User-oriented Load Balance in Software-Defined Campus WLANs

Jie Feng, Liqiang Zhao, Chen Chen, Zhiyuan Ren, Jianbo Du
State Key Laboratory of Integrated Service Networks, Xidian University
Xi'an, Shaanxi 710071, China
E-mail: 784852087@qq.com, lqzhao@mail.xidian.edu.cn, cc2000@mail.xidian.edu.cn, zyren@s-an.org, dujianboo@163.com

***Abstract*—In this paper, we propose a novel concept of virtual resource chain for Software Defined Campus WLANs (SD-WLANs). A typical SD-WLAN is composed of three layers, i.e., infrastructure, access control and application layers. Firstly, the network control plane is decoupled from the forwarding plane, and soft-defined access points (SD-APs) merely execute the forwarding rules according to the instructions from access controllers (ACs). Secondly, with a global view of the network, AC could abstract, encapsulate, and virtualize the physical resources (e.g., computing, storage, and radio resources). Finally, by binding all the resources between the infrastructure and access control layers, isolated virtual resource chains are built up and simultaneously mapped to northbound interface (NBI) to accommodate various services at the application layer. Benefiting from the virtual resource chain, a user-oriented load balance scheme is presented in this paper. In SD-WLANs, the load of APs could be balanced according to the user's demands, and our proposed user-oriented load balance scheme could be easily invoked only by coding at the application layer. Experiment results demonstrate that our scheme is able to distribute mobile stations among all the APs and increase the average system throughput, and thus to increase the flexibility and availability of networks.**

Keywords—Software-Defined Campus WLAN ; Access Controller; Load Balance; Software Defined Access Point.

I. INTRODUCTION

Deployments of IEEE 802.11x wireless local area networks (WLANs) in campus have increased sharply due to the growing number of client devices, such as smart-phones and tablets. In order to guarantee better quality of experience (QoE) with the rapid growth of user scale and traffic loads, more and more access points (APs) and access controllers (ACs) have to be employed in conventional WLANs to provide a variety of network functions or services. Some possible solutions have been proposed by vendors, but most of these solutions are proprietary and complex, which may hinder the deployment of new services.

Software-defined networking (SDN) [1] is a new network paradigm that could make networks more flexible and manageable, where the network control is decoupled from forwarding and is directly programmable. Generally, SDN is complemented by network function virtualization (NFV) [2], which advocates the virtualization of network functions as software modules running on standardized IT infrastructure, and this complementation can improve the flexibility and simplicity of networks and service delivery.

Users are not evenly distributed so that some APs tend to suffer from heavy load, while their adjacent APs may carry only light load. Load imbalance among APs can incur congestion, so it is undesirable in WLANs. In [3], a centralized operation agent collects the load information of all APs, and manages their respective coverage sizes from the overall network level to control the load distribution among APs. A cooperative Load Balancing (LB) paradigm is implemented in [4] by leveraging both centralized and distributed resource-allocation algorithms to provide a high area spectral efficiency (ASE) and the highest throughput. In [5], Ong et al. present a comparison of three dynamic load distribution algorithms, viz., predictive load balancing (PLB), predictive quality of service (QoS) balancing (PQB), reactive QoS balancing (RQB), and analyze the performance of different algorithms. These load balance algorithms may improve the network performance in some aspects, while each of them is dedicated to a specific scenario.

In this paper, we propose software-defined campus WLANs (SD-WLANs)–an open SDN framework for campus WLANs. A novel virtual resource chain bound all the resources in network is developed to support various network applications. Benefiting from the virtual resource chain, a user-oriented load balance scheme is implemented to improve the resource utilization.

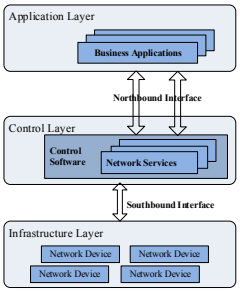
The remaining of this paper is organized as follows. Section II presents the related works on software-defined WLANs. The description of open SD-WLANs framework and virtual resource chain is provided in Section III. The user-oriented load balance is elaborated in Section IV. The experiment results are shown in Section V. Finally, Section VI presents our conclusions.

