

Programming the Home and Enterprise WiFi with OpenSDWN

Julius Schulz-Zander
TU Berlin
Berlin, Germany
julius@inet.tu-berlin.de

Stefan Schmid
Deutsche Telekom Innovation
Laboratories / TU Berlin
Berlin, Germany
stefan@inet.tu-berlin.de

Carlos Mayer
TU Berlin
Berlin, Germany
carlosnmayer@gmail.com

Anja Feldmann
TU Berlin
Berlin, Germany
anja@inet.tu-berlin.de

Bogdan Ciobotaru
TU Berlin
Berlin, Germany
bogdan.ciobotaru1@gmail.com

Roberto Riggio
CREATE-NET
Trento, Italy
rriggio@create-net.org

ABSTRACT

The quickly growing demand for wireless networks and the numerous application-specific requirements stand in stark contrast to today's inflexible management and operation of WiFi networks. In this paper, we present and evaluate OPENSNDWN, a novel WiFi architecture based on an SDN/NFV approach. OPENSNDWN exploits datapath programmability to enable service differentiation and fine-grained transmission control, facilitating the prioritization of critical applications. OPENSNDWN implements per-client virtual access points and per-client virtual middleboxes, to render network functions more flexible and support mobility and seamless migration. OPENSNDWN can also be used to out-source the control over the home network to a participatory interface or to an Internet Service Provider.

Categories and Subject Descriptors

C.2.3 [Computer Communication Networks]: Network Operations; C.2.1 [Computer Communication Networks]: Network Architecture and Design

Keywords

Network Function Virtualization; Software-Defined Networking; Programmable RAN; Enterprise WLANs; Wi-Fi

1. INTRODUCTION

In this demo, we show the potential and benefits of introducing programmability and virtualization in *wireless networks*. Wireless networks are very different from wired networks—the domain where SDN/NFV has been studied most intensively so far—as communication happens over a shared medium whose characteristics can change quickly over time and in an unpredictable manner, and as users are often mobile and associations dynamic. WiFi

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: <http://dx.doi.org/10.1145/2785956.2790037>

networks offer several unique knobs to influence the probability of successful transmissions, such as transmission rate and power, as well as retry chains. This introduces opportunities for a fine-grained and application specific transmission control, e.g., for service differentiation.

Our proposed architecture, OPENSNDWN [2], is based on a unified, programmable control plane which allows to manage both the datapath as well as the virtualized middleboxes. Specifically, OPENSNDWN implements per-client virtual access points and per-client virtual middleboxes, to render network functions more flexible and support mobility and seamless migration of network processing functions.

This demo presents interesting use cases of OPENSNDWN: (1) OPENSNDWN enables service differentiation capabilities, and allows administrators or users to specify application and flow priorities. These priorities are implemented using a fine-grained transmission control. (2) Using its per-client virtual access points and virtual middleboxes, OPENSNDWN supports seamless user mobility, as well as flexible function allocation (e.g., function collocation at night to save energy). (3) Network functions such as firewalls and NATs can be deployed flexibly, e.g., outside user premises. (4) OPENSNDWN also introduces flexibilities in terms of network control: the system exposes a participatory interface à la [1], where users can indicate priorities for their applications. The control can also be outsourced to an Internet Service Provider (ISP), e.g., for troubleshooting.

2. THE OPENSNDWN SYSTEM

OPENSNDWN is based on an SDN+NFV (a.k.a. *SDNv2*) approach and consists of the following components:

1. **Unified Programmability and Abstractions:** The logically centralized control plane unifies SDN and NFV through programmatic abstractions. That is, OPENSNDWN virtualizes both access points and virtualized middleboxes (see Figure 1(b)), which facilitates an easy handling and migration of per-client state, also beyond CPE boundaries. The OPENSNDWN abstractions can be seen as an extension of Odin [3] to NFV: Odin's *LVAP* concept abstracts the complexities of the IEEE 802.11 protocol stack (client associations, authentication, and handovers), and enables the unified slicing of both the wired and wireless portions of the network by encapsulating the client's Openflow state. OPENSNDWN additionally introduces per-client virtual middleboxes, short

