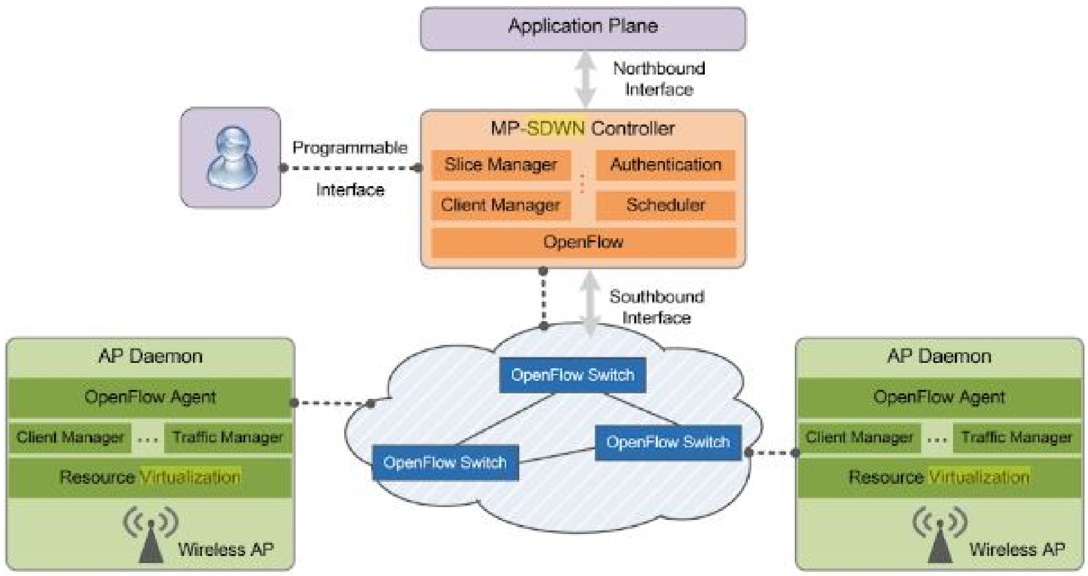

Fig. 4. Conceptual Architecture of OF enabled AP

5 SIGNIFICANT RESEARCH EFFORTS

For the last few years, SD-WLANs has gained a lot of attention from the researchers. Many published research work has focused on identifying the applications of SD-WLANs.

OpenRoads [Yap et al. 2010] is the first project that enabled SD-WLANs. OpenRoads architecture consists of three layers that are physical, slicing and control layers. Physical layer constitutes the OF enabled devices. Control layer is in-charge of configuring these physical devices. NOX controller is used for network orchestration. The slicing layer intercepts the OF packets thus provides the slicing capability so that different network administrators can implement different policies according to there needs on the same physical network.

Suresh et al. developed Odin [Schulz-Zander et al. 2014b; Suresh et al. 2012] which introduced programmability in WLANs. Odin architecture is developed on top of the SDN framework that simplifies the implementation of high-level enterprise WLAN services, such as authentication, authorization, and accounting (AAA). This architecture enables the SDN controller to take control of client's association with the access point. Usually, with the change of network parameters like RSSI the user may want to change its association to another AP. Thus, the last hop connecting a user to the WLAN is varying. One of the key components introduced by Odin framework is the light virtual access points (LVAPs). With these LVAPs, Odin provides programmers with a stable link from users to APs. Every user has its unique BSSID, and a virtual access point is created based on this BSSID and the user is associated with his own Virtual AP. This virtual AP is named as LVAP. A physical AP can host multiple LVAPs. Odin uses Floodlight controller and runs a master on top of this controller as an Application API, and multiple agents are installed on open-flow APs. The southbound interface makes use of a TCP connection to enable communication between the agents and master. Load balancing among APs can be achieved easily using this architecture. Moreover, it is easy to achieve mobility without degrading the QoS with this framework as the user don't have to re-associate with the new assigned AP thus making the hand-off process much faster.

This work has a few limitations such as the load condition AP is not considered while the association and disassociation decisions are made, Odin runs over single controller thus, subject to the single point of failure and prioritizing applications running on a single controller is not possible. The LVAPs can be shifted to any other physical AP within the same subnet and operating on the same channel. Despite its limitations, ODIN framework provided the basis for new techniques to be applied in SD-WLANs. Many research work in the field of SD-WLANs is carried out using LVAP abstraction or is using the idea of virtual access points (VAPs).

CloudMAC [Vestin et al. 2012; Vestin and Kassler 2015] is SD-WLAN based management platform. Like Odin, CloudMAC implements a Split-MAC approach with the added processing of WLAN MAC layer frames using VAPs within a co-located Cloud. The physical APs are thin WLAN APs with the sole purpose to send out IEEE 802.11 MAC layer ACKs to standard WLAN clients and tunnel MAC layer frames towards the VAP via SDN-enabled enterprise WLAN. With experiments, it is proved that this architecture has the same performance as the legacy WLANs. The main contribution of this architecture is opening paths for new SDN-based applications for WLAN networks.

In [Rangisetti et al. 2014], authors leveraged the Odin framework and implemented a load-aware hand-off algorithm over centralized SDN controller. The controller keeps track of the load at each AP. The controller is also aware of the load imposed by the client that needs hand-off. The client is handed-off to the new AP only if the load at new AP is after the hand-off remains within threshold limits. To make the framework load-aware new functionality was added to the existing ODIN framework. The controller gathers the cumulative load of each AP, and also the load of clients. Another important feature introduced in this research work is channel switch announcement (CSA) message which enables the client to hand-off to a different operating channel.

