Traffic Aware Service Differentiation in Software Defined Wi-Fi Networks for Health care

Dr. Sohaib Manzoor[0000−0002−9880−6090]

Electrical Engineering Department
Mirpur University of Science and Technology, MIrpur AJK, Pakistan, 10250
{sohaib.ee}@must.edu.pk

Abstract. The IEEE 802.11 wireless local area networks (WLANs) have grown rapidly in the past decade causing difficulties for the network administrator to handle the huge amount of flow requests. This is due to the simple WiFi network architecture implementation, low costs and high bandwidth link capacity. A new network paradigm is introduced recently to overcome the afore mentioned issues, which is software defined networks (SDN). In SDN the data and control plane are separated and the network management is done by the centralized controller rather than the transfer devices in the data plane. To maximize the efficiency and reliability of the network resources, optimization techniques are needed so that, there is no burden on any single device and the workload is balanced among all the network devices.

This study focuses on optimizing load balancing in software defined WiFi networks for healthcare. In order to understand WiFi networks, optimization of IEEE 802.11 distributed coordination function (DCF) is performed, which addresses an open issue to make the RTS threshold parameter self adaptive. The study further deals with the understanding of the wireless load definition in a software defined WiFi network (SD-WiFi). Proper tunning of the system parameters is performed to avoid contention in wireless transmission. Finally combining the optimization techniques of WiFi and wireless load, load balancing in control and data plane is achieved. In the control plane we take a distributed approach where the controllers are placed in a two-tier arrangement. In the data plane, the load on the wireless access points (APs) is balanced by designing a suitable algorithm through OpenFlow. Handover times are also reduced while balancing the load among the APs, through a software defined approach. Performance evaluation is supposed to be obtained by performing simulations in NS-3, OMNeT++ and also by designing a Linux based testbed. In order to understand the tradeoffs between latency, throughput, workload handled by the controllers, entropy and fairness among APs, the comparison will be made with previous optimization techniques such as flow stealer, switch migration, and studies using testbed prototypes. We are hopeful to utilize the proposed scheme in order to achieve improved quality-of-service for real time applications on cloud based high density software defined WiFi end architecture with heavy amount of loading.

Keywords: WiFi, SDN, Optimization, APs, IEEE 802.11 DCF, Healthcare, Load Balancing, Handovers, NS-3, OMNeT++, Testbed

Introduction

In the past decades the WiFi networks have been considered as the key access networks and due to this reason, there has been a rapid growth in the mobile held devices causing serious pressure on network operators regarding performance satisfaction. In WiFi an important protocol in the medium access control (MAC) is the distributed coordination function (DCF). DCF has two way handshake which is the basic access and the four way handshake which is the RTS/CTS mechanism. Recently Dai has proposed a unified analytical framework for IEEE 802.11 DCF [1] which is different from what Bianchi proposed in 2000 [2]. The difference is based on the Markov model that is employed by Dai to study the behavior of head of line (HOL) packets. The simplicity and the power of the model has been checked through NS-2.

RTS/CTS requires an exchange of request to send and clear to send messages in prior to reserve the channel as opposed to the basic access. The exchange of messages may degrade the network performance for small packet sizes. A RTS threshold parameter is required that switches to the appropriate network mode without deteriorating the performance. Many studies have been done but the various network assumptions make it difficult to switch to an appropriate mode.

Researchers perform experiments using simulations, emulation and testbed tools in order to study the behavior of real time systems. To-date there are many performance evaluation tools used by students and researchers. The comparison among the tools is shown in Fig.1. The simulators are the least realistic in compared to testbed tools such as PlanetLab. NS-3 and OMNeT++ are the emerging simulators but very few studies have considered there PHY/MAC layer validation due the complexity involved.

Researchers have used different definitions for wireless load in the literature. The load has been approximated from the channel access delay in the unsaturated network [3]. The researcher have also considered the time intensity, the measure of time when a certain AP remains busy for a particular time [4]. The most common definition of load used by the researchers is the number of wireless devices connected to a certain AP, but this cannot also characterize load as the load condition for different wireless stations may be different. So there is a dire need to understand load and the parameters that make load different for different wireless stations.

In the traditional networks the route in the network is learned by the protocols such as Open Shortest Path First (OSPF) or Extended Interior Gateway Routing Protocol (EIGRP). When the route is learned, the node makes flow tables and then takes forwarding decisions. Apart from these distributed protocols the nodes such as switches and routers also run protocols such as Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LADP) that are useful to in showing the information of the neighbor nodes. The control decisions among the nodes is distributed. The control plane distribution makes the net-

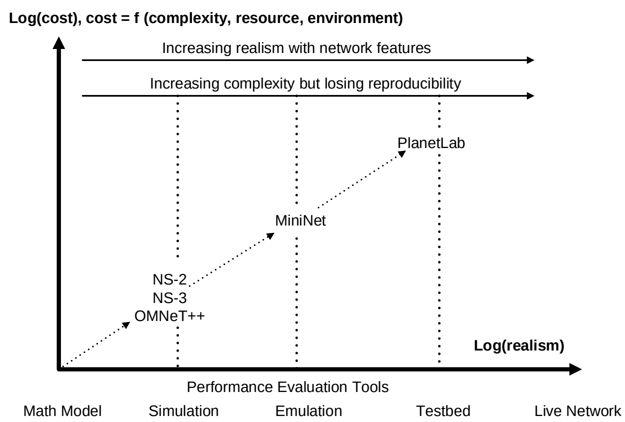


Fig. 1: Performance evaluation tools.

work petrified and this results in lack of modularization and abstraction layer in the network, which discourages the programmability and leads to no innovation of the network. Moreover, the vendor-oriented devices are not supporting the deployment of new services and only limited services can be used.

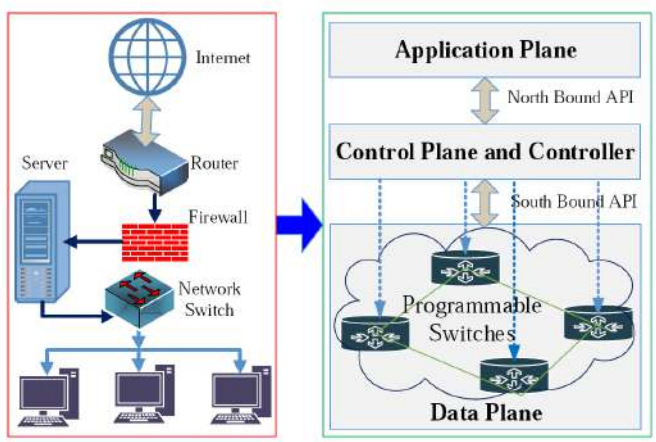


Fig. 2: Comparison between traditional networks and SDN framework.

Software Defined Networking (SDN) [5] is a new network paradigm that have goals to manage the network conveniently with no hardware modifications at the client side. The sole purpose of SDN is to transform the complex network architecture into basic, simple and manageable blocks. Recent studies have shown that the traditional networks are not capable of handling the increased demands as all components that are vertically integrated thus making a very complex architecture which is hard to understand and manage [6] [7] [8] [9]. The traditional networks only follow the vendor specified policies and are not flexible for a dynamic environment [6] [8] [9]. Fig.2 shows the comparison between the traditional WiFi networks and the SDN framework.

The prime entities in a SDN framework are controller and switches. Controllers are placed in the control plane and take control of the network and control decisions where as switches are placed in the data plane and work according to the decisions made by the controller. In the traditional network when there is a system update then whole of the system needs to be configured where as in the SDN only a software update is required [10]. The architecture of SDN is presented in Fig.3. There are three planes: application plane, control plane and data plane.

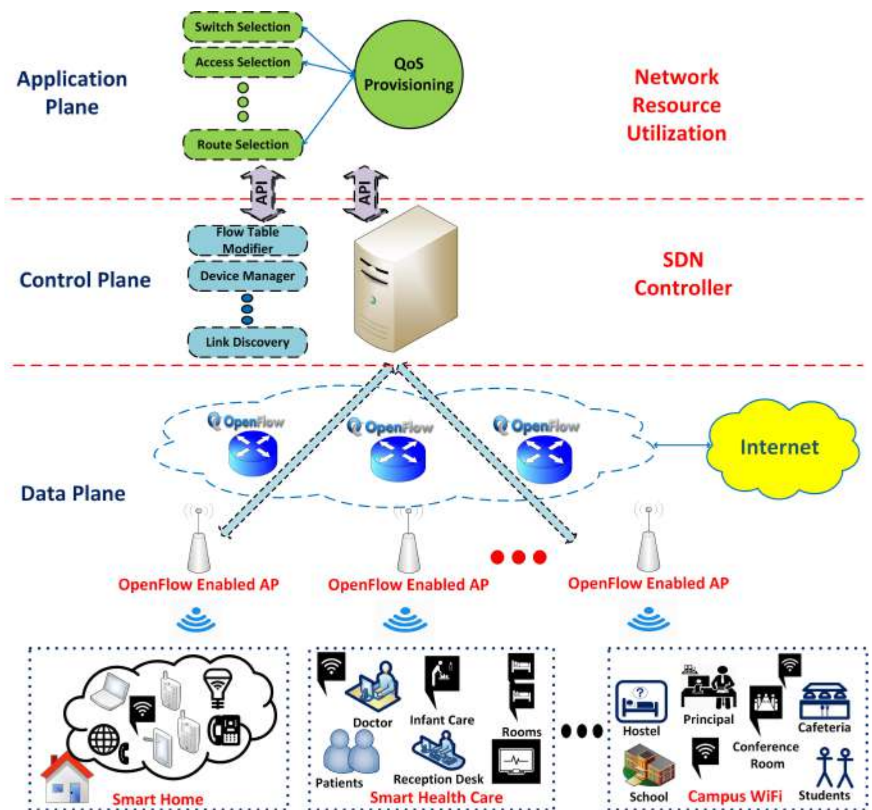


Fig. 3: SDN architecture.

