SDWLAN: A Flexible Architecture of Enterprise WLAN for client-unaware fast AP handoff

Dong Zhao*, Ming Zhu*, Ming Xu*
* College of Computer, National University of Defense Technology, Changsha, China
Email:{dongzhao, zhuming, xuming}@nudt.edu.cn

***Abstract*—Currently, enterprise WLAN is facing rapid growth of user scale and traffic load, as well as constantly emerging new features. However, traditional enterprise WLAN architecture suffers from poor flexibility and the lack of coordination between wireless access points (APs) and wired backbone. Inspired by the emerging idea of *Software-Defined Networking (SDN)*, we proposed *SDWLAN*, an alternative architecture for enterprise WLAN. The salient features of SDWLAN are twofold. First, most of 802.11 AP functions are decoupled from scattered devices and centralized in a controller, leaving some simplified devices (i.e., *wireless access switches, or WASes*) manipulated by the controller through extended OpenFlow protocol. Second, the control of APs and wired backbone are consolidated to provide a unified network control platform. By reorganizing 802.11 AP's functional modules, SDWLAN can achieve remarkable flexibility. Benefiting from the extended OpenFlow protocol and the unified control platform, we proposed a client-unaware fast AP handoff mechanism in SDWLAN. Simulation results demonstrated that AP handoff operation in SDWLAN leads to negligible throughput fluctuation of on-going connection compared to traditional architecture with 802.11 standard handoff mechanism. Furthermore, SDWLAN requires no modification to existing 802.11 clients, which make our solution practical.**

Keywords*—*SDN, enterprise WLAN, AP handoff

I. Introduction

In recent years, IEEE 802.11 enterprise WLANs, are widely deployed in public places such as campus and airport, providing Internet-access service. As the rapid expansion of user scale, the increase of traffic load, and the emergence of various application demands, constant innovations are needed in enterprise WLANs.

Today's enterprise WLAN, however, follows an inflexible architecture with limited extensibility, hindering real deployment of new features. In enterprise WLAN, access points (APs) are interconnected through wired backbone and there are one or more AP controllers (ACs) conducting management and control over these APs (operators usually deploy multiple ACs to perform regional management, as shown in Fig. 1). In exiting solutions [1], [2] that follow the "AP + AC" architecture, adding new feature often involves the update of a large number of APs, which is a complex and error-prone task.

In addition, today's enterprise WLAN don't consider the coordination between wireless APs and wired backbone. Although AC is usually integrated with switch or router, acting as a "gateway" between APs and external network, the function of controlling APs and "gateway" function (e.g., switching and routing) are implemented independently. In addition, AC can not control other wired switches or routers.

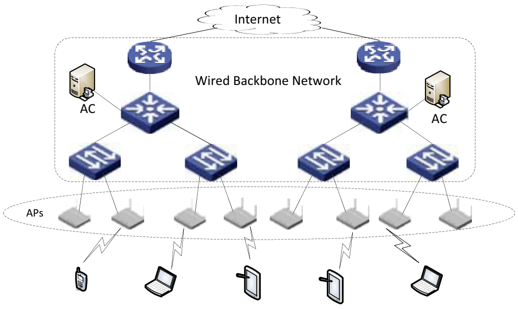


Fig. 1. Typical "AP + AC" enterprise WLAN with a wired backbone network composed of several switches/routers, a number of APs and two ACs.

The emerging Software-Defined Networking (SDN) is a revolutionary networking paradigm that makes processing and forwarding of packets more flexible and programmable. The core of SDN is decoupling of packet forwarding hardware and control decision software. It moves control logic from switches/routers to a central controller, while keeping switches/routers very simple. SDN, along with its supporting technology OpenFlow [3], offers a promising alternative architecture for enterprise WLAN to address the aforementioned issues.

In this paper, we present **SDWLAN** (**S**oftware-**D**efined **W**ireless **L**ocal **A**rea **N**etwork), a new architecture for enterprise WLANs. SDWLAN takes advantage of SDN by reorganizing 802.11 AP MAC-layer and consolidates the control of wireless APs and wired backbone.