COAP [Patro and Banerjee 2014] is mainly focused on home networks. It is an effort to develop a vendor-neutral controller based framework to manage the home APs with the intent to improve the QoE for home users by mitigating the interference caused by WiFi (neighboring APs) and non-WiFi (e.g., microwave ovens, cordless phones, etc.) devices. Solutions like [Dli [n. d.]] already exists but these proprietary solutions do not leverage the coordination between neighboring APs thus are unable to reduce the interference between them. The Authors suggested that the WiFi is provided by the building management as any other utility like water and electricity so that it has a centrally managed architecture. Thus, enabling the use of SDN controller to enhance the QoE by mitigating interference.

SWAN [Lei et al. 2014] architecture introduce programmability by using SDN to efficiently manage the WLANs which enables seamless hand-offs and load-balancing in WLANs. SWAN architecture makes use of a logically centralized SDN controller, and an agent API installed on each of the physical APs. A set of applications are running on top of the controller to provide the necessary network services. The controller has the global view of the network. A custom protocol is used by the controller to instruct agents to send statistics. MAC layer is divided into controller and agents using wifi split-MAC. This architecture makes use of software AP (SAP) to take control of clients' association with the AP. Every client is associated with its unique SAP, which appears as a regular IEEE 802.11 AP that handles the general IEEE 802.11 handshakes with the UE. Each SAP has a unique BSSID in order to distinguish it from the rest of the SAPs. The clients are logically isolated with respect to IEEE 802.11 MAC. This helps the controller to take control of user association. A physical AP can maintain multiple SAPs. These SAPs can migrate from one physical AP to another while maintaining the association of the clients. If this migration of SAP is made fast enough, the client does not require re-associate to new physical AP. This architecture has been tested for use cases like Seamless Hand-off and Load Balancing.

In AeroFlux [Schulz-Zander et al. 2014a] the authors introduced a way to reduce the latency caused by cloud-based controllers by introducing the near-sighted controllers (NSC) to handle the time-critical events. These NSCs are placed very closed to the data plane. These NSCs control localized events thus providing a quicker response as they are placed close to the APs. The global controller is typically cloud-based and handles the events which require the global view of the network. The GC handles events which inherently global, or not time-critical [Schulz-zander and Schmid 2014] such as, authentication, mobility management, global policy verification and implementation, load-balancing, and network monitoring. In addition, the global controller is also suited to manage middleboxes (MBs, such as firewalls and intrusion detection systems).

[Zhao et al. 2014a] presented SDWLAN to introduce client unaware hand-off. The client is in the illusion of connection to one big AP which helps in seamless mobility between APs, whereas, actually it is communicating with multiple APs. Instead of using an agent to work with standard OF protocol SDWLAN uses extended OF protocol. All the functionalities required to control MAC layer of AP is embedded in the standard OF protocol.

BeHop [Yiakoumis et al. 2014] is a testbed designed by the authors in order to evaluate the functionality of different WiFi network management techniques. Behop is a general purpose framework that can be used for experimental testing of association control, transmission power adjustment, etc. BeHop can be easily integrated into the production network thus providing the benefit of a real-world deployment and giving the flexibility to run experiments. The basic design of BeHop testbed includes commodity APs running OpenWRT and OVS along with some custom WiFi extensions. The APs are connected to the OpenFlow enabled LAN switch. The controller is based on POX OpenFlow controller.

In [Luengo et al. 2015] the authors have designed and implemented a testbed for the software-defined integrated wired-wireless network. In this work, the authors used commodity off-the-shelf networking equipment to design a test-bed that can be used to evaluate different scenarios on SDN based both wired and wireless networks in the real-world traffic environment.

Ethanol [Moura et al. 2015] is proposed as an architecture for dense 1EEE 802.11 networks where the number of APs and clients is very high, e.g. Enterprise networks or campus networks. The basic aim of authors is to design a framework that allows the operators to run services according to the requirements of the network. In addition to forwarding Ethanol controller can handle functions like node mobility, authentication, virtual networking, QoS, and user-localization. The Ethanol architecture comprises two types of devices. One is the controller (POX) that can run on a computer in the wired network or can be cloud-based. Second is Ethanol APs which are commodity wireless routers modified to run Ethanol code (commercial home APs running user-level OpenFlow Software Pantou 1.0). This architecture has been tested for load-aware hand-offs and QoS. Ethanol architecture provides the proof that we can enforce the QoS policies in WLANs using OpenFlow.

OpenSDWN [Schulz-Zander et al. 2015] leverages the LVAP abstraction from Odin and extends it. This architecture makes use of the benefits of SDN and NFV to manage the home and enterprise WiFi networks. OpenSDWN is a unified network architecture that can manage both wired and wireless data paths. Furthermore, by the use of network function virtualization (NFV) the network functions like firewalls and NATs can be deployed flexibly. In addition, this architecture allows operators to specify priorities for applications and flows on wired and wireless portions of the network. OpenSDWN introduces user-specific virtual middleboxes and relies on a Bro Intrusion Detection System (IDS) to identify and classify flows thus enabling service differentiation.

[Sen and Sivalingam 2015] presented a mobility management model with latency quiet lower than the traditional WLANs. The controller shifts the association state of the STA from one AP to another AP as it detects the received power of the STA is less than the set threshold value. In this way, the STA is moved to new AP without the need of re-association and discovery. This eliminates the delays caused by mobility in traditional WLANs.

In [Riggio et al. 2015a] the authors presented high-level programming abstractions for WiFi networks. The proposed model allows the implementation of new features and services as software modules thus hiding away the implementation complexities of the underlying technology. The proposed abstraction was tested by building a basic Mobility Manager as a sample Network App. The Network App is a Python-based module that can be loaded at run-time.