The data plane consists of devices such as access points, switches and routers. Virtual switches such as Indigo, Pica8, Nettle, OpenFlow and physical switches coexist in this layer [9] [10] [11] [12]. The main objective of the data plane is to forward the packets according to assigned rules and policies. In the control plane resides a controller that controls the overall functions of SDN. This layer acts as a bridge between the application and data/infrastructure layer. The control layer is responsible to manage the traffic flow and have certain duties such as routing, forwarding of flows, and dropping of packets through programming [13] [14]. In the distributed environment these controllers communicate with each other through east bound and west bound interfaces. The control layer and data plane communicate with each other through south-bound application programming interface (API) such as OpenFlow. The topmost layer is the application plane. Network virtualization, intrusion detection System, intrusion prevention system, mobility management, load balancing and firewall implementation are some examples of applications handled by this plane. This layer communicates with the control layer through northbound API [12].

SDN can bring solution to many problems existing in a WiFi network. These issues include load balancing, mobility management, resource discovery allocation, control signaling, management and operations of a network and security. There also some non-technical issues such as law and regulations etc. [15]. In comparison to the wired network, where the bandwidth is allocated according to the hardware basis, it is difficult to control the behavior of the WiFi network. The problems exist due to the stochastic fluctuation of wireless channel and broadcast nature of wireless communication.

A few studies have extended their work towards WLANs with SDN. The first work was the OpenRoads project [16], where the user did not have to worry about the network details, he roamed freely between WiFi and cellular network having the advantages of seamless handover techniques. Then came the OpenRadio project where the PHY and MAC layers were made programmable so that a software abstraction level was formed between the hardware and the programmer. The wireless protocols were divided intro basic logical and processing blocks. Logical blocks were used to guide or map a path for processing blocks to transverse and processing blocks were used to manipulate the analogue signals. So implementation of digital processors was made possible. The SDN into WiFi is also introduced by Odin similar to VAP introduced by CloudsMAC [17]. In SDN the light virtual access points (LVAP) provides a abstraction layer to the programmer and a virtual unchanging link so that the user can connect to the APs. There has been a number of advancements made by SDN and OpenRoads but still there is a need to enhance the SDN performance.

Flow stealer, a lightweight load balancing for distributed SDN controllers is proposed. The component load detection identifies the load of controller; flow stealing is done only when the controller is idle for assignment of switches. Management of distributed controllers were involved into cluster formation for balancing load via external server using weight computation by round robin scheduling method. In distributed multi-controller SDN architecture, a centralized con-

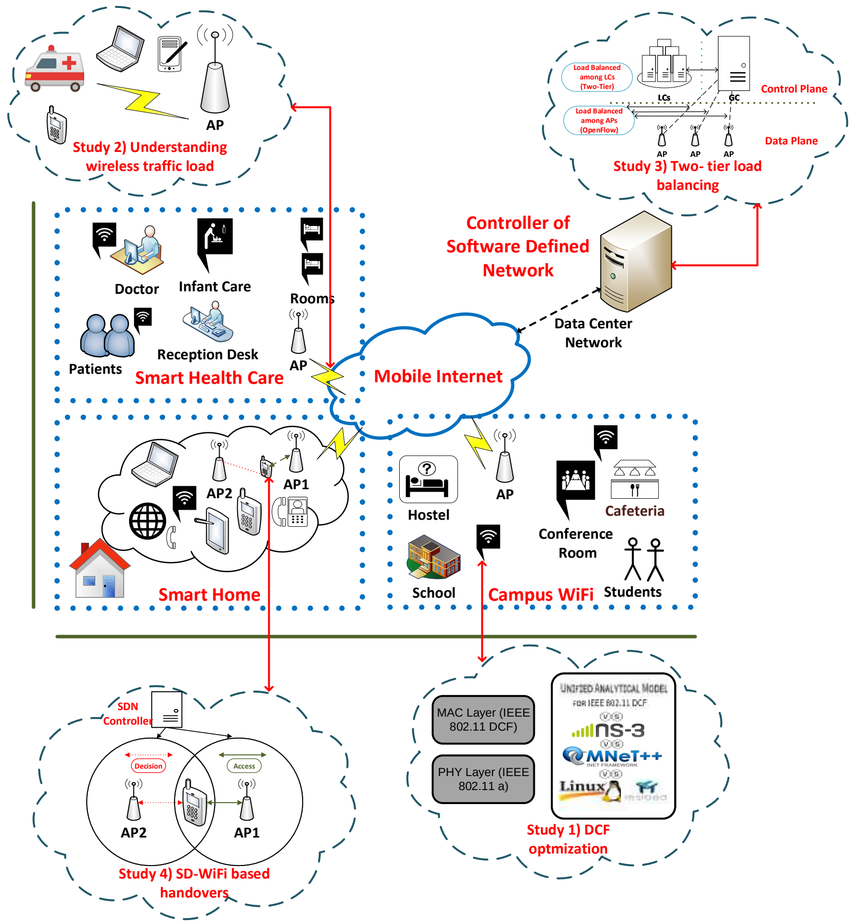


Fig. 4: Proposed SD-WiFi architecture.

troller is assigned for monitoring other controllers connected to it. Centralized controller monitors in term of CPU, traffic and latency. Load is balanced for proving users with faster response to the requested application service.Parameterized wildcard rules were written to partition servers according to weighted values and assign minimum loaded controllers [18].

The proposed SD-WiFi architecture is shown in Fig.4. The wireless stations get access to the network through OpenFlow enabled APs. These APs are controlled by a centralized controller. Many WiFi protocols are neglected by the APs as the forwarding behavior is controlled by a centralized controller. SDN allows a fast deployment of new services while enabling virtualization and storage. Many applications such as mobility control and load balancing are running on top of the controller allowing the wireless stations to move freely among the APs without noticing re-association with other APs.

1 Motivation

It is expected that in the near future the wireless traffic will exceed the wired traffic [19]. The wireless control devices including the WiFi will have an increased share in the total IP traffic of about 66 percent [20]. The Fig.5 shows the Ericson's mobility report on the increase used of the smart devices by 2019. This dramatic use of the media and wireless networks suggests improvement of the wireless network infrastructure. If we look at the infrastructure of the todays network, it is not sufficient enough to handle the increased demand. There is an increased complexity due to the mode of operation. The operational costs have increased and the innovation process is slowed down [21]. We need to make the networks more agile, scalable and flexible so that it can cope up with the user demand and managing and tweaking the network becomes easy [22].

A few studies have considered the validation of NS-3 and OMNeT++ MAC layer due to its complexity. A number of studies that validate the physical layer and the MAC layer in NS-3 and OMNeT++ are [23–28]. Baldo et al [29] presented the validation of NS-3 MAC through a test bed and Malekzadeh et al [30] validated the OMNeT++ MAC model using a test bed. Validation of MAC layer in NS-3 is also done by varying number of nodes [31]. Number of studies have focused on the optimal RTS threshold for IEEE 802.11 WLANs [32–36]. A set of complex non-linear equations based on the classic Bianchi's model [2] were used to calculate the RTS threshold [33,35]. The results show that the RTS threshold increase when the number of nodes decrease or the data rate increase [34,35]. Various network assumptions and implicit nature of solution make the relationship of RTS to key system parameters uncertain, leading to tremendous difficulties in switching between basic and RTS/CTS mechanisms in practical WLANs.

The application of WiFi networks, have attracted many researchers due to the facile connectivity anywhere and anytime. WiFi networks found in public areas such as campus, enterprise and health care support large number of wireless stations, causing performance degradation at APs [37]. The 802.11 protocols are the most widely adopted standards in the wireless local area networks (WLAN)

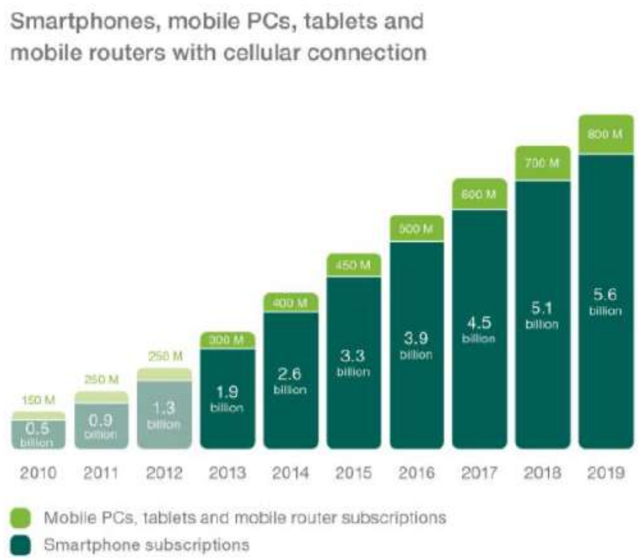


Fig. 5: Ericsson mobility report. [19]

family. Research efforts have been made to see the performance evaluation of the IEEE 802.11 Distributed Coordination Function (DCF) [38] with limited network parameters. Techniques have been presented to balance load in high density SD-WiFi [39] without defining the wireless load. In smart health sector, QoS for wireless transmission of ECG is enhanced and the co-existence of wide body area networks (WBAN) and WLAN is made possible by controlling window size of a WLAN [40]. Still the interference and collisions among the packets led to the degradation of the wireless transmission.

