

RESEARCH

Open Access



# Terminal handover in software-defined WLANs

Zhaohui Chen<sup>1</sup>, Zhaoyang Luo<sup>1</sup>, Xiaohui Duan<sup>1</sup> and Lianming Zhang<sup>1,2\*</sup>

## Abstract

With the development of wireless LAN (WLAN) technology, wireless access points (APs) are deployed increasingly more and the number of mobile terminals is increasing. It is normal for a mobile terminal to hand over its AP. The traditional terminal handover method cannot effectively control the handover delay and balance the network load due to the inherent drawbacks such as closed encapsulation and the lack of centralized control. In this paper, we introduce the innovative network architecture of software-defined networking (SDN), propose a software-defined WLAN architecture, and propose a seamless handover method and a load balancing handover method of the software-defined WLAN. Experimental results show that the proposed handover method effectively reduces the handover delay, balances the network load, and improves the quality of service (QoS) and resource utilization of the network.

**Keywords:** SDN, Software-defined WLAN, Seamless handover, Load balancing

## 1 Introduction

### 1.1 Background and motivation

With the deepening of social information processes and the rapid development of wireless communication technology, the demand for wireless networks, the continuous growth of traffic, and the continuous expansion of coverage areas have promoted the massive deployment of wireless LANs (WLANs) and the intensive use of wireless access points (APs) [1]. In a network environment where APs are densely deployed, to ensure the user's network experience, the APs must be switched with the terminals. However, the terminal handover method based on the traditional WLAN has been unable to meet the increasing demands of users for high-quality, high-rate services of the network. In traditional WLAN architectures such as fat APs and thin APs [2], the collection of network information is a relatively difficult process, it is impossible to coordinate scheduling, and load migration

is difficult to achieve. As a result, the handover is blind, and the traffic scheduling is unbalanced. The quality of service (QoS) provided by the wireless network and the resource utilization rate of the network cannot be guaranteed.

The emergence of software-defined networking (SDN) [3–9] provides a new way to thoroughly solve this problem. Its main idea is to decouple the control function of the forwarding function of the network and hand over the control to the centralized controller. Because of the global perspective of the controller and the open programmable interface of the network, users can obtain the state information on all the underlying devices and realize traffic scheduling according to their own needs. The network becomes more flexible and intelligent. The terminal handover in SDN-based WLANs can effectively improve the QoS and the resource utilization of WLANs. As a new computing paradigm, software-defined networking can also be applied to cellular networks [10], D2D heterogeneous networks [11], MESH networks [12], cloud computing [13], and edge computing [14] to facilitate resource allocation, provide service quality assurance, and improve user experience.

\*Correspondence: [zm@hunnu.edu.cn](mailto:zm@hunnu.edu.cn)

<sup>1</sup>College of Information Science and Engineering, Hunan Normal University, No.36, Lushan Road, 410081 Changsha, China

<sup>2</sup>Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, No.36, Lushan Road, Changsha, China

## 1.2 Related work

To solve the above problems, many researchers have devoted themselves to researching SDN-based WLAN architecture and terminal handover methods [15–29]. Odin [15] enabled the seamless handover of terminals by performing virtual access point (VAP) migration. However, the scheme is just a simple functional demonstration. It omits related processes such as authentication in the access process. It cannot meet the needs of practical applications. CloudMAC [16] is also based on the virtual AP to transfer traffic between APs by controlling the data forwarding behavior of the forwarding equipment with the SDN controller. Although CloudMAC realizes the fast handover of terminals, the handover method requires that all the terminals connected with the source VAP are switched to the destination VAP as a whole. A single terminal cannot be flexibly switched. The SDWLAN architecture [17] and One Big AP [18] handover method of this architecture migrates the control function of the AP to the SDN controller. The controller conducts the data forwarding operation according to the deployed strategy. The controller needs to virtualize one VAP for all APs. All the packets need to be processed by the controller. It is easy about the controller to become the performance bottleneck in the network. In [19], Rangiseti et al. proposed a novel load-aware hand-off algorithm for SDN-based WLAN systems which considers traffic load of APs in addition to received signal strength at wireless clients to solve load imbalance among APs and offer seamless mobility. Kiran et al. [20] proposed a seamless handover method of WLAN terminals based on an SDN, referred to as the KYA method. However, the KYA method is not comprehensive enough to judge the performance of APs based on their throughput. It cannot reflect the performance of APs as a whole. In a wireless network of a complex environment, the choice of the destination AP is not necessarily accurate. In [21], Sequeira et al. presented the development and testing of an open enterprise Wi-Fi solution based on virtual APs, managed by a central WLAN controller, and it allows seamless handovers between APs in different channels, maintaining the QoS of real-time services. Gilani et al. [22] presented a novel mobility management scheme based on logical AP that keeps a connection with the user terminal during hand-off triggered by either the user or the SDN controller of seamless mobility. Nahida et al. [23] introduced a new SDN-based Wi-Fi architecture and proposed an AP load balance based handover algorithm. Zahid et al. [24] studied software-defined Wi-Fi networks to understand the ability of SDWN to effectively hide the handover delay between AP of the adoption of the SDWN architecture of Wi-Fi networks. In [25], the authors presented an SDN approach to design and implement a rate/link adaptation (RA) framework of WLAN. The framework provides

support for real-time RA applications and flexibility to satisfy various degrees of QoS or quality of experience (QoE) requirements. The authors in [26] introduced an improved IEEE 802.11 architecture utilizing SDNs, and the proposed architecture allows communications during device movements without losing a QoS. The fast seamless handover of QoS enables efficient usage of radio resources in large networks. In [27], Coronado et al. introduced a flexible, programmable, and open source SDN platform for heterogeneous 5G radio access networks (RANs). The platform allows network programmers to deploy complex management tasks as policies on top of a programmable centralized controller. Coronado et al. [28] presented a centralized approach to manage multicast flows based on SDN principles and reported on a proof-of-concept implementation that can be used in enterprise WLANs. Sen and Sivalingam [29] presented the testbed implementation details of a seamless handover mechanism for a software-defined enterprise WLAN framework. The authors of [30] presented an overview of SDWLAN architectures and provided a qualitative comparison in terms of features such as programmability and virtualization.

The above research solutions all solve the problem of the terminal handover method to a certain extent. However, they cannot flexibly implement the handover of a single terminal, or they cannot accurately judge the performance of the APs.

## 1.3 Contributions

To effectively solve the problem of terminal handover problem of the network, improve the QoS of networks, and improve the utilization of network resources, we propose an SDN-based WLAN architecture and implement a seamless handover method based on this architecture of this paper. The main contributions to this paper can be summarized as follows.

- An SDN-based WLAN architecture is proposed, which can realize seamless handovers and load balancing handover. The proposed architecture has the characteristics of separated control and forwarding, centralized control, and open programming, making the network flexible and controllable. The controller has a global perspective. Users can write applications based on their own needs.
- A seamless handover method is proposed. The method first effectively controls the handover delay by streamlining the access procedure of the handover and reduces the data propagation delay by releasing the flow table in advance, allowing seamless handovers to be realized. The experimental results show that the method reduces the overall delay of

terminal handovers and improves the QoS of the network.

- A load-balancing handover method is proposed. The global perspective and centralized control ability of the controller are used to collect, control, and analyze the information. According to the information provided by the controller, the load balancing module uses the algorithm proposed in this paper to calculate the weights and formulate the handover strategy. The controller realizes the handover of the terminal according to the handover strategy of the load balancing module. The experimental results show that the method improves the throughput of the system and improves the utilization of network resources.