[Sood et al. 2015] has presented a client-side AP association procedure in order to assure the QoS as well as mitigating the greedy user problem. In this scheme, the global controller is responsible to send the network statistics to the client. The local controller in the client device then selects the least loaded AP based on these statistics. If the RSSI of the least loaded AP is not adequate the client will search for the second least loaded AP. This cycle will continue until both the conditions are met and the association will take place. However, making changes or installing a software at the client device is arguable and may not be feasible with the public wireless networks as well as for the guest users in Enterprise networks.

[Stiti et al. 2015] is also an attempt to design SDN based Virtual WAPs (wireless access points) by virtualizing the whole system thus making APs more scalable and flexible while maintaining the performance equivalent to legacy physical APs. The virtual WAP is a virtual machine running on a physical AP. It has virtual interfaces that allow the virtual WAP to communicate with the physical interfaces. Each virtual WAPs has complete isolation and run on same physical AP.

In [Han et al. 2016] authors has designed and implemented an adaptive load balancing technique. This work is a complement to the ODIN framework with the added features of calculating the load on each AP and thus making an intelligent decision of migrating hosts to the new AP with good RSS.

RADIator [Cwalinski and Koenig 2016] provides fine-grained control over wireless traffic. Selective forwarding of the MAC frames is carried out in order to reduce the bandwidth consumption. The RADIator architecture is designed to enable different applications such as Geo-localization, energy optimization, and interference detection and mitigation, etc.

[Shin et al. 2016] used ONOS controller to gain control of the user access and prioritize the network traffic in case of congestion thus making sure the QoS for highly prioritized users.

LegoFi [Schulz-Zander et al. 2016] presents an approach to modularizing the WiFi functions thus enabling the flexibility of deploying these functions where they are needed in the network. Functions like packet deduplication, load-balancing, en/decryption, and mobility management are modularized so that they can be implemented flexibly where needed. The authors argue that all the network functions should be modularized and placed in the network where they are most suited. Some functions are time critical so need to be placed very closed to user end while others may require network-wide view and thus can be placed at the controller or in the cloud.

ISD-WiFi [Tu et al. 2016] is an intelligent architecture for enterprise WLANs addressing mobility management and interference management problems with a flexible architecture to meet the needs of future networks.

VALI [Senapathi and Marina 2016] is an SDN based management framework for the public WLANs. This system makes use of the LVAPs with the added functionality to prioritize the traffic of a certain type whereas providing remaining network resources for other applications.

[Sang et al. 2017] used SDN and 802.11u amendment of IEEE 802.11 protocol to tackle the load balancing problem in WLANs. This paper presents a novel load balancing architecture based on the client-network collaboration model. SDN applications monitor the load on each AP and the information is exchanged between network and clients. A novel algorithm is used for AP selection based on the information gathered.

In [Yang et al. 2017] authors proposed a novel mobility architecture using ONOS (Open Networking Operating System) for IPv6 hosts. ONOS is an SDN OS for the service provider and mission-critical networks. ONOS is specifically built to provide the high availability, scale-out, and performance the networks are demanding, based on the distributed architecture.

RFlow+ [Jang et al. 2017] is a WLAN management and monitoring framework that is capable of supporting both short-term and long-term applications and enforce treatments based on the requirements of applications.

In [Singh and Pandey 2017] authors have proposed a mobility management solution for IP-Based WiFi networks. The proposed solution is capable of providing both layer-2 and layer-3 based mobility management. Authors have utilized Mininet-WiFi to emulate their proposed solution with Floodlight controller.

[Gilani et al. 2017] presented another solution for seamless hand-off management as well as a load-balancing solution for WLAN networks. This solution is not just based on RSSI value to calculate load as in traditional networks. Relying on RSSI value only can lead to unbalanced load in the network. Rather, this solution takes into account the workload of each AP and mobility based and hand-off decisions are taken incorporating the AP load situation.

With the use of VAP, [Sequeira et al. 2017] has developed a framework that allows inter-channel hand-off without any modification at the client side. The hand-off takes place at walking speed.

[Luengo et al. 2017] authors have designed a Wireless User Management system (WUMS)to solve the Load balancing problem reactively as well as provide the mobility management solution. WUMS keeps track of the load on the network. As it detects unbalanced load WUMS looks for under-loaded APs and migrate users to these APs in order to restore the load-balanced condition. Although this system has high efficiency but requires the installation of the application on user devices which is usually not feasible.

[Carmo et al. 2017] presented an entirely different concept for high availability of WLAN network by use of network slicing. With the network virtualization and SDN, the network resources can be shared amongst the general public thus providing enhanced connectivity for the users and unlocking the resources of the WLANs under-utilized networks.

6 DISCUSSION AND OPEN RESEARCH DIRECTIONS

A lot of research efforts have been made in implementation of SD-WLANs. However, a need to design a solution that can replace legacy WLAN is yet not been implemented. Many gaps need to be filled out before fully replacing legacy WLANs with SD-WLANs. The research efforts focus on certain problem and try to solve it. However, a consolidate design that can integrate these solutions in order to combine their strengths yet needs to be build. Like a centralized controller that is combined with data plane programmability and network function virtualization to support better resource allocation along with energy efficiency, best channel selection etc. Moreover, the proposed solutions are not sufficiently evaluated due to lack of data and benchmarks. Designing better experimentation tools and simulation models for SD-WLANs can help us better understand the properties of solutions designed and will enhance the development of better solutions.