Flow prioritization have not been taken into account to deal with packets with higher priorities. In a multi-controller network the future prediction of load for a a specific controller needs special attention. Designing an algorithm using Markov chain models can help in balancing the loads among the multi-controllers. When large number of users connect to the same AP, the contention will be much more and the throughput performance will decrease [38]. Due to the distributed architecture of the traditional WiFi networks, users imbalance distribution and handover decisions are tamed with only client-side or AP-side information. In this study, we propose the load balancing and handover problem for WiFi networks by taking advantages of SDN to improve network performance. Based on the network controller harvesting a total view of the network, we design and implement a load balancing and a handover algorithm using both client-side and AP side information.

2 Problem Statement

SD-WiFi is a combination of WiFi network devices and SDN solutions. This network has to provide services to different smart devices. The wireless devices specially in a smart environment such as tablets, smart phones etc. need high bandwidth due to increased used of internet based services and applications. To satisfy the demand of the users, it is really difficult because video streaming, online gaming and VoIP are the services which need uninterrupted connections and priority-based processing.

The impact of RTS threshold on throughput was studied in multi-rate networks [41]. The simulations were performed for various rates in IEEE 802.11 ”b” standard. The main findings of the paper were the analysis of when to increase or decrease the RTS threshold for a specific data rate or number of nodes. No optimal tuning of RTS threshold was performed. Modeling and simulation of IEEE 802.11 ”g” standard was performed in OMNeT++ simulator [28]. The performance evaluation was performed using round trip times between two ping messages neglecting major performance metrics such as throughput versus the number of nodes, packet payloads, cutoff phase and initial backoff window sizes.

Researchers have arrogated the channel access delay in an unsaturated network to approximate the load [3]. Recent studies have also considered the traffic intensity, a measure of time when a certain AP remains busy for certain period of time, as a matrix to measure load [4]. The literature consists of many definitions of wireless traffic load, where the most common one is number of wireless stations associated with the APs, but this cannot fully characterize the traffic load as the traffic conditions for each wireless stations may not be the same [42]. To understand the traffic load better we study the parameters such as packet payload and packet generation times, which make the load different for each wireless station.

An SDN enabled WiFi network architecture was designed with two-tier load balancing [43]. Here the load balancing took place at the AP and controller side. APs reported capacity, load and association table information to the controller. Depending on these reports the controllers made a fairness index. The association list is de-associated using Last-In-First-Out (LIFO) policy, there fore this induced delay as the first user has to wait longer than the last user. Selvi et al. [44] proposed cooperative load balancing scheme for the hierarchal SDN controllers (COLBAS), where super controller handles the flow request of other users. This technique needed repeated installation of the rules and created higher delay in processing packets. Due to the distributed architecture of the traditional WiFi networks, users imbalance distribution and handover decisions are tamed with only client-side or AP-side information. Based on the network controller harvesting a total view of the network, we design and implement a load balancing and a handover algorithm using both client-side and AP side information.

Most of the handover studies in the past rely on the received signal strength indicator (RSSI) [45]. Handover that is based on RSSI values do not guarantee the clients performance and network throughput. Other techniques [46] need client side modification that needs huge amount of cost and trouble and is not suitable in real world solution. The problem formulation is depicted in Fig.6.

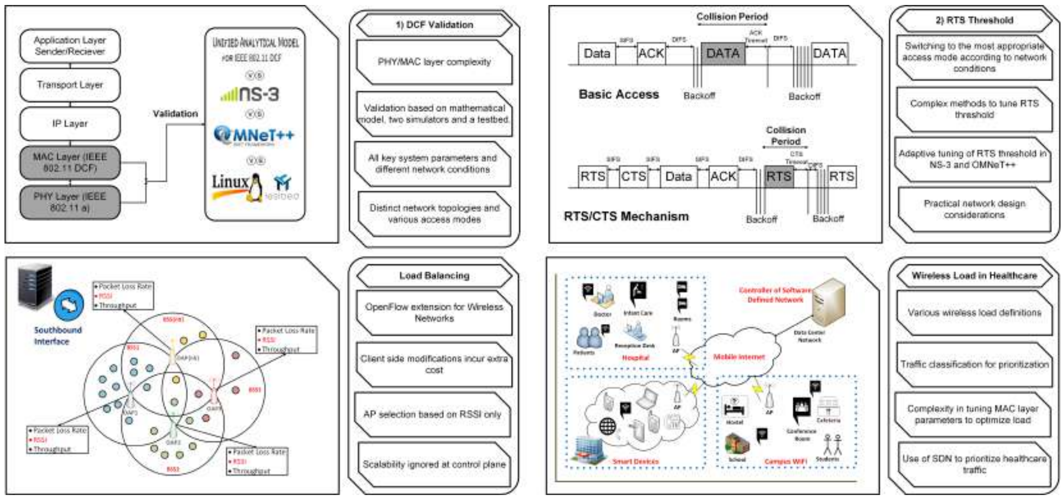


Fig. 6: Problem formulation.

3 Research Objectives

The main objective of the research is to analyze and optimize the WiFi networks, understand wireless load and tune the system parameters for least collisions in wireless transmission, achieve load balancing in data and control plane, maintain traffic priorities and to optimize the handover solutions in SD-WiFi. The proposed architecture will resolve the issues encountered in previous optimization techniques such as switch migration, RTS threshold adjustment, flow stealer approach and modified constraint particle swarm optimization techniques. The performance will be measured in terms of throughput, latency and workload handled by the controllers.

To summarize the research goal the following questions will be answered:

1. How to achieve maximum WiFi throughput performance by optimizing network parameters such as number of nodes, cutoff phase, initial backoff window size, transmission rate and RTS threshold?
2. How to analyze and understand wireless traffic load and it correlation to network parameters for a contention free transmission in software defined WiFi network?
3. How to improve the degree of load balancing and handover times by taking advantage of SDN capabilities in a WiFi network?

To answer these questions, we will design a network framework that will integrate the SDN and WiFi network architecture. A series of experiments to measure the network throughput, latency, and workload handled by the SDN controllers will be designed. These results will help to further evaluate the performance of software defined approach.

3.1 Research Scope

The research is centered on understanding and optimizing the software defined WiFi networks using mathematical models, algorithms, simulations and testbeds. We also investigate the advantages of integrating the SDN into the WiFi networks.

3.2 Challenges

There are number of challenges that might occur in the research. Our research touches the areas like WiFi networks, software defined networking. These areas need deep understanding before they can be integrated together. The SDN control plane and data plane bridges all these areas together and requires more study in order to define the parameters to characterize the input of our WiFi networks. The project relies on source codes and algorithms which induces the time challenge. There is another workload challenge of making a testbed and running all the experiments.

Background In this chapter we present the related work regarding the WiFi networks, the evolution of software defined networks and the past research that has been done on software defined WiFi networks.

4 WiFi Networks

The WiFi networks can be found in smart homes, public areas, school campuses, healthcare etc. The WiFi networks allow the communication between two end hosts through wireless signals. WiFi networks can be categorized into three parts: 1) A service provided that provides wired connection, 2) an access point (AP), 3) wireless devices that connect to the APs through a wireless connection. As the AP provide a wireless connection, so it can be called as a wireless Ethernet adapter. A conventional WiFi network is depicted in Fig.7.

Nine kinds of services are defined by the IEEE 802.11 architecture. The services can be divided into two groups. One is the station (STAs) services and the other is the distribution services. The STA services include delivery of data, privacy, authentication and de-authentication. The distribution services include association, re-association, disassociation, distribution and integration.

The impact of RTS threshold in IEEE 802.11 networks has been studied for multi rates [41]. The analysis presented showed when to increase or decrease the RTS parameter for specific data rates or number of nodes. No optimal tunning of RTS threshold has been performed. Performance of DCF MAC has also been

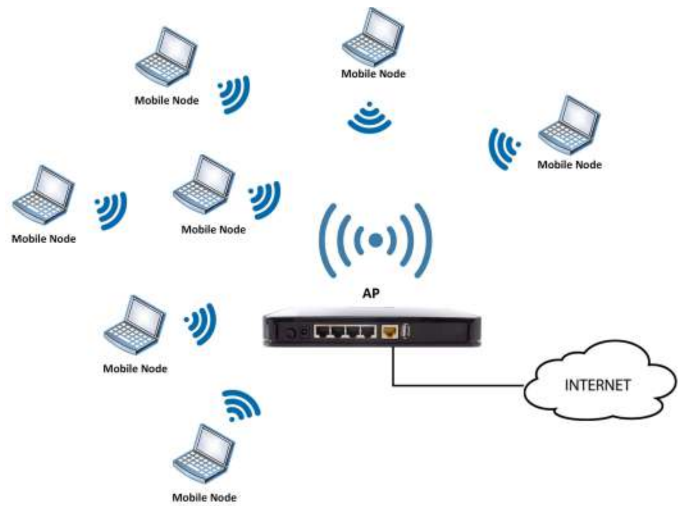


Fig. 7: A conventional WiFi network.

performed for varying RTS values [47]. QoS parameters such as end to end delay, media access delay and retransmission attempts have been evaluated for fixed RTS values such as 128, 256 or 1024 bytes. The study of how to adjust the RTS parameter automatically is made in [48]. The relationship of RTS to number of nodes is presented while neglecting other key network parameters such as packet arrival rate and packet lengths are ignored.