The remainder of this paper is organized as follows. In Section 2, an SDN-based WLAN architecture is proposed, and the working mechanism of each module and the design of each component is described in detail. In Section 3, a seamless handover method and a load balancing handover method is proposed and the main ideas and working processes of the two handover methods are described. In Section 4, the algorithm is deployed in an SDN controller, and the performance of

the algorithm is tested. Finally, concluding remarks are given.

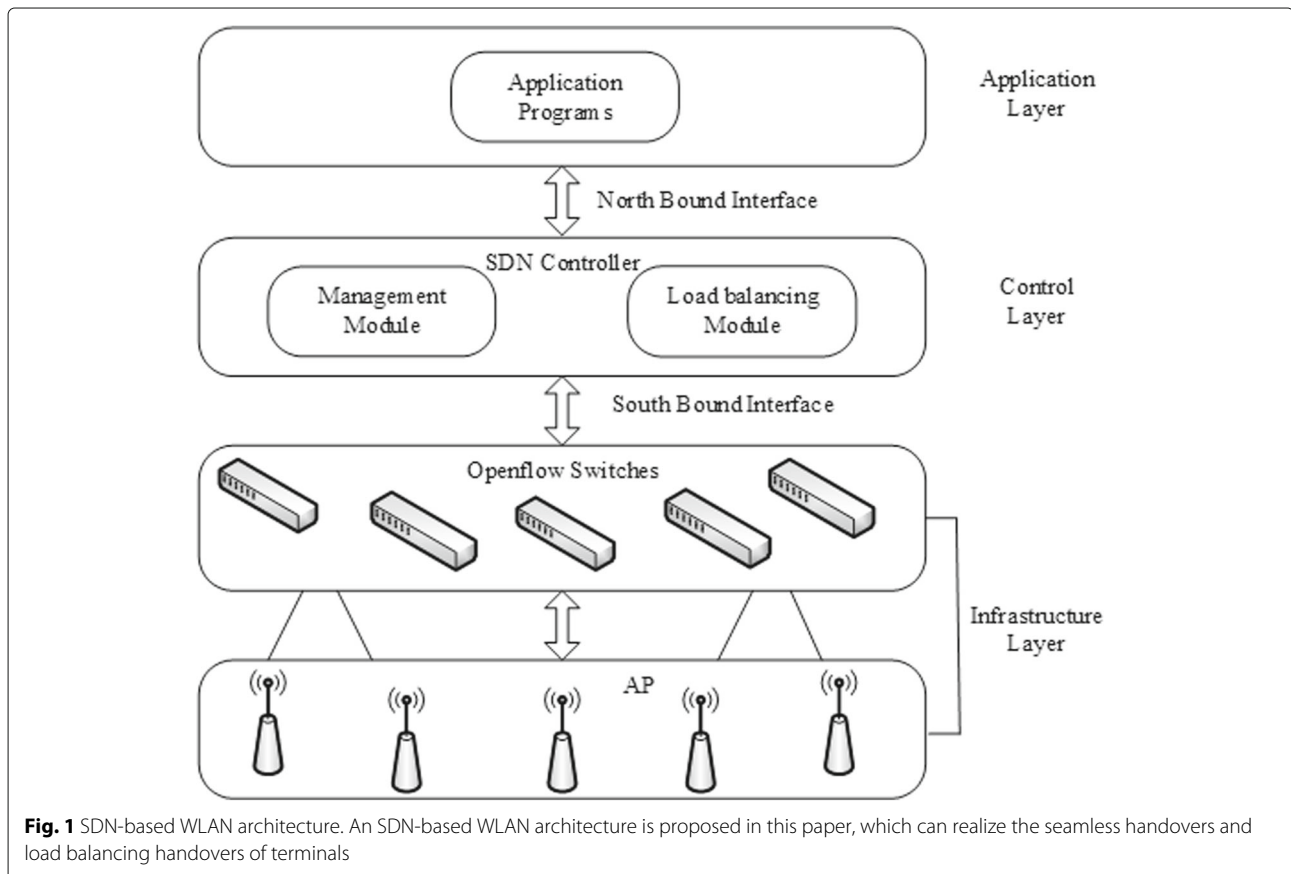
## 2 Network architecture

An SDN-based WLAN architecture is proposed in this paper, which can realize the seamless handovers and load balancing handovers of terminals, as shown in Fig. 1. The network architecture is divided into three layers: the infrastructure layer, the control layer, and the application layer. The infrastructure layer is composed of APs and OpenFlow switches, the control layer is composed of SDN controllers, and the application layer is composed of applications realized by independent programming. The architecture is centered on the control layer. The control layer interacts with the infrastructure layer through the southbound interface and performs data interaction with the application layer through the northbound interface.

### 2.1 Components

#### 2.1.1 AP

The APs in this architecture only retain the data forwarding function, and their logic control functions are moved to the controller. The current OpenFlow protocol [6] does not fully consider the characteristics of a wireless network and the controller cannot manage the APs. It is necessary



to establish a separate communication channel between the controller and each AP and the controller collects the state information on each AP through that channel, thereby implementing control of the APs.

### 2.1.2 Switch

The OpenFlow switch, similar to an AP, has its original control functions stripped and moved to the controller, retaining only the data forwarding function. Such a switch is equivalent to a pure data channel.

### 2.1.3 Controller

The controller is a centralized and unified logic control unit. It is responsible for the collection of device status information, the management of OpenFlow switches and APs, and the delivery of flow tables. The controller also provides a programmable interface, which is convenient for users to write applications according to their own needs.

## 2.2 Module function

### 2.2.1 Management module

The module is mainly used to manage APs and terminals. The APs in the network are stripped of the management control function, and each AP connects to a different number of mobile terminals. There needs to be a module to manage the large number of APs and terminals in the network, so a management module is deployed in the controller to manage the APs and terminals. The module is divided into two submodules: the AP management module and the terminal management module. The main functions of the AP management module are to discover and configure APs and to manage and collect the status information of each AP. The module mainly controls the authentication process of its access network.

### 2.2.2 Load balancing module

The module is deployed in the application layer to realize the concept of load balancing, and it is divided into the load sensing module and the handover decision module. The former mainly obtains the state information of each AP through the management module deployed in the controller. The latter mainly obtains the state information of the network according to the load sensing module, and it uses the AP weight algorithm proposed in this paper to assign a weight to each AP and its corresponding terminal. According to the weights, the traffic in the network is scheduled, and the terminal is switched, which balances the network load.

## 3 Methods for terminal handover

With the above architecture, we propose a seamless handover method and load balancing handover method, which can effectively improve the QoS of networks and the utilization of network resources.

### 3.1 Seamless handover

The delay caused by terminal handover is an important factor affecting users' Internet experience [31]. In proposed SDN-based WLAN architecture, since the terminal has been connected to the network, its related information has been stored in the controller, and the scanning and authentication process has been omitted without causing delay. In addition to the handover delay in the terminal handover process, the network delay mainly occurs to four processes: queuing, node processing, transmission, and propagation. Here, the delays in the first three processes are on the order for microseconds, and the delay in the propagation process is on the order milliseconds. Therefore, this paper mainly considers the handover delay and propagation delay.

#### 3.1.1 Handover delay

The traditional handover method usually takes hundreds of milliseconds [32]. It may not cause serious problems of some services, but it may seriously restrict real-time applications such as voice over Internet protocol (VoIP) or online games.

