Implementation and Evaluation of Hybrid Wired/Wireless SDN Switch

Jiayang CHEN[†a)], *Nonmember*, Kien NGUYEN[†,††b)], *Member*, *and* Hiroo SEKIYA[†c)], *Fellow*

SUMMARY A hybrid wired/wireless software-defined networking (SDN) switch is a networking device that can switch wired and wireless frames (e.g., Ethernet and IEEE 802.11) with SDN capability. This paper presents an implementation of such an SDN switch using an SDN controller (i.e., POX), Open vSwitch (OVS), a Wi-Fi interface (with IEEE 802.11n standard, and a 1 Gbps Ethernet one. The switch can operate when the Wi-Fi interface is in a client or access point (AP) mode. The switch can smoothly integrate the AP mode's interface. With the client mode, we develop an SDN module on the SDN controller to solve the addresses MAC address mismatch to effectively managed wireless communication. To validate the hybrid SDN switch, we build the switch with an IEEE 802.11n interface (in the client and access point modes) utilizing the 2.4 GHz band and a 1 Gbps Ethernet one. We then experiment with switching packets between a wired and a Wi-Fi network to show the switching capability when the Wi-Fi link is in the client and AP modes. In the evaluation with the client mode, the TCP and UDP throughput are approximately 98.60 Mbps and 106.41 Mbps, respectively. With the Wi-Fi interface in the AP mode, we compare the switch performance with a hotspot created by the Ubuntu Network Manager. The results show that the throughput and round-trip time (RTT) values are comparable, indicating the SDN will not cause significant overhead.

key words: *Hybrid switch, SDN, Ethernet, Wireless network*

1. Introduction

With the increase of mobile devices and applications, there is a growing demand for high-quality and flexible communication networks that can accommodate wired and wireless technologies [1], [2]. Wired communications offer low latency and high reliability but often require cables or lack mobility support [3]. In contrast, wireless networks excel in addressing these challenges. However, managing both types of networks separately can lead to complexity and inefficiency. To address the issue, a hybrid SDN switch that leverages SDN capabilities represents a promising solution. A hybrid wired/wireless SDN switch is a networking device that integrates both wired and wireless packet switching capabilities into a single switch. These switches can also employ SDN technology's programmability and centralized control [4] (e.g., with the assistance of the OpenFlow protocol).

Regarding the wireless network, this research focuses on the popular Wi-Fi/IEEE 802.11 technology. Normally, a Wi-Fi link of the hybrid switch can operate in two main modes: client mode and AP mode. In the former mode, integrating a Wi-Fi interface to an SDN switch raises the issue of covering Ethernet ports by a Wi-Fi client. That is because the Wi-Fi client's MAC address may overwrite that of the Ethernet port (i.e., the default settings of SDN switch), as highlighted in the previous research [5]. To solve the problem, we propose leveraging SDN to modify the packet headers. In the latter mode of Wi-Fi interface, the hybrid switch operates similarly to a Wi-Fi AP, providing connectivity and packet forwarding for Wi-Fi clients.

In this work, we implemented and evaluated a hybrid SDN switch with SDN controller, an Ethernet and a Wi-Fi interface (i.e., operating in the client and the AP modes). The Wi-Fi interface uses IEEE 802.11n, operating at the 2.4 GHz band, and the Ethernet interface has a capability of 1 Gbps. To show the switching capability, we measured TCP and UDP throughput in a simple scenario (i.e., switching from a wired to a wireless network). The results illustrate that the hybrid SDN switch achieved approximately 98.60 Mbps for TCP and around 106.41 Mbps for UDP. Moreover, we compared the hybrid SDN switch when the Wi-Fi interface is in the AP mode with a hotspot created by the Ubuntu Network Manager. The results show comparable throughput and RTT values, indicating the insignificant overhead of SDN.

The remainder of this paper is as follows. Section 2 includes related works. Section 3 and Section 4 present our methodology to construct the hybrid switch and evaluation results, respectively. Finally, Section 5 concludes the paper.