For the validation of NS-3 and OMNeT++ PHY/MAC layers few studies can be found. The IEEE 802.11 "b" standard is validated in the NS-3 against the results obtained from the wireless network emulator [23]. The authors have focused on the validation of the physical layer ignoring the aspects of the MAC layer. The reliability of OMNeT++ in IEEE 802.11 was determined through a testbed study [30]. Here the performance metrics taken were the packet loss ratio and throughput. DoS attacks were the main concern to the authors. Network conditions such as saturated and unsaturated networks were ignored.

5 Software Defined Networks

The help of various supporting technologies lead to the idea of SDN in early 1990s [49] [50] [51]. The idea of separating the data and control plane came from the network control point implemented in telephone network where there is a fragmentation between data and control plane. This proposed idea was cost effective and provided secure solution. Active Networks [52] [53] provided the idea of bring programmability into networks. Infrastructure (VINI) [54], and programmable router switch [52] and [55] are the examples of active networking models offering flexibility to control tasks and events. These models were discovered so early that could not be implemented due to lack of hardware and infrastructure support.

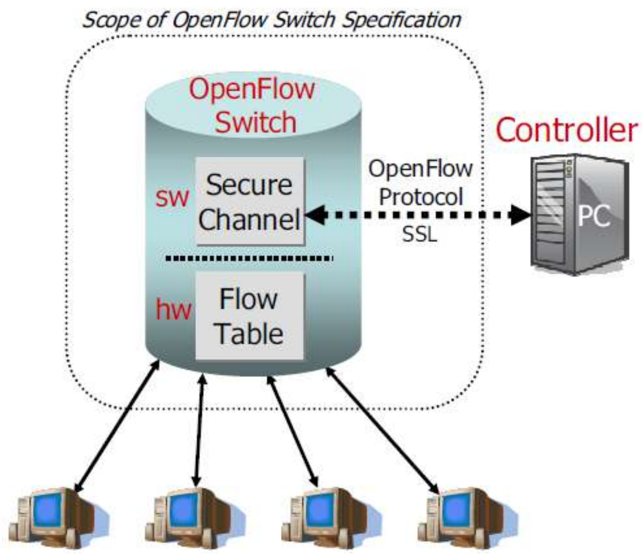


Fig. 8: OpenFlow architecture.

5.1 OpenFlow

The OpenFlow switch has two major components and that are flow table and a secure channel. The flow table is used for forwarding and packet lookup. The channel has the responsibilities to communicate between the channel and switch. The flow table has number of components such as flow entries, actions to match the packets and activity counters. If the match entry is found then a specific action is performed. If no match entry is found then the packet is forwarded to the controller for further processing. The controllers actually deal with those packets which do not have any further flows entries in the switch. The Fig.8 explains the overall architecture of OpenFlow enabled switch.

OpenFlow also provided an option of OpenFlow protocol that was used to customize the flow table entries in the routers and switches. It helped the controllers great deal as the partition into production and research flows was made easy. This gave the researchers an ease to control the flows. This was actually a revolution towards a new era that supported designing new protocols, alternatives to IP and addressing schemes and security models. The traffic on the production network is treated the same as it has been treated today [16].

Fig.9 demonstrates that how a flow can be processed in SDN. When a new packet arrives the switch searches for its entry, if the entry is not found the packet is forwarded to the controller and now by using the flow modification message the controller will install entry in it. After the entry is installed by the controller the packet is forwarded towards the host. Switches are implemented in a platform that supports customized hardware that can be programmed such as Linux, NETFPGA, and OpenWRT [56].

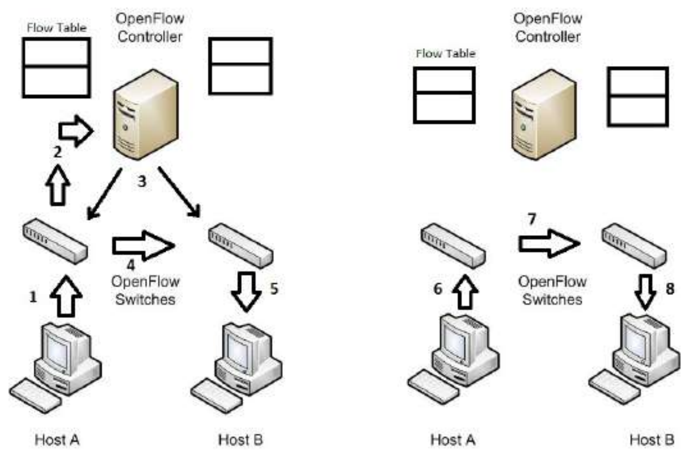


Fig. 9: Arrival of a new flow at OpenFlow switch/AP.

6 Software Defined WiFi Networks

There is been research works done in order to bring programmability to SD-WiFi networks. In a real time, networking scenario, the station connects to the AP with the highest signal strength indicator (RSSI). There comes a situation when many users connect to a single AP and the other APs may get under-utilized. These scenarios lead to unbalanced patterns of traffic and required proper load balancing applications. There is protocol called as Distributed Coordination Function (DCF) protocol, where the users using WiFi share the channel in distributed manner. When the AP gets more user connected to it this lead to more severity in connection and eventually the network gets degraded [38]. In the past there have been some works on load balancing where certain load balancing algorithms have been designed to assist the stations to connect to appropriate APs. In the traditional WiFi networks the distributed architecture is tamed with only AP or client side information [57].

6.1 SD-WiFi in Healthcare

Load metric is define to be the number of wireless stations associated with the access points [39]. The actual characterization of load for each wireless station is ignored. Lei at al [58] proposed a load definition from the channel busy time ratio. The average network allocation vector (NAV) is used to help determine load in WLANs. A deeper study of what factors cause NAV in keeping a busy duration for all mobile hearing nodes is not discussed, further the understanding of load performance of DCF is not presented. Load is balanced by prioritizing the flows and having a dynamic two-tier architecture for multiple controllers [59]. QoS support and limitations for WLANs are presented in [60]. The support includes achieving high throughputs by using latest IEEE 802.11 standards where data rates upto 7 Gbps can be achieved. The MAC limitations include the first come first serve (FCFS) creating fairness problems hence resulting in unfair bandwidth share. The real time performance depiction of DCF with varying load is neglected. A unified framework for the performance of IEEE 802.11 DCF has been proposed in [61]. Discrete time Markov chains are used to study the behavior of IEEE 802.11 a standard. The performance metrics include throughput variation with respect to window size, cutoff phase and packet payloads. No study has been made to understand the effect of packet loss and ping times with actual load definitions. A MAC protocol has been proposed in a full duplex enabled WLAN [62]. Throughput analysis has been performed in wireless collision detection scenario taking into account the MIMO techniques at PHY layer. A deeper research regarding factors leading to higher collision and load at PHY and MAC layer still needs to be carried out.

6.2 Load Balancing in SD-WiFi

OpenFlow devices in SDN checks load periodically with respect to time intervals for supporting QoS parameters [63]. Controllers load is noticed in terms of CPU utilization. Deployment of switches and connectivity are characterized as graphs to manage the devices. Karush-Kuhn-Tucker (KKT) condition is used to optimize the load by predicting states of controllers. Load balancing approaches were effectively supported in multiple SDN controllers that maximize network performances [64] [65] [66]. Many practical problems in SDN are resolved when implemented using OpenFlow protocol. Over several research works held in SDN, load balancing remains as hot spot issue in different applications. Control plane load is managed in the implementation of cCluster [67].

6.3 Handover in SD-WiFi

Most of the handover studies in the past rely on the received signal strength indicator (RSSI) [45]. Handover that is based on RSSI values do not guarantee the clients performance and network throughput. Other techniques [46] need client side modification that needs huge amount of cost and trouble and is not suitable in real world solution. ODIN makes use of the SDN and virtualize the physical AP by virtual APs [17]. Each user is assigned with a unique basic service set identification (BSSID). All the physical APs are controlled by the OpenFlow controller and the virtual APs are transferred to the new physical AP when a handover is performed. Although ODIN makes use of the SDN but scalability is still an issue. If the number of clients increase on each AP since each user has a different virtual AP so this add extra amount of information to be handled.

The major differences between the proposed four optimization techniques in SD-WiFi and the previous techniques are shown in Table 1.

Table 1: Comparison between proposed and previous techniques

Optimization Technique in SD-WiFi	Related Works	Research Gaps	Proposed Solution
IEEE 802.11 DCF	Pei. G [23]	MAC layer validation ignored.	PHY and MAC layers taken into account.
	Malekzadeh,Mina [30]	Unsaturated and saturated networks are ignored.	All network conditions considered.
	Wang.Jianxin [41]	Optimal tunning of RTS parameter ignored.	Optimal tunning of RTS threshold performed.
	Sheu.Shiann-Tsong [48]	Packet arrival rates and packet lenghts are ignored	Packet arrival rates and packet lenghts considered.
Wireless Traffic Load in Healthcare	Chen.Ze [39]	Realization of load for each wireless-station is ignored.	Load definition for each wireless station.
	Malik.Aqsa [60]	Based on first come first serve the fairness is ignored. DCF variation to load is ignored.	Fairness among all wireless stations. DCF throughput evaluation to load and network parameters considered.
	Yin.Yachao [61]	Load definition ignored. Packet loss and ping times ignored.	Wireless load defined. Packet loss and generation times included.
	Kaneko.Megumi [62]	Collisions at PHY layer ignored.	Collisions taken into account in wireless transmission.
Load Balancing	Singh.Abhishek [63]	Controllers load not fully characterized.	CPU utilization memory taken for each controller.
	Song.Ping [64]	Random load balance without considering future load of controllers.	Periodic load balancing for each controller.
	Qiu.Kun [67]	Clustering of controllers done without maintaining load matrix.	Load matrix for each local controller maintained by global controller.
	Ma.Yi-Wei [66]	Flow prioritization is ignored.	Flow prioritization maintained by SVM and flow-classifiers.
Handovers	Fakuda.Yutaka [45]	Handovers only based RSSI values.	Distance and mobility considered for handovers.
	Murty.Rohan [46]	Client side modifications required.	No client side modification required.
	Suresh.Lalith [17]	Extra LVAPs incurred extra information to be handled.	SDN controllers helps in reducing the association and deassociation messages.