The handover method proposed in this paper mainly reduces the handover delay by optimizing the terminal access process. Configuring the network, we set the service set identifiers (SSIDs) and basic service set identifier (BSSIDs) of all APs to be the same. In wireless networks, terminals identify APs by identifying the SSIDs and BSSIDs; therefore, when the terminal interacts with the wireless network, the data frames sending to the wireless network can be monitored by some APs with consistent SSIDs and BSSIDs. From the perspective of the terminal, there is only one logical "big AP" in the network, and there are no multiple independent physical APs in reality. In the SDN-based WLAN architecture, the authentication function of an AP is stripped and moved to the controller of its control function, so the AP is no longer responsible for the authentication operation of the terminal. The terminal authentication component is deployed in the management module of the controller of the architecture. After all the terminals perform the authentication operation through the controller when they access the network of the first time, the identity information on the terminal will be saved for the terminal authentication component of the controller. When the terminal reconnects to the network, the controller will compare its identity information. If the terminal authentication component already has the terminal information, there is no need for authentication.

#### 3.1.2 Propagation delay

When the terminal performs the handover, the terminal associated with the AP has changed, and the flow

table in the AP is still updated in the future. Therefore, in the packet forwarding process, there is a phenomenon where the matching cannot be performed, as shown in Fig. 2.

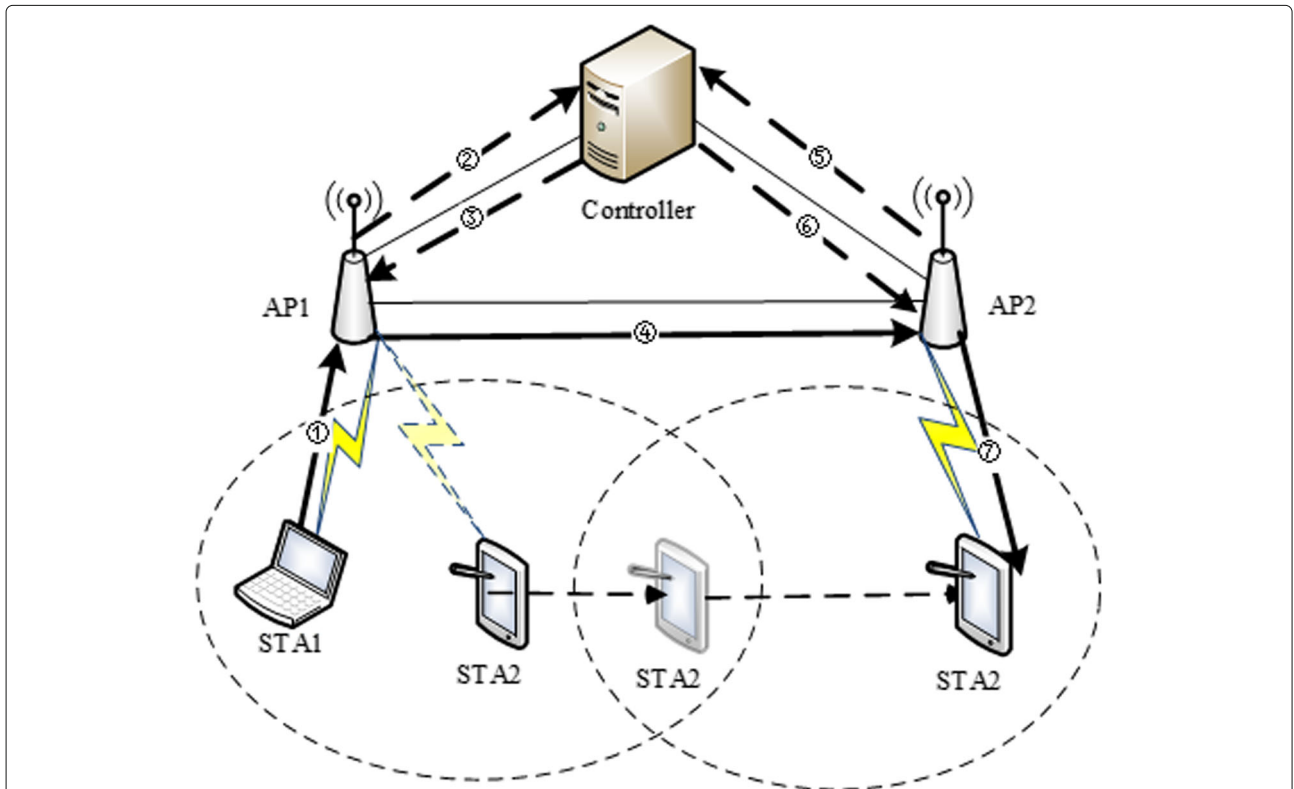
The server STA1 communicates with the client STA2. Both STA1 and STA2 are in the coverage area of AP1 and are connected to it. STA2 moves to the coverage area of AP2. When the handover condition is met, the association with STA2 and AP1 is canceled, and STA2 establishes a connection with AP2. Process ① is the transmission of the data packet sent by STA1 to STA2 to AP1. At this time, AP1 is disconnected from STA2, and there is no matching entry into the flow table. According to the OpenFlow standard protocol, once the switch receives the packet, when there is no match in the flow table in the switch, the received packet is sent to the controller by using the Packet-In event, and then the packet forwarding operation is performed according to the flow table sent by the switch. When the packet is sent to the switch and there is no match in the flow table of the switch, the received packet is sent to the controller by using the Packet-In event, and then the packet forwarding operation is performed according to the flow table sent by the packet. In process ②, the AP1 uses the Packet-In

event to send the data packet to the controller. In process ③, the controller sends a flow table to guide AP1 to perform packet forwarding operations. Process ④ involves AP1 forwarding the data packet. Process ⑤, similar to process ②, reports the Packet-In event to the controller. In process ⑥, the flow table is delivered in the same manner as in process ③. Process ⑦ involves AP2 forwarding the data packet to STA2 according to the flow table. In this case, this working process will have a corresponding delay.

Assume that the corresponding delays of each link in Fig. 2 are  $T_1, T_2, T_3, T_4, T_5, T_6,$  and  $T_7,$  respectively. The wireless link distance between STA1 and AP1 and between STA2 and AP2 is  $D$ ; the wireless transmission rate is  $C$ ; the wired link distance between the AP1, AP2, and the controller is  $L$ ; and the transmission rate is  $V$ .

The delay of the wireless link is  $T_w = \frac{D}{C}$ , and the delay of the wired link is  $T_x = \frac{L}{V}$ . When the flow entries in the AP match, the delay is

$$T_y = T_1 + T_4 + T_7 = \frac{2D}{C} + \frac{L}{V}. \tag{1}$$



**Fig. 2** Packet forwarding process. When the terminal performs the handover, the terminal associated with the AP has changed, and the flow table in the AP is still updated in the future. Therefore, in the packet forwarding process, there is a phenomenon where the matching cannot be performed

When there is no match in the flow table in the AP, the delay is

$$T_z = \sum_{i=1}^7 T_i = \frac{2D}{C} + \frac{5L}{V}. \quad (2)$$

From Eqs. 1 and 2, it can be seen that when there is no matching term in the AP, the propagation delay is  $\frac{4L}{V}$  more than that of flow table item matching, which is four times that of the wired link delay  $T_x$ . We effectively reduce the propagation delay by using the pre-transmit flow table strategy: when STA2 reaches the critical point of the coverage area of the two APs, it updates the flow table of the source AP and the destination AP before the handover is completed so that the packets sent from STA1 can be sent from STA1 without triggering the Packet-In event, thus saving  $T_2$ ,  $T_3$ ,  $T_5$ , and  $T_6$  and reducing the delay by  $\frac{4L}{V}$ .  $T_1$ ,  $T_4$ , and  $T_7$  are the inherent delays of the network.

