

Software Defined Network for Qos Enhancement in Mobile Wi-Fi Network



Sounni Hind, El kamoun Najib, Lakrami Fatima

Abstract: *Wi-Fi has become one of the dominant technology for direct access to the Internet due to its expanded services offering, its simplicity and its high-bandwidth provisioning that gives a capacity to support high traffic. Software-defined network (SDN) architecture simplified network management through an automatically programmed network. It provides a separation between the control plan (Controller) and the data plan (Switch) functions of the networks, this allows network optimization and fast response to network changes without the need to manually reconfigure existing infrastructure or purchase new hardware. In this paper, we have proceeded by studying the exchanges during a mobility within a WI-FI-based SDN network, then a network performance evaluation has been performed in order to analyze the impact of introducing the SDN concept on the performance of a mobile Wi-Fi network. For all proposed scenarios, we notice that the Wi-Fi-Based SDN architecture gives better results than in the case of the traditional Wi-Fi network.*

Index Terms: SDN; 802.11; mobility; Mininet, performance evaluation.

I. INTRODUCTION

In recent years, the progress in wireless technologies have completely revolutionize customer's communication habits. Wireless technologies have become an essential need of consumers' daily lives, and its effect will become even more important in the future; especially, with the heavy use of smartphones and the wider deployment of wireless networks that have boosted the access to multimedia applications. Recently, the popularity of Wireless Local Area Networks (WLAN) has increased significantly because of their ability to provide high mobility support, flexibility and ease of use along with reduced cost of installation and maintenance.

For current applications running on IP-based wireless

networks; mobility is one of the important requirement. In a Wi-Fi context [1], mobility refers to the ability of a client device to roam from one access point (AP) to another while maintaining an active network connection. Despite the large number of research works in mobility management, whether it is between homogenous or heterogeneous technologies, to be in on local or extended location, the problems continue to persist... In fact, while speaking of mobility, several levels are evoked. Mobility can occur in many layers of OSI model, however, most research works in Wlan mobility are focusing on level 2 and level 3 mobility [2]. A Level 2 (Link layer) mobility is involved when a terminal is re-associated with a new AP from to the same subnetwork. When the target AP is belonging to a different network it is considered as a level 3 mobility (IP layer) [2]. Actually, level 3 mobility can occur only after a successful level 2 mobility.

The challenges encountered in achieving transparent connectivity can be attributed to the failure of underlying network devices to dynamically adjust their activities in accordance with the changes induced by mobility. The deployment of mobility in Wi-Fi networks faces some critical issues. Firstly, latency, especially when the node starts the re-association procedure to connect to a new AP. Secondly during the handover the node needs an authentication procedure; this authentication is a time-limited task and the network connection will be lost in the case of timeout. Another issue can occur when a user selects an AP on the basis of the strongest received signal strength indication (RSSI) this AP may be congested, and the user could experience low throughput. Wi-Fi network did not include a load balancing mechanism and this has an impact on throughput.

Recently, software-defined networks (SDNs) [3] have been more widely adopted as paradigm [4] for new network communication that allow the separation of network traffic management (control plane) from the transmission of network packets (data plan). In fact, one of the main advantages of the SDN is that it allows to control and precisely define the flows necessary to identify and link applications. The mission of the SDN is to provide an abstraction layer of the network and to present it as a single system.

Up to now, many different types of SDN controllers have been developed with different programming languages and feature sets. These controllers are deployed in both industry and academia. The Open flow protocol is supported by most of these controllers to program routing instructions on the data level.

Manuscript published on 30 September 2019

* Correspondence Author

Mohammad Zarour*, College of Computer and Information Sciences, Prince Sultan University , Rafha Street, Riyadh, Saudi Arabia, E-Mail: mzarour@psu.edu.sa

Norah Alhammad, College of Computer and Information Sciences, Prince Sultan University , Rafha Street, Riyadh, Saudi Arabia, E-Mail: norah.a.alhammad@psu.edu.sa

Mamdouh Alezezi, College of Computer and Information Sciences, Prince Sultan University , Rafha Street, Riyadh, Saudi Arabia, E-Mail: malenzi@psu.edu.sa

Khalid Alsarayrah, College of Computer and Information Sciences, Prince Sultan University , Rafha Street, Riyadh, Saudi Arabia, E-Mail: khalidtoriki@psu.edu.sa

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Most popular controllers are: Ryu, Floodlight, ONOS and OpenDayLight, POX, NOX [4, 5, 6, 7]. Table I represents the different characteristics of each controller.