SDWLAN provides a flexible architecture for enterprise WLAN. In SDWLAN, adding new feature only need to update software in the controller-side. In addition, the consolidation of wireless APs and wired backbone provides a global-view platform for control and management, which promises to achieve the integration and optimized utilization of network resources from a global viewpoint.

Based on SDWLAN, we proposed a client-unaware fast AP handoff mechanism. The proposed AP handoff mechanism benefits from the reorganization of 802.11 AP MAC-layer and the unified control platform. We demonstrated the efficiency of the proposed fast AP handoff mechanism by measuring the effect of handoff operation on client's on-going TCP/UDP session. The experiment results also demonstrate the flexibility and extensibility afforded by the SDWLAN architecture.

SDWLAN don't require any modification to exiting 802.11 clients. We preserved as many features of 802.11 standard as possible, which means that little hardware modifications to state-of-the-art hardware are needed.

Specifically, the main contribution of this paper includes:

- We propose SDWLAN, a new architecture for enterprise WLANs. SDWLAN provides a flexible and enhanced platform for enterprise WLAN. The consolidation of wired and wireless resources benefits network innovation. SDWLAN requires no modification to client.
- We develop a client-unaware fast AP handoff mechanism in SDWLAN. We take advantages of the extension to OpenFlow protocol and the unified control platform of wireless APs and wired backbone to achieve low-latency AP handoff.
- We implement basic SDWLAN framework and measure the proposed fast AP handoff mechanism. The results not only demonstrates that the proposed fast AP handoff mechanism can significantly reduce the effect of AP handoff on user's experience, but embodies the feasibility and efficiency of SDWLAN.

II. SDWLAN Architecture

The SDWLAN's goal is to support flexible and fine-grained control over enterprise WLAN in a scalable manner. We aim to achieve this without requiring any change to clients. In this section, we first introduce basic elements in SDWLAN, and then describe how 802.11 AP MAC-layer functions are reorganized in SDWLAN to realize the decoupling MAC-layer management policy and packet processing. And next we demystify the new device, "*wireless access switch (WAS)*", and the extended OpenFlow protocol.

A. Overview

As shown in Fig. 2, SDWLAN consists of a central controller, a set of *wireless access switches (WASes)*, a OpenFlow-based wired backbone network which is composed of several OpenFlow-enabled *wired backbone switches (WBSes)*, and some optional appliances. All WASes and WBSes are at the discretion of the controller. Any function or emerging feature is implemented as application running atop controller. "Virtual AP Management" is example of application.

Wired backbone switches (WBSes) are just OpenFlow-enabled switches that can be communicated with and controlled through standard OpenFlow protocol. They constitute the wired backbone network, interconnecting WASes and external network. The wired backbone acts as a uniform routing fabric that capitalizes on IP and Ethernet technology to deliver packets between WAS and external network (Internet). Any packet destined for a client should be delivered to the client's associated WAS.

Here we use the coined term *wireless access switches (WAS)* rather than access point (AP). This is because traditional AP MAC-layer have been reorganized in SDWLAN. We decompose 802.11 AP MAC-layer into several functional modules, and move most of them from original AP and place them onto controller side, leaving a simplified device: *wireless access switches (WAS)*. In addition to standard OpenFlow protocol, the controller can control WASes through an extended OpenFlow protocol (mentioned in section II-C) to support the reorganization of 802.11 AP MAC-layer function.

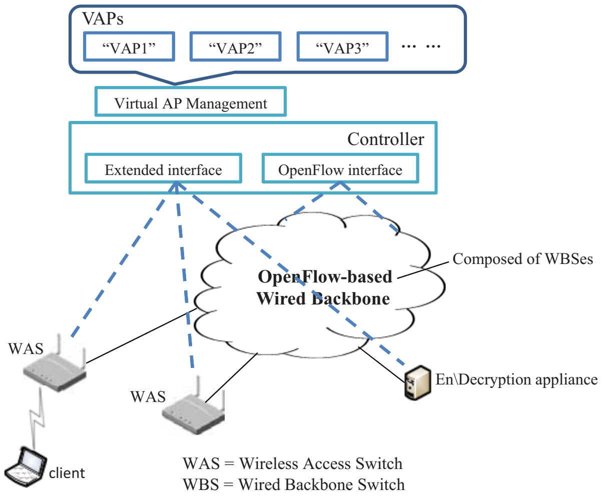


Fig. 2. SDWLAN architecture consists of a centralized controller, a set of wireless access switches (WASes), a OpenFlow-based backbone network composed of several OpenFlow-enabled wired backbone switches (WBSes), and some optional appliances.