Methodology

7 Optimizing Load Balancing

In this thesis work we will propose four studies that relate to WiFi and SDN integration. In the first study we simulate and optimize the DCF parameters for an optimal network throughput performance. In the second study we understand the wireless load definition and its correlation to the network parameters. In the the third study we design a dynamic two-tier load balancing architecture to achieve load balancing in SD-WiFi. In the fourth study we minimize the handover times between the wireless stations and the APs using the software defined approach.

7.1 IEEE 802.11 DCF Optimization

The PHY and MAC layers of NS-3 and OMNeT++ are proposed to be validated by an unified analytical model [1] shown in Fig.10, which incorporates the fundamental features of IEEE 802.11 DCF networks.

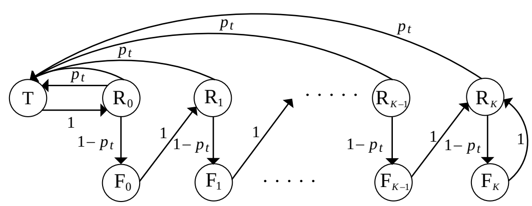


Fig. 10: Unified analytical framework for IEEE 802.11 DCF

The infrastructure and ad-hoc mode are used for validation of PHY/MAC layers of NS-3 and OMNeT++. In the ad-hoc mode each nodes act as a receiver and a transmitter where as in infrastructure mode the nodes get the access to the network through an AP by following beacon transmission, active scanning and authentication.

The network sum rate $\hat{D}$ is given by the Eq.1.

$$\hat{D}=\hat{\lambda}_{out}\cdot\frac{\frac{8PL}{R_D\sigma}}{\tau_T}\cdot R_D=\frac{-8PL\cdot p_A\ln p_A}{\sigma(1+\tau_F-\tau_F p_A-(\tau_T-\tau_F)p_A\ln p_A)}, \tag{1}$$

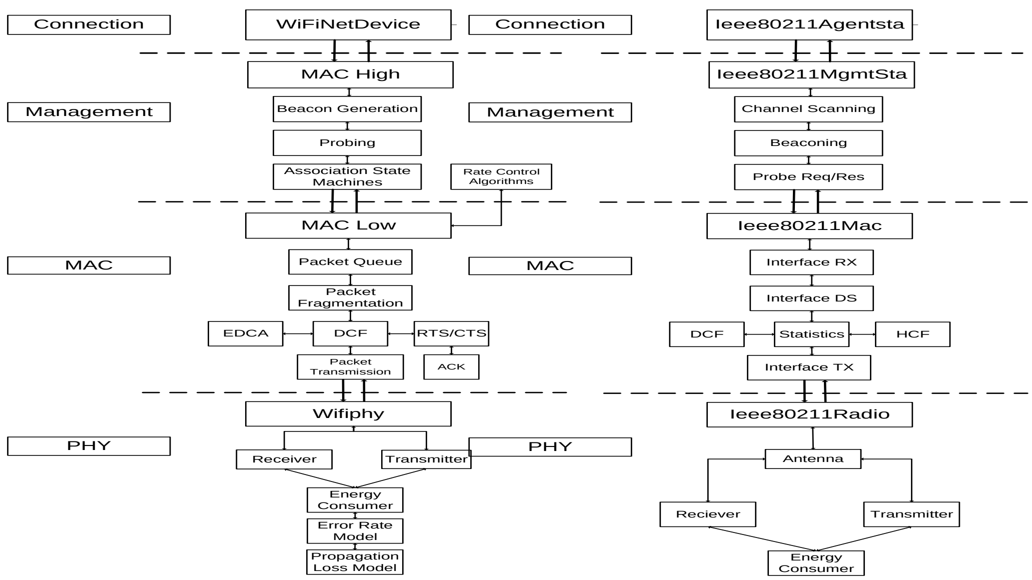
p_A is the non-zero root of the fixed-point equation presenting the steady-state probability that the HOL packets are transmitted successfully in an idle channel p:

$$p=\exp\left\{-\frac{2n}{W\cdot\left(\frac{p}{2p-1}+\left(1-\frac{p}{2p-1}\right)(2(1-p))^K\right)}\right\}, \tag{2}$$

The RTS threshold parameter is proposed to be formulated as shown in Eq. 3 and optimized which guarantees maximum network throughput.

$$RT^{*}=\frac{\frac{RTS}{R_B}(1-p_A)-(\frac{CTS}{R_B}+\frac{\mathrm{SIFS}}{4}+\frac{\mathrm{PH}}{4})p_A\ln p_A}{1-p_A+p_A\ln p_A}\cdot R_D-\mathrm{MH}, \tag{3}$$

Further we compare the performance and credibility of the WiFi modules of NS-3 and OMNeT++ as shown in Fig.11,



(a) NS-3. (b) OMNeT++.

Fig. 11: WiFi modules.

against a Linux based testbed depicted in Fig.12. The testbed built is Linux-3.2.71 kernel based with mac80211 driver support. NETGEAR WNDA3200 USB wireless cards based on the Atheros AR9002U-2NX chipsets, are used as wireless nodes. AR9002U-2NX chipsets use the Ath9k drivers. The iw [68] configure tool is used to communicate to mac80211 module.

The connection of the components in a testbed is depicted in Fig.13.

7.2 Optimizing Wireless Traffic Load

To understand the wireless load and traffic priorities, IEEE 802.11 a/b/g standard will be implemented in an infrastructure mode, where the fundamental network information will be shared to wireless stations by APs. The traffic route in healthcare is depicted in Fig.14.

The AP transmits the beacon frames continuously so that the association is made. 30 wireless stations will be contending for the wireless medium and transmitting in ideal conditions, i.e., there is no hidden stations and the channel is error free. So the assumptions made are 1) the hidden terminal effect is ignored and 2) packets are lost only due to collisions. The overall depiction of wireless transmission is shown in Fig. 15.

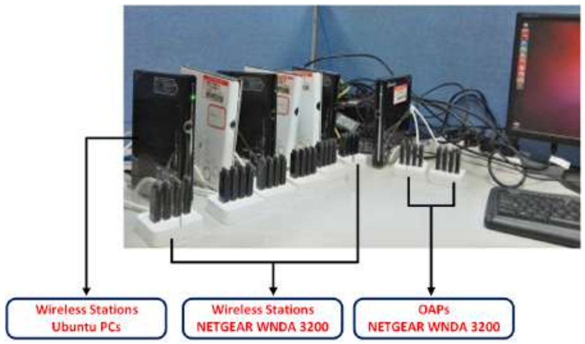


Fig. 12: Testbed setup

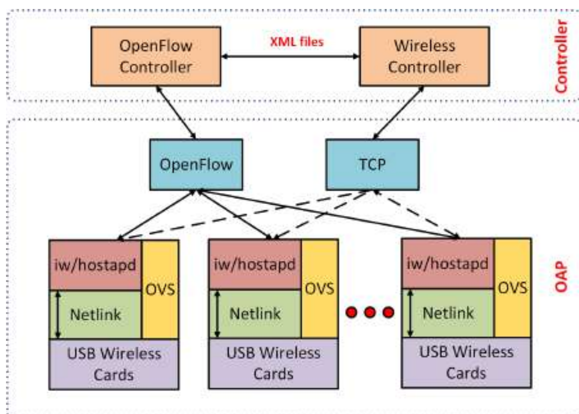


Fig. 13: Connections in the testbed.

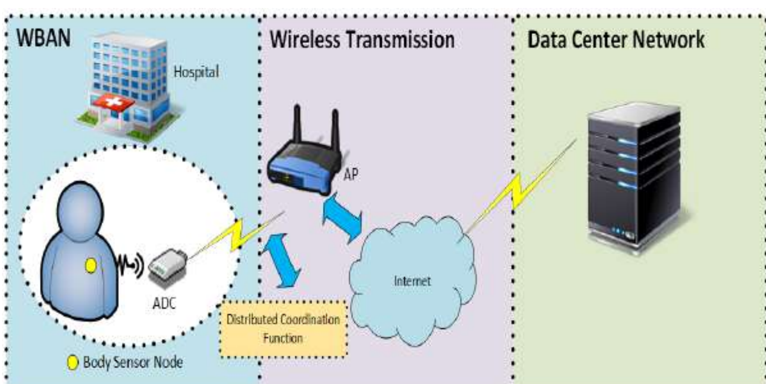


Fig. 14: Traffic route for healthcare in SD-WiFi

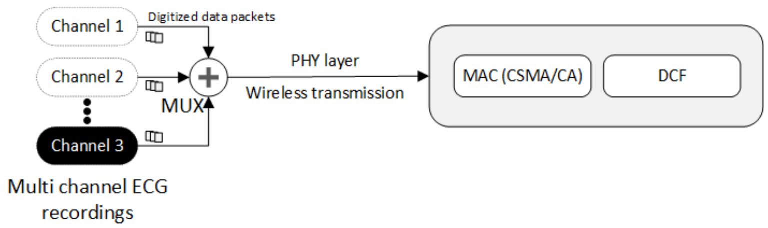


Fig. 15: Wireless transmission

7.3 Load Balancing in Data and Control Plane

The proposed SD-WiFi architecture will consist of a flow classification module, a dynamic two-tier load balancing module and an (AHP, MCM or T2FPSO) module for prioritizing flow requests and dispatching them to appropriate local controllers. WiFi access points will be responsible to classify the incoming flows from access networks (i.e., smart home networks and wireless local area networks (WLAN)), into two queues of real-time and non-real-time flow requests. These request queues will be processed into controllers. AHP, MCM or T2FPSO will prioritize requests based on three features of flows including flow size, service type and delay constraint. The highest priority flows will be assigned to idle cluster for fast processing by the global controller. Flows with some delay tolerance will be processed next. The goals of load balancing are two-fold: 1) migrate the controller in the overloaded cluster; 2) allocate prioritized flows towards less-loaded controllers. Algorithm in Fig.16b explains load balancing among controllers.