As for home networks, an architecture can be designed in which subscriber of an Internet service can attain roaming. All the APs can be remotely managed by the controller of Internet service provider (ISP) thus allowing the ISP to manage all the APs. The ISP can thus authenticate the users even if they are out of their homes in the coverage area of another AP managed by the same ISP. Thus a user will have access to the Internet via Wi-Fi using his own account. This architecture can also help in mitigating the interference from neighboring APs provided they are controlled by the same controller (may or may not be by same ISP).

Moreover, on one hand, SDN has promised not only to solve existing network issues but to also allow new techniques and methods for more advanced networks. While on the other hand, has also introduced new SDN related issues.

Furthermore, the concept of Knowledge Defined Networking (KDN) has vast benefits as have been pointed out in [Mestres et al. 2017]. In our view, WLANs can also gain vast benefits from self-adapting and self-optimizing networks. A few proposals like [López-Raventós et al. 2018] are a proof of concept. Big data mining and Machine Learning has the ability to collect huge amount of information that can in-turn be used to improve network performance and management.

7 CONCLUSIONS

Legacy WLAN networks are vertically coupled and are highly inflexible. The emerging SDN paradigm holds a great promise to break down this vertical integration and thus providing the necessary flexibility. In this paper, we have presented a brief survey on the benefits that WLANs can achieve by adopting SDN. The new paradigms, such as SD-WLANs, are emerging aiming to fulfill the requirements of future networks. We have, in detail, discussed the requirements and implementation details of SDN with all the components required for designing the SD-WLAN based network. Further, we have discussed the on going research efforts to implement SD-WLANs in order to solve the problems that limits the usability of WLANs.

REFERENCES

[n. d.]. DIR-826L Cloud Gigabit Router N600 _ D-Link UK. http://www.dlink.com/uk/en/products/dir-826l-cloud-gigabit-router-n600

[n. d.]. Open Networking Foundation is an operator led consortium leveraging SDN, NFV and Cloud technologies to transform operator networks and business models. https://www.opennetworking.org/

[n. d.]. Open vSwitch. http://openvswitch.org/

[n. d.]. OpenWRT. https://openwrt.org/

[n. d.]. Pantou: OpenFlow-1.3-for-OpenWRT. https://github.com/CPqD/ofsoftswitch13/wiki/OpenFlow-1.3-for-OpenWRT

2017. NS-3. http://ns3-code.com/ns3-sdn-projects/

2017. Raspberry Pi. https://www.raspberrypi.org/

2017. Trema. https://trema.github.io/trema/

Ed. A. Doria, Ed. J. Hadi Salim, Ed. R. Haad, Ed. H. Khosravi, Ed. W. Wang, L. Dong, R. Gopal, and J. Halpern. 2010. *Forwarding and Control Element Separation (ForCES) Protocol Specification.* Technical Report Category: Standards Track. 1–124 pages.

Rodolfo Alvizu, Guido Maier, Navin Kukreja, Achille Pattavina, Roberto Morro, Alessandro Capello, and Carlo Cavazzoni. 2017. Comprehensive survey on T-SDN: Software-defined Networking for Transport Networks. *IEEE Communications Surveys and Tutorials* 19, 4 (2017), 2232–2283. https://doi.org/10.1109/COMST.2017.2715220

Fetia Bannour, Sami Souihi, and Abdelhamid Mellouk. 2017. Distributed SDN Control: Survey, Taxonomy and Challenges. *IEEE Communications Surveys & Tutorials* c (2017), 1–1. https://doi.org/10.1109/COMST.2017.2782482

BellTechlogix. [n. d.]. THE REAL BENEFITS OF BYOD. ([n. d.]).

Nikos Bizanis and Fernando A. Kuipers. 2016. SDN and Virtualization Solutions for the Internet of Things: A Survey. *IEEE Access* 4 (2016), 5591–5606. https://doi.org/10.1109/ACCESS.2016.2607786

Maxweel S. Carmo, Sandino Jardim, Thalyson De Souza, Augusto V. Neto, Rui Aguiar, and Daniel Corujo. 2017. Towards enhanced connectivity through WLAN slicing. *Wireless Telecommunications Symposium* (2017). https://doi.org/10.1109/WTS.2017.7943550

Claude Chaudet and Yoram Haddad. 2013. Wireless software defined networks: Challenges and opportunities. In *2013 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems, COMCAS 2013.* 21–23. https://doi.org/10.1109/COMCAS.2013.6685237

Jiangyong Chen, Xianghan Zheng, and Chunming Rong. 2015. Survey on software-defined networking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9106, 1 (2015), 115–124. https://doi.org/10.1007/978-3-319-28430-9_9

CISCO. 2015. *The Zettabyte Era: Trends and Analysis.* Technical Report May. 1–29 pages. https://doi.org/1465272001812119

Cisco. 2017. Cisco Meraki. https://meraki.cisco.com/

Cisco Systems. 2017. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016âĂŞ2021 White Paper. (2017), 2016–2021 pages. https://doi.org/1465272001663118 arXiv:1454457600809267

Salvatore Costanzo, Laura Galluccio, Giacomo Morabito, and Sergio Palazzo. 2012. Software defined wireless networks: Unbridling SDNs. In *Proceedings - European Workshop on Software Defined Networks, EWSDN 2012.* 1–6. https://doi.org/10.1109/EWSDN.2012.12

Ar Andrew R Curtis, Jeffrey C Jc Jeffrey C Mogul, Jean Tourrilhes, Praveen Yalagandula, Puneet Sharma, and Sujata Banerjee. 2011b. DevoFlow: scaling flow management for high-performance networks. In *Proceedings of the {ACM} {SIGCOMM} 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Toronto, {ON}, Canada, August 15-19, 2011*. 254–265. https://doi.org/10.1145/2043164.2018466