II. BACKGROUNDS AND RELATED WORK

As shown in figure 1, SDN consists of three layers, i.e., infrastructure, control and application layers. The infrastructure layer has the ability of providing resources for the controller via southbound interfaces (SBIs). The application layer sends the network requirements to the controller plane via northbound interfaces (NBIs). SDN

978-1-5090-1698-3/16/$31.00 ©2016 IEEE

controller exerts low-level control functions over the network elements via SBIs, and provides relevant information up to the application layer.



基于Odin，这个文章提出了在SDWLAN下的考虑负载的切换算法。除了考虑接收信号强度外还考虑AP上的流量负载。解决了负载不均衡并提供了无缝的移动性

Figure 1. Framework of SDN

Odin [6] is a programmable framework for simplifying association decisions of clients in WLANs, which allows network applications and services to be deployed on a centralized controller. Based on Odin, [7] proposes a load-aware hand-off algorithm in SD-WLANs, which considers traffic load of APs in addition to receiving signal strength at wireless clients to solve load imbalance among APs and offer seamless mobility. Similarly, [8] presents a software-defined enterprise WLAN based on the OpenFlow-based AP and controller. These schemes are proposed to resolve some specific problems like hand-off in WLANs, but they cannot deal with some other problems, such as wireless resource virtualization and management.

III. VIRTUAL RESOURCE CHAIN BASED FRAMEWORK FOR SD-WLANS

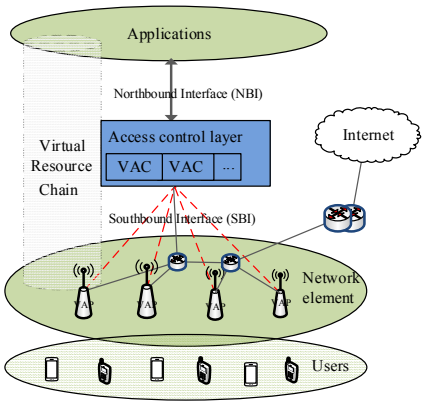


Figure 2. SD-WLANs Framework

This section, we present a virtual resource chain based on open framework for campus WLANs, including virtual access controller (VACs), virtual access point (VAP) and the virtual resource chain (VRC), as shown in figure 2.

A. Vitual access controller

VACs are virtual machines running as normal programs inside a host operating system (OS). VACs can orchestrate applications' demands based on limited network resources. Firstly, the VACs can abstract details of the hardware resources (i.e., network, computing, and storage) in the local access controller, and slice them into isolated virtual resources. Secondly, northbound interfaces bound the virtual resources are called by upper-layer application. In addition, the VACs visit the infrastructure layer via SBI to control the network devices.

B. Vitual access point

VAP is an abstraction of the physical AP, which consists of the virtual expression of radio resources (e.g., bandwidth), the operating status (e.g., packet loss rate), and the hardware resources (i.e., computing, and storage) of AP. It runs in the physical AP and exposes a necessary interface (i.e., SBI) for VACs to orchestrate resources.

C. Vitual resource chain

By abstracting the infrastructure and access control layer respectively, VAPs and VACs are developed for the application layer. And to obtain a global view of all the resources at different layers, a high-level abstraction, i.e., VRC is proposed.

As VRC masks the details of virtual entities, the implementation of the resource virtualization and management is transparent to the application layer. More importantly, the combination of the infrastructure and access control layers facilitates the dynamic migration of applications. That is, the network applications could be deployed anywhere more than running on a particular platform (e.g., the central server). If they can get the interface of virtual resource chain, the applications can migrate between different network devices and different logical layers.

IV. USER-ORIENTED LOAD BALANCE

In this section, we introduce the implementation of VAP the extended OpenFlow (i.e., exOpenFlow) protocol, VAC for load balance, respectively.

A. Implementation of VAP

Based on light virtual access point (LVAP) [6], we develop VAP in a commercial AP (i.e., D-link) embedded the OpenWrt operation system. LVAP virtualizes association state and is separated from the physical AP, and each LVAP can also be treated as a virtual client.

A VAP is composed of three parts, i.e., channel monitoring, load status monitoring, and exOpenFlow agent. Here we emphasize on the channel monitoring and load-monitoring module to support the load balance.