In Algorithm for load balancing among the APs as shown in Fig.16a, MAXLOAD is the maximum threshold value for each APs load. If one APs load in the network is larger than MAXLOAD, the Adaptive Connection and Handoff (ACH) algorithm will be triggered. MAXDIFF is the maximum difference value between any two APs load. If there is a difference value between two APs load in this network, we consider the load distribution is unbalanced and the ACH algorithm will be triggered. This algorithm runs periodically based on the information collected by the network controller. The access process for stations is considered first; then, if the load of the network approaches imbalance, the hand off process will be initiated. Both AP-side information and STA-side information are needed as the input for the ACH algorithm. The simulation topologies for both load balancing schemes is shown in Fig.17.

7.4 Optimizing Handovers in SD-WiFi

The same SDN platform as used for the load balancing is extended for depicting how the handover time is reduced. The centralized SDN controller has an overview of the entire network so it can easily deploy the new handover policies in an AP and reduce the hassle of handover in a traditional network by neglecting some unimportance messages between the wireless stations and the APs. The concept can be visualized in Fig.18.

Algorithm 1 The ACH Load Balancing Algorithi

The network controller collects information STA and AP.
for STA_j in $STAs$ **do**
  Construct a AP set UA_j, which are the ser STA_j
  Get $APMIN_j$ from UA_j which has th $STAs$ associated.
  Connect STA_j to $APMIN_j$.
  Update network parameters, $Load_{max}$ and
  if $Load_{max}$ $>$ $MAXLOAD$ or $(L$ $Load_{min}) > MAXDIFF$ **then**
    Get $APload_{max}$, the AP with maxin associated.
    Pick the weakest signal STA_t in $APloa$ ciated $STAs$.
    Construct set UA_t for STA_t.
    Find AP_m from UA_t, the minimum ST ated.
    Handoff STA_t from $APload_{max}$ to AP
  else
    continue;
  end if
end for

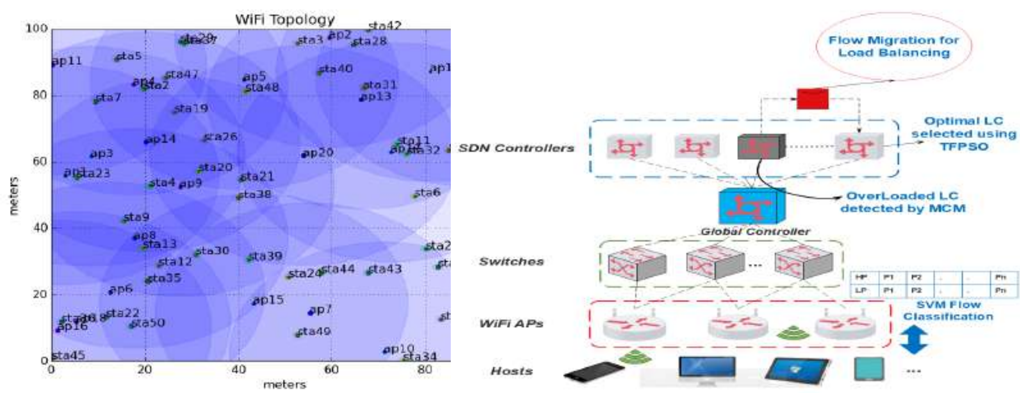
(a) Data plane load balancing.

Algorithm 1 The Proposed Load Balancing Algorithm

Receive all flows from users
for $allflows$ **do**
  Initialize flow size, delay constraint.
  Feed flows into SVM classifier
  Classify flows into HP and LP
  Send flows to GC through switches based on priority
end for
Assign flows to LCs randomly
for $allLCs$ **do**
  Compute current load on LCs using eqn 5
  Compute probability of state transition for each LC using eqn 3
  Construct transmission matrix
  Compute future load using eqn 4
end for
if $LCisoverloaded$ **then**
  Migrate flows from that LC
  Initialize all particles with fitness value
  Initialize pbest & gbest values
  for $allparticles$ **do**
    Compute fitness value for each particle using fuzzy logic
    Update pbest & gbest in each iteration
    If maximum epochs meet
    Stop.
  end for
else
  repeat the process;
end if

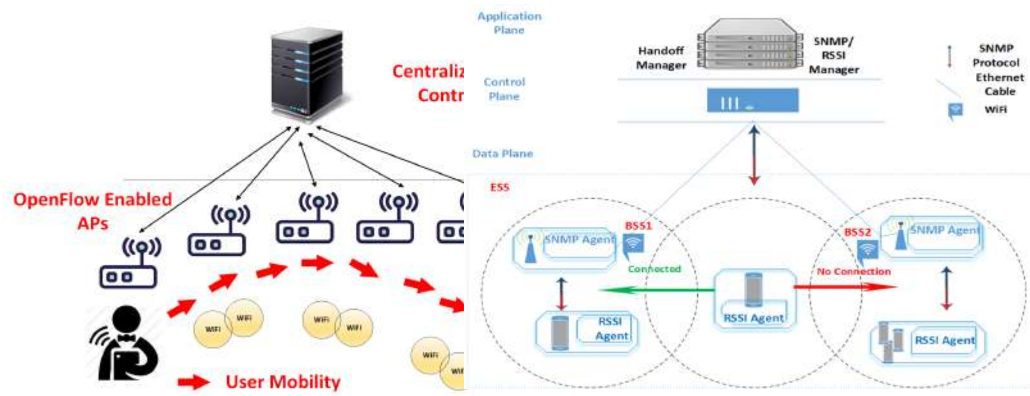
(b) Control plane load balancing.

Fig. 16: Load balancing algorithms in SD-WiFi.



(a) Data plane load balancing.
(b) Control plane load balancing.

Fig. 17: Simulation topologies.



(a) Overview of handovers in SD-WiFi.

(b) Proposed architecture for handovers in SD-WiFi.

Fig. 18: Handovers in SD-WiFi.

8 Evaluation

For the evaluation we will use the key system parameters such as number of nodes n, window size W, cutoff phase K, transmission rates R_D and inter arrival time λ against network sum rate. In optimizing the RTS threshold, we use the standard and optimal setting for different network sizes and packet payloads against the network sum rate. In some preliminary results the comparison made is between the throughput $\hat{D}$ and the initial backoff window size W in an infrastructure mode as shown in Fig. 19a. The RTS threshold performance is evaluated by a comparison made between between throughput and packet payload for n = 50 as shown in Fig.19b. The optimal settings show improved results.

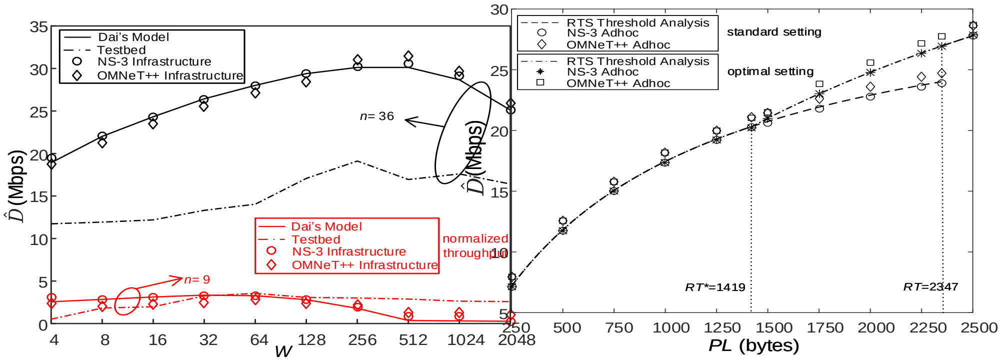
For understanding traffic load and maintaining traffic priorities we use the packet generation times and packet payloads. The preliminary results for latency vs no of stations using different packet generation times is shown in Fig.20a. Similarly packet loss rate vs no of stations for different packet payloads in shown in Fig.20b.

In the load balancing and handover study we perform the evaluation of the end-user's QoS, by considering throughput, delay, work handled by the controllers, and Jain's index among the APs. The QoS performance will be examined for both non-real-time and real-time traffic. The performance evaluation for load balancing in SD-WiFi is shown in Fig.21.

The simulation parameters proposed to be used in the four studies are given in Table2.

Research Schedule The Table 3 below shows the research schedule I have undertaken and will follow to complete my PhD on time.

