Testbed Implementation for Routing WLAN Traffic in Software Defined Wireless Mesh Network

Won Jin Lee, Jung Wan Shin, Hwi Young Lee, and Min Young Chung
College of Information and Communication Engineering, Sungkyunkwan University
300, Suwon, 16419, Republic of Korea
Email: {reign0208, withmyjw, lhy152, mychung} @skku.edu

***Abstract*—Software defined network (SDN) is a promising network technology that separates the control plane from data plane. The network function can be defined by using open application programming interface (Open API) in the SDN. Therefore, the network operator can embed the network function in the SDN controller by developing the application source code. Since the SDN brings benefit of network especially in management perspective, it is considered applying SDN to the wireless mesh network (WMN) which has to manage a lot of routers in the networks. In this paper, the implementation of software defined-wireless mesh network (SD-WMN) is introduced. We develops both the AP (Access Point) on the open source wireless router (OpenWRT) platform. We also develops the open virtual switch (OVS) on the Raspbian platform in order to provide the interface between SDN controller and AP. Using SDN controller, the WMN topology can be seem in the graphic user interface (GUI) of SDN. Finally, we measure the performance of AP in our WMN testbed environment.**

I. Introduction

Software defined network (SDN) is a promising network paradigm, and it has received a lot of attention for the future networks[1]-[5]. In SDN, data plane and control plane are separated for centralized network which enable to manage entire network devices via open interfaces. Since all network devices connect to the SDN controller physically or virtually, it is easy to manage the network system. SDN controller provides open interface which is called open application programming interface (Open API) between the controller and network devices in order to exchange the control information. Through the Open API, specific programmable software can be embedded in network device in order to manage the entire network system using OpenFlow protocol[3]-[7].

Wireless mesh network (WMN) is a subset of the wireless ad-hoc network. In ad-hoc network, each router forwards their packet to its neighbor, and the routing can be changed dynamically depends on the link state or link quality between routers[8]. Similarly, WMN consider complex network environment with a lot of routers. In the mesh network, the router is considered with little or no mobility and only occasional route fluctuations occur. In WMN, one of the router acts as a gateway which is directly connected to the Internet. Mesh routers forward their traffic to and from the gateway using wireless medium in order to access the Internet. Mesh router exchange the control information to update the topology, so WMN can operate well even if one of mesh router is malfunctioning or additional mesh router is added.

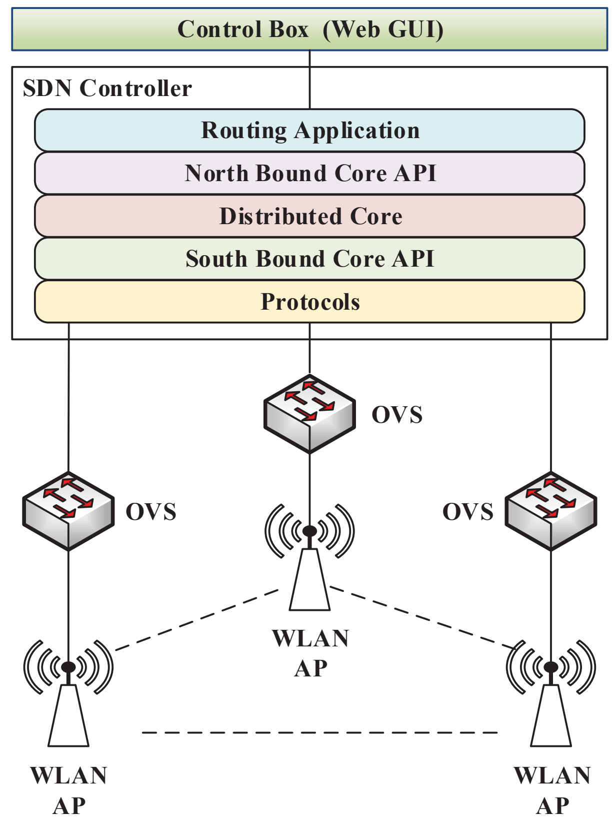

Fig. 1. The structure of SDN controller in WLAN environment

The combination of WMN and SDN has been studied by different industry and academic institution[9]-[11]. Since traditional WMN is difficult to manage and upgrade[11], a logically centralized SDN controller provides not only the maintenance but also management the WMN. Therefore, we aim at developing practical test environment of software defined-wireless mesh network (SD-WMN) in this paper. System architecture of SD-WMN is shown in Figure 1. The SDN controller exchanges control message with wireless local area network (WLAN) access point (AP) via open virtual switch (OVS). Therefore all the WLAN APs can be managed by centralized SDN controller. In order to provide the interface between SDN controller and network devices, OVS is also developed in our testbed.