We essentially merge two controllers: the WAS controller and the WBS controller. This comes out with a unified control platform of wireless APs (WAS in practice) and wired backbone. Using the interface provided by controller, we can develop various applications to coordinate the behavior of WASes and WBSes to improve network performance more efficiently and implement some functions which are difficult to realize in traditional architecture (e.g., fast AP handoff as mentioned in section III).

Operator can deploy some additional appliances at any location to aid the completion of specific function. For example, We will introduce en/decryption appliances to support client-unaware fast WAS handoff (seen in section III). We can define our proprietary protocol between the application and corresponding appliances to exchange message between them.

B. The Reorganization of 802.11 AP's MAC-layer

According to 802.11 standard [4], we decompose 802.11 AP's MAC-layer function into the following functional modules (as shown in Fig. 3 (a)):

1) **Beaconing & Probe Response**: AP should constantly generate beacon packets. Some clients may broadcast probe request and AP is supposed to respond to them with probe respond packet. Both beacon and probe response are to claim the AP's existence and covey information about the AP's configuration and features.

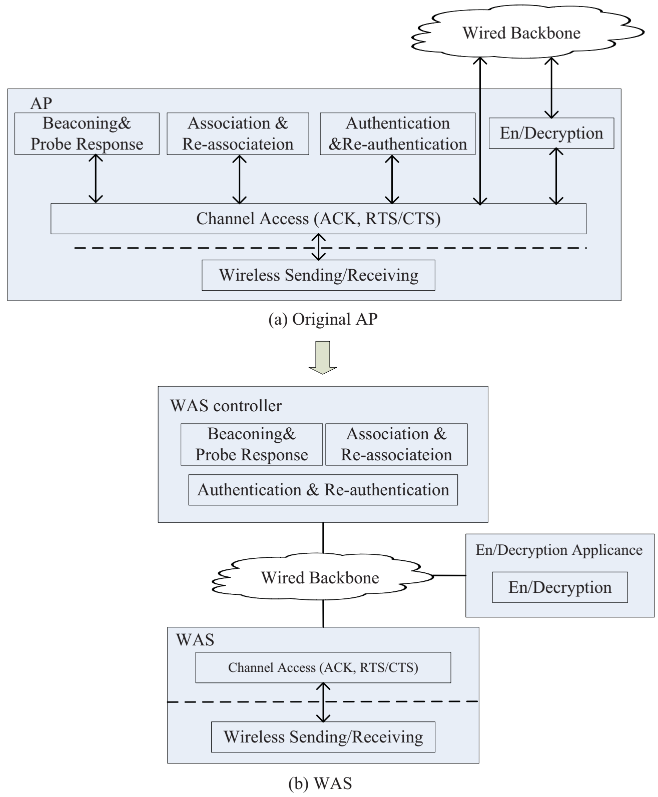
2) **Association & Re-association**: Client should associate with an AP by sending association request. If the client's identity is verified, AP will respond to the client by sending associate response. Reassociation is used when client determine

to associate with a new AP. In effect, association is a procedure to register client to wired backbone, so that the client's traffic can be correctly routed to client's AP.

3) **Authentication & Re-authentication**: when a client wants to associate with an AP, the AP should initiate authentication challenge and verify client's identity. In enterprise environment, AP plays an intermediate role to forward authentication request and response packets between clients and remote authentication (RADIUS) server.

4) **En/Decryption**: In a secure enterprise WLAN, user's traffic are encrypted to be securely transmission in open air to keep confidentiality. So AP should encrypt packet data frame being sent to clients and decrypt data frame received from clients.