2. Related Work

SDN allows the network to be centrally managed, which makes it cost-effective, flexible, and efficient for resource management [6], [7]. In SDN, an SDN switch or a data plane is an essential component. The SDN switches have been extensively studied and widely deployed. In [8], the authors focused on the performance of SDN switches, like the security when facing attacks from the network. In [9], [10], SDN switches are used to provide guaranteed QoS and QoE, respectively. They used SDN switches in combination with micropayment. However, in the previous works, the typical SDN switch is based on wired switches that do not accommodate wireless networks.

There are efforts to extend SDN to wireless networks. Compared with wired networks, wireless networks allow users to connect to the network and access resources with-

[†]The authors are with Graduate School of Science and Engineering, Chiba University, 1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan. K. Nguyen is the corresponding author.
[††]The author is with the Institute for Advanced Academic Research, Chiba University 1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan.
a) E-mail: cega0749@chiba-u.jp
b) E-mail: nguyen@chiba-u.jp
c) E-mail: sekiya@faculty.chiba-u.jp

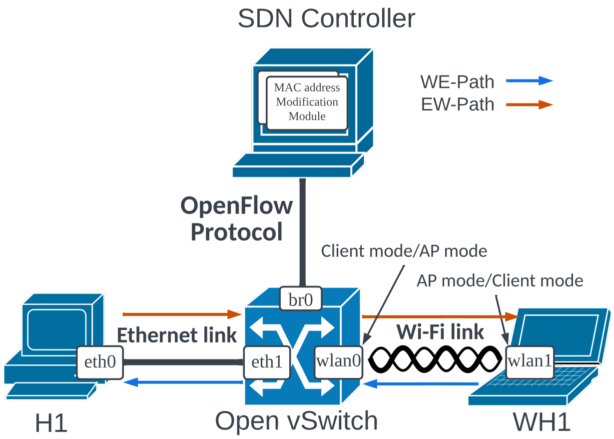


Fig. 1: Illustrating an SDN hybrid switch with Wi-Fi link in client and AP modes

out using physical cables. In [11], the author built the SDN-based APs to compare the POX and floodlight controller performance by measuring convergence time, packet loss rate, bandwidth, and network jitter. In [12], the author has implemented SDN using OpenFlow for wireless scenarios by incorporating the mobility functions and WiFi interfaces in the UAV-assisted networks. These two research were conducted based on the emulator, mininet-wifi, since they did not have the SDN switches that support wireless scenarios. The hybrid wired/wireless SDN switch has also attracted attention SDN will benefit both networks. In [13], the authors proposed an SDN-based wired/wireless hybrid network architecture of cloud data centers that realizes centralized management and flexible scheduling. The main disadvantage is that it only contains theoretical analysis without actual implementation. We will provide a method for a hybrid wired/wireless SDN switch.

3. Methodology

Implementing a hybrid switch involves utilizing various components, including an SDN controller, OVS, a Wi-Fi interface, and an Ethernet adapter. The hybrid switch in a simple scenario is depicted in Fig. 1. $eth0$ and $eth1$ are Ethernet ports, $wlan0$ and $wlan1$ are Wi-Fi interfaces, and $br0$ is the internal port connected with the SDN controller. The core of the hybrid switch is the OVS, which is integrated with the wired port ($eth1$) (Ethernet port) and the Wi-Fi interface ($wlan0$) using Wi-Fi technology connected with the SDN controller. In the simple scenario, we establish a wireless connection between the host and the hybrid switch while another host is connected to the hybrid switch via a wired connection. We define the WE-Path as the path that traffic goes from the wireless side to the Ethernet side and the EW-Path as the traffic goes reverse way. To establish the Ethernet link, we added the Ethernet port to the switch, connecting it to another device with cables. Additionally, we integrated the Wi-Fi interface into the switch. The Wi-Fi link after can be in the client or AP mode.