The handover process of the seamless handover method under the architecture proposed in this paper is as follows.

(1) The terminal and AP1 complete the access process and then access the network.

(2) The terminal moves from the signal range of AP1 to the overlapping area of the signal range between AP1 and AP2.

(3) When the handover trigger condition is reached, the controller first selects the AP with the best comprehensive performance according to the policy to serve the terminal, assuming that the comprehensive performance of AP2 is better than AP1. Then, it performs the handover operation and at the same time issues instructions to the management modules of the two APs to perform the following two steps: ① update the flow table of the AP1, stop providing services to the terminal, and no longer receive and send the terminal data. ② Update the flow table of AP2, and make the data packet of the terminal completely responsible for receiving and transmitting the data of AP2.

(4) The terminal accesses the AP2.

Compared with the terminal handover method in the traditional architecture, the seamless handover method proposed in this paper is carried out under the guidance of the controller. The terminal does not need to participate and only needs to perform an authentication process when accessing the network for the first time. There is no need to reauthenticate and associate with the destination AP during the handover process, which reduces the time consumption during the handover process. Furthermore, using the “early flow table” to avoid the Packet-In event reduces the time consumption of the data dissemination process.

### 3.2 Load balancing handover

This paper proposes a load balancing terminal handover method. The main idea is as follows. The management

modules use the global perspective of the controller to collect the underlying APs and all the mobile terminal's state information, such as the parameters affecting the AP performance, and passes it to the load balancing module; then, the load balancing module performs weight calculation of each AP according to the state information collected by the algorithm determines the merits of each AP's comprehensive performance according to the weight value, and finally makes a handover decision, selecting the best-performing AP. This handover method is used to balance the network load and improve the overall efficiency of the network.

#### 3.2.1 Parameters affecting AP performance

In general, the parameters that affect the performance of APs are the signal strength  $S$  and the number of associated terminals  $N$ . The greater the received signal strength is, the better the transmission quality of the wireless link. In the traditional terminal handover, the signal strength is selected to judge the AP performance as the handover basis, the number of associated terminals can reflect the AP load to some extent, and, in general, the terminals associated with the same AP contend for the same channel to communicate. The ability of each terminal to contend for a channel is the same. The greater the number of APs associated with a terminal is, the greater the channel competition, and the greater the impact of the resulting collision on performance.

However, the two parameters of the signal strength and the number of access terminals cannot fully reflect the performance of an AP. The reasons are as follows.

(1) An AP with a strong signal in the network will often attract a large number of terminal accesses, causing its load to increase and the QoS of the network to worsen.

(2) The terminals associated with an AP are not necessarily working, and some of the associated terminals may be idle; therefore, the performance of an AP with a large number of access terminals is not necessarily worse than that of an AP with fewer access terminals.

To fully reflect the comprehensive performance of an AP, a parameter that reflects the true load of the AP should be added. The AP throughput is introduced in the load balancing handover strategy, and the throughput can reflect the load of the AP. The larger the throughput is, the larger the load, the worse the performance of the AP, and the lower the number of terminals that can be accessed.

#### 3.2.2 AP weight

Assume that the signal strength is  $S$ ,  $\bar{S}$  is the average of the signal strength  $S$ , we have

$$S = \alpha \bar{S}_t + (1 - \alpha) \bar{S}_{t-1}. \quad (3)$$

where  $\alpha$  is a smoothing factor.

The throughput of the AP will show some regularity. Analyzing the throughput of an AP over some time can more accurately determine the load status of the AP. In this paper, the AP historical traffic of the latest  $n$  unit time  $t_0$  of the AP is counted. The unit flow of the AP in the  $i$ th  $t_0$  time is

$$\theta(i) = \frac{F_i^{T_x} + F_i^{R_x}}{t_0}, \quad (4)$$

where  $F_i^{T_x}$  indicates the traffic sent by the AP in the  $i$ th unit at time  $t_0$  in the queue and  $F_i^{R_x}$  indicates the traffic received by the AP in the  $i$ th unit at time  $t_0$  in the queue.

From Eq. 4, we can obtain that the unit traffic (throughput)  $\tilde{\theta}$  of the AP is

$$\tilde{\theta} = \frac{\sum_{i=1}^n \theta(i)}{n}. \quad (5)$$

Because the number of terminals associated with an AP and the historical throughput of the AP are the parameters that reflect the load condition of the AP, we propose a load index that reflects the load of the AP by the two parameters of the throughput and the number of terminals. The load index of  $AP_k$  in the network is  $L_k$ , which is calculated as

$$L_k = \frac{\tilde{\theta}_k}{\theta_{\max}} + \frac{N_k}{N_{\max}}, \quad (6)$$

where  $\tilde{\theta}_k$  is the historical throughput of  $AP_k$  over some time.  $\theta_{\max}$  is the maximum throughput that  $AP_k$  can withstand,  $N_k$  is the number of terminals already associated with  $AP_k$ , and  $N_{\max}$  is the maximum number of terminals with which  $AP_k$  can associate.  $\theta_{\max}$  and  $N_{\max}$  can be obtained by testing.

The larger the load index of an AP is, the greater the load, which means that the worse the performance of the AP is, the lower the number of STAs that can be associated with it. On the contrary, the smaller the load index of the AP is, the smaller the load, and the better its performance. Therefore, the performance and weight of an AP should be negatively correlated with the load index.

In general, the greater the received signal strength is, the better the transmission quality of the wireless link, so the AP performance is positively correlated with the signal strength.

Therefore, it is concluded that the performance weight  $W$  of an AP is

$$W = \frac{S_k}{L_k}. \quad (7)$$

To sum up, we have

$$W = \frac{\alpha \bar{S}_t + (1 - \alpha) \bar{S}_{t-1}}{\frac{\tilde{\theta}_k}{\theta_{\max}} + \frac{N_k}{N_{\max}}}. \quad (8)$$

The main strategy of the load balancing handover method is that the controller monitors the AP through its global perspective and gives the corresponding weights to each AP according to the collected AP state information. The AP with a strong signal, a small number of associated STAs, and low historical throughput is given a large weight, and then the controller selects the AP with the best performance for the STA to serve it according to the weight size of each AP.

The load balancing handover method is mainly realized by the controller that controls the access flow of the STA and the handover process. When the STA accesses the network for the first time, the controller first authenticates it. Then, it queries the AP weight table, selects the AP with the largest weight corresponding to the STA to provide the service, and completes the access process of the STA. In view of the mobility of the STA and the change in the network service, the load of the AP in the network is also changing with time, and the controller updates the weight table of the AP in real-time based on the state information of the AP collected in the network. When it is found that the AP serving an STA is not the AP with the best performance corresponding to the STA, the AP with the best performance, that is, the highest weight will be found for the STA, and the handover will be performed. To ensure that the controller always selects the best-performing AP for an STA to provide services and ensure the maximization of network resource utilization, the AP weight algorithm of the load balancing handover is given, as shown in Algorithm 1.