TABLE I. SDN CONTROLLER'S CHARACTERISTICS

Controller	Language	OpenFlow version	Multithreading	Virtualization
Ryu	Python	1.0, 1.2, 1.3, 1.4, 1.5	No	Mininet/OVS
Floodlight	Java	1.0, 1.1, 1.2, 1.3, 1.4,	Yes	Mininet/OVS
OpenDayLight	Java	1.0, 1.3	Yes	Mininet/OVS
POX	Python	1.0	No	Mininet/OVS
NOX	C++	1.0, 1.1, 1.2, 1.3,	No	Mininet/OVS

In support of Software Defined Network (SDN) research [9], many techniques are used to evaluate performance, network testing and debugging; emulation is used to running real code under realistic network conditions and gives rich experiments and fast prototyping cycles. As with wired SDN experiences, an emulation tool is required for the testing and analysis of software-defined wireless networks (SDWNs) [12] wish to become an emerging and prominent branch of SDN research. To this end, we used Mininet emulator [10]. Only a few alternatives are available based on current simulators (NS-2, NS-3 and OMNet ++ [11]).

In this article, we introduce a Wi-Fi-based SDN to achieve seamless connectivity in mobility scenarios. Our contributions can be synthesized as follow:

1. Challenges identification when mobility occurs.
2. Describe the exchanges between MN, AP and SDN controller.
3. Implementation of a platform to emulate the wireless network based on Mininet-Wi-Fi [12].
4. Evaluation of the Wi-Fi-based SDN for UDP application in terms of delay, throughput, and packet loss.

The remainder of the paper is structured as follows. Section II provides SDN and Openflow overview. Section III gives details about emulation setup and presents the results obtained. Section V outlines conclusion and future works.

II. SDN AND OPENFLOW OVERVIEW

Software-Defined Networking is a new interesting paradigm that helps to overcome network ossification by bringing in programmability. This approach promises to simplify network management, configuration and offers scalability benefits. In traditional network devices, packet transmission (data plan) and routing decisions (control plan) are executed on the same device. An OpenFlow switch is used to separate these two functions. The data plan still remains on the switch, while all routing decisions are transferred to the OpenFlow controller. Based on its global view of the network, the controller provides optimal programming of the data plan transfer behavior. The OpenFlow controller and switches are based on a common set of APIs and control messages to communicate with each other. The SDN

paradigm is composed of three logical plans, “fig. 1” represents the SDN architecture:

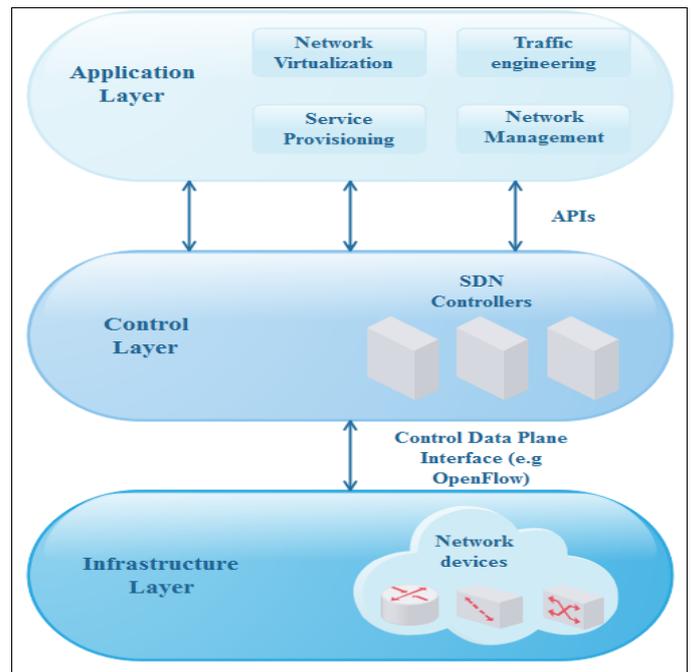


Fig. 1. SDN architecture

- The application plan: offers to the user a list of services to manage the network, on which the control plan will act.
- The control plane: Is the the network part that includes the signaling and routing traffic. The aim of the control plan is to control and manage the infrastructure equipment, and to link it with the applications. It is composed of one or several controllers. This plan offers several abstractions and additional services such as algorithms for calculating paths and the discovery of authenticated hosts or users.
- Data plan: The data plan represents the physical infrastructure that includes user traffic. This infrastructure is composed of a set of interconnected equipment. Unlike traditional infrastructure, SDN data plan devices have no control functionality within the network that allows them to perform autonomous decision-making.

The SDN architecture offers a set of application programming interfaces (APIs) that allow network devices to be programmed via multiple languages which simplifies the implementation of common network services such as routing, security, access control, QoS, bandwidth, traffic engineering, management, energy efficiency and various policy management forms.

The SDN provides a faster and more efficient process than traditional network architectures, it also simplifies the management of large networks by assigning all monitoring and decision-making processes to the SDN controller. The SDN is based on two main elements: the controller and the SDN switches. OpenFlow messages are also used to establish flows and install them on switches.

The switches and their controllers communicate through the OpenFlow protocol which is used to define control messages (for example: received packets, the packet to be switched, and the switch statistics).

The OpenFlow switch is the equipment used to classify packets based on well-defined rules and corresponding actions “fig. 2”. It has a flow table; each entry in the flow table includes a set of packet fields and actions (for example, switching these packets, without changing the fields, or drop). The OpenFlow switch sends to the controller all packets for which it has no equivalent entry in its flow table. The controller then decides how to process this packet, delete or add an entry to the flow table, and how to transmit similar packets in the future. Rules/actions are defined by the Controller(s) using policies, these policies can be changed and modified easily and rapidly through the installation of new Controller software [13].

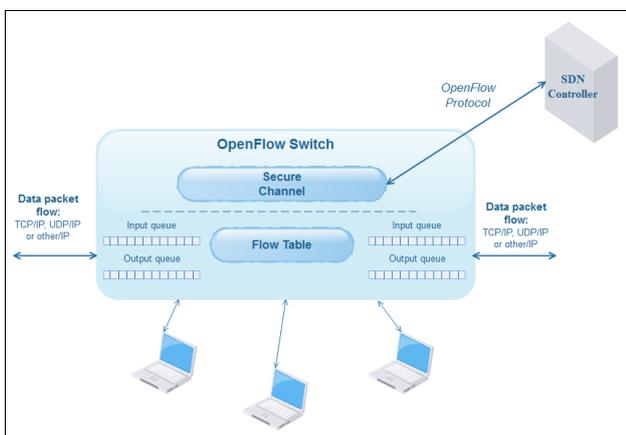


Fig. 2. The OpenFlow Switch

Openflow offers great flexibility in network flow routing. It is based on a paradigm that allows the switches to match not only the header fields of the layer 2 packets, but also the layer 3 and layer 4 fields. These flexibilities can be used in several fields, for example to deploy fine-grained traffic engineering[14], to enforce complex network policies[15], also to enhance resource utilization in wide area networks[16], and also to provide network virtualization in data centers.

III. MOBILITY MANAGEMENT IN WI-FI-BASED SDN

In the case of Wi-Fi-based SDN networks [17], the OpenFlow Switch resides at the access point itself, which allows the data plan to be placed directly on the access point. In fact, in a multi-vendor environment, SDN protocols provide the capability for interoperability between controllers and access points, making it possible to manage control and data plans. Another advantage of the SDN is that it allows centralized control of the network this can facilitates network optimization but also poses critical challenges in terms of scalability and robustness [18].