3.1 $Wlan0$ in the client mode

To let $wlan0$ communicate with an AP (i.e., $wlan1$ is in the AP mode), we use the wpa_supplicant tool. We created a configuration file, including information including SSID and password. After running, the tool establishes a Wi-Fi link from $wlan0$ to WH1, which uses the hotspot (running on $wlan1$) created by a network manager, supporting protocols such as Wi-Fi Protected Access (WPA) or Wi-Fi Protected Access 2 (WPA2). However, After creating the client mode, we encountered an issue related to physical connectivity. The wireless link did not communicate appropriately because the switch could not handle the physical and link layer tasks for the wireless connection [5]. When we added the Wi-Fi interface to become a port of the switch, the port's MAC address became the wireless link's. We call this a mismatch between MAC addresses, making it impossible for the Wi-Fi interface to communicate with the Open vSwitch. To solve the problem, referring to the algorithm in our previous paper [14], we modify the packet header in the SDN controller and then send the packets to the corresponding port of the hybrid switch to enable communication between different devices.

3.2 $Wlan0$ in the AP mode

To use the AP mode on $wlan0$ in the hybrid switch, there are two ways: using additional software (i.e., hostapd) or the hotspot created by the Ubuntu Network Manager. o use hostapd, the configuration file containing the information of interface, SSID, password, Wi-Fi channel, Wi-Fi standard, and so on, is necessary. Wpa_command is also necessary to create the wireless link, but the configuration file and the commands should be used in the client, similar to the procedure in the client mode. Both client and AP modes can be applied in implementing a hybrid switch. In addition, both modes are necessary to connect more hybrid switches.

4. Evaluation

4.1 Experimental setup

The evaluation setup is as the one illustrated in Fig. 1. In the figure, the central component of the hybrid switch is an OVS, which is locally connected to the SDN controller. Additionally, it connects to a wireless host (WH1), a host (H1) through Wi-Fi and Ethernet interface.The Wi-Fi interfaces we used for the interface ($wlan1$) in WH1 and that ($wlan0$) in the hybrid switch are TP-Link Archer T9UH and TP-Link Archer T3U, respectively. The Wi-Fi connection is IEEE 802.11n with the 2.4 GHz frequency band's channel one. In the client mode, AP is set to WH1 with the SSID to be 'chen'. The hosts (H1 and WH1) with different links can communicate after establishing the topology and solving the MAC address mismatch problem. In the AP mode, WH1 becomes

Table 1: Software and Tools

Name	version
Network Manager	1.36.6
Open vSwitch	2.17.5
wpa_supplicant	2.10-devel
hostapd	2.10-devel
Ubuntu system	22.04.2
iperf3	3.9
sockperf	3.10-0

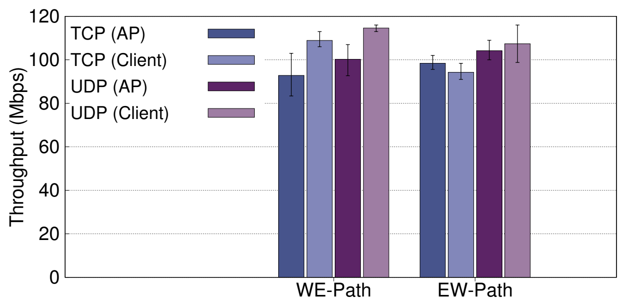


Fig. 2: Switching capacity in different Wi-Fi modes for different paths

the client, and $wlan0$ works in the AP mode. The version of software and tools we used in the evaluation are in Table 1.

4.2 Switching capability

This experiment aims to evaluate the switching capability of the hybrid switch with TCP and UDP traffic when the Wi-Fi interface is in the client and AP modes. More specifically, we sent traffic from WH1 to H1 (WE-Path) and from H1 to WH1 (EW-Path), with $wlan0$ being in the client or AP mode. First, we ran iperf3 to measure the throughput of TCP and UDP to show the switching capability. In the client mode, the iperf3 client started from the WH1 (WE-link scenario) and communicated to the iperf3 server on H1 in 10 seconds, and we repeated this experiment 10 times. We recorded the results of every experiment and calculated the average values, minimum and maximum. Subsequently, we experimented with $wlan0$ becoming the AP mode under the WE-Path and EW-Path scenarios. We then compared the throughput results in different modes and for different paths.

