Off-the-Shelf Software-defined Wi-Fi Networks

Seppo Hätönen*, Petri Savolainen†, Ashwin Rao*, Hannu Flinck+, Sasu Tarkoma*†
*University of Helsinki, †Helsinki Institute for Information Technology HIIT, +Nokia Bell Labs

ABSTRACT

Wi-Fi networks were one of the first use-cases for Software-defined networking (SDN). However, to deploy a software-defined Wi-Fi network today, one has to rely on research prototypes with availability, documentation, hardware requirements, and scalability issues. To alleviate this situation, we demonstrate two simple techniques to bring SDN functionality to existing Wi-Fi networks and discuss their benefits and short-comings. Researchers can use our techniques to convert their existing Wi-Fi testbeds into software defined Wi-Fi testbeds. Our two techniques thus significantly lower the barrier-to-entry for deploying software-defined Wi-Fi networks.

CCS Concepts

•**Networks** → **Wireless local area networks;**

1. INTRODUCTION

Wi-Fi is becoming synonymous with Internet connectivity. However, in spite of its growing importance, Wi-Fi has received limited attention in the SDN community. Wi-Fi access points (APs) including those created using open-source solutions, such as *OpenWrt* [2] and *hostapd* [1], act as bridges or hubs between Wi-Fi clients. These Wi-Fi APs cannot take intelligent forwarding decisions because they cannot be programmed to process the complex match/action rules used by SDN switches such as Open vSwitch (*OVS*) [4].

In this paper, we present two simple techniques, namely Intelligent Edge and Thin Edge, that bring SDN functionality to Wi-Fi networks. These techniques leverage on wireless isolation and SDN switches such as *OVS* to simplify the integration of Wi-Fi networks and SDN.

SIGCOMM '16, August 22–26, 2016, Florianopolis, Brazil

 ISBN 978-1-4503-4193-6/16/08…$15.00

DOI: http://dx.doi.org/10.1145/2934872.2959071

The Intelligent Edge technique empowers devices running *OpenWrt* with *OVS*. In contrast, the Thin Edge technique allows existing Wi-Fi APs which support wireless isolation to offload the flow management to SDN switches and be integrated as-is into SDN.
Our key contributions are as follows.

• Our two techniques enable existing SDN controllers to manage the flows traversing Wi-Fi networks built using off-the-shelf components. Using these components, *we significantly lower the barrier-to-entry to deploy and experiment on software-defined Wi-Fi networks.*

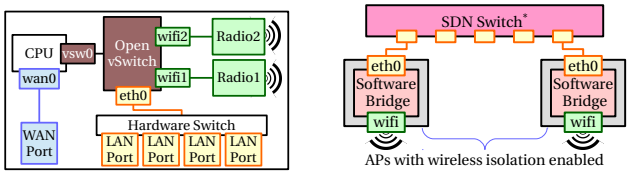
• Our Intelligent Edge technique allows SDN controllers to leverage the processing power of the APs for managing the edge of Wi-Fi networks. Furthermore, by combining *OVS* with *OpenWrt*, this technique opens avenues for SDN research on existing Wi-Fi testbeds which use devices running *OpenWrt*.

• Our Thin Edge technique offloads the flow management to either SDN switches or hosts running *OVS*. This technique is useful for managing Wi-Fi networks and testbeds which use APs and routers that either cannot support SDN, or are not powerful enough to run *OVS* and process the complex match-action rules supported by OpenFlow [3] and other SDN protocols.

The seminal work on bringing SDN to Wi-Fi networks was OpenRoads [7], which used protocols such as the Simple Network Management Protocol (SNMP) to manage the Wi-Fi APs. Based on the insights of OpenRoads, several solutions such as BeHop [8], ÆtherFlow [6], and OpenSDWN [5] have been proposed. However, these solutions suffer from deployability issues which make it hard to convert existing Wi-Fi networks and testbeds into software-defined Wi-Fi networks. We therefore focus on enabling SDN in Wi-Fi networks built using off-the-shelf components.

2. OUR SOLUTION

We present two techniques to create software defined Wi-Fi networks using off-the-shelf components. Our techniques build on Wireless Isolation, a feature that configures an AP to process all the wireless frames in the network stack instead of just bridging wireless clients associated with that AP. Wireless Isolation is supported by many APs including those running *OpenWrt*, and some enterprise APs such as the Cisco 1131 AP.



(a) **Intelligent Edge**. (b) **Thin Edge**.