---

#### Algorithm 1 AP weight algorithm

---

**Input:** AP weight table

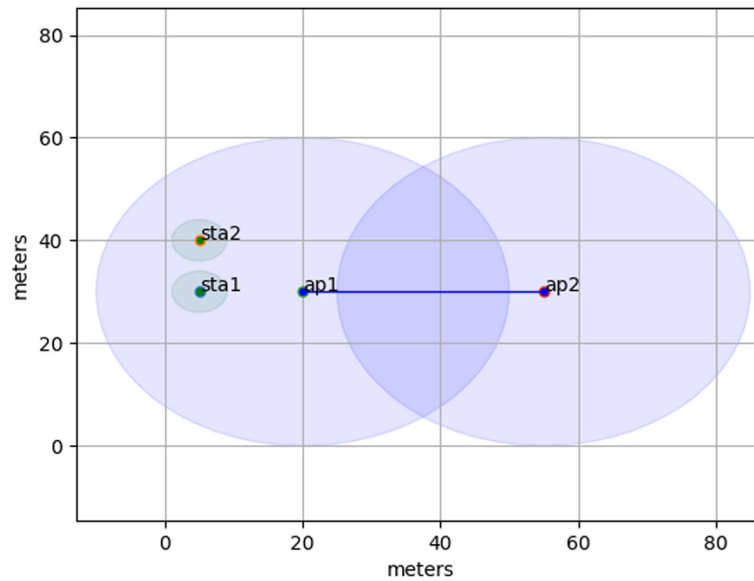
**Output:** Destination AP

```

1: for  $STA_i$  in the AP weight table do
2:   Destination AP = Current  $AP_x$ 
3:    $W_{imax} = W_{ix}$ 
4:   if the number of the APs in the APs list that can be
      connected by the STA is not 0 then
5:     for  $AP_j$  in the connectable APs list of the terminal
      do
6:       if  $W_{ij} > W_{imax}$  then
7:          $W_{imax} = W_{ij}$ 
8:         Destination AP =  $AP_j$ 
9:       end if
10:    end for
11:   end if
12: end for
13: if Destination AP ==  $AP_x$  then
14:    $STA_i$  does not need to hand over
15: else
16:    $STA_i$  needs to handover to destination AP
17: end if

```

---



**Fig. 3** Network topology with two APs and two terminals before moving. A topology consisting of one controller, two APs, and two terminals is constructed

#### 4 The experimental results and analysis

To verify whether the seamless handover method and load balancing handover method are effective and can meet the ideal performance requirements. We used an emulator for SDNs: Mininet-WiFi 2.2.0d1 with OpenFlow 1.3 [33]. The controller uses RYU v4.27. The performance of the handover method is compared with that of the KYA method in [20] concerning the aspects of transmission rate and delay.

To evaluate the performance of the proposed handover method, we select two indicators that are the most intuitive to reflect network performance: transmission rate and delay.

##### 4.1 Seamless handover

A topology consisting of one controller, two APs, and two terminals is constructed on the Mininet-WiFi platform, as shown in Fig. 3.

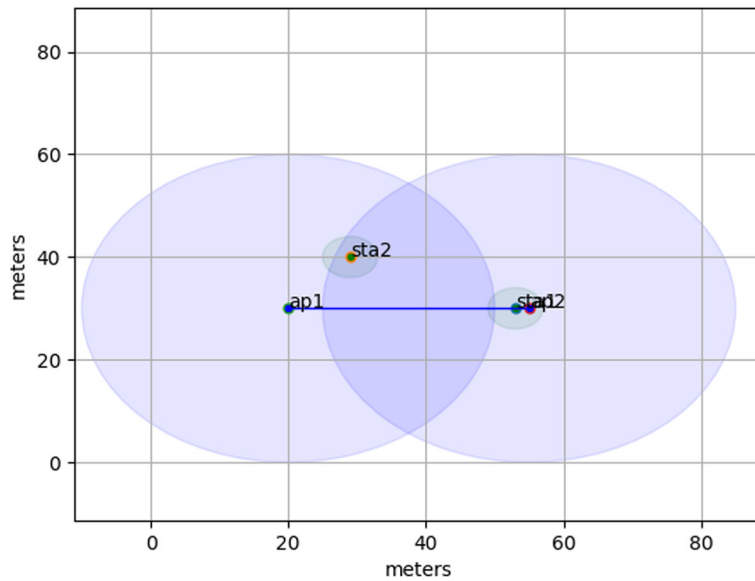
The moving speeds of  $2\text{ m/s}$  and  $1\text{ m/s}$  are respectively given to STA1 and STA2, and horizontal movement occurs to the right for 24 s. The positions after moving are shown in Fig. 4.

The selected signal strength value of the proposed method is used as the basis for terminal handover. STA1 is set as the client, STA2 is set as the server, the Iperf tool is used to record the transmission rate of STA1 during the moving, and then the Ping command is used to record the delays generated during the movement of STA1 and STA2. The experimental results of the two handover methods are then compared, as shown in Figs. 5 and 6.

It can be seen from Fig. 5 that the speed of STA1 is basically unchanged in the first 4 s because STA1 starts to move after recording for 4 s. At  $t = 12\text{ s}$ , the transmission rate of the two curves reaches the maximum value because STA2 is close to AP1 and STA1 has moved to the AP1 position at this time. Here, signal strength is the highest, so the speed is the fastest. When moving to  $t = 21\text{ s}$ , the transmission rates of both curves are reduced to their minimums because both of the handover methods are based on the signal strength. Also, at  $t = 21\text{ s}$ , STA1 has moved to the middle point of AP1 and AP2, and the signal strengths of AP1 and AP2 are the same. Therefore, both handover methods are terminal handovers here. By contrast, the traditional handover method reduces the transmission rate to 0 and maintains the disconnected state for about 3 s. The rate of the proposed seamless handover strategy is reduced, but the connection is not disconnected. When the moving reaches about 28 s, STA1 stops moving, so the transmission rate is unchanged.

From Fig. 6, the terminal starts to move from  $t = 4\text{ s}$ . When  $t = 12\text{ s}$ , STA1 moves to AP1, where the signal strength is the largest and the delay is the smallest. When  $t = 21\text{ s}$ , STA1 moves to the handover point. At this time, the terminal handover is performed, and the delay is increased; the traditional handover method is disconnected and maintained for about 3 s, and the proposed seamless handover method only has a delay of about 1 s and is not disconnected. When  $t = 30\text{ s}$ , the moving is stopped, and the delay remains unchanged.





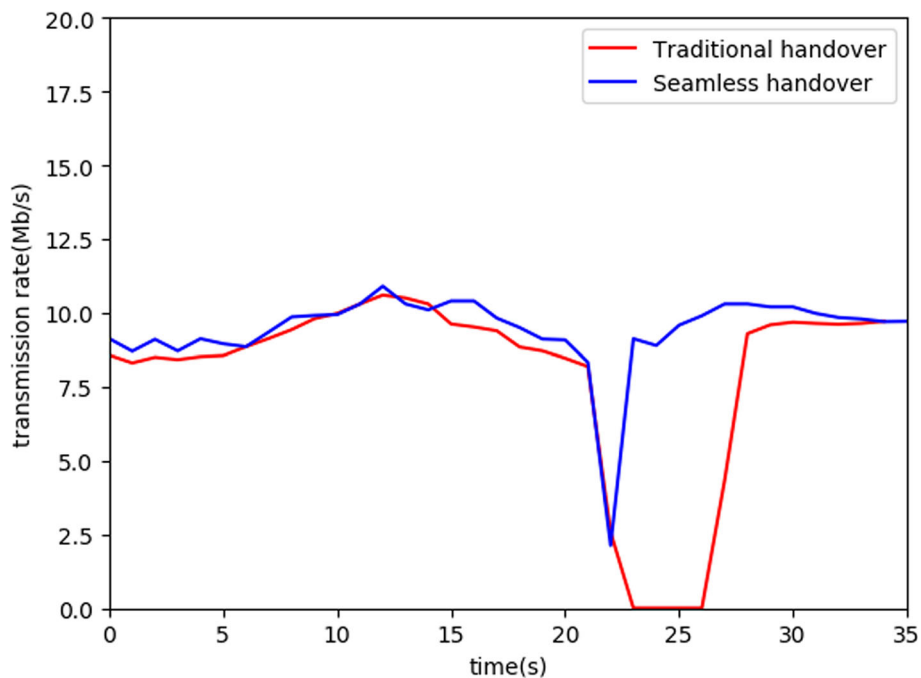
**Fig. 4** Network topology with two APs and two terminals after moving. The motion speeds of 2m/s and 1m/s are respectively given to STA1 and STA2, and horizontal movement occurs to the right for 24s

The seamless handover method maintains the QoS of the network when the terminal is switched, and the delay is about 12.4 ms, which is less than 20 ms mentioned in [31]. The user has no perception of the delay, and the method is superior to the traditional handover method.