The evaluation results in Fig. 2 show the switching capability of our hybrid switch. The results indicate that the average switching capability in these two modes is approximately 98.60 Mbps and 106.41 Mbps in terms of throughput with TCP and UDP, respectively. Additionally, in the client mode, the throughput for the WE-Path is greater than that of the EW-Path, whereas, in the AP mode, the trend is reversed. This observation is sensible as the download link typically exhibits higher throughput than the upload link. Moreover, UDP throughput is more significant than TCP throughput due to that UDP is a connectionless protocol and lacks congestion control and retransmission mechanisms, resulting in higher throughput.

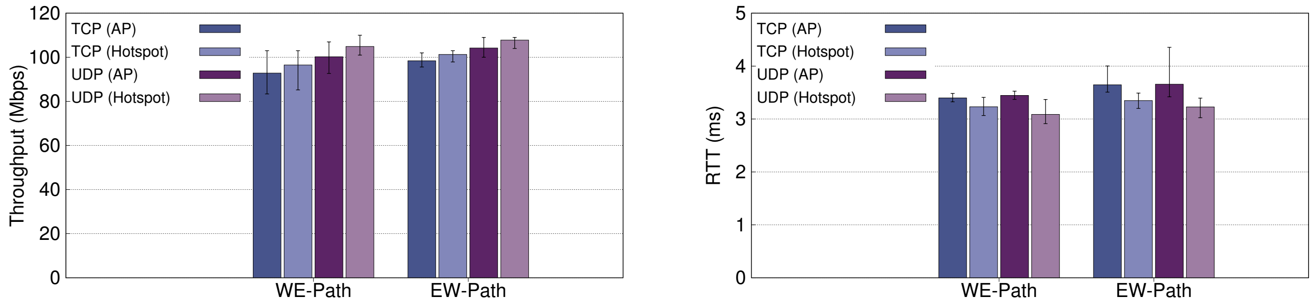
4.3 Performance comparison with the hotspot

We evaluated the performance of our hybrid switch with $wlan0$ being the AP (master) mode and compared the performance with the hotspot using the Ubuntu Network Manager. We used iperf3 to measure the TCP and UDP throughput in this scenario. In an experiment, iperf3 ran in 10 seconds. Moreover, each experiment is repeated 10 times. We also used sockperf, a network benchmarking tool used to measure and analyze the performance of TCP and UDP traffic (such as RTT) to measure the RTT values. We also experimented with one run in 10 seconds and repeated each ten times.

The TCP and UDP throughput of iperf3 experiments are shown in Fig. 3a. Considering the throughput results, we observed that for WE-link, the hotspot mode has slightly higher TCP throughput (3.68 Mbps for WE-Path and 2.86 Mbps for EW-Path) and UDP throughput(4.65 Mbps for WE-Path and 3.6 Mbps for the EW-Path) than that of the hybrid switch when the Wi-Fi interface is in AP mode. The results also show that the throughput of the EW-Path is more significant than that of the WE-Path, and the UDP throughput is higher than the TCP throughput. Next, we examined the RTT results for TCP and UDP using sockperf, shown in Fig. 3b. The existence of the extra Ethernet link caused RTT to increase, ranging from 0.16 ms to 0.43 ms in different situations. This is also reasonable since the Ethernet link may cause extra paths, increasing the RTT. By comparing these results, we can conclude that the switch can switch packets between the wireless network and the Ethernet without significantly degrading the network's throughput or introducing substantial delays. As a result, the SDN will not cause significant overhead.