Andrew R. Curtis, Wonho Kim, and Praveen Yalagandula. 2011a. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. In *Proceedings - IEEE INFOCOM*. 1629–1637. https://doi.org/10.1109/INFCOM.2011.5934956

Radoslaw Cwalinski and Hartmut Koenig. 2016. RADIator - An approach for controllable wireless networks. In *IEEE NETSOFT 2016 - 2016 IEEE NetSoft Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services*. 260–268. https://doi.org/10.1109/NETSOFT.2016.7502421

Tooska Dargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, and Mauro Conti. 2017. A Survey on the Security of Stateful SDN Data Planes. *IEEE Communications Surveys and Tutorials* 19, 3 (2017), 1701–1725. https://doi.org/10.1109/COMST.2017.2689819

Rogerio Leao Santos De Oliveira, Christiane Marie Schweitzer, Ailton Akira Shinoda, and Ligia Rodrigues Prete. 2014. Using Mininet for emulation and prototyping Software-Defined Networks. *2014 IEEE Colombian Conference on Communications and Computing, COLCOM 2014 - Conference Proceedings* (2014). https://doi.org/10.1109/ColComCon.2014.6860404

Peter Dely and Andreas Kassler. 2009. KAUMesh Demo : A MultiRadio MultiChannel Mesh Testbed. http://www.cs.kau.se/~pdely/papers/meshdemo1.pdf

Behnam Dezfouli, Vahid Esmaeelzadeh, Jaykumar Sheth, and Marjan Radi. 2018. A Review of Software-Defined WLANs: Architectures and Central Control Mechanisms. *IEEE Communications Surveys and Tutorials* July (2018), 1–33. https://doi.org/10.1109/COMST.2018.2868692 arXiv:1809.00121

Ozgur Ercetin. 2008. Association Games in IEEE 802.11 Wireless Local Area Networks. *Wireless Communications, IEEE Transactions on* 7, 12 (2008), 5136–5143. https://doi.org/10.1109/T-WC.2008.071418

Estinet-Technologies. 2013. EstiNet Network Simulator and Emulator. September (2013), 110–117. http://www.estinet.com/products.php?lv1=1{&}sn=2

Dave Evans. 2011. *The Internet of Things - How the Next Evolution of the Internet is Changing Everything*. Technical Report April. 1–11 pages. https://doi.org/10.1109/IEEESTD.2007.373646 arXiv:arXiv:1011.1669v3

Ramon R. Fontes, Samira Afzal, Samuel H.B. Brito, Mateus A.S. Santos, and Christian Esteve Rothenberg. 2015. Mininet-WiFi: Emulating software-defined wireless networks. In *Proceedings of the 11th International Conference on Network and Service Management, CNSM 2015*. 384–389. https://doi.org/10.1109/CNSM.2015.7367387

Fortinet. 2017. Fortinet. https://www.fortinet.com/

Open Networking Foundation. 2015. Principles and Practices for Securing Software - Defined Networks. January (2015).

Syed Mushhad M. Gilani, Tang Hong, Wenqiang Jin, Guofeng Zhao, H. Meng Heang, and Chuan Xu. 2017. Mobility management in IEEE 802.11 WLAN using SDN/NFV technologies. *EURASIP Journal on Wireless Communications and Networking* 2017, 1 (2017), 67. https://doi.org/10.1186/s13638-017-0856-9

Fabrizio Granelli, Anteneh A. Gebremariam, Muhammad Usman, Filippo Cugini, Veroniki Stamati, Marios Alitska, and Periklis Chatzimisios. 2015. Software defined and virtualized wireless access in future wireless networks: Scenarios and standards. *IEEE Communications Magazine* 53, 6 (2015), 26–34. https://doi.org/10.1109/MCOM.2015.7120042

Natasha Gude, Nicira Networks, Teemu Koponen Hiit, Justin Pettit, Ben Pfaff, Martín Casado, Nick Mckeown, and Scott Shenker. 2008. NOX: Towards an Operating System for Networks. In *ACM SIGCOMM Computer Communication Review*, Vol. 38. 105 – 110.

Y. Han, K. Yang, X. Lu, and D. Zhou. 2016. An adaptive load balancing application for software-defined enterprise WLANs. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. 281–286. https://doi.org/10.1109/ICTC.2016.7763486

Israat Tanzeena Haque and Nael Abu-Ghazaleh. 2016. Wireless Software Defined Networking: A Survey and Taxonomy. *IEEE Communications Surveys and Tutorials* 18, 4 (2016), 2713–2737. https://doi.org/10.1109/COMST.2016.2571118

Sofia Naning Hertiana, Adit Kurniawan, Hendrawan, and Udjianna Sekteria Pasaribu. 2017. Path Associativity Centralized Explicit Congestion Control (PACEC) for SDN. In *3rd International Conference on Control, Electronics, Renewable Energy, and Communications 2017*. 18–23.

Mitsunobu Homma and Norihiko Shinomiya. 2017. Cycle-based traffic load balancing method for clustered network with multiple SDN controllers. *TENCON 2017 - 2017 IEEE Region 10 Conference* (2017), 1843–1846.