The channel-monitoring module is used to estimate the signal to interference and noise ratio (SINR). If $SINR_{ij}$ denotes the SINR of the i-th user equipment (UE_i) associated with the j-th AP (AP_j), we get

$$SINR_{ij} = \frac{g_{ij}P_j}{\sum_{k \in A_i \cap k \neq j} g_{ik}P_{k+1}N_0} \quad (1)$$

where g_{ij} is the channel gain from AP_j to UE_i, P_j is the AP's transmit power, N_0 is the power of additive Gaussian white noise, and A_i is the set of APs covering the user j. $P_j \in [P_{min}, P_{max}]$, where P_{min} and P_{max} are the minimum and maximum

transmission power, respectively. User i reports the channel quality indicator (CQI) to the VAC, thus the VAC can adjust the resources allocation.

There are various load or admission control algorithms, which design suitable load matrixes to estimate the available network capacity accurately. The common load metric is based on channel utilization, which estimates the fraction of channel occupation time per observation interval. Accordingly, to obtain, the parameter of an AP's load, the channel utilization, we first estimate the average transmission rate of each link between an AP and its users. As each bit rate level is determined by a range of SINR, we could get

$$r_{ij} = f(G, P) \tag{2}$$

where r_{ij} is the bit rate between UE_i and AP_j, G is the channel gain matrix with the rows representing users and the columns representing APs, and P is the vector of transmit powers. And then we have

$$L_i = \frac{\sum_{i=1}^{N} t_i}{T} = \frac{\sum_{i=1}^{N} \frac{l_i}{r_{ij}}}{T} \tag{3}$$

where L_i is defined as the channel utilization with transmission or reception of N frames in a given time period T (in seconds). Its calculation includes all data and control frame. Therefore, we need to calculate duration t_i by the frame length l_i and PHY data rate r_{ij} for each MAC frame.

To quantify both QoS explicitly and wireless channel variations implicitly, we consider link layer measurement about packet delay *PD* and packet loss rate *PLR* as the QoS load metrics. Therefore, the QoS balancing algorithms can be implemented by the QoS load metrics.

B. Implementation of VAP Extened openflow (exOpenFlow)

It has been argued that programming WLANs using OpenFlow don't cover the 802.11 MAC functions currently (e.g., handling association logic, and configuring wireless interfaces). So we develop the exOpenFlow, which considers the core component of SD-WLANs as a standard communication protocol between the infrastructure and access control layers. The new matching fields and actions (or called instructions) are implemented in OXM format.

New matching field: *lvap_bssid*. When the controller determines to move a LVAP to an old SD-AP, or remove a LVAP from the SD-AP, the flow tables containing this match field will be sent to the corresponding SD-APs.

New actions: *rm_lvap, im_lvap, get_vap_property, set_vap_property.* The new actions ***rm_lvap, im_lvap*** are developed to implement some mobile management tasks. As for the ***get_vap_property*** and ***set_vap_property***, they are the key actions for VACs to get and set properties for the VAPs.

C. VAC for load blance

As illustrated in figure 3, our solution provides different load balance services for load balance entity and load estimation entity.

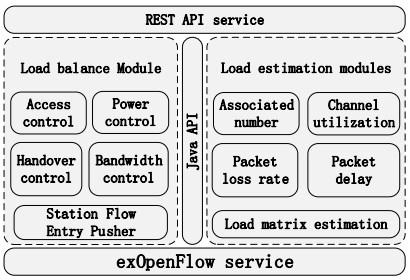


Figure 3. Framework of load balance VAC

The admission control regulates the network load by operations under the saturation threshold to protect QoS of existing flows. Handover control is implemented by removing and spawning LVAPs to balance the associated clients among APs. Inspired by Yigal Bejerano's work [3], we develop the power control module to conduct load balance by adjusting the coverage area of APs. The bandwidth control module is designed to relieve the influence of mass data transmission, which avoids the unnecessary handovers. The load contribution indicator (LCI) illustrates the load contribution of a client to the AP's load, which can be expressed in equation (3).