Actually, mobility protocols run in SDN network. In this section, we give an example of providing mobility in a Wi-Fi-based SDN which will allow us to solve several problems: overload, transfer delays, and to reduce the packet

packet loss compared to level 3 mobility in a traditional Wi-Fi network. SDN Wi-Fi Mobility is based on two essential elements, the OpenFlow controller and Access Point:

- The OpenFlow controller is one of the main functional entities in SDN mobility, its role is to make a decision on how to process the packet, delete or add an entry to the flow table of all Access Point in the SDN network.
- The OpenFlow Switches reside at the Access Point, they are in charge of the MN mobility and the exchange of OpenFlow messages with the controller via the OpenFlow protocol.

In order to clarify how SDN Wi-Fi Mobility works; “fig. 3” introduces a sequence diagram that describes all exchanges (steps, processes) that are implicated in the decision-making process carried out by the Controller. In this work, the mobility process can be divided into two steps: MN registration and MN handover. These two procedures are established by transmission of OpenFlow messages to notify MN events and to update the routing path. In the “fig. 3” it is assumed that the first access point is to which the MN is currently connected and the second access point is which the MN will be connected later.

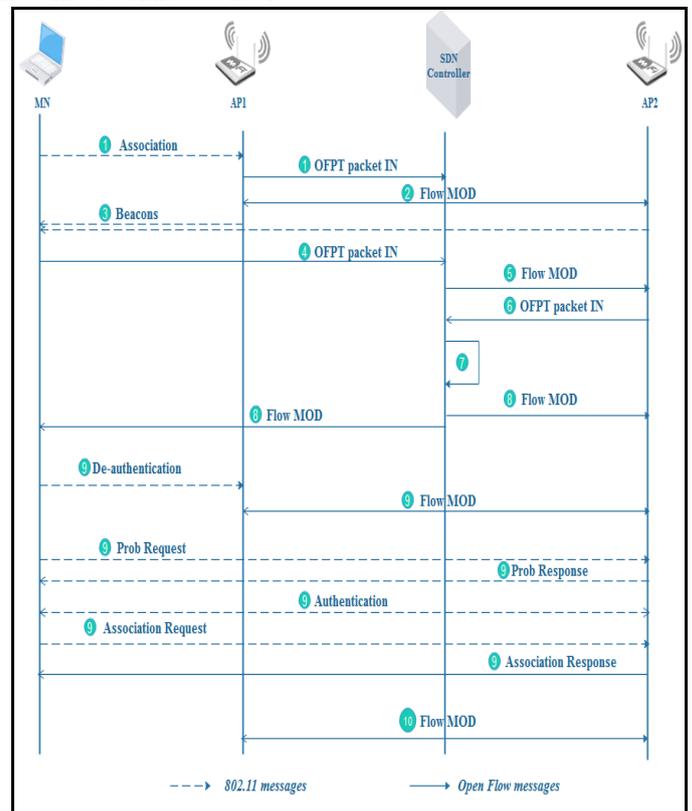


Fig. 3. SDN Wi-Fi Exchanges

1. When the AP detects MN association, it sends the OFPT PACKET IN message to its controller to inform that an MN has joined in the SDN WI-Fi network.
2. Once the controller has received this message, he treats it and sends FLOW MOD message to all the other APs to add the MN routing flow.
3. As usual, MN captured beacon frames that are continuously broadcast by the APs.



4. The MN extracts the RSSI values from the AP and sends it to the controller for comparison with a defined threshold to check the signal strength (OFPT PACKET IN send RSSI).

5. The Controller also periodically requests information regarding traffic at each access point (FLOW MOD getApInfo).

6. The APs transmit the responses to the controller (OFPT PACKET IN). All this traffic information are collected in order to select a destination access point. These activities are carried out continuously until the threshold is reached.

7. Once the threshold is reached, the controller decides that the MN must perform a handover to another AP. After comparing the RSSI values themselves, the controller choose AP with the highest RSSI value as destination.