Figure 1: **Our Intelligent Edge and Thin Edge techniques**. *For the Intelligent Edge, OpenWrt's software bridge is replaced with an OVS and wireless isolation is enabled on the AP; a patched* hostapd *uses the OVS while allowing Wi-Fi clients to encrypt their Wi-Fi frames. For the Thin Edge, Wireless Isolation is enabled on the APs and the forwarding decisions are taken on an external SDN switch; all the data encapsulated in Wi-Fi frames traverse the external SDN switch.*

Intelligent Edge. In this technique, we run *OVS* on an AP running *OpenWrt*. As shown in Figure 1(a), we replace the software bridge created by *OpenWrt* with an *OVS*; all the interfaces that used the software bridge now use the *OVS*. This step is not enough for the *OVS* to manage flows between two Wi-Fi clients associated with the AP. To manage this Wi-Fi traffic, we configure the Wi-Fi interfaces to send all the packets encapsulated inside the Wi-Fi frames to the *OVS*. We also use a patched *hostapd* to use the *OVS* while allowing Wi-Fi clients to encrypt their Wi-Fi frames. The Intelligent Edge technique also supports multiple (virtual) Wi-Fi networks on the same AP. Each virtual network appears as a logical Wi-Fi interface on *OpenWrt*, which is added to the *OVS*. These interfaces can be treated as virtual APs and the network flows are controlled by *OVS*.

Thin Edge. In this technique, an external SDN switch manages all the flows traversing the Wi-Fi APs. Thin Edge can be used with APs which support Wireless Isolation but do not support custom firmware such as *OpenWrt*, allow installing an SDN switch, or have hardware limitations such as CPU or flash memory size. Furthermore, as this technique only requires Wireless Isolation, many older devices which support *OpenWrt* can now be used for SDN research. At the same time, enterprise equipment which do not support custom firmware but support Wireless Isolation can also be integrated into these testbeds. Thus the Thin Edge can bring SDN to existing Wi-Fi networks or testbeds. In this technique, the AP acts as a remote Wi-Fi interface for the SDN switch and each Wi-Fi network of the AP becomes a port on that switch. A key shortcoming of this technique is that all the Wi-Fi traffic needs to traverse an external switch; the flows are not routed optimally and the network can become congested.

Discussion. Our techniques combine widely deployed technologies, Wireless Isolation and SDN switches such as *OVS*, and can therefore be used in almost all existing networks including Wi-Fi testbeds and also enterprise networks. Furthermore, our techniques enable networks running legacy Wi-Fi devices such as Wi-Fi testbeds to be powered by SDN, thus lowering the barrier-to-entry for doing SDN research on Wi-Fi networks.

3. DEMO DESCRIPTION

We use an *OpenWrt* AP with *OVS* for the Intelligent Edge, and an enterprise AP (Cisco 1131 Autonomous AP) without any firmware changes for the Thin Edge.

We demonstrate our techniques using Google Chromecast. A Chromecast device is typically visible to all clients in its same network, which may not be desirable in campus or enterprise networks. Our two techniques that leverage on SDN and Wireless Isolation restrict access to the Chromecast device to a given Wi-Fi client. In its default mode, Wireless Isolation blocks all Wi-Fi clients associated to a given AP from communicating with each other. We use SDN to extend Wireless Isolation to selectively enable a given Wi-Fi client to communicate with the Chromecast device.

We use Chromecast to exemplify that our techniques can be used to programmatically manage the Wi-Fi flows in smart spaces. Furthermore, we demonstrate that bringing SDN to Wi-Fi networks does not require specialized hardware, large changes in devices or software, and can be accomplished using existing devices whether they support custom firmware or not. Our techniques open existing Wi-Fi testbeds for experiments on next generation applications which leverage the programmability offered by SDN. The steps required for using our two techniques for bringing SDN to Wi-Fi networks are documented on the following web page: https://wiki.helsinki.fi/display/WiFiSDN/

To conclude, our techniques can be used to convert existing Wi-Fi networks and testbeds created using off-the-shelf devices into software-defined Wi-Fi networks.

Acknowledgments. This work has been supported by the Tekes 5GTrek research project, Digile Cyber Trust project, Academy of Finland Cloud Security Services project, and the Nokia Center for Advanced Research (NCAR).

4. REFERENCES

[1] hostapd: IEEE 802.11 AP. http://w1.fi/hostapd/.
[2] OpenWrt. https://openwrt.org/.
[3] McKeown, N., et al. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM CCR.* (2008).
[4] Pfaff, B., et al. The Design and Implementation of Open vSwitch. In *Proc. of NSDI*, (2015).
[5] Schulz-Zander, J., et al. OpenSDWN: Programmatic Control over Home and Enterprise WiFi. In *Proc. of ACM SOSR '15* (2015).
[6] Yan, M., et al. Ætherflow: Principled Wireless Support in SDN. In *Proc. of IEEE ICNP* (2015).
[7] Yap, K.-K., et al. OpenRoads: Empowering Research in Mobile Networks. *ACM SIGCOMM CCR.* (2010).
[8] Yiakoumis, Y., et al. BeHop: A Testbed for Dense WiFi Networks. In *Proc. of WiNTECH '14* (2014).