978-1-4673-9991-3/16/$31.00 ©2016 IEEE

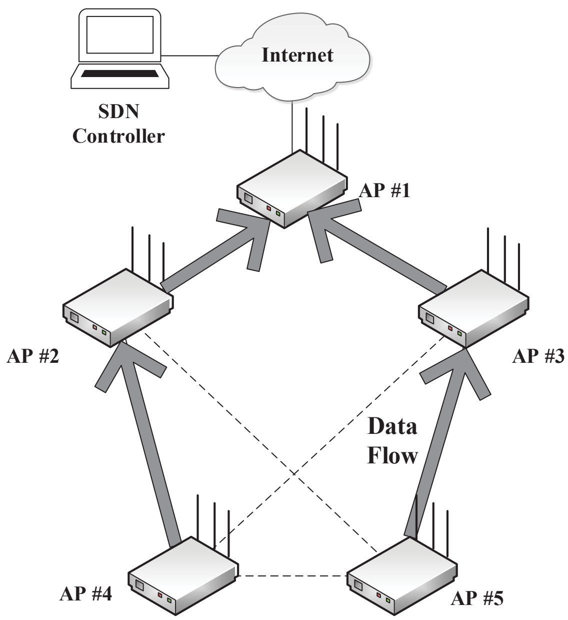

Fig. 2. The overview of our testbed environment

3个树莓派和2个AP

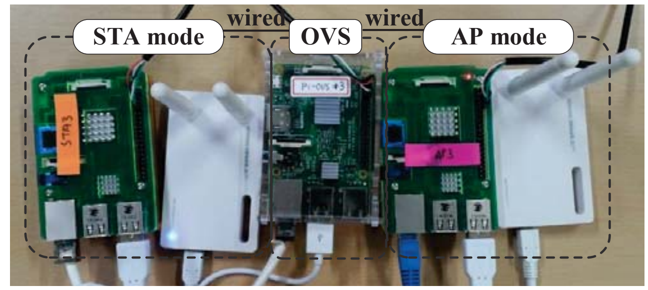

Fig. 3. The composition of WLAN AP using Raspberry Pi

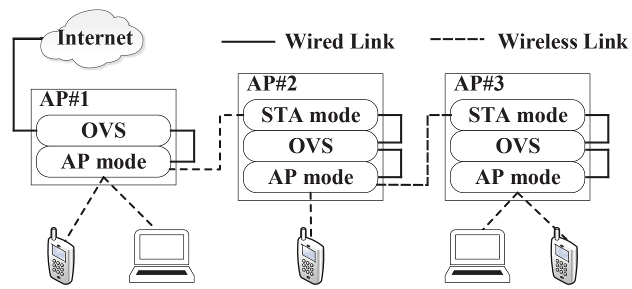

Fig. 4. Wired and wireless link connection of our network devices

The reminder of this paper is organized as follows. Section 2 presents the implementation of SD-WMN. The experimental results are discussed in Section 3. Then, this paper conclude in Section 4. Finally, we introduce our future works to enhance our current testbed for SD-WMN in Section 5.

II. Implementation of SD-WMN

In this section, the environment of our testbed for SD-WMN is introduced. Figure 2 shows deployment of our testbed environment. Our testbed consists with a SDN controller and five WLAN APs. Among all the APs, only AP1 is connected to the Internet with Ethernet cable. The other APs are not connected to the Internet directly, however, they can access the Internet via AP1 by transmitting the packet in wireless medium. All APs can change their routing path only if it has more than one neighbor AP. For example, the AP5 in Fig. 2 can change its routing to the AP3 or AP4 dynamically in order to achieve higher performance for both transmitting and receiving the packet.

One AP entity is consisted with three raspberry pi devices as following: AP mode open source wireless router (OpenWRT), STA mode OpenWRT, and OVS which is shown in Fig 3. AP mode OpenWRT provides the WLAN service to the client such as smart phone, lap top, etc. STA mode OpenWRT associates with adjacent AP mode OpenWRT. Since STA mode OpenWRT connects to the AP mode OpenWRT, traffic can be go through between the AP mode and STA mode of OpenWRT. The OVS provides interface between the SDN controller and its subordinate network devices by using OpenFlow. OpenFlow is a flow-oriented protocol which enables direct access to and manipulation of forwarding plane of network device.