5. Conclusions

In this paper, we implemented the hybrid wired/wireless SDN switch, capable of switching packets between wired and wireless networks with SDN capability. We have realized the implementation with OVS, Wi-Fi interface (with IEEE 802.11n standard), 1 Gbps Ethernet interface, and SDN controller. The switch can operate when the Wi-Fi interface is in a client or access point (AP) mode. The switch can smoothly integrate the AP mode's interface. To show the switching capability, we evaluated and compared the TCP and UDP throughput in client and AP modes. The experiments revealed that the hybrid switch exhibited a switching capability of approximately 98.60 Mbps for TCP throughput and around 106.41 Mbps for UDP throughput. Then, we also compared the performance in the AP mode with a hotspot created by the Ubuntu Network Manager. By comparing the performance of the AP mode with that of the hotspot, we can conclude that the hybrid switch effectively forwards packets between wireless and Ethernet networks without experiencing significant degradation in throughput or delays. As a result, the SDN will not cause significant overhead. In fu-



(a) (b)

Fig. 3: Performance comparison (AP: the switch with Wi-Fi's AP mode; hotspot: hotspot of Ubuntu Network Manager)

ture work, we plan to add more Wi-Fi interfaces and Ethernet ports into this switch to test its switching capability in larger-scale networks.

Acknowledgment

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) under Grant 20H0417, 23H03377 in part by the Japan Science and Technology Agency (JST) through the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2107.

References

[1] M. K. Atiq, R. Muzaffar, . Seijo, I. Val, and H.-P. Bernhard, "When ieee 802.11 and 5g meet time-sensitive networking," *IEEE Open J. Ind. Electron. Soc.*, vol. 3, pp. 14–36, 2022.

[2] Y. Takei, S. Nakatsukasa, M. Nishiguchi, and H. Ohnishi, "Poster: Fine-grained network identification and control combining routers and programmable switches," in *Proc. IEEE LANMAN*, pp. 1–4, 2021.

[3] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation ethernet, iiot, and 5g," *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[4] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on sdn and nfv for next generation iot," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5706–5712, 2020.

[5] K. Nguyen, P. L. Nguyen, Z. Li, and H. Sekiya, "Empowering 5g mobile devices with network softwarization," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 2492–2501, 2021.

[6] N. Ashodia and K. Makadiya, "Detection of ddos attacks in sdn using machine learning," in *Proc. IEEE ICEARS*, pp. 1322–1327, 2022.

[7] J. Ren, J. Li, H. Liu, and T. Qin, "Task offloading strategy with emergency handling and blockchain security in sdn-empowered and fog-assisted healthcare iot," *Tsinghua Sci. Technol.*, vol. 27, no. 4, pp. 760–776, 2022.

[8] U. Tupakula, V. Varadharajan, and K. K. Karmakar, "Attack detection on the software defined networking switches," in *Proc. IEEE NetSoft*, pp. 262–266, 2020.

[9] H. Masaki, K. Nguyen, and H. Sekiya, "Fine-grained qos provisioning with micropayments in wireless networks," *NOLTA, IEICE*, vol. 14, no. 1, pp. 50–65, 2023.

[10] J. Chen, H. Masaki, K. Nguyen, and H. Sekiya, "Qoe provisioning system for voip and video streaming using software-defined networking and iota micropayment," *IEICE Communications Express*, vol. X12-B, no. 12, 2023.

[11] A. El Sayed Fathy Ahmed and H. A. Elsayed, "Performance comparison of sdn wireless network under floodlight and pox controllers," in *Proc. IEEE ICEENG*, pp. 91–95, 2022.

[12] C. Singhal and K. Rahul, "Efficient qos provisioning using sdn for end-to-end data delivery in uav assisted network," in *Proc. IEEE ANTS*, pp. 1–6, 2019.

[13] T. Jing, L. Shidong, L. Chuan, B. Xiande, and Z. Gang, "A sdn-based wired/wireless hybrid network architecture of cloud data center," in *Proc. IEEE ITNEC*, vol. 5, pp. 232–236, 2021.

[14] J. Chen, K. Nguyen, and H. Sekiya, "Seamless integration of wireless interface to sdn switch," in *Proc. IEEE APNOMS*, pp. 219–222, 2023.