AeroFlux: A Near-Sighted Controller Architecture for Software-Defined Wireless Networks

Julius Schulz-Zander[†] , Nadi Sarrar[†] , and Stefan Schmid[‡†]

[†] TU Berlin, Germany [‡] Telekom Innovation Laboratories, Germany

Abstract

Applying the concept of SDN to WiFi networks is challenging, since wireless networks feature many peculiarities and knobs that often do not exist in wired networks: obviously, WiFi communicates over a shared medium, with all its implications, e.g., higher packet loss and hidden or exposed terminals. Moreover, wireless links can be operated in a number of different regimes, e.g., transmission rate and power settings can be adjusted, RTS/CTS mechanisms can be used.

Indeed, due to the non-stationary characteristic of the wireless channel, permanently adjusting settings such as transmission rate and power is crucial for the performance of WiFi networks and brings significant benefits in the service quality, e.g., through reducing the packet loss probability. Today's rate and power control is mainly done on the WiFi device itself. But it is rarely optimized to the application-layer demands and their diverse traffic requirements, e.g., their individual sensitivity to packet loss or jitter. Therefore, if SDN for wireless can provide mechanisms to control the WiFi-specific transmission settings on a per-slice, per-client, and per-flow level, traffic and application-aware optimizations are feasible.

This however requires that controllers frequently collect link characteristics and, accordingly, adjust transmission settings in a timely manner. As a reference, the standard rate control mechanism in the Linux kernel adjusts the transmission rate on a wireless link based on transmission success probability statistics every 100 ms. Leaving rate control (and power control accordingly) to a centralized controller comes with a risk of overloading the control plane, or of adding too much latency, while there is limited benefit in maintaining these statistics globally. For instance, the coherence time (also a function of the client mobility) can easily exceed the expected time of the successful transmission of multiple data frames [2], rendering optimized control difficult.

In this paper, we suggest a 2-tiered approach for the design of a wireless SDN control plane. Our design, called AeroFlux, handles frequent, localized events close to where they originate, i.e., close to the data plane, by relying on *Near-Sighted Controllers* (`NSC`) [3, 4, 7]. Global events, which require a broader picture of the network's state, are handled by the *Global Controller* (`GC`). More specifically, `GC` takes care of network functions that require global visibility, such as mobility management and load balancing, whereas `NSC`s control per-client or per-flow transmission settings such as rate and power based on transmission status feedback information exported by the Access Points (AP), which include the rates for best throughput and best transmission probability. Put differently, we enable the global controller to offload latency-critical or high-load tasks from the tier-1 control plane to the `NSC`s. This reduces the load on the `GC` and lowers the latency of critical control plane operations.

As a result, with AeroFlux, we realize a scalable wireless SDN architecture which can support large enterprise and carrier WiFi deployments with low-latency programmatic control of fine-grained WiFi-specific transmission settings. The AeroFlux design introduces a set of new trade-offs and optimization opportunities which allow for advancements in the use of the shared wireless medium, and, as a result, in the user's quality of experience. For instance, our prototype's per-flow control allows application-aware service differentiation by prioritizing multimedia streams (§2). Another key feature of AeroFlux is that it does not require modifications to today's hardware and works on top of commodity WiFi equipment.

1 The AeroFlux Architecture

AeroFlux uses a 2-tiered control plane: the Global Control plane `GC` and the Near-Sighted Control plane `NSC`. Figure 1 depicts the high level interactions of the architecture's building blocks. The `GC` is logically centralized, e.g., a set of redundant controllers deployed in data centers, whereas the `NSC`s are located closer to where they are needed, e.g., close to the wireless APs. On the APs runs a Radio Agent (`RA`), which hosts the Light Virtual Access Points (LVAPs) [6] that abstract the specifics of the 802.11 protocol, such as association and authentication state. Furthermore, LVAPs store per-client OpenFlow and WiFi Datapath Transmission (`WDTX`) rules. In the following, we describe the different elements in more detail.

Global Controller (`GC`): The global controller handles events which are not time-critical [7] or events belonging to inherently global tasks [4]. Examples include: authentication, wide-area mobility management, global policy verification (including loop-free forwarding sets), client load balancing, and applications for intrusion detection or network monitoring. In addition, the global controller is best suited to man-

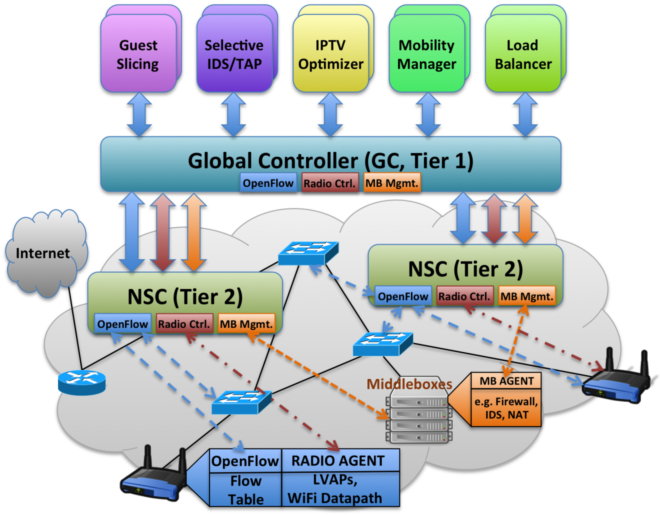

Figure 1: AeroFlux: NSCs are deployed close to the APs.

age middleboxes (firewalls, etc.), including their instantiation and the re-routing of flows for middlebox traversal. The GC also controls the NSCs, and uses them to offload selected network functions which do not rely on a global network state.