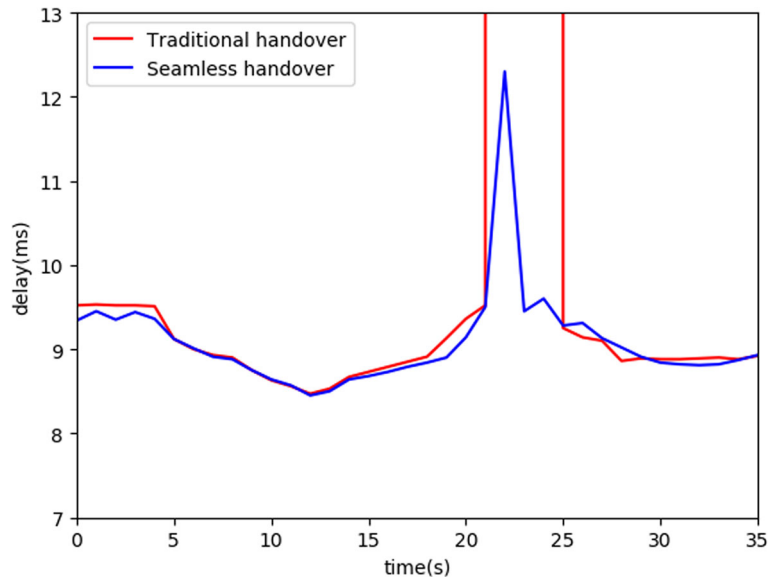
#### 4.2 Load balancing handover

The topology with a controller, two APs, and eight terminals is constructed on the Mininet-WiFi platform, as shown in Fig. 7.

The speed for STA1 is 4 m/s, and the moving time is 50 s when moving from the starting position (50, 150, 0) to the end (250, 150, 0), as shown in Fig. 8.



**Fig. 5** Transmission rate for traditional and seamless handovers. The experimental results of the two handover mechanisms are then compared

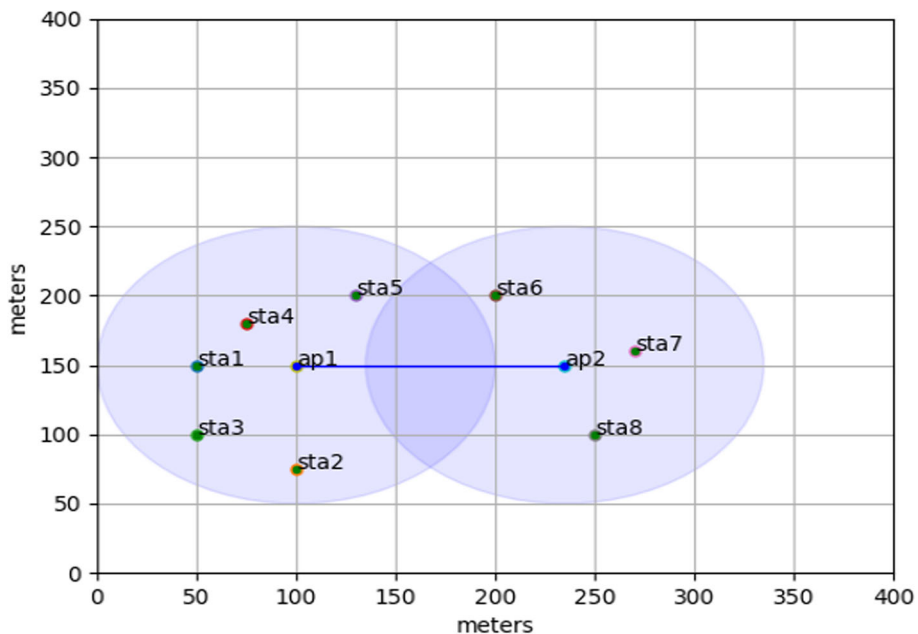


**Fig. 6** Delay for traditional and seamless handovers. The experimental results of the two handover mechanisms are then compared

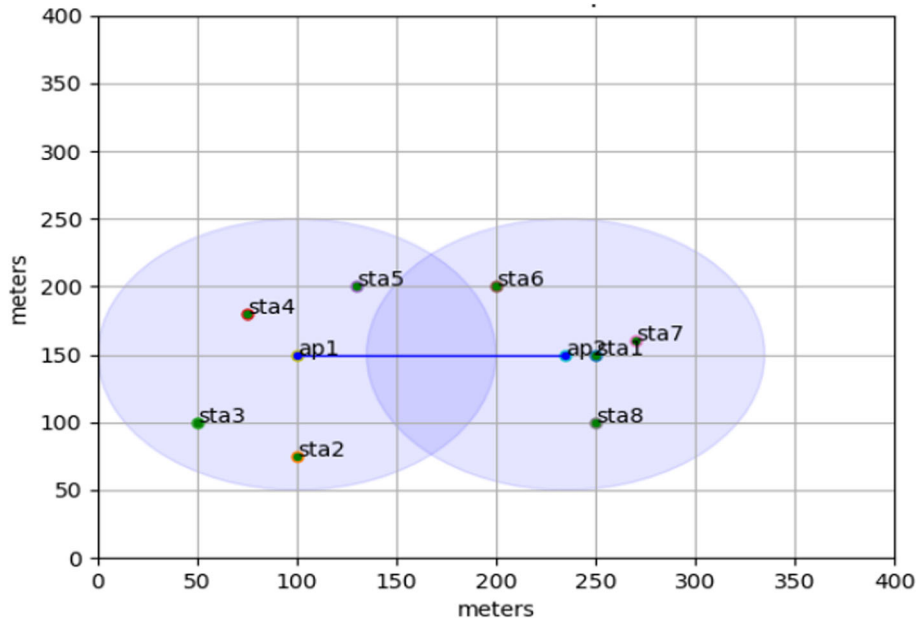
STA1 is set as the Iperf client, and STA4 is set as the Iperf server. Iperf is used to record the transmission rate of STA1 during moving, and then the Ping command is used to record the delays caused by STA1 and STA4 during motion. The results of the transmission rate and delay are shown in Figs. 9 and 10, respectively.