In this paper, we develop the WLAN AP with raspberry pi 2 model B and OpenWRT in our works. Raspberry pi is a single board computer which is equipped with ARM central processing unit (CPU). OpenWRT is a kind of operating system (OS), which provides router platform based on the Linux[12]-[13]. OpenWRT also provides its package which is necessary for routing functions. Some packages which enable to perform WLAN AP of the raspberry pi is installed on OpenWRT. In this work, the N600UA of IPtime is used for WLAN card to provide the WLAN service. Since OpenWRT provides the web GUI named LUCI, it is easy to set the wireless setting such as frequency band, channel, and transmission power of AP. Then by selecting the mode of OpenWRT, the OpenWRT performs AP mode or STA mode.

OVS is a open source implementation of a distributed virtual switch. OVS provides a switching stack for hardware virtualization and connectivity between the SDN controller and network devices. Raspbian is a Debian-based computer operating system for Raspberry Pi. After installing the Raspbian on the Raspberry Pi, it is necessary to install some packages to perform the OVS such as *brcompat* and *datapath-source*. Then, add the OVS bridge and port to generate virtual interface. By using the IP of the SDN controller and port number, execute the command on the command line interface (CLI). Then, the connectivity between OVS and SDN controller is generated.

We consists our SD-WMN testbed in our lavatory, and in our testbed the open network operating system (ONOS) is used for the SDN controller. Figure 5 shows that the web GUI of ONOS which contains five OVSs, APs, STAs, and lots of clients. The