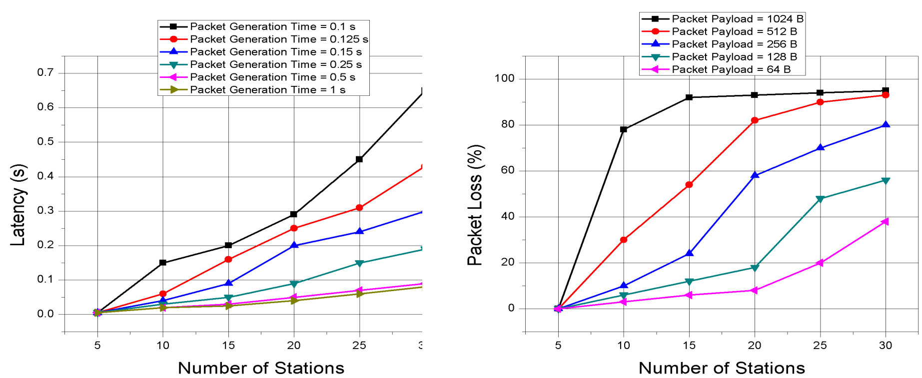
Summary In this thesis work, we will study the optimization techniques in software defined WiFi networks. In doing so first we will present a detailed



(a) Throughput vs window size.

(b) Throughput vs packet payload for n=50.

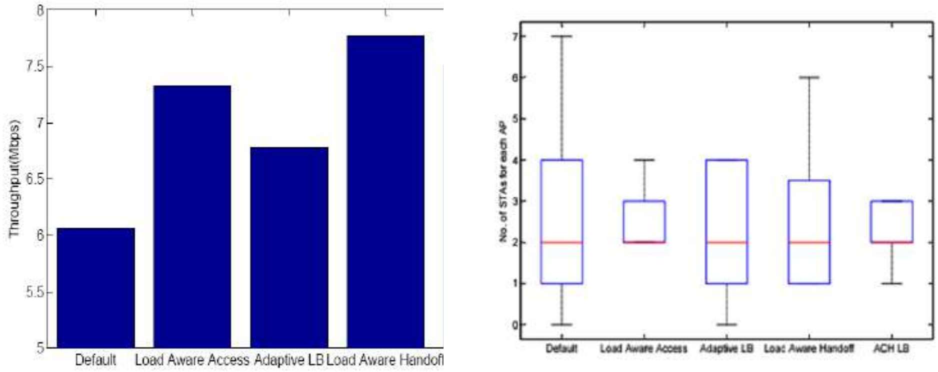
Fig. 19: IEEE 802.11 DCF optimization results.



(a) Latency vs no of stations.

(b) Packet loss rate vs no of stations.

Fig. 20: Wireless load for healthcare in SD-WiFi.



(a) Throughput vs load balancing schemes. (b) Fairness among APs.

Fig. 21: Load balancing performance evaluation.

Table 2: Simulation Parameters

Area	250 to 1000 m*m	SDN Controller	Flood Light, NOX, POX
OpenFlow enabled APs	1-8	Wireless Stations	20-100
Radio Transmitter Power	9.0 mW	Radio Receiver Sensitivity	-85 dbm
Carrier Frequency	2.4 GHz	MAC Header	224 bits
PYH Header	192 bits	DIFS	50 us
SIFS	10 us	Slot Time	20 us
Transmission Coverage	300 m	MTU	1500 Bytes
Packets Transmitted/s	500-1000	Retry Limit	7
Mobility	25mps $\pm$ 8mps	CW max	1023
CW min	31	Packet Generation Time	100-200 us
Simulation Time	200-300s	Packet Payload	512 B - 1024 B

Table 3: Research Schedule

Phase	Time	Task
1	Sep 2016 - July 2017	Course work and literature review
2	July 2017 - July 2018	Evaluation, analysis and 3 conference publications
3	Sep 2018 - Jan 2019	Enhanced experimentation and 2 conference publications
4	Jan 2019 - June 2019	Journal paper formulation
5	June 2019 - Dec 2019	Proposal defense and 2nd journal paper
6	Jan 2020 - March 2020	Dissertation writing
7	March 2020	Thesis submission
8	May 2020	Thesis defense

analysis of the WiFi networks and the solutions to obtain optimum throughput performance in the IEEE 802.11 WLANs for healthcare scenario. Then we understand the definition of wireless load and its correlation to the system parameters that effect the wireless load between APs and wireless stations. In the third step, we will design and implement a load balancing scheme consisting of flow classification and cluster-based load balancing, to solve the overloading problem in controllers and OpenFlow enabled APs. Clusters will be constructed with respect to the load in terms of processing flows. The clustering by the global controller will support to allocate flows and balance load. The deployment of our proposed scheme in SD-WiFi will support higher throughputs and lower delays. Finally the handover time between a wireless station and an AP is reduced by incorporating SDN features. The centralized view of the network makes it easier to perform quick handover in a SD-WiFi. We hope to have simulation results better than previous related studies. This proposed scheme can be applied to achieve improved quality-of-service for real-time applications on cloud-based software defined WiFi networks with heavy loading.

Acknowledgment

The research project titled "Traffic Aware Service Differentiation in Software Defined Wi-Fi Networks for Health care" was successfully completed in the Wireless Networking Research and Development Lab under the Pakistan Engineering Council (PEC) Annual Award of the Final Year Design Projects (FYDPs) for the year 2022-23. The project was supervised by Engr. Dr. Sohaib Manzoor.

References

1. Dai, L., Sun, X.: A unified analysis of IEEE 802.11 DCF networks: Stability, throughput, and delay. IEEE Transactions on Mobile Computing **12**(8), 1558–1572 (2013)
2. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on selected areas in communications **18** (2000)
3. Ma, Y., et al.: Multi-hop multi-AP multi-channel cooperation for high efficiency WLAN. In: PIMRC. pp. 1–7. IEEE (2016)
4. Han, Y.: Load-Aware Traffic Control in Software-Defined Enterprise Wireless Local Area Networks. Ph.D. thesis, University of Essex (2018)
5. Committee, O.M.E., et al.: Software-defined networking: The new norm for networks. ONF White Paper (2012)
6. Goransson, P., Black, C., Culver, T.: Software defined networks: a comprehensive approach. Morgan Kaufmann (2016)
7. McKeown, N.: Software-defined networking. INFOCOM keynote talk **17**(2), 30–32 (2009)
8. Nadeau, T.D., Gray, K.: SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. " O'Reilly Media, Inc." (2013)
9. Hu, F., Hao, Q., Bao, K.: A survey on software-defined network and OpenFlow: From concept to implementation. IEEE Communications Surveys & Tutorials **16**(4), 2181–2206 (2014)

10. Kreutz, D., Ramos, F.M., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., Uhlig, S.: Software-defined networking: A comprehensive survey. Proceedings of the IEEE **103**(1), 14–76 (2015)
11. Stallings, W.: Software-defined networks and openflow. The internet protocol Journal **16**(1), 2–14 (2013)
12. Voellmy, A., Hudak, P.: Nettle: Taking the sting out of programming network routers. In: International Symposium on Practical Aspects of Declarative Languages. pp. 235–249. Springer (2011)
13. Dhamecha, K., Trivedi, B.: SDN issues-a survey. International Journal of Computer Applications **73**(18) (2013)
14. Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., Shenker, S.: NOX: towards an operating system for networks. ACM SIGCOMM Computer Communication Review **38**(3), 105–110 (2008)
15. Nunes, B.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: Past, present, and future of programmable networks. IEEE, Communications Surveys & Tutorials **16**(3), 1617–1634 (2014)
16. Yap, K.K., Kobayashi, M., Sherwood, R., Huang, T.Y., Chan, M., Handigol, N., McKeown, N.: OpenRoads: Empowering research in mobile networks. SIGCOMM Computer and Communications **40**(1), 125–126 (Jan 2010). https://doi.org/10.1145/1672308.1672331, http://doi.acm.org/10.1145/1672308.1672331
17. Suresh, L., Schulz-Zander, J., Merz, R., Feldmann, A., Vazao, T.: Towards programmable enterprise WLANS with Odin. In: Proceedings of the first workshop on Hot topics in software defined networks. pp. 115–120. ACM (2012)
18. Lin, T.L., Kuo, C.H., Chang, H.Y., Chang, W.K., Lin, Y.Y.: A parameterized wildcard method based on SDN for server load balancing. In: International Conference on Networking and Network Applications (NaNA). pp. 383–386 (July 2016). https://doi.org/10.1109/NaNA.2016.74
19. Ericsson mobility report 2015, accessed on 2016-03-25
20. Cisco: Cisco visual networking index (January 2016)
21. Feamster, N., Rexford, J., Zegura, E.: The road to SDN. Queue **11**(12), 20 (2013)
22. Koponen, T., Shenker, S., Balakrishnan, H., Feamster, N., Ganichev, I., Ghodsi, A., Godfrey, P.B., McKeown, N., Parulkar, G., Raghavan, B., Rexford, J., Arianfar, S., Kuptsov, D.: Architecting for innovation. SIGCOMM Comput. Commun. Rev. **41**(3), 24–36 (Jul 2011). https://doi.org/10.1145/2002250.2002256, http://doi.acm.org/10.1145/2002250.2002256
23. Pei, G., et al.: Validation of ns-3 802.11b PHY model (2009), http://www.nsnam.org/ pei/80211b.pdf
24. Barsocchi, P., Oligeri, G., Potorti, F.: Validation for 802.11b wireless channel measurements. Tech. rep., Consiglio Nazionale delle Ricerche (2006)
25. Fuxjager, P., et al.: Collecting broken frames: Error statistics in IEEE 802.11 b/g links. In: IEEE WiOPT (2008)
26. Di Stefano, A., Scaglione, A., Terrazzino, G., Tinnirello, I., Ammirata, V., Scalia, L., Bianchi, G., Giaconia, C.: On the fidelity of IEEE 802.11 commercial cards. In: IEEE WICON. pp. 10–17 (July 2005)
27. Bredel, M., Bergner, M.: On the accuracy of IEEE 802.11 g wireless LAN simulations using omnet++. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques. p. 81. ICST (Institute for Computer Sciences, Social-Informatics and Technology (2009)