5) **ACK & RTS/CTS**: ACK and RTS/CTS mechanism is the feature of 802.11 wireless MAC that wired MAC don't have. Every 802.11 unicast frame (data frame or management frame) should be responded with an ACK frame. A transmitted data frame without receiving ACK is believed to be failure and should be retransmitted. RTS/CTS handshake is optional mechanism used to clear wireless channel when sending long data frame.


Wired Backbone
AP
Beaconing& Probe Response
Association & Re-associateion
Authentication &Re-authentication
En/Decryption
Channel Access (ACK, RTS/CTS)
Wireless Sending/Receiving
WAS controller
Beaconing& Probe Response
Association & Re-associateion
Authentication & Re-authentication
Wired Backbone
En/Decryption Applicance
En/Decryption
WAS
Channel Access (ACK, RTS/CTS)
Wireless Sending/Receiving

(a) Original AP

(b) WAS

Fig. 3. The reorganization of 802.11 AP's MAC-layer. Most of original AP's functional modules are lifted to controller, while "ACK & RTS/CTS" module, together with PHY-layer 802.11 wireless sending and receiving function, remains in WAS.

We reorganize 802.11 AP's MAC-layer. As shown in Fig. 3 (b), the following functional modules are moved to the controller: *"Beaconing & Probe response", "Association & Re-association", "Authentication & Re-authentication", and "En/Decryption"*. In contrast, *"ACK & RTS/CTS"* module, together with PHY-layer 802.11 wireless signal sending/receiving function, remains in WAS. As a result, most 802.11 AP MAC-layer intelligence has been removed from original AP. That is why we replace original term "AP" with the coined term "WAS".

The controller take over these WASes and instruct them through extended OpenFlow interface. Specifically, the controller guides WASes whether or not to respond to a packet with ACK. We will detail the extension of OpenFlow in the following section. The synergy of controller, WASes, and some optional appliances realizes complete 802.11 AP MAC-layer function. We implemented a Virtual AP management application that coordinates the configuration and packets processing of WASes, WBSes and related appliances. We will detail the VAP management application in following section.

C. Wireless Access Switch and Extended OpenFlow

As shown in Fig. 4, a WAS is equipped with a wireless NIC and a wired Ethernet NIC. Except for the wireless NIC, WAS has no difference with wired switch. Naturally, WAS supports standard OpenFlow protocol just as a OpenFlow-enabled switch. So the WAS controller can communicate with WASes through standard OpenFlow protocol.

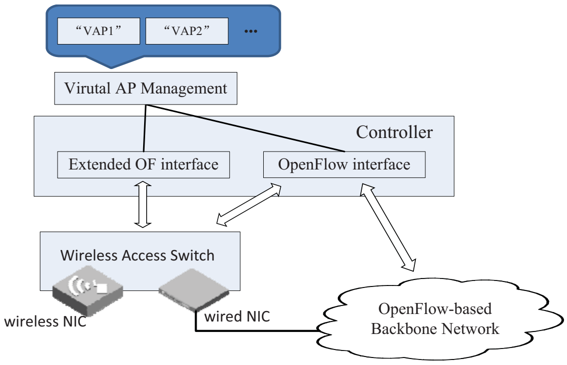


Fig. 4. Wireless Access Switch (WAS) has two NICs: one wireless NIC and one wired NIC. The controller uses extended OpenFlow interface to communicate with WAS.

Apart from standard OpenFlow, we proposed an extra interface used between WAS and controller, i.e., extensions of OpenFlow protocol. The extensions are critical to implement the proposed reorganization of AP function. Our extensions are based on the latest OpenFlow specification version 1.4.0 [5].

1) **New matching field: BSSID.** 802.11 MAC-layer packet has a specific field "BSSID" that is used to identify AP. MAC-layer packet from Ethernet to wireless client should be converted into 802.11-specific format by adding the field and filing in it with the AP's BSSID before forwarding to the client. Conversely, packet from wireless client to Ethernet will be checked if the value of "BSSID" field is exactly AP's BSSID and then converted into Ethernet format. We extend OpenFlow specification by adding this new field so that controller can instruct WAS to match BSSID field of received packet. An

fringe benefit is that, in conjunction with and rewrite BSSID of output packet as needed.