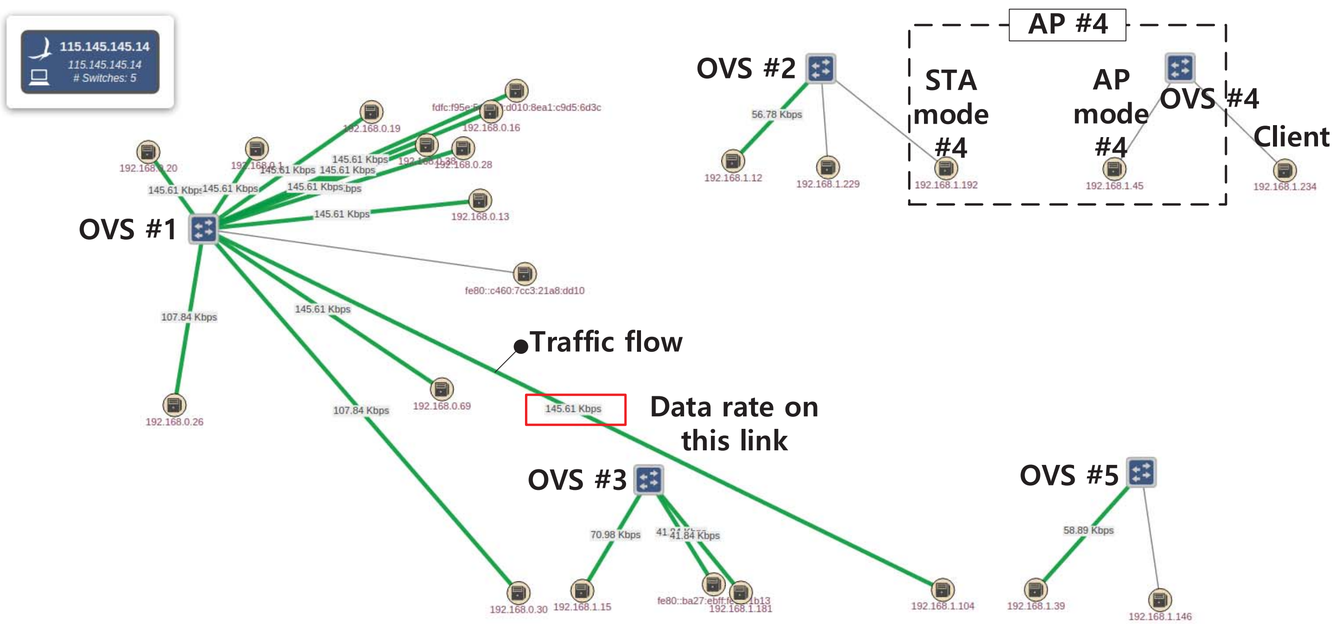

Fig. 5. Our testbed environment in ONOS web graphic user interface

OVS1 connects to the Internet via Ethernet hub which connects to several clients such as PC and smart phone in our lavatory. Therefore, it seems to lots of devices are connected to the OVS1. STA and AP mode OpenWRT connect to the OVS, and it seems like a host in the ONOS web GUI. The interest thing is that the STA is connected to neighbor OVS of which the STA associated with. For example the STA4 connects to the OVS2 in ONOS web GUI, even though it connects to the OVS4 with the Ethernet cable. Most impressive thing is the that the ONOS web GUI can shows the data traffic with green bold line as shown in Fig 5. Furthermore, the ONOS web GUI provides the data rate of each link for all the hop between network devices[14],[15].

By using powerful management function of the ONOS controller, the mesh network can be easily managed such as topology update, traffic monitoring, and realizing of routing algorithm practically. In order to make it feasible, the interface should be set up between ONOS controller and WLAN AP. Currently we focus on the method for providing the connectivity between ONOS controller and WLAN AP by using socket programming. After setting up the interface, it is expected that the practical SD-WMN works can be feasible.

III. Experimental Results

In this section, we shows the performance of APs in our testbed. In this experimental test, five APs are deployed in our laboratory (580 cm x 820 cm) consisting wireless mesh network as shown in Fig. 6. Among these APs, only AP1 connects to the Internet, and STA mode OpenWRTs of AP2 and AP3 associated with AP mode OpenWRT of AP1. Therefore, AP2 and AP3 forward their packet to and from the AP1 through wireless relay transmission. Similarly, AP4 and AP5 associate with AP2 and AP3 respectively. Therefore, all of the packet in this environment deliver to the AP1 in order to provide the Internet service to the client.

这种拓扑下，AP4和AP5的速率会很慢

We measure the performance of AP with smart phone application named BenchBee for practical measurements[16]. Table 1 shows the results in our experimental test. We took average of 10 trials for all the results. When smart phone associates with AP1, the performance shows the best with whole parameters compare with some other APs. Since the number of hop between smart phone and AP1 is lowest, it is possible to achieve maximum throughput of the performance both up-link and down-link cases. However, when the smart phone associate with AP2 or AP3, the throughput decrease significantly. The delay and loss rate increase with 2-hop APs. It is the reason the number of hop between smart phone and AP1 increases. Same as this, when the smart phone associate with AP4 or AP5, the performance of APs decrease, and it shows the lowest performance since the number of hop is maximum.

IV. Conclusion

In this paper, we developed the testbed for wireless mesh network based on the SDN. In SDN, centralized controller easily manage the network by separating its control plane from the data plane. Thus, applying SDN concept to the wireless mesh network which generally have a lot of nodes brings advantage for managing the networks. In this paper practical SD-WMN testbed is introduced. The ONOS controller is used for SDN controller, and the AP is developed using Raspberry Pi on the OpenWRT platform. In order to provide the interface between controller and AP, the OVS is also developed using Raspberry Pi on the Raspbian platform. Furthermore, we showed the ONOS web GUI with our SD-WMN environment consist with 5 APs. However, we could not show the experimental result

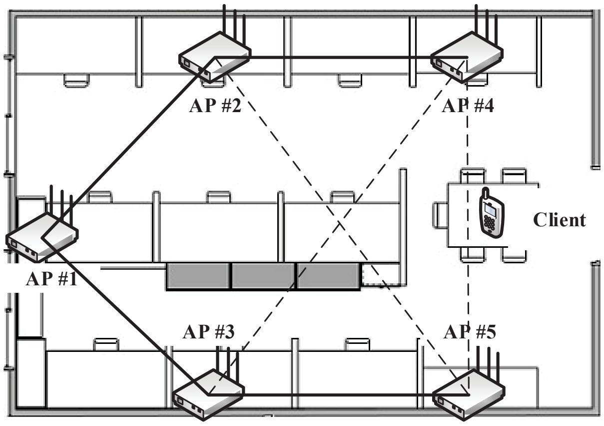

Fig. 6. The deployment of WLAN test in practical environment

TABLE I
EXPERIMENTAL RESULTS OF WLAN AP PERFORMANCE

Para-meters	Throughput [Mbps]		Delay [ms]			Loss Rate[%]
	Down Link	Up Link	AVG	MIN	MAX	
AP#1	5.74	6.75	51.96	7.00	188.75	0.25
AP#2	1.01	2.35	70.04	12.63	244.13	1.25
AP#3	0.84	2.24	84.65	12.75	296.75	9.50
AP#4	0.67	0.79	93.58	12.25	301.50	5.75
AP#5	0.60	0.78	95.43	12.54	311.21	6.13

with ONOS based WMN at current. So, we are focus on the interaction between ONOS controller and AP, and put this work for the future works.