Tao Huang, F. Richard Yu, Chen Zhang, Jiang Liu, Jiao Zhang, and Yunjie Liu. 2017. A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. *IEEE Communications Surveys and Tutorials* 19, 2 (2017), 891–917. https://doi.org/10.1109/COMST.2016.2630047

RhongHo Jang, DongGyu Cho, Youngtae Noh, and DaeHun Nyang. 2017. RFlow+: An SDN-based WLAN monitoring and management framework. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

Amit P. Jardosh, Konstantina Papagiannaki, Elizabeth M. Belding, Kevin C. Almeroth, Gianluca Iannaccone, and Bapi Vinnakota. 2009. Green WLANs: On-demand WLAN infrastructures. *Mobile Networks and Applications* 14, 6 (2009), 798–814. https://doi.org/10.1007/s11036-008-0123-8

Faris Keti and Shavan Askar. 2015. Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. In *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, Vol. 2015-Octob. 205–210. https://doi.org/10.1109/ISMS.2015.46

M. A. Khan, S. Leng, Wang Xiang, and Kun Yang. 2015. Architecture of heterogeneous wireless access networks: A short survey. In *TENCON 2015 - 2015 IEEE Region 10 Conference*. 1–6. https://doi.org/10.1109/TENCON.2015.7372905

Diego Kreutz, Fernando M V Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proc. IEEE* 103, 1 (jan 2015), 14–76. https://doi.org/10.1109/JPROC.2014.2371999 arXiv:1406.0440

Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A network in a laptop. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks - Hotnets '10*. 1–6. https://doi.org/10.1145/1868447.1868466

Hwi Young Lee, Young Min Kwon, Jung Wan Shin, Won Jin Lee, and Min Young Chung. 2016. High Availability WLANs Based on Software-Defined Networking. In *AICT 2016 : The Twelfth Advanced International Conference on Telecommunications High*. 24–28.

T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang. 2014. SWAN: An SDN based campus WLAN framework. In *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)*. 1–5. https://doi.org/10.1109/VITAE.2014.6934489

L. Li and Q. Xu. 2017. Load balancing researches in SDN: A survey. In *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*. 403–408. https://doi.org/10.1109/ICEIEC.2017.8076592

Linux Foundation Project. 2017. OPENDAYLIGHT. https://www.opendaylight.org/

Álvaro López-Raventós, Francesc Wilhelmi, Sergio Barrachina-Muñoz, and Boris Bellalta. 2018. Machine Learning and Software Defined Networks for High-Density WLANs. *CoRR* abs/1804.05534 (2018).

Eduardo Luengo, Shahram ShahHeydari, and Khalil El-Khatib. 2017. SDN-based wireless user management system. *Telecommunication Systems* 65, 4 (2017), 755–769. https://doi.org/10.1007/s11235-016-0264-1

Eduardo Luengo, Christopher Weber, Shahram Shah Heydari, and Khalil El-Khatib. 2015. Design and Implementation of a Software-Defined Integrated Wired-Wireless Network Testbed. In *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access*. 47–52. https://doi.org/10.1145/2810362.2826697

Murphy McCauley. [n. d.]. POX. https://github.com/noxrepo/pox/

Nick Mckeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. In *ACM SIGCOMM Computer Communication Review*, Vol. 38. ACM, 69–74.

Albert Mestres, Alberto Rodriguez-natal, Josep Carner, Eduard Alarcn, Marc Sol, Victor Munts-mulero, Sharon Barkai, Mike J Hibbett, Giovani Estrada, Florin Coras, Vina Ermagan, Hugo Latapie, Chris Cassar, John Evans, Fabio Maino, Jean Walrand, Albert Cabellos, Albert Mestres, Alberto Rodriguez-natal, Josep Carner, Eduard Alarcn, Marc Sol, Victor Munts-mulero, Sharon Barkai, Mike J Hibbett, Giovani Estrada, Florin Coras, Vina Ermagan, Hugo Latapie, and Chris Cassar. 2017. Knowledge-Defined Networking. In *ACM SIGCOMM Computer Communication Review*, Vol. 47.

Oliver Michel and Eric Keller. 2017. SDN in wide-area networks: A survey. In *2017 4th International Conference on Software Defined Systems, SDS 2017*. 37–42. https://doi.org/10.1109/SDS.2017.7939138

Henrique Moura, Gabriel V C Bessa, Marcos A M Vieira, and Daniel F Macedo. 2015. Ethanol: Software Defined Networking for 802.11 Wireless Networks. In *IFIP/IEEE*. 388–396. https://doi.org/10.1109/INM.2015.7140315

Yogesh Mundada, Rob Sherwood, Nick Feamster, and Georgia Tech. 2009. An OpenFlow Switch Element for Click. In *Architecture*. 1–1. https://doi.org/10.1.1.214.1115

Gilbert N. Nde and Rahamatullah Khondoker. 2016. SDN testing and debugging tools: A survey. In *2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016*. 631–635. https://doi.org/10.1109/ICIEV.2016.7760078

The Open Networking Lab (ON.Lab). 2014. *Introducing ONOS - a SDN network operating system for Service Providers !* Technical Report. 14 pages.

Farzaneh Pakzad, Siamak Layeghy, and Marius Portmann. 2017. Evaluation of Mininet-WiFi integration via ns-3. *26th International Telecommunication Networks and Applications Conference, ITNAC 2016* (2017), 243–248. https://doi.org/10.1109/ATNAC.2016.7878816

Ashish Patro and Suman Banerjee. 2014. COAP: A Software-defined Approach for Home WLAN Management Through an Open API. *Proceedings of the 9th ACM Workshop on Mobility in the Evolving Internet Architecture* (2014), 31–36. https://doi.org/10.1145/2645892.2645902

Ben Pfaff and Bruce Davie. 2013. The open vSwitch database management protocol. (2013).