2) **New actions: MAC_ACK and CTS_ACK.** Due to the unreliability of wireless channel, the sending of an 802.11 unicast packet should be followed by an ACK. By adding new action "MAC_ACK", the controller can instruct WAS whether to respond to a received packet with ACK or not. Similarly, RTS/CTS is a specific mechanism to 802.11 MAC-layer transmission, the action "CTS_ACK" enable controller to determine whether a WAS should respond a specified RTS handshake request.

D. *Controller and Application*

As mentioned above, SDWLAN provides controller an extended OpenFlow interface to communicate with WASes and WBSes so that we can develop applications to coordinate the behavior of WASes and WBSes. We categorize applications into three class: (1) RF-related applications, (2) packet processing applications and (3) the combination of the above.

RF-related applications improve network performance by tuning (statically or dynamically) wireless NIC's operation parameters. Prior works like channel assignment, power control, rate adaptation fall into this class. They can be easily ported to SDWLAN.

Packet processing applications achieve certain objective by performing appropriate processing on packets in WAS and WBS. Specifically, we take full advantage of the new matching field of "BSSID" and the new-added actions ("MAC_ACK" and 'CTS_ACK") to implement some functions which are very hard to achieve in traditional enterprise WLAN architecture.

It is promising to develop some applications that jointly optimize the configuration of RF-related parameters and packets processing strategy. However, the development such applications is still an open area and will be our future work.

III. Client-unaware Fast AP handoff in SDWLAN

A. *"One Big AP" illusion*

Similar with the 'VirtualCell' proposed by Meru [6], we aim to let APs operate on the same channel and form a bigger coverage area relative to a single AP. We refer to this as "One big AP" illusion in that client is tricked into believing that it is always interacting with one AP, while it is actually communicating with many different APs (As shown in Fig. 5). Meru's solution is proprietary and the details are not available.

The key to achieve "One Big AP" illusion is client-unaware fast AP handoff mechanism. AP handoff should not affect client's on-going traffic. Due to the broadcast nature of wireless signal, client's packets can be received by several WASes simultaneously, we need to ensure that at any time there is one and only one WAS sending the ACK or CTS response to client.

We developed an application "Virtual AP Management", which takes advantage of SDWLAN architecture to realize such "One Big AP" illusion. Specially, "Virtual AP Management" is to coordinate packets processing of WASes and WBSes, as well as some appliances involved. "Virtual AP Management" application creates several virtual AP (VAP) on controller, and maintains some basic information of virtual AP, including SSID, MAC address, authentication methods, the maximum number of associated clients, and so on. The application also stores every client's association context.

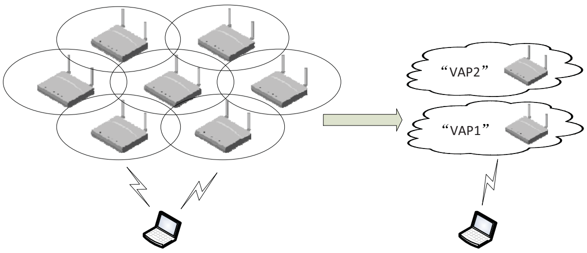


Fig. 5. "One Big AP" illusion: client is tricked into believing that it is always interacting with one AP, while they are in fact communicating with many different APs.

The control logic on controller is insufficient to implement complete functions of 802.11 AP. We deployed several appliances (e.g., en/decryption appliances) distributed around wired backbone. Their task is to encrypt and decrypt 802.11 MAC-layer packets on behalf of virtual APs. By installing rules on WASes and WBSes involved along specified path, encrypted wireless packets are encapsulated by WASes and directed to these appliances. The appliances are responsible for decrypting these packets. Similarly, packets destined for clients are also routed to these appliances to be encrypted and then sent to their corresponding WASes. So that, packets from/to the clients should travel along a well-designed path.