28. Sarkar, N.I., Membarth, R.: Modeling and simulation of IEEE 802.11 g using OMNeT++. In: Handbook of Research on Discrete Event Simulation Environments: Technologies and Applications, pp. 379–397. IGI Global (2010)
29. Baldo, N., Requena-Esteso, M., Núñez-Martínez, J., Portolès-Comeras, M., Nin-Guerrero, J., Dini, P., Mangues-Bafalluy, J.: Validation of the IEEE 802.11 MAC model in the ns3 simulator using the extreme testbed. In: ICST SimuTools. p. 64 (2010)
30. Malekzadeh, M., Ghani, A.A.A., Subramaniam, S., Desa, J.M.: Validating reliability of OMNeT++ in wireless networks DoS attacks: Simulation vs. testbed. IJ Network Security **13**(1), 13–21 (2011)
31. Patidar, R., Roy, S., Henderson, T.R., Mehrnoush, M.: Validation of Wi-Fi network simulation on ns-3. Tech. rep., University of Washington (2017)
32. Yan, S., Zhuo, Y., Wu, S.: An adaptive RTS threshold adjust algorithm based on minimum energy consumption in IEEE 802.11 DCF. In: International Conference on Communication Technology Proceedings, 2003. ICCT 2003. vol. 2, pp. 1210–1214. IEEE (2003)
33. Chatzimisios, P., Boucouvalas, A., Vitsas, V.: Optimisation of RTS/CTS handshake in IEEE 802.11 Wireless LANs for maximum performance. In: IEEE Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. pp. 270–275. IEEE (2004)
34. Kong, Z.n., Tsang, D.H., Bensaou, B.: Adaptive RTS/CTS mechanism for IEEE 802.11 WLANs to achieve optimal performance. In: 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577). vol. 1, pp. 185–190. IEEE (2004)
35. Zhang, L., Shu, Y.: RTS threshold self-tuning algorithm based on delay analysis on 802.11 DCF. In: Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005. vol. 1, pp. 627–630. IEEE (2005)
36. Liu, J., Guo, W., Xiao, B.L., Huang, F.: RTS threshold adjustement algorithm for IEEE 802.11 DCF. In: 2006 6th International Conference on ITS Telecommunications. pp. 654–658. IEEE (2006)
37. Zhang, C., et al.: Characterizing interference in a campus WiFi network via mobile crowd sensing. In: Springer International Conference on Collaborative Computing: Networking, Applications and Worksharing (2015)
38. Gao, Y., Dai, L., Hei, X.: Throughput optimization of multi-bss ieee 802.11 networks with universal frequency reuse. IEEE Transactions on Communications **65**(8), 3399–3414 (2017)
39. Chen, Z., Manzoor, S., Gao, Y., Hei, X.: Achieving load balancing in high-density software defined WiFi networks. In: IEEE International Conference on Frontiers of Information Technology (FIT). pp. 206–211 (2017)
40. Cho, B.M., et al.: Fairness-aware radio resource management for medical interoperability between WBAN and WLAN. Annals of Telecommunications (2016)
41. Wang, J., Li, B.: The impact of RTS threshold on the performance of multi-rate network. Computer Journal (2009)
42. Han, Y., et al.: An adaptive load balancing application for software-defined enterprise wlans. In: ICTC. IEEE (2016)
43. Lin, Y.D., Wang, C.C., Lu, Y.J., Lai, Y.C., Yang, H.C.: Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks. Wireless Networks pp. 1–13 (2017)
44. Selvi, H., Zeydan, E., Gur, G., Alagoz, F.: Load balancing in OpenFlow-enabled switches for wireless access traffic aggregation. In: IEEE/IFIP Network Op-

erations and Management Symposium (NOMS). pp. 1013–1014 (April 2016). https://doi.org/10.1109/NOMS.2016.7502949
45. Fukuda, Y., Honjo, M., Oie, Y.: Development of access point selection architecture with avoiding interference for WLANs. In: 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications. pp. 1–5. IEEE (2006)
46. Murty, R., Padhye, J., Wolman, A., Welsh, M.: Dyson: An architecture for extensible wireless LANs. In: Usenix annual technical conference (2010)
47. Singh, H., Kaur, A., Sharma, A., Sharma, V.: Performance optimization of DCF-MAC standard using enhanced RTS threshold under impact of IEEE 802.11 n WLAN. In: 2015 Fifth International Conference on Advanced Computing & Communication Technologies. pp. 421–424. IEEE (2015)
48. Sheu, S.T., Chen, T., Chen, J., Ye, F.: The impact of RTS threshold on IEEE 802.11 MAC protocol. In: Ninth International Conference on Parallel and Distributed Systems, 2002. Proceedings. pp. 267–272. IEEE (2002)
49. Greenberg, A., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., Zhang, H.: A clean slate 4D approach to network control and management. SIGCOMM Comput. Commun. Rev. **35**(5), 41–54 (Oct 2005). https://doi.org/10.1145/1096536.1096541, http://doi.acm.org/10.1145/1096536.1096541
50. Casado, M., Freedman, M.J., Pettit, J., Luo, J., McKeown, N., Shenker, S.: Ethane: taking control of the enterprise. In: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 1–12. SIGCOMM '07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1282380.1282382, http://doi.acm.org/10.1145/1282380.1282382
51. Feamster, N., Rexford, J.: Getting students' hands dirty with clean-slate networking. SIGCOMM-Computer Communication Review **41**(4), 531 (2011)
52. Bhattacharjee, S., Calvert, K.L., Zegura, E.W.: An architecture for active networking. In: High Performance Networking VII, pp. 265–279. Springer (1997)
53. Tennenhouse, D.L., Wetherall, D.J.: Towards an active network architecture. In: DARPA Active Networks Conference and Exposition, 2002. Proceedings. pp. 2–15. IEEE (2002)
54. Bavier, A., Feamster, N., Huang, M., Peterson, L., Rexford, J.: In VINI veritas: realistic and controlled network experimentation. SIGCOMM Comput. Commun. Rev. **36**(4), 3–14 (Aug 2006). https://doi.org/10.1145/1151659.1159916, http://doi.acm.org/10.1145/1151659.1159916
55. Alexander, D.S., Arbaugh, W.A., Hicks, M.W., Kakkar, P., Keromytis, A.D., Moore, J.T., Gunter, C.A., Nettles, S.M., Smith, J.M.: The switchware active network architecture. IEEE network **12**(3), 29–36 (1998)
56. Sezer, S., Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., Rao, N.: Are we ready for sdn? implementation challenges for software-defined networks. IEEE Communications Magazine **51**(7), – (2013). https://doi.org/10.1109/MCOM.2013.6553676
57. Khondoker, R., Zaalouk, A., Marx, R., Bayarou, K.: Feature-based comparison and selection of software defined networking (SDN) controllers. In: World Congress on Computer Applications and Information Systems (WCCAIS) (2014)
58. Lei, J., Tao, J., Huang, J., Xia, Y.: A differentiated reservation MAC protocol for achieving fairness and efficiency in multi-rate IEEE 802.11 WLANs. IEEE Access **7**, 12133–12145 (2019)

59. Manzoor, S., Hei, X., Cheng, W.: Towards dynamic two-tier load balancing for software defined WiFi networks. In: International Conference on Communication and Information Systems (ICCIS) (Nov 2017)
60. Malik, A., Qadir, J., Ahmad, B., Yau, K.L.A., Ullah, U.: QoS in IEEE 802.11-based wireless networks: a contemporary review. Journal of Network and Computer Applications **55**, 24–46 (2015)
61. Yin, Y., Gao, Y., Hei, X.: Performance evaluation of a unified IEEE 802.11 DCF model in NS-3. In: 11th EAI International Conference on Simulation Tools and Techniques (SIMUtools). IEEE (2018)
62. Kaneko, M.: Throughput analysis of CSMA with imperfect collision detection in full duplex-enabled WLAN. IEEE Wireless Communications Letters **6**(4), 490–493 (2017)
63. Singh, A., Tiwray, M., Kumar, R., Misra, R.: Load balancing among wide-area SDN controllers. In: International Conference on Information Technology (ICIT). pp. 104–109 (Dec 2016). https://doi.org/10.1109/ICIT.2016.032
64. Song, P., Liu, Y., Liu, T., Qian, D.: Flow Stealer: lightweight load balancing by stealing flows in distributed sdn controllers. Science China Information Sciences **60**(3), 032202 (2017)
65. Hikichi, K., Soumiya, T., Yamada, A.: Dynamic application load balancing in distributed SDN controller. In: 18th Asia-Pacific Network Operations and Management Symposium (APNOMS). pp. 1–6 (Oct 2016). https://doi.org/10.1109/APNOMS.2016.7737208
66. Ma, Y.W., Chen, J.L., Tsai, Y.H., Cheng, K.H., Hung, W.C.: Load-balancing multiple controllers mechanism for software-defined networking. Wireless Personal Communications **94**(4), 3549–3574 (2017)
67. Qiu, K., Tu, R., Huang, S., Zhao, J., Wang, X.: cCluster: A highly scalable and elastic openflow control plane. In: IEEE 23rd International Symposium on Quality of Service (IWQoS). pp. 165–170 (June 2015). https://doi.org/10.1109/IWQoS.2015.7404728
68. iw: https://wireless.wiki.kernel.org/en/users/documentation/iw