Load estimation entity consists of several parameters associated with channel impairments and network congestions. We choose *PD* and *PLR* as the critical QoS parameters for multimedia traffic to reliably select the best access point according to the QoS requirements of different traffic classes. The channel utilization and associated the number of users implement the load balance algorithms without QoS requirements. All these parameters are estimated through the load matrix estimation module, and the required data is obtained by invoking the exOpenFlow API.

Algorithm 1 User-oriented Load Balance Algorithm

---

1 VAC sets initial load balance subscription (thresholds of the load indicator) in APs
2 APs send updated loads statistics (load matrix) to VAC periodically
4 ***loop***
VAC analyses loads statistics of APs when load indicator exceeds the threshold
5 ***if*** (all AP's channel utilazation > threshold)
VAC invokes the access and bandwidth control module
6 ***else***
VAC analyses the LCI in overloaded APs
7 ***if*** (some clients' LCI > threshold)
VAC invokes the bandwidth control module
8 ***else***
VAC invokes the handover control module
9 ***end if***
10 ***end loop***

Algorithm1 describes the proposed load balance algorithm running on the VAC. Due to space constraints, we choose channel utilization as the load indicator of APs, and the balance algorithm is based on handover and bandwidth control.

As for the QoS-balanced application, only when service QoS cannot be satisfied or a better quality access point is

required will a balance algorithm be triggered. We choose QoS satisfaction factor (QSF) [5] as the indicator.

V. EXPERIMENTAL RESULTS

A. Experiment scenarios

In our experiments, two SD-APs are deployed on the same floor, and all the switches are OpenFlow enabled switches. All SD-APs are implemented using OpenWrt embedded Linux package and equipped the wireless network card using Atheros AR9220 chipset. Floodlight is the access controller in our experiments, and we add some modules to support resource virtualization. We also extend the OpenFlow1.3 in Open- vSwitch, which is embedded by the OpenWrt to support exOpenFlow and the VAP virtualization. The load offered by clients is generated by *iperf* client/server tools, and all the clients run as *iperf* clients.

We set a fixed threshold of load in AP. When the load of AP exceeds this limit, the load balance algorithm will be triggered. The initial threshold of load is fixed at 0.9 for all APs in the WLAN system, and the threshold of LCI for clients is 0.05. If the load of AP is above 0.9, a new client will not be served by this AP. At VAC, updating L_i value of the AP is used for detecting trigger of load balance events. In our algorithm, the first condition ensures that the number of unnecessary handover is the minimum, and at the same time a good QoS can be guaranteed for all the clients by providing sufficient bandwidth. The second condition avoids occurring the Ping-Pong effects between the APs. When load difference between APs is below the load hysteresis margin, and all the APs are suffering heavy load, the handover algorithm will abort.

B. Performance of our algorithm

For this experiment, the APs are deployed with some of the clients being deployed in overlapping area. Initially, one client only connects to AP1, and the rest are associated with AP2, thus creating the imbalance between APs. This experiment runs duration of 600 seconds and obtains an average value for every 20 seconds, and the first 200 seconds is warming up time for the experiment, which has no load balance control. Only then can we start the load balance control in VAC.

In figure 4, channel utilization represents the load indicator of AP, which illustrates the channel busy time. The figure shows the load of AP2 is about three times greater than that of the AP1 on the first 200 seconds，and then the load balance algorithm is triggered. With some of the clients handing off to AP1, the load of AP1 and AP2 shrinks towards equality, and both APs are under the threshold.

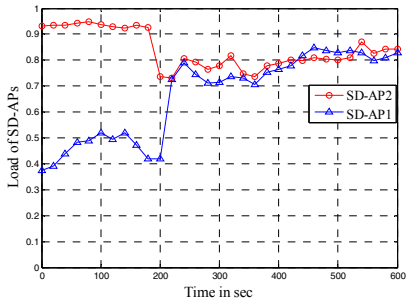


Figure 4. Load of APs during the experiment

The average QoS indicator of clients in our experiment is shown in Fig.5. In the first 200 seconds, the average packet loss rate of clients connecting to AP1 is much lower than the ones associated with AP2, which means the clients connecting to AP2 are suffering from poor network service. And then the load balance algorithm takes effect. For the clients to connect to AP1, the average packet loss rate grows from about 0.4% to about 1% , because some of the clients connecting to AP2 hand off to AP1, increasing the load of AP1. Thus, the average QoS indicator of clients connecting to AP2 is improved after the clients hand off to AP1.