Fig. 6 illustrates a client's flow distribution in SDWLAN. You can found that the packets of client are categorized into three class of flow. 802.11 MAC-layer management packets such as association-relevant and authentication-relevant packets are sent to controller via OpenFlow channel (red dotted line). Encrypted packets (green dashed line) are first sent to specified an en/decyption appliance, where key is kept and packets are decrypted, and then sent to their destination. Decrypted traffic is represented by green solid line. Plaintext (unencrypted) traffic flow (blue solid line) is directly routed the destination (remote network). It's no hard to see that the controller can easily achieve this by carefully installing appropriate rules in WBSes involved through OpenFlow interface.

B. *Client-unaware fast WAS handoff*

A roaming client have to handoff between different APs to maintain continuous connectivity. In addition, some AP association strategies are proposed to dynamically adjusting AP association decision to achieve AP load balance [7]–[9] or QoS gurantee [8], [10]. Reducing AP handoff delay is a critical issue to improve users' experience in enterprise WLAN. Traditional AP handoff process takes considerable time of one second or so. In this section we attempt to take advantage of SDWLAN to design a client-unaware fast AP[1] handoff mechanism.

[1] As explained in prior section, the coined term "WAS" replaces "AP" in SDWLAN. However, we still use the term WAS and AP interchangeably for easy of exposition. In this section, when we refer to "AP handoff in SDWLAN", it means "WAS handoff in SDWLAN" in essentially.

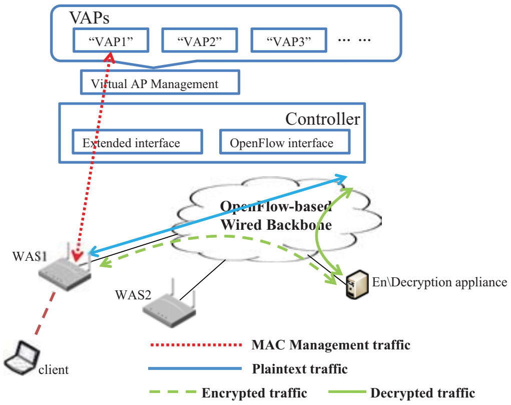


Fig. 6. The flow of a client. Red dotted line represents flow of 802.11 MAC-layer management traffic, green dashed line represents flow of encrypted packets traffic, green solid line represents the decrypted packets traffic, and blue solid line represents unencrypted packets traffic.

Ingress Port	BSSID	MAC Dst	MAC Src	…	Actions
Wireless NIC	BSSIDA	…	MACA	…	ACK, Forward to wired NIC, …
Wired NIC	…	MACA	…	…	Set "BSSID" to ***BSSIDA***, Forward to wireless NIC, …

Fig. 7. Rules installed on client A's serving WAS. The first rule means that any packet, which is received from the wireless NIC, whose "BSSID" field is $BSSID_A$, and whose MAC source address is client A's MAC address will be responded to with an ACK and forwarded to the wired NIC. The second rules means that any packet which is received from the wired NIC, and whose MAC destination address is client A's MAC address will be forwarded to the wireless NIC with the "BSSID" field set to be $BSSID_A$.

The handoff procedure involves two critical steps: changing client's serving WAS and steering client's traffic from old WAS to new WAS. The first step needs to coordinate both old WAS and new WAS to ensure there is one and only one WAS serving the client at any time. The second step is to ensure packets destined for the client can be forwarded to its new WAS after the change of serving WAS. In WAS handoff, client is supposed to be unaware of the change of its serving WAS and any MAC-layer re-authentication and re-association will never be triggered. So the client can be tricked into believing that it is always communicating with the very single AP. Any significant interruption of on-going session won't be caused as long as the background operation in WAS, WBS, and controller can be done in a short time.

Change of serving WAS. In SDWLAN, client is virtually associated with virtual AP stored on controller. When a client joins in the network, the "virtual AP management application" takes over association determination, and client's association state information is stored on controller after authentication. WAS acts as an agent of virtual AP, providing service on behalf of the virtual AP. So, we call it "*serving*" WAS rather than "*associated*" WAS here. The change of serving WAS in SDWLAN is equivalent to the change of associated AP in traditional architecture.