8. A FLOW MOD message is sent by the controller to inform the APs of its decision, it also sends a message to the destination AP to install the flow (installFlow). Thus, when the MN migrates to the destination AP, the appropriate flow is already installed on it. At the same time, the controller starts the handoff process by sending to the MN a message (connecttoSelectedAP) containing the SSID of the destination AP.

9. When the MN is detached from AP1, it sends a De-authentication frame which indicate that it is disconnected from the source AP. The controller immediately sends the FLOW MOD message to all APs to delete the routing flow of MN. Second, MN deliver a Probe Request Frame in which it specify SSID of the destination AP. Consequently, only the concerned AP responds with the Probe Response Frame. Then, the MN association to the destination AP is completed and the remaining handoff frames are exchanged.

10. Once this message is received, the controller processes it and sends the FLOW MOD message to all APs to add the MN routing flow. Then the connection between MN and CN continues.

IV. EMULATION SETUP AND RESULTS

This paper studies the impact of introducing an SDN architecture in home and enterprise Wi-Fi networks. The proposed scenario includes three access points AP1, AP2, and AP3, two Openflow enabled switches S1 and S2, a fixed host that acts as a server and a mobile node (MN) that moves from one AP to another exchanging UDP traffic “fig. 4”.

To emulate scenarios, we used mininet-Wi-Fi [10]. This platform supports different mobility models such as random and linear waypoints. These mobility models can be considered to emulate mobile nodes moving from one access point to another.

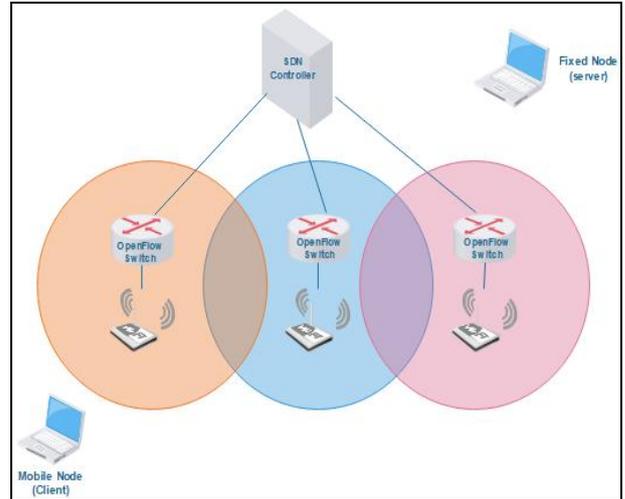


Fig. 4 . Emulated scenario

The emulation scenarios will be applied in two cases of study: In traditional Wi-Fi architecture and in Wi-Fi-based SDN architecture to analyze the results. This will help to determine which model behaves more efficient and fast even when mobility speed and traffic load (number of packets) are increased. In this evaluation, we use the same topology for both networks. The simulation parameters are mentioned in the table below:

TABLE II. SIMULATION PARAMETERS

Mobility model	Straight line (net. StartMobility and net.mobility methods)
Type of traffic	UDP
Type of controller	POX
Access point configuration	802.11n, TX-range=30 meters, TX-power=13, frequency=2.4GHz
Mobility speeds	1m/s, 5m/s, 10m/s
Emulation platform	Mininet-Wi-Fi
Packet size	75 00 bytes, 15 000 bytes, 30 000 bytes, 45 000 bytes.

As traffic generator, we used Iperf, which is a common tool for performance evaluation. We have executed the Iperf and made mobile node as a client and the fixed node as a server for different UDP traffic (20 packets per second for a different size). Results (Delay, throughput and packet loss) are obtained directly using Iperf terminal:

- Delay refers to the time of packet transmission across a network from source to destination.
- Packet loss is defined as the number of lost message, usually in percentage.
- Throughput indicates the average data successful rate transmission over a channel, measured in bits per second (bps).

All graphs are plotted against the mobility speed in both, traditional Wi-Fi network and Wi-Fi-based SDN; and it may be observed that the second architecture gives better results than the first one in all different scenarios even when increasing mobility speed and traffic load.