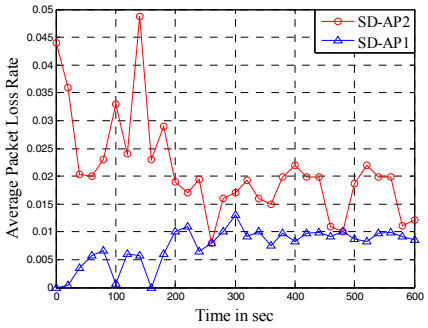


Figure 5. Average packet loss rate of clients

Figure 6 shows the average throughput of clients connecting to AP1 and AP2. At the beginning, average throughput of clients connecting to AP1 is about 34Mbps, while AP2 is around 13Mbps. Some of the clients hand off to AP1 at about 200 seconds, and then the average throughput of all clients connecting to AP2 increases to about 22Mbps. Although the average throughput of clients connecting to AP1 drops to about 20Mbps, the average throughput of all the clients is increased greatly.

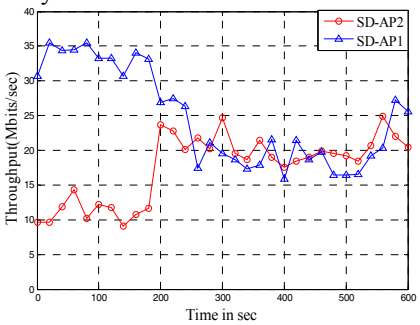


Figure 6. Average throughput of clients

VI. CONCLUSION

In this paper, we propose the SD-WLANs and virtual resource chain aiming at providing the developers with expressive tools to control the state of the network while hiding away the implementation details of the underlying technology. The proposed virtual resource chain catches the four fundamental elements that compose a network control loop, namely REST API, resource pool, exOpenFlow and VAP API. And a user-oriented load balance scheme is developed to provide a cost-effective, efficient and transparent method to expand the bandwidth of network devices, to increase the system throughput, and to enhance the data process capability, thus to increase the flexibility and availability of networks in the end.

For future work, firstly, we shall develop more applications to evaluate our designed platform. Secondly, we shall evaluate the effectiveness of hardware resources (e.g., computing and storage) on VACs.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China (No. 61372070), Natural Science Basic Research Plan in Shaanxi Province of China (2015JM6324), Hong Kong, Macao and Taiwan Science & Technology Cooperation Program of China (2014DFT10320), and the 111 Project (B08038).

REFERENCES

[1] "Software-Defined Networking: The new norm for networks," White Paper, Open Networking Foundation (ONF), April 2012.

[2] ETSI, "Network Functions Virtualization (NFV); Architectural Framework," GS NFV 002 (v. 1.1.1), Oct. 2013.

[3] Bejerano, Y., Seung-jae Han, "Cell Breathing Techniques for Load Balancing in Wireless LANs", IEEE Transactions on Mobile Computing, Pages: 735-749, 2009.

[4] Xuan Li; Rong Zhang; Hanzo, L., "Cooperative Load Balancing in Hybrid Visible Light Communications and WiFi", IEEE Transactions on Communications, Pages: 1319-1329, 2015.

[5] Eng Hwee Ong, Khan, J.Y., Mahata, K. "On Dynamic Load Distribution Algorithms for Multi-AP WLAN under Diverse Conditions" in Wireless Communications and Networking Conference (WCNC), Pages: 1 – 6, 2010.

[6] 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.

[7] A.K. Rangisetti, B.H. Bhopabhai, B.P. Kumar and B.R. Tamma, "Load-aware Hand-offs in Software Defined Wireless LANs" in international Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 685-690, 2014.

[8] W.–S. Kim, S.–H. Chung, C.-W. Ahn, and M.-R. Do, "Sesmless Handoff and Performace Anomaly Reduction Schemes based on OpenFlow Acess Points," in Advanced Information Networking and Applications Workshops, pp. 316-321, 2014.