V. FUTURE WORKS

The main goal of our future works is the enhancement of our current testbed for SD-WMN. First of all using the interface between centralized SDN controller and APs, the SDN and AP have to exchange control message. Based on this control information, the SDN controller can manage the WMN. After that, we will test the well-known routing algorithm such as Dijkstra's algorithm or distance vector algorithm. Then we will proposed a novel routing algorithm which consider the traffic load of each AP. Finally, we will show the advantage of centralized SD-WMN.

ACKNOWLEDGMENT

Prof. Min Young Chung is the corresponding author (e-mail: mychung@skku.edu).

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0190-15-2013, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks)

REFERENCES

[1] Marco Canini, et al, "Software-Defined Networks: Incremental Deployment with Panopticon," IEEE Computer, Vol. 47, No. 11, pp. 56-60, Nov. 2014.

[2] Yong LI and Min Chen, "Software-Defined Network Function Virtualization: A Survey," IEEE Access, Vol. 3, pp. 2542-2553, Dec. 2015.

[3] Sakir, Sezer, et al, "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communications Magazine, Vol. 51, No. 7, pp. 36-43, July 2013.

[4] Fei Hu, Qi Hao, and Ke Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," IEEE Communications Surveys & Tutorials, Vol. 16, No. 5, pp. 2181-2206, May 2014.

[5] Donghai Zhu, et al, "Towards Effective Intra-Flow Network Coding in Software Defined Wireless Mesh Networks," IEEE Computer Communication and Networks (ICCCN) 24th International Conference on, pp. 1-8, Aug. 2015.

[6] Andrea Detti, et al, "Wireless mesh software defined networks (wmSDN)," IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob), pp. 89-95, Oct. 2013.

[7] Fan Yang, et al, "OpenFlow-based load balancing for wireless mesh infrastructure," IEEE Consumer Communications and Networking Conference (CCNC), pp. 444-449, Jan. 2014.

[8] Ian F. Akyildiz and Xudong Wang. "A survey on wireless mesh networks," IEEE Communications Magazine, Vol. 43, No. 9, Sep. 2005.

[9] F. Houeto, S. Pierre, R. Beaubrun, and Y. Lemieux, "Reliability and cost evaluation of third-generation wireless access network topologies: a case study," IEEE Transactions on Reliability, Vol. 51, No. 2, pp. 229-239, Jun 2002.

[10] Y. D. Lin, et al. "Design issues and experimental studies of wireless LAN mesh," IEEE Wireless Communications, Vol. 17, No. 2, pp. 32-40, Apr. 2010.

[11] H. Huang, P. Li, S. Guo, and W. Zhuang, "Software-defined wireless mesh networks: architecture and traffic orchestration," IEEE Network, Vol. 29, No. 4, pp. 24-30, July 2015.

[12] C. E. Palazzi, M. Brunati, and M. Roccetti, "An OpenWRT solution for future wireless homes," IEEE Multimedia and Expo (ICME), pp. 1701-1706, July 2010.

[13] M. Kciuk, "OpenWRT operating system based controllers for mobile robot and building automation system students projects realization," IEEE Research and Education in Mechatronics (REM), pp. 1-4, Sep. 2014.

[14] B. Grasic , et al, "Use of open source operating system and TCP/IP connectivity in urban environmental monitoring. In: Industrial Technology," 2003 IEEE International Conference on. Vol. 2, pp. 1257-1261, Dec. 2003.

[15] Zhiming Tan, et al, "An implementation of open source operating system on multiprocessor system-on-a-chip," Consumer Electronics IEEE Transactions on, Vol. 52, No. 3, pp. 1118-1123, Aug. 2006.

[16] Zhang, Lin, et al. "UNOTest: An efficient traffic performance test platform in heterogeneous networks," IEEE Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, pp. 1-5, June 2013.