Near-Sighted Controller (NSC): The tier-2 control plane includes a number of NSCs, located close to the wireless APs to perform time-critical and load-intensive network functions which do not require a global network view. Essentially all functions exported by the Radio Agent running on the APs themselves (see below) can be used by SDN applications offloaded onto the NSCs. The NSCs also implement an interface for middlebox life-cycle management (delegated by the GC).

Radio Agent (RA): The Radio Agent hosts the LVAPs, an abstraction that addresses the specific requirements of the IEEE 802.11 protocol. LVAPs are basically per-client APs that simplify the handling of client associations, authentication, handovers and slicing. LVAPs also store client specific WiFi datapath transmission (WDTX) rules which are tied to OpenFlow rules; the former defines the transmission attributes on the wireless medium and the latter is needed for the integration with the wired portion of the network. The RA exposes all functionality of the LVAP abstraction and the WDTX rules to the NSC. Furthermore, RAs provide an interface to network applications to selectively receive IEEE 802.11 frames.

WiFi Datapath Transmission (WDTX) Rule: The WiFi datapath transmission rules extend OpenFlow rules by defining per-flow transmission properties of an IEEE 802.11 frame. WDTXs are stored in a table on a WiFi AP and are linked to OpenFlow flow entries. With that, NSCs can control the transmission properties of a flow through OpenFlow rules. Currently a WDTX entry controls the following transmission properties: rate, retries, power, usage of ACK and of RTS/CTS. However, WDTX rules are extensible and could also incorporate transmission properties such as the transmission chain or even the antenna used for transmissions. With OpenRadio [1], our system could benefit from a clean-slate programmable network dataplane.

2 Use Cases

To motivate the potential benefits of SDN for wireless networks in general and AeroFlux in particular, we consider two case studies: (1) live video streaming and (2) selective TAPs / intrusion detection system.

Live Video Streaming: Ensuring good transmission quality of live video streams is a challenging task in today's wireless networks. A control application can use admission control and flow classification to limit the number of parallel video flows to ensure sufficient resources. More interestingly, the AeroFlux control plane can also decide to transmit keyframes of a video stream at high probability transmission settings (e.g., lower rate, higher power), while the other, less-critical packets can be transmitted using more optimistic rate/power settings, optimized for throughput, as the loss of a non-keyframe does not negatively impact the video quality as much. This can be achieved through flow tagging by a middlebox [5].

Selective Intrusion Detection System: To detect malicious behavior in a WiFi deployment, an IDS application can selectively monitor IEEE 802.11 frames, such as suspicious CTS frames that block the usage of the medium, from all or only a subset of the APs. The latter is used to detect a CTS jamming attack and to switch off the virtual carrier sensing. Furthermore, AeroFlux can benefit from traditional IDS systems such as the Bro Network Security Monitor by routing or selectively duplicating flows for intrusion analysis [8].

3 Early Prototype and Outlook

We implement the Near-Sighted Controllers and Radio Agents within our existing Odin-framework [6] (from which AeroFlux also inherits the LVAP concept), which is based on Floodlight. The tasks of managing LVAPs for client mobility, load-balancing, and network slicing are well suited for the global part of the control plane and are implemented at the Odin Controller (publicly available as open source). In our prototype the WDTX entries are kept and handled within the Linux kernel for performance reasons. A deployment in the 100-node Berlin Open Wireless Lab (BOWL) is ongoing.

Acknowledgments: This work is a first study towards a distributed control plane for the architecture of the FP7 EU project UNIFY.

References

[1] M. Bansal, J. Mehlman, S. Katti, and P. Levis. OpenRadio: a programmable wireless dataplane. In *HotSDN '12*.

[2] H. Jung et al. React: Rate adaptation using coherence time in 802.11 WLANs. *Comput. Commun. '11*.

[3] B. Heller, R. Sherwood, and N. McKeown. The controller placement problem. In *HotSDN '12*.

[4] S. Schmid and J. Suomela. Exploiting locality in distributed SDN control. In *HotSDN '13*.

[5] S.K. Fayazbakhsh et al. Enforcing network-wide policies in the presence of dynamic middlebox actions using flowtags. In *NSDI '14*.

[6] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao. Towards programmable enterprise WLANS with Odin. In *HotSDN '12*.

[7] Y. Hassas et al. Kandoo: a framework for efficient and scalable offloading of control applications. In *HotSDN '12*.

[8] Y. Zhang et al. Online identification of hierarchical heavy hitters: algorithms, evaluation, and applications. In *IMC '04*.