As showed in the following figures “fig. 5, 6, 7”, Packet loss increases and throughput decreases when mobility speed has increased from 0 m/s to 10 m/s. For delay, it increases when the position of the station changes from fixed to mobile, and decreases when the mobility speed increases from 1 m/s to 10 m/s. It has to be noticed that 18 graphs are plotted (representing the results of the 12 scenarios) as shown in the figures “fig. 8, 9, 10”. The figures “fig. 5, 6, 7” represents a detailed graphs for one UDP traffic (45000 bytes).

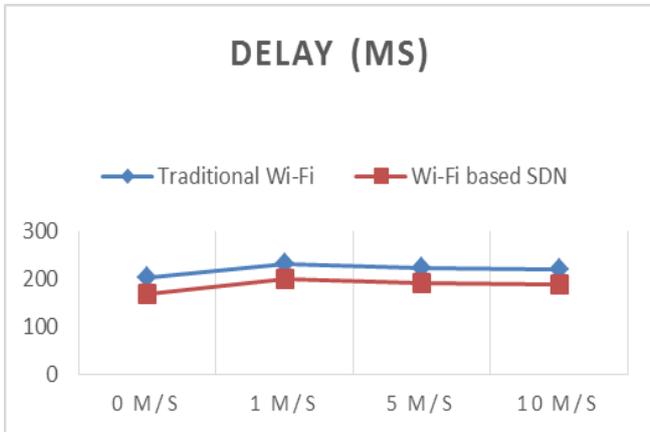


Fig. 5. Delay for 45000 bytes

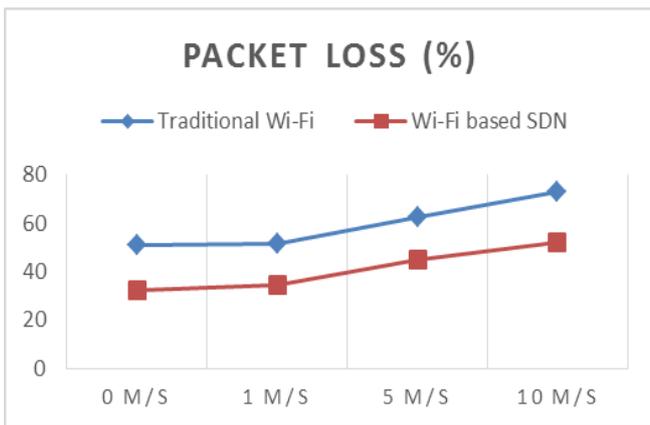


Fig. 6. Packet loss for 45000 bytes

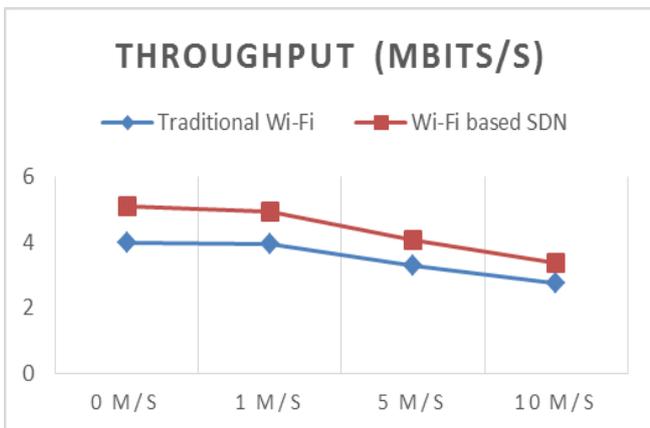


Fig. 7. Throughput for 45000 bytes

traffic increased in the two cases, fixed and mobile client. We can also observe a degradation of the network performance for both scenarios when the mobility speed increases, which is totally normal due to Wi-Fi transmission features.