Ben Pfaff, Justin Pettit, Keith Amidon, Martin Casado, Teemu Koponen, and Scott Shenker. 2009. Extending Networking into the Virtualization Layer.. In *Hotnets*.

Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, Keith Amidon, Awake Networks, and Martín Casado. 2015. The Design and Implementation of Open vSwitch. In *Nsdi*. 117–130. https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/pfaff

Project Floodlight. [n. d.]. Project Floodlight. http://www.projectfloodlight.org/floodlight/

Anil Kumar Rangisetti, Baldaniya Hardik Bhopabhai, B. Pradeep Kumar, and Bheemarjuna Reddy Tamma. 2014. Load-aware hand-offs in software defined wireless LANs. In *International Conference on Wireless and Mobile Computing, Networking and Communications*. 685–690. https://doi.org/10.1109/WiMOB.2014.6962245

Daniel Raumer, Lukas Schwaighofer, and Georg Carle. 2014. MonSamp: an SDN Application for QoS Monitoring. In *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, Vol. 2. 961–968. https://doi.org/10.15439/2014F175

Roberto Riggio, Karina Mabell Gomez, Tinku Rasheed, Julius Schulz-Zander, Slawomir Kuklinski, and Mahesh K. Marina. 2015a. Programming Software-Defined wireless networks. In *Proceedings of the 10th International Conference on Network and Service Management, CNSM 2014*. 118–126. https://doi.org/10.1109/CNSM.2014.7014148

Roberto Riggio, Mahesh K. Marina, Julius Schulz-Zander, Slawomir Kuklinski, and Tinku Rasheed. 2015b. Programming Abstractions for Software-Defined Wireless Networks. *IEEE Transactions on Network and Service Management* 12, 2 (2015), 146–162. https://doi.org/10.1109/TNSM.2015.2417772 arXiv:1411.2852v1

Ryu. [n. d.]. Ryu. http://ryu.readthedocs.io/en/latest/index.html

Xiaokang Sang, Qian Wu, and Hewu Li. 2017. Client-network collaborative load balancing mechanism for WLAN based on SDN and 802.11u. In *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017*. 506–511. https://doi.org/10.1109/IWCMC.2017.7986337

Julius Schulz-Zander, Carlos Mayer, Bogdan Ciobotaru, Stefan Schmid, and Anja Feldmann. 2015. OpenSDWN: Programmatic Control over Home and Enterprise WiFi. In *Sosr*. 16:1–16:12. https://doi.org/10.1145/2774993.2775002

Julius Schulz-Zander, Nadi Sarrar, and Stefan Schmid. 2014a. AeroFlux: A Near-Sighted Controller Architecture for Software-Defined Wireless Networks. In *Presented as part of the Open Networking Summit 2014 (ONS 2014)*. USENIX, Santa Clara, CA.

Julius Schulz-zander and Stefan Schmid. 2014. Towards a Scalable and Near-Sighted Control Plane Architecture for WiFi SDNs. In *HotSDN 2014*. 217–218. https://doi.org/10.1145/2620728.2620772

Julius Schulz-Zander, Stefan Schmid, James Kempf, Roberto Riggio, and Anja Feldmann. 2016. LegoFi the WiFi building blocks! The case for a modular WiFi architecture. *2016 ACM Workshop on Mobility in the Evolving Internet Architecture, MobiArch 2016* 03-07-Octo (2016), 7–12. https://doi.org/10.1145/2980137.2980142

Julius Schulz-Zander, Lalith Suresh, Nadi Sarrar, and Anja Feldmann. 2014b. Programmatic Orchestration of WiFi Networks. In *USENIX ATC*. 347–358. https://doi.org/10.1109/JPROC.2014.2371999 arXiv:arXiv:1011.1669v3

Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. 2013. SDN security: A survey. *SDN4FNS 2013 - 2013 Workshop on Software Defined Networks for Future Networks and Services* (2013). https://doi.org/10.1109/SDN4FNS.2013.6702553

SDxCentral. 2017. Network Virtualization Report: SDN Controllers, Cloud Networking and More - Online Edition. https://www.sdxcentral.com/reports/2017/network-virtualization-sdn-controller/chapter-3-network-virtualization-adoption-and-vendor-landscape/

Arkadeep Sen and Krishna M. Sivalingam. 2015. An SDN framework for seamless mobility in enterprise WLANs. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC* 2015-Decem (2015), 1985–1990. https://doi.org/10.1109/PIMRC.2015.7343624 arXiv:arXiv:1011.1669v3

Sivaprakash Senapathi and Mahesh K Marina. 2016. Poster : VALI âĂŞ An SDN-based Management Framework for Public Wireless LANs. In *Mobicom 2016*. 443–444.

Luis Sequeira, Juan Luis de la Cruz, José Ruiz-Mas, Jose Saldana, Julián Fernandez-Navajas, and José Almodovar. 2017. Building an SDN enterprise WLAN based on virtual APs. *IEEE Communications Letters* 21, 2 (2017), 374–377.