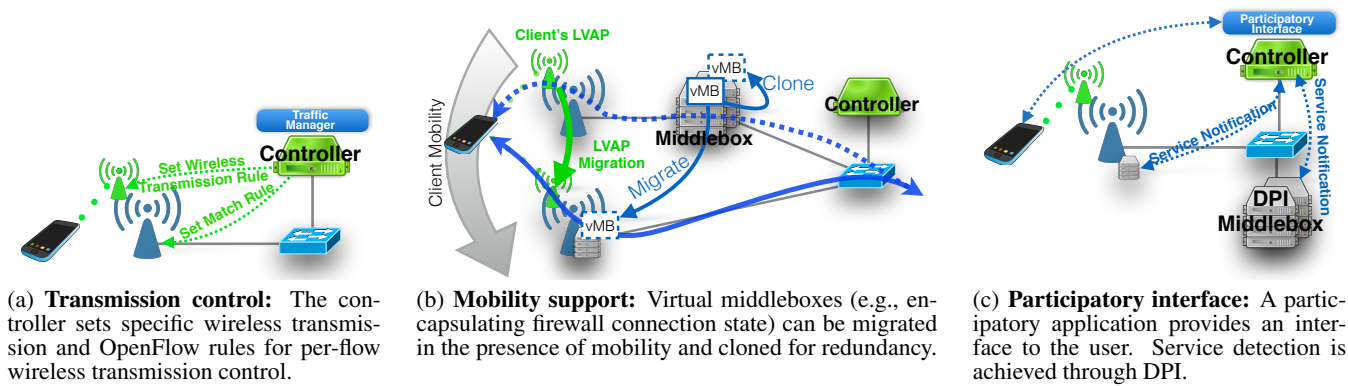


Figure 1: Three basic operations supported by OPENSOWN.

vMBs, which can be transferred seamlessly across the network. Specifically, a vMB encapsulates the client’s MB state as a virtual MB object. Thus, OPENSOWN achieves control logic isolation as SDN/NFV applications running on top the controller can only operate on their respective LVAPs and vMBs.

- Programmable Datapath:** The programmable datapath allows us to specify per-flow transmission settings, as shown in Figure 1(a). The settings include transmission power, transmission rate as well as tailored retry chains. It is even possible to differentiate between different packets of the *same flow* (5-tuple): for instance, key frames of a live stream may be given higher priority. This is achieved by using an IDS such as Bro for packet classification and *tagging*: transmission settings are chosen depending on the tag.
- Participatory Interface:** OPENSOWN’s participatory interface allows us to define flow priorities as well as priorities over customers. The chosen priorities are translated by the controller into meaningful network policies. Priorities can be adjusted anytime. Figure 1(c) depicts the participatory interface.

3. DEMO

User-Defined Service Differentiation

OPENSOWN offers visibility into the network’s state and supports a fine-grained transmission control, by allowing administrators and users to set per-flow and per-packet specific transmission settings (such as transmission rate, power, retransmission and RTC/CTS strategy). For instance, as we will demonstrate, OPENSOWN can protect latency-sensitive flows (e.g., live streaming) from competing with background traffic (e.g., a Dropbox synchronization). Especially given today’s trend to deploy more and more wireless devices in the user’s premises, traffic can significantly interfere, e.g., an unimportant system update for a device can easily interfere with requested *on demand* services such as Spotify or Netflix, resulting in poor performance.

Mobility and migration

By virtualizing not only the per-client access points, but also the middleboxes, OPENSOWN supports both seamless user mobility and a dynamic resource allocation. The more dynamic resource management introduced by OPENSOWN enables the adjustment and migration of resources and functionality with the user, e.g., for flexibly scaling up or down resources depending on the demand. By collocating network functions, e.g., at night, also energy may be

saved. The firewall state migration service is a reactive application triggered through external events to move state between MB instances. For example, when Odin [3] detects a client with a higher RSSI at a new AP, a handover event is generated and the client’s stateful firewall state migrated to the AP before the handover is executed. The firewall state migration service then decides whether the state associated with the mobile user needs to be migrated and executes the operation. The application keeps a mapping between APs and firewalls. If the client is moving over to an AP that corresponds to a different stateful firewall than the current, a migration of the client’s flow state is performed.

Smart Direct Multicast Service

OPENSOWN can also be used in conjunction with group abstractions. Multicast is a standard group communication abstraction, and with the advent of IPv6, the fraction of multicast traffic is likely to grow in the future, e.g., (IPv6 realizes broadcast over multicast and mDNS to broadcast features to neighboring stations.

In IEEE 802.11, multicasts are typically sent at basic rate. However, wireless networks may benefit from a direct multicast service (DMS) which has the potential of reducing the transmission time over regular multicast, by sending 802.11 packets as unicast. Unfortunately, DMS requires a client to signal its DMS capabilities to the AP, which is the reason why DMS is rarely used in 802.11 networks today.

With OPENSOWN, a controller can detect the number of subscriptions for a particular multicast service and control the transmission accordingly. Specifically, a controller can install an OpenFlow rule to switch from multicast to unicast for the transmission. Moreover, OPENSOWN allows to assign *WDTX* group-based transmission rules, e.g., transmit a stream of multicast data at the *maximum common transmission rate*.

Acknowledgments. Research supported by the Federal Ministry of Education and Research (BMBF) Software Campus “SDWN” Project Grant (Reference number 01IS12056).

4. REFERENCES

- [1] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Participatory networking: An api for application control of sdn. *SIGCOMM Comput. Commun. Rev.*, 43(4):327–338, Aug. 2013.
- [2] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann. OpenSDWN: Programmatic Control over Home and Enterprise WiFi. In *SOSR ’15*.
- [3] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz. Programmatic orchestration of wifi networks. In *USENIX ATC ’14*.