As shown in Fig. 9, the performance of the KYA method [20] is reduced to its lowest when  $t = 20\text{ s}$  because the

only handover basis for its terminal handover is the AP throughput and the throughput of AP1 is larger than that of AP2. Therefore, once STA1 enters the signal coverage of AP2, the handover is performed. Moreover, the load balancing handover method proposed in this paper reduces the transmission rate to its lowest at  $t = 28\text{ s}$  because the handover is based on the AP weight, which is determined by the three parameters of the signal strength, the number



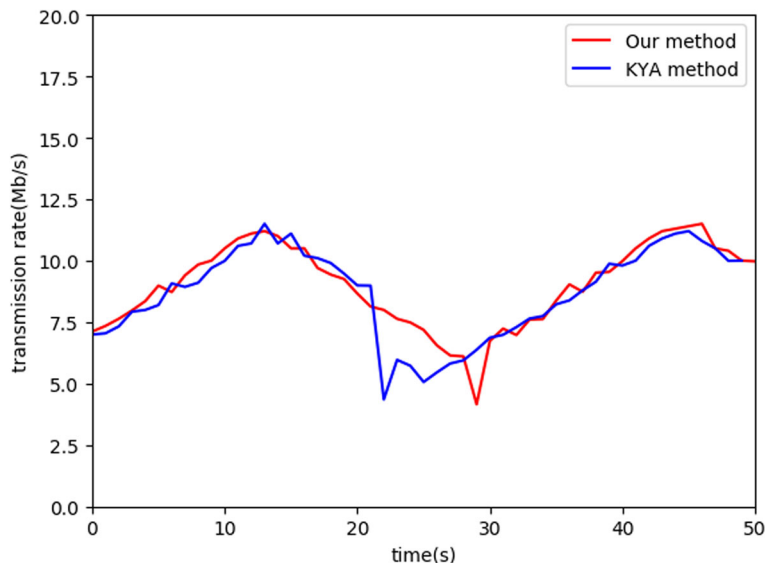
**Fig. 7** Network topology with two APs and eight terminals before moving. The topology with a controller, two APs, and eight terminals is constructed



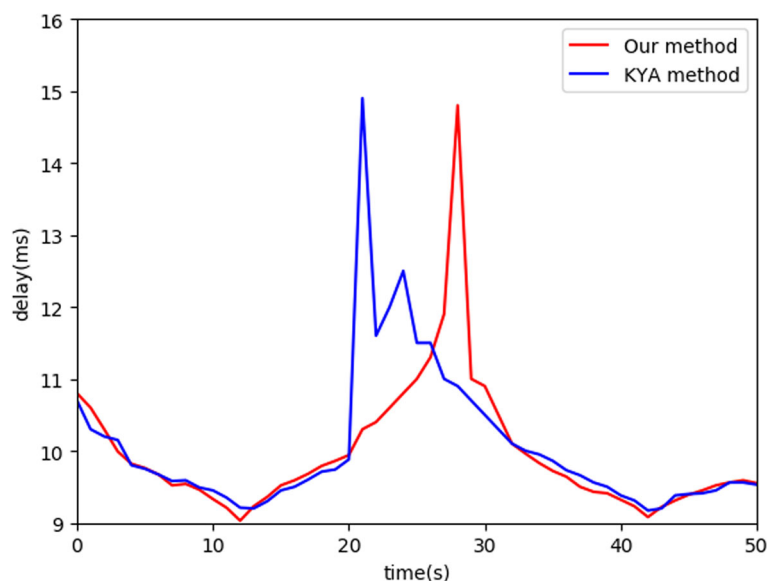
**Fig. 8** Network topology with two APs and eight terminals after moving. The speed for STA1 is 4m/s, and the motion time is 50s when moving from the starting position (50, 150, 0) to the end (250, 150, 0)

of connected terminals and the throughput; moreover, at  $t = 28 s$ , AP1 and AP2 have equal weights. The two curves are in agreement between 0 and 20 s and 30 s and 50 s. The transmission rate of STA1 will only have small fluctuations. During the 20 s to 30 s, due to the different handover methods, the transmission rate of STA1 will fluctuate greatly during handover. The transmission rate of STA1 under the handover method is higher than that of the KYA method.

As shown in Fig. 10, the time delay of the KYA method reaches the maximum when  $t = 20 s$  because STA1 is executing the handover at this time; by contrast, the delay of the handover method proposed in this paper reaches the maximum when  $t = 28 s$  because the proposed handover method is executing the terminal handover. The two curves are in agreement between 0 and 20 s and 30 s and 50 s. During 20 s to 30 s, due to the different handover methods, the transmission rate of STA1 will fluctuate



**Fig. 9** Transmission rate for our and KYA methods. The results of the transmission rate are shown



**Fig. 10** Delay for our and KYA methods. The results of the delay are shown

greatly during handing over. The delay of STA1 under the handover method is higher than that of the KYA method.

## 5 Conclusions

In this paper, we propose an architecture for SDN-based WLAN and also the mechanisms for (1) seamless handover and (2) load balancing by utilizing the SDN paradigm. For the seamless handover case, we propose to reduce the data propagation delay by the early release of the OpenFlow table. For the load balancing handover case, we propose the algorithm that weights AP based on 3 factors. The largest weight is given to AP with strong signals, a small number of associated stations, and low historical throughput. The experiment using Mininet-WiFi emulator is presented, which is overall sufficient. However, when a large number of terminals are moving in a large-scale network where APs are densely deployed, it is easy to cause the terminals to switch back and forth between the two APs, spending more system resources and causing a decline in network performance. How to prevent the terminal from switching back and forth between the two access points when the smart terminal and the AP are dense has become a problem that must be considered to improve the utilization of network resources, and it is also a work that needs further research.

### Abbreviations

LAN: Local area network; WLAN: Wireless local area network; AP: Access point; SDN: Software-defined networking; QoS: Quality of service; VAP: Virtual access point; SDWLAN: Software-defined wireless local area network; RA: Rate adaptation; QoE: Quality of experience; RAN: Radio access network; VoIP: Voice over internet protocol; SSID: Service set identifier; BSSID: Basic service set identifier

### Acknowledgements

The authors thank the person who provided meticulous and valuable suggestions for improving the paper.

### Authors' contributions

ZC is the main writer of this paper. ZC and LZ proposed the main idea. XD completed the experiments. ZC and XD analyzed the results. ZC, ZL, XD, and LZ discussed feasibility analysis. LZ is the corresponding author of this paper. All authors read and approved the final manuscript.

### Funding

This study was supported by the National Natural Science Foundation of China (No. 61572191).

### Availability of data and materials

Not applicable

### Competing interests

The authors declare that they have no competing interests.

Received: 29 November 2019 Accepted: 6 March 2020

Published online: 23 March 2020

### References

1. L. Zhang, K. Wang, D. Xuan, K. Yang, Optimal task allocation in near-far computing enhanced C-RAN for wireless big data processing. *IEEE Wirel. Commun.* **25**(1), 50–55 (2018). <https://doi.org/10.1109/MWC.2018.1700188>
2. H. W. Hsu, K. L. Huang, Y. C. Kao, S. C. Tsai, Y. B. Lin, Deploying WLAN service with OpenFlow technology. *Int. J. Netw. Manag.* **27**(3), e1970 (2017). <https://doi.org/10.1002/nem.1970>
3. Y. Hu, T. Peng, L. Zhang, Software-defined congestion control algorithm for IP networks. *Sci. Program.*, 3579540 (2017). <https://doi.org/10.1155/2017/3579540>
4. L. Zhang, Q. Deng, Y. Su, Y. Hu, A box-covering-based routing algorithm for large-scale SDNs. *IEEE Access.* **5**, 4048–4056 (2017). <https://doi.org/10.1109/access.2017.2682501>
5. D. Yin, L. Zhang, K. Yang, A DDoS attack detection and mitigation with software-defined Internet of Things framework. *IEEE Access.* **6**, 24694–24705 (2018). <https://doi.org/10.1109/ACCESS.2018.2831284>
6. OpenFlow Specifications. <https://www.opennetworking.org/software-defined-standards/specifications/>. Accessed 27 Nov 2019