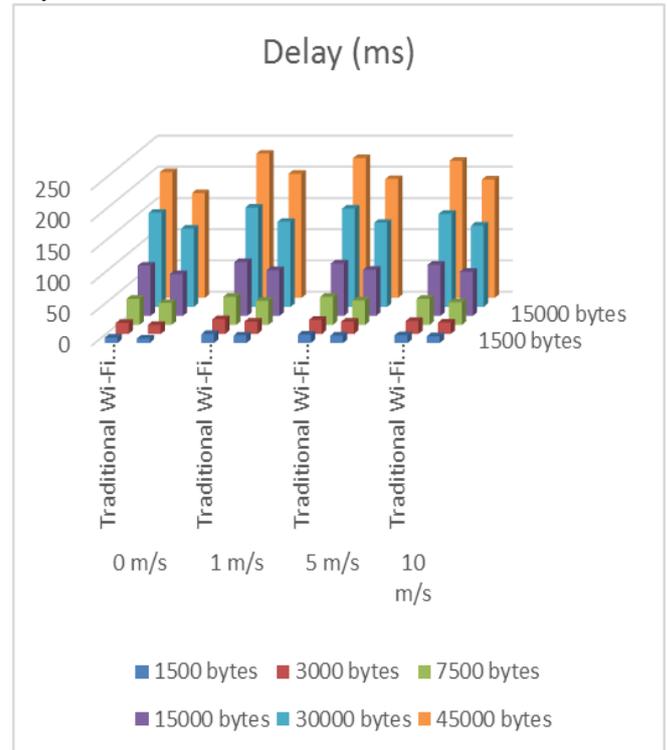


Fig. 8. UDP traffic Delay (ms)

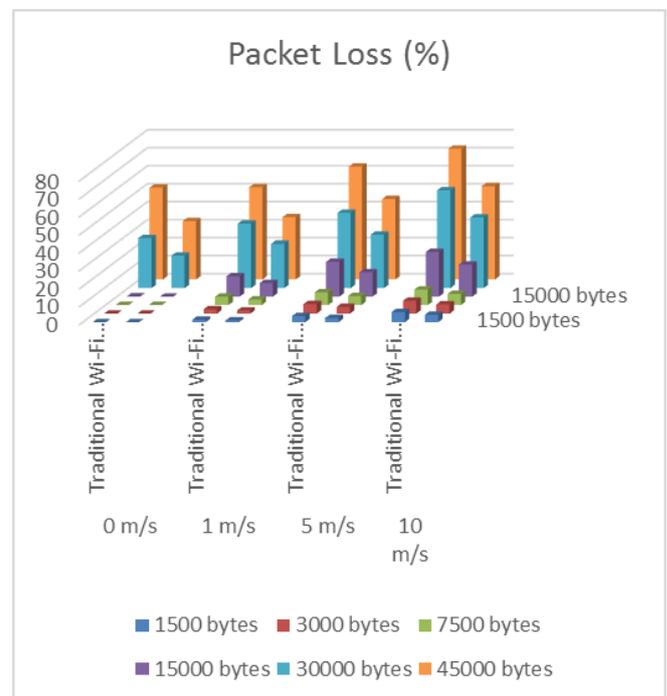


Fig. 9. UDP traffic Packet Loss (%)

Fig. 10. UDP traffic (7500 bytes) Throughput (Mbps/s)
Improvement percentage is calculated when SDN technology is implemented for fixed and mobile scenarios and for different mobility speed. Results show that the introduction of SDN gives better results for fixed than mobile scenarios.

“Fig. 8, 9, 10” summarize results of delay, packet loss and throughput for all emulated scenarios. From the results obtained, we deduce that a significant improvement of Wi-Fi network performance has been achieved with the introduction of the SDN. The quality of service is improved even when



When mobility speed is increased results deteriorates as can be seen in table III.