Sakir Sezer, Sandra Scott-Hayward, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Cisco Systems Jim Finnegan, Niel Viljoen, Netronome Marc Miller, Navneet Rao, and Scott-hayward Layout. 2013. INTRODUCTION: WHAT IS SOFTWARE-DEFINED NETWORKING? FUTURE CARRIER NETWORKS Are We Ready for SDN? Implementation Challenges for Software-Defined Networks BACKGROUND: WHY SDN? *Future Carrier Networks* 51, 7 (2013), 36–43. https://doi.org/10.1109/MCOM.2013.6553676 arXiv:arXiv:1406.0440v1

Jung Wan Shin, Hwi Young Lee, Won Jin Lee, and Min Young Chung. 2016. Access control with ONOS controller in the SDN based WLAN testbed. *International Conference on Ubiquitous and Future Networks, ICUFN* 2016-Augus (2016), 656–660. https://doi.org/10.1109/ICUFN.2016.7537115

Myung Ki Shin, Ki Hyuk Nam, and Hyoung Jun Kim. 2012. Software-defined networking (SDN): A reference architecture and open APIs. In *International Conference on ICT Convergence*. 360–361. https://doi.org/10.1109/ICTC.2012.6386859 arXiv:arXiv:1011.1669v3

Krishna Vijay Kumar Singh and Mayank Pandey. 2017. Software-defined mobility in IP based Wi-Fi networks: Design proposal and future directions. *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems, ANTS 2016* (2017). https://doi.org/10.1109/ANTS.2016.7947808

IEEE Computer Society. 2010. *IEEE Standard for information technology – Amendment 6: wireless access in vehicular environments*. 1–51 pages.

Haoyu Song. 2013. Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane. In *HotSDN '13*. 127–132. https://doi.org/10.1145/2491185.2491190 arXiv:arXiv:1405.0060v1

W.K. Soo, T.-C. Ling, A.H. Maw, and S.T. Win. 2018. Survey on load-balancing methods in 802.11 infrastructure mode wireless networks for improving quality of service. *Comput. Surveys* 51, 2 (2018). https://doi.org/10.1145/3172868

Keshav Sood, Shigang Liu, Shui Yu, and Yong Xiang. 2015. Dynamic Access Point Association Using Software Defined Networking. 226–231.

Dorothy Stanley, Pat Calhoun, and Michael Montemurro. 2009. Control and provisioning of wireless access points (CAPWAP) protocol specification. (2009).

Oussama Stiti, Othmen Braham, and Guy Pujolle. 2015. Virtual openflow-based SDN Wi-Fi access point. In *2015 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 1–3. https://doi.org/10.1109/GIIS.2015.7347190

Junho Suh, Ted Taekyoung Kwon, Colin Dixon, Wes Felter, and John Carter. 2014. OpenSample: A low-latency, sampling-based measurement platform for commodity SDN. In *Proceedings - International Conference on Distributed Computing Systems*. 228–237. https://doi.org/10.1109/ICDCS.2014.31

Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. 2012. Towards programmable enterprise WLANs with Odin. In *HotSDN'12*. 115–120. https://doi.org/10.1145/2342441.2342465

Amin Tootoonchian and Yashar Ganjali. 2010. HyperFlow: A Distributed Control Plane for OpenFlow. In *Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking (INM/WREN'10)*. USENIX Association, Berkeley, CA, USA, 3–3.

Celio Trois, Marcos D. Del Fabro, Luis C.E. De Bona, and Magnos Martinello. 2016. A Survey on SDN Programming Languages: Toward a Taxonomy. *IEEE Communications Surveys and Tutorials* 18, 4 (2016), 2687–2712. https://doi.org/10.1109/COMST.2016.2553778

Daikai Tu, Zhifeng Zhao, and Honggang Zhang. 2016. ISD-WiFi: An intelligent SDN based solution for enterprise WLANs. *2016 8th International Conference on Wireless Communications and Signal Processing, WCSP 2016* (2016), 1–6. https://doi.org/10.1109/WCSP.2016.7752738

Jonathan Vestin, Peter Dely, Andreas Kassler, Nico Bayer, Hans Einsiedler, and Christoph Peylo. 2012. ACM MobiCom 2012 Poster : CloudMAC - Towards Software Defined WLANs. *Mobile Computing and Communications Review* 16, 4 (2012), 42—-45. https://doi.org/10.1145/2436196.2436217

Jonathan Vestin and Andreas Kassler. 2015. QoS Management for WiFi MAC Layer Processing in the Cloud: Demo Description. *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks* (2015), 173–174. https://doi.org/10.1145/2815317.2815343

Yunjung Yang, Seil Jeon, Daecheon Kim, and Hyunseung Choo. 2017. Design and implementation of SDN-based lightweight IPv6 mobility manager in ONOS. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication - IMCOM '17*. 1–5. https://doi.org/10.1145/3022227.3022333

Kok-Kiong Yap, Rob Sherwood, Masayoshi Kobayashi, Te-Yuan Huang, Michael Chan, Nikhil Handigol, Nick Mckeown, and Guru Parulkar. 2010. Blueprint for Introducing Innovation into Wireless Mobile Networks. In *ACM VISA*. 25—-32.

Yiannis Yiakoumis, Manu Bansal, Adam Covington, Johan van Reijendam, Sachin Katti, and Nick McKeown. 2014. BeHop: a testbed for dense WiFi networks. In *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH '14*, Vol. 18. 1–8. https://doi.org/10.1145/2643230.2643233

Minlan Yu, Lavanya Jose, and R Miao. 2013. Software defined traffic measurement with opensketch. In *... 10th USENIX Symposium on Networked Systems ...*. 29–42.

Dong Zhao, Ming Zhu, and Ming Xu. 2014a. SDWLAN: A flexible architecture of enterprise WLAN for client-unaware fast AP handoff. In *5th International Conference on Computing Communication and Networking Technologies, ICCCNT 2014*. https://doi.org/10.1109/ICCCNT.2014.6963100

Dong Zhao, Ming Zhu, and Ming Xu. 2014b. Supporting 'One Big AP' illusion in enterprise WLAN: An SDN-based solution. In *2014 6th International Conference on Wireless Communications and Signal Processing, WCSP 2014*. https://doi.org/10.1109/WCSP.2014.6992121