7. K. Wang, K. Yang, H.-H. Chen, L. Zhang, Computation diversity in emerging networking paradigms. *IEEE Wirel. Commun.* **24**(1), 88–94 (2017). <https://doi.org/10.1109/MWC.2017.1600161WC>
8. S. Zhu, Z. Sun, Y. Lu, L. Zhang, Y. Wei, G. Min, Centralized QoS routing using network calculus for SDN-based streaming media networks. *IEEE Access.* **7**, 146566–146576 (2019). <https://doi.org/10.1109/ACCESS.2019.2943518>
9. X. Zhong, L. Zhang, Y. Wei, Dynamic load-balancing vertical control for large-scale software-defined Internet of Things. *IEEE Access.* **7**, 140769–140780 (2019). <https://doi.org/10.1109/ACCESS.2019.2943173>
10. P. Dong, K. Gao, J. Xie, W. Tang, N. Xiong, A. Vasilakos, Receiver-side TCP countermeasure in cellular networks. *Sensors.* **19**(12), ID 2791 (2019). <https://doi.org/10.3390/s19122791>
11. Z. Kuang, G. Liu, G. Li, X. Deng, Energy efficient resource allocation algorithm in energy harvesting-based D2D heterogeneous networks. *IEEE Internet Things J.* **6**(1), 557–567 (2019). <https://doi.org/10.1109/jiot.2018.2842738>
12. X. Deng, J. Luo, L. He, Q. Liu, X. Li, L. Cai, Cooperative channel allocation and scheduling in multi-interface wireless mesh networks. *Peer-to-Peer Netw. Appl.* **12**(1), 1–12 (2019). <https://doi.org/10.1007/s12083-017-0619-8>
13. H. Ma, H. Zhu, K. Li, W. Tang, Collaborative optimization of service composition for data-intensive applications in a hybrid cloud. *IEEE Trans. Parallel Distrib. Syst.* **30**(5), 1022–1035 (2019). <https://doi.org/10.1109/TPDS.2018.2879603>
14. J. Luo, X. Deng, H. Zhang, H. Qi, QoE-driven computation offloading for edge computing. *J. Syst. Archit.* **97**, 34–39 (2019). <https://doi.org/10.1016/j.sysarc.2019.01.019>
15. L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, T. Vazao, in *Paper presented at the 1st ACM International Workshop on Hot Topics in Software Defined Networks*. Towards programmable enterprise WLANs with Odin, (Helsinki, 2012)
16. J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, C. Peylo, CloudMAC: towards software defined WLANs. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **16**(4), 42–45 (2012). <https://doi.org/10.1145/2348543.2348592>
17. D. Zhao, M. Zhu, M. Xu, in *Paper presented at Fifth International Conference on Computing, Communication and Network, Communication and Networking Technologies*. SDWLAN: a flexible architecture of enterprise WLAN for client-unaware fast AP handoff, (Hefei, 2014)
18. D. Zhao, M. Zhu, M. Xu, in *Paper presented at Sixth International Conference on Wireless Communications and Signal Processing*. Supporting "One Big AP" illusion in enterprise WLAN: An SDN-based solution, (Hefei, 2014). <https://doi.org/10.1109/wcsp.2014.6992121>
19. A. K. Rangiseti, H. B. Baldaniya, B. P. Kumar, B. R. Tamma, in *Paper presented at IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*. Load-aware hand-offs in software defined wireless LANs, (Larnaca, 2014). <https://doi.org/10.1109/wimob.2014.6962245>
20. N. Kiran, C. Yin, Z. Akram, in *Paper presented at IEEE International Conference on Network Infrastructure and Digital Content*. AP load balance based handover in software defined WiFi systems, (Beijing, 2016). <https://doi.org/10.1109/icnidc.2016.7974526>
21. L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, J. Almodovar, Building an SDN enterprise WLAN based on virtual APs. *IEEE Commun. Lett.* **21**(2), 374–377 (2017). <https://doi.org/10.1109/LCOMM.2016.2623602>
22. S. M. M. Gilani, T. Hong, W. Jin, G. Zhao, H. M. Heang, C. Xu, Mobility management in IEEE 802.11 WLAN using SDN/NFV technologies. *EURASIP J. Wirel. Commun. Netw.* **67** (2017). <https://doi.org/10.1186/s13638-017-0856-9>
23. K. Nahida, C. Yin, Y. Hu, Z. A. Arain, C. Pan, I. Khan, Y. Zhang, G. M. S. Rahman, Handover based on AP load in software defined Wi-Fi systems. *J. Commun. Netw.* **19**(6), 596–604 (2017)
24. T. Zahid, X. Hei, W. Cheng, A. Ahmad, P. Maruf, On the tradeoff between performance and programmability for software defined WiFi networks. *Wirel. Commun. Mob. Comput.*, 1083575 (2018). <https://doi.org/10.1155/2018/1083575>
25. L. Li, G. Oikonomou, M. Beach, R. Nejabati, D. Simeonidou, in *Paper presented at IEEE International Conference on Communications*. An SDN agent-enabled rate adaptation framework for WLAN, (Shanghai, 2019). <https://doi.org/10.1109/icc.2019.8761424>
26. K. Kostal, R. Bencel, M. Ries, P. Truchly, I. Kotuliak, High performance SDN WLAN architecture. *Sensors.* **19**(8), 1880 (2019). <https://doi.org/10.3390/s19081880>
27. E. Coronado, S. N. Khan, R. Riggio, 5G-EmPOWER: a software-defined networking platform for 5G radio access networks. *IEEE Trans. Netw. Serv. Manag.* **16**(2), 715–728 (2019). <https://doi.org/10.1109/TNSM.2019.2908675>
28. E. Coronado, E. T. Garriga, J. Villalon, A. Garrido, L. Goratti, R. Riggio, SDN@Play: Software-defined multicasting in enterprise WLANs. *IEEE Commun. Mag.* **57**(7), 85–91 (2019). <https://doi.org/10.1109/MCOM.2019.1800502>
29. A. Sen, K. M. Sivalingam, Testbed evaluation of a seamless handover mechanism for an SDN-based enterprise WLAN. *Sadhana Acad. Proc. Eng. Sci.* **44**(12), 243 (2019). <https://doi.org/10.1007/s12046-019-1229-3>
30. B. Dezfouli, V. Esmaealzadeh, J. Sheth, M. Radi, A review of software-defined WLANs: architectures and central control mechanisms. *IEEE Commun. Surv. Tutor.* **21**(1), 431–463 (2019). <https://doi.org/10.1109/COMST.2018.2868692>
31. I. Purushothaman, S. Roy, FastScan: a handoff scheme for voice over IEEE 802.11 WLANs. *Wirel. Netw.* **16**(7), 2049–2063 (2010). <https://doi.org/10.1007/s11276-010-0243-5>
32. A. Mishra, M. Shin, W. Arbaugh, An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM SIGCOMM Comput. Commun. Rev.* **33**(2), 93–102 (2003). <https://doi.org/10.1145/956981.956990>
33. R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, C. E. Rothenberg, in *Paper presented at the 2015 11th International Conference on Network and Service Management*. Mininet-WiFi: emulating software-defined wireless networks, (Barcelona, 2015). <https://doi.org/10.1109/cnsm.2015.7367387>

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)