The first time a client joins in the network, virtual AP will assign the client a unique BSSID, and the serving WAS will communicate with the client using the assigned BSSID. Suppose client A is assigned $BSSID_A$, we only need to install two rules (shown in Fig. 7) on a WAS to let that WAS become client A's serving WAS. Obviously, the change of serving WAS can be easily achieved by deleting the two rules from old WAS and installing them on new WAS.

Route update in wired backbone. Steering the client's traffic from old WAS to new WAS is essentially to update route in wired backbone. OpenFlow excels at such kind of tasks and the controller can easily achieve this by installing appropriate rules in WBSes involved. Since WAS handoff is initiated by controller, we can proactively install rules to update route in wired backbone before updating rules on old WAS and new WAS. This can significantly eliminate the latency of route update in wired backbone.

IV. Evaluation

In this section, we evaluated the performance of fast WAS handoff in SDWLAN through extensive packet level simulations. Since our contribution focus on fast AP handoff mechanism, rather than handoff policy, we artificially executed handoff procedure in SDWLAN and traditional WLAN. We setup UDP and TCP connection between client and remote server node, and measured the instant throughput overtime. We calculate the instant throughput every 0.2 seconds, and the results are shown in Fig. 8 and Fig. 9.

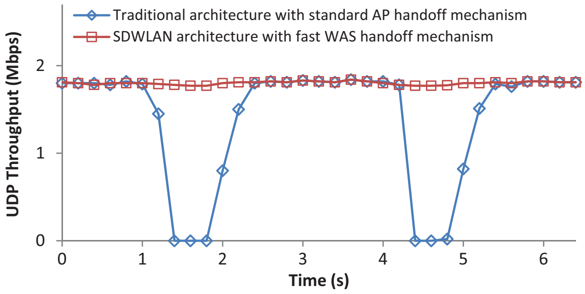


Fig. 8. Effect of AP handoff procedure on UDP instant throughput over time (calculate every 0.2 second). Handoff occurred 3 seconds and 6 seconds after the beginning the experiment respectively.

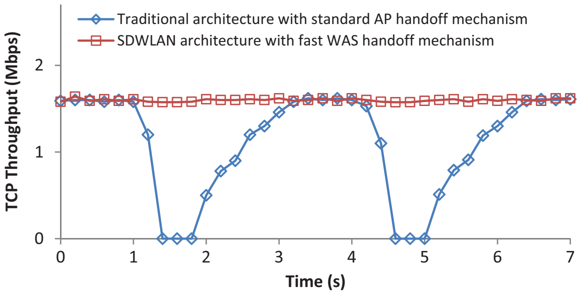


Fig. 9. Effect of AP handoff procedure on TCP instant throughput over time(calculate every 0.2 second). Handoff occurred 3 seconds and 6 seconds after the beginning of the experiment.

By comparing the effect of AP handoff mechanism on UDP and TCP throughput in traditional WLAN and SDWLAN, we found that 802.11 standard AP handoff mechanism in traditional architecture lead to a relatively long time interruption of TCP/UDP session. It takes nearly one second to recover from such interruption caused by AP handoff (For TCP, it takes over one second due to TCP's "Slow Start" mechanism). In Traditional WLAN, 802.11 standard AP handoff involves MAC-layer re-authentication and re-association operation, MAC address relearning (for layer-2 handof), and DHCP procedure or mobile IP register signaling (for layer-3 handoff). On the contrary, with the proposed fast WAS handoff mechanism in SDWLAN eliminates these procedures. So we found that in SDWLAN with fast WAS handoff mechanism, both UDP and TCP throughput are slightly affected by handoff operation. It indicates that AP handoff operation performed by controller can be done very quickly, incurring very short duration of communication interruption.

V. Related Work

Attempts to apply SDN to WLAN can be found in [11], [12] and [13]. [11] and [12] proposed to use OpenFlow to monitor traffic flows and provide a GUI for administrators to control traffic flows. [12] aimed to realize network virtualization by OpenFlow, and slice network according to user's requirements or application characteristics. But [12] does not allow to control the IEEE 802.11 MAC layer.