TABLE III. IMPROVEMENT PERCENTAGE

	0 m/s	1 m/s	5 m/s	10 m/s
Delay	16,8 %	14,6 %	13,7 %	13,1 %
Throughput	27,7 %	25,4 %	23,8 %	22,9 %
Packet Loss	35,9 %	32,2 %	29,7 %	28 %

V.CONCLUSION

The SDN enabled by OpenFlow decouples the control of the network from its data plane, and therefore it allows network designers and operators to simplify network operations and deploy innovative network applications. In this paper, we first have identified the challenges when mobility occurs, and the need of SDN to solve these issues; next, we described the different exchanges between SDN controller and other Wi-Fi equipment; finally, we emulated and compared traditional Wi-Fi architecture to Wi-Fi-based SDN architecture in the case of mobility using UDP for different traffic size and mobility speed. The outcomes show that the scenarios based on SDN provided a better results and transparent mobility for Wi-Fi over IP networks. We emulated our scenarios using mininet-Wi-Fi. The results obtained are very promising and suggest the applicability of the second scenario. However, there are many possibilities and ideas for improvement that will be discussed in a future work.

REFERENCES

1. H. Gacanin and A. Ligata, "Wi-Fi self-organizing networks: Challenges and use cases," IEEE Communications Magazine, vol. 55, no. 7, pp. 158–164, 2017. View at Publisher • View at Google Scholar •View at Scopus.
2. H. Sounni, N. Elkamoun, F. Lakrami "Performance Evaluation of Wireless 802.11n Using Level 2 and Level 3 Mobility", Indian Journal of Science and Technology, Vol 11(14), DOI: 10.17485/ijst/2018/v11i14/120616, April 2018.
3. Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," 2012.
4. D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, January 2015.
5. B. Lantz, B. Heller and N. McKeown, "A network in a laptop: rapid prototyping for software defined networks," in The Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, October 2010, pp. 1–6.
6. D. Klein and M. Jarschel, "An OpenFlow extension for the OMNeT++ INET framework," in 6th International ICST Conference on Simulation Tools and Techniques (SimuTools '13), March 2013, pp. 322–329.
7. D. Pitt, "Open Networking Foundation," 2012. [Online]. Available: <http://opennetworking.org>.
8. N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. "NOX: Towards an operating system for networks". SIGCOMM Comput. Commun. Rev., 38(3):105–110, Jul 2008.
9. B. Heller, R. Sherwood, and N. McKeown. "The controller placement problem". In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 7–12, New York, NY, USA, 2012. ACM.

10. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. "OpenFlow: enabling innovation in campus networks". SIGCOMM Comput. Commun. Rev., 38(2):69–74, Mar 2008.
11. Network Development and Deployment Initiative (NDDI). "Open Science, Scholarship, and Services Exchange (OS3E)". <http://www.internet2.edu/network/ose/>, 2012.
12. R. Fontes, S. Afzal, S. Brito, M. Santos, and C. Esteve Rothenberg, "Mininet-Wi-Fi: emulating Software-Defined wireless networks," in 2nd International Workshop on Management of SDN and NFV Systems, 2015(ManSDN/NFV 2015), (Barcelona, Spain), Nov. 2015.
13. Le Protocole OpenFlow dans l'Architecture SDN (Software Defined Network). www.effort.com
14. A. Mendiola, J. Astorga, E. Jacob, M. Higuero," A Survey on the Contributions of Software-Defined Networking to Traffic Engineering". IEEE communications surveys & tutorials, vol. 19, no. 2, second quarter 2017.
15. B. Devin, A. Gupta, T. Darrell, P. Abbeel, S. Levine. "Learning Modular Neural Network Policies for Multi-Task and Multi-Robot Transfer". ArXiv: 1609.07088v1 [cs. LG] 22 Sep 2016.
16. J. Peta"ja"rvi, K. Mikhaylov, M. Pettissalo. "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. International Journal of Distributed Sensor Networks"; 2017, Vol. 13(3), DOI: 10.1177/1550147717699412.
17. M. Aqsa, Q. Junaid, A. Basharat, Y. Kok-Lim Alvin, U. Ubaid. "QoS in IEEE 802.11-based Wireless Networks: A Contemporary Survey". ArXiv: 1411.2852v1 [cs.NI] 11 Nov 2014.
18. A. Nguyen and T. Kim. "An SDN-Based Connectivity Control System for Wi-Fi Devices". Article ID 9359878, 10 pages. Volume 2018.