In Odin [13], users' association states are kept on a central controller and AP is responsible for authentication and beaconing. Odin introduce LVAP, which records a user's association context. With a user's LVAP, AP can communicate with the user. AP handoff in odin is realized by removing LVAP from old AP and spawning it in new AP. This inevitably takes quite some time, while AP handoff in SDWLAN have no such overhead. Moreover, since LVAP contains client's key, client's mobility may lead to key scattering throughout several APs. This increases security risk. There is no such issue in SDWLAN because client's key is stored in only one en/decryption appliance.

CloudMAC [14] also lifts MAC-layer management function onto central controller. However, CloudMAC don't mention how to unified the wired controller and wireless controller. Moreover, due to the lack of OpenFlow extension, CloudMAC only supports switching all the associated clients from one AP to a new AP at the same time, which don't satisfy the requirement of per-client AP handoff.

VI. Conclusion

Today's enterprise WLAN suffers from inflexible architecture and lack of coordination between wireless APs and wired backbone. Inspired by the emerging idea of "Software-Defined Networking" (SDN), we proposed *SDWLAN*, an alternative architecture for enterprise WLAN. In SDWLAN, most of 802.11 MAC-layer function have been extracted from original APs and put onto a central controller and the control of wireless APs and wired backbone are consolidated. In SDWLAN, adding new feature only needs to update software in the controller-side rather than update many APs. This greatly increasing the flexibility and extensibility of enterprise WLAN. Benefiting from SDWLAN, we proposed a client-unaware fast AP handoff mechanism, and demonstrated by experiment that it can significantly reduce the adverse effect of AP handoff procedure on user's on-going transmission, compared with 802.11 standard AP handoff mechanism in traditional architecture. Moreover, SDWLAN requires no modification to existing 802.11 clients, which make the solution practical.

Acknowledgment

This work was partially supported by the National Science Foundation of China under Grant No. 61379144 (2014-2017). The authors wish to thank Prof. Jiannong Cao, from the Hong Kong Polytechnic University, for his suggestion on SDN and paper writing.

References

[1] Cisco, "Cisco wireless control system." [Online]. Available: http://www.cisco.com/c/en/us/products/wireless/wireless-control-system/index.html.

[2] Aruba, "Enterprise solutions from Aruba networks," Aruba Networks. [Online]. Available: http://www.arubanetworks.com/solutions/enterprise.php

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[4] *802.11-2007: IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11-2007, 2007.

[5] "OpenFlow switch specification version 1.4.0," Open Networking Foundation (ONF), October 2013. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

[6] "Virtual cells: The only scalable multi-channel deployment," White Paper, Meru Networks, Aug 2009.

[7] C. Yue, G. Xue, H. Zhu, J. Yu, and M. Li, "S3: Characterizing sociality for user-friendly steady load balancing in enterprise WLANs," in *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 491–499, 2013.

[8] Y. Bejerano, S.-J. Han, and L. Li, "Fairness and load balancing in wireless lans using association control," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 3, pp. 560–573, Jun. 2007.

[9] R. Murty, J. Padhye, A. Wolman, and M. Welsh, "Dyson: An architecture for extensible wireless lans," in *Proceedings of USENIX Annual Technical Conference (USENIX ATC)*, pp. 10–15, 2010.

[10] I. Jabri, N. Krommenacker, T. Divoux, and A. Soudani, "IEEE 802.11 load balancing: An approach for QoS enhancement," *International Journal of Wireless Information Networks*, vol. 15, no. 1, pp. 16–30, 2008.

[11] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A. Moore, A. Koliousis, and J. Sventek, "Control and understanding: Owning your home network," in *Proceedings of International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pp. 1–10, 2012.

[12] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown, "Slicing home networks," in *Proceedings of ACM SIGCOMM Workshop on Home networks (HomeNets)*, pp. 1–6, 2011.

[13] L.Suresh, J.Schulz-Zander, R.Merz, A.Feldmann, and T.Vazao, "Towards programmable enterprise WLANs with Odin," in *Proceedings of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networks (HotSDN)*, pp. 49–54, 2012.

[14] P. Dely, A. Kassler, J. Vestin, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: An OpenFlow-based architecture for 802.11 MAC Layer processing in the cloud," in *Proceedings of IEEE Broadband Wireless Access Workshop*, 2012.