Adaptive Load-balancing Scheme Through Wireless SDN-based Association Control

Chia-Ying Lin, Wan-Ping Tsai, Meng-Hsun Tsai and Yun-Zhan Cai
Department of Computer Science and Information Engineering, National Cheng Kung University,
Tainan, Taiwan
Email: {a711186, wanping}@imslab.csie.ncku.edu.tw, tsaimh@csie.ncku.edu.tw, F74039017@mail.ncku.edu.tw

***Abstract*—As the popularity of mobile applications, Wi-Fi has become one of the major access methods for many users. When a large amount of users move in public places together, such as classrooms or meeting rooms, severe load imbalance of Wi-Fi Access Points (APs) and unfair bandwidth allocation are likely to occur consequently.**

Software Defined Networking (SDN) is a new networking paradigm which allows the network administrators to write programs for controlling the behaviors of network devices. In this paper, we propose an adaptive load balancing scheme through association control in wireless software defined network. The proposed scheme consists of an event detection mechanism and an adaptive load balancing algorithm on controller. In our algorithm, controller can derive an optimal association solution based on the traffic load and number of users on each AP. To observe the effects of population distribution and user mobility, we propose a simulation model to investigate the performance in terms of average AP load, user bandwidth and user throughput. Our simulation results show that our scheme has better performance than other three methods and performs better in imbalanced environment.

***Index Terms*—association control; IEEE 802.11 network; load balancing; SDN**

I. Introduction

In recent years, the number of intelligent mobile devices has rapidly increased [1]. The penetration rate of wireless Internet has already increased to 91.5% in 2014, which was 14% eight years ago [2]. To meet the growth of user traffic load in Wireless Local Area Network (WLAN), more and more Wi-Fi Access Points (APs) are constructed around us. There are currently about 47 million APs worldwide, and the number of APs is expected to be seven times in 2018 [3].

In some occasions (conference, classroom, etc.), hundreds of wireless devices attempt to associate with one AP in a short time. AP overloading infers low throughput problem for users and load imbalance problem in WLANs. In these occasions, load balancing becomes a critical issue. However, there are some difficulties in current wireless architecture. First, due to lack of flexibility of the legacy management mechanisms, it is difficult to add new features (e.g., new association control mechanism) into currently deployed APs. Second, many personnel costs are required to configure the forwarding rules of all the APs and to avoid conflicts among these rules. Last, the signal of an AP is usually interfered by the other nearby APs owing to the lack of cooperation among APs. All we need is a more resilient, scalable network architecture with centralized control.

Software Defined Network (SDN) is a new networking paradigm that provides a more flexible and programmable architecture. SDN separates the control modules from the infrastructure layer to the centralized controllers. In SDN, OpenFlow [4] is proposed to allow the controllers to control behaviors of switches. OpenFlow realizes that the switches from different vendors can easily cooperate with each other, and thus provides more choices in selecting infrastructure components (e.g., APs, switches). Besides, the SDN allows the network administrators to write programs on the controllers. Hence, we use the programmable architecture to apply our association control mechanism into the wireless network resiliently and easily.

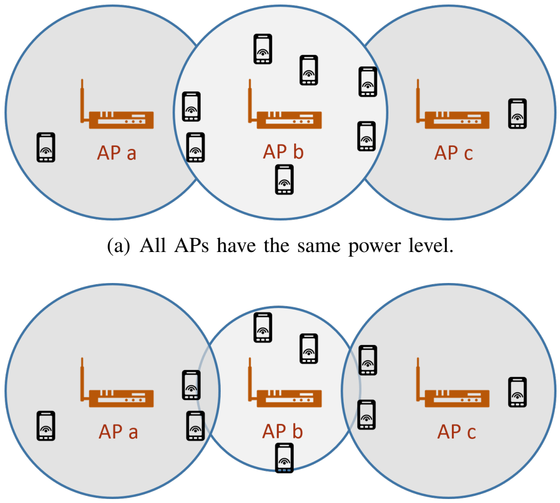
Based on SDN, we propose an adaptive load balancing scheme for Wi-Fi APs through association control. By the concept of SDN, we use the controller to collect the load of APs and user connection situations. Based on the global vision of the controller, we design an algorithm to do association control as follows. When the controller detects that the load of APs is imbalanced, the controller adaptively configures the beacon power of relevant APs. In this case, some users may change association from APs with weak power to APs with strong power. Based on the power adjustment, we propose adaptive load balancing scheme to improve the performance in WLANs.

II. Related Works

In this section, we describe existing association control schemes and load balancing schemes in literature.

The strongest signal first (SSF) method is the traditional association mechanism defined in IEEE 802.11 standards [5]. When a device detects more than one AP, the device selects the AP with the largest received signal strength indicator (RSSI) value for association. This method is also widely adopted in handoff schemes [6], [7]. The main problem of these schemes is the load imbalance of APs. When too many users associate to an AP with strong signal at the same time, these users may incur worse performance from the overloaded AP.

Least load first (LLF) is the most straightforward load balancing scheme, where users select the AP with least number of users [8]. In [9], Balachandran et al. proposed that users associate to the AP which can provide sufficient bandwidth



(a) All APs have the same power level.

(b) AP b transmits with the lowest power level.

Fig. 1. Using Cell Breathing Method for Imbalanced Situation.

based on the users' bandwidth requirement. In [10], Bejerano et al. further consider the fairness among all users in association control.

The cell breathing concept has been studied mostly in code devision multiple access (CDMA) cellular network. In [11] and [12], cell breathing was applied to IEEE 802.11 WLANs. Figure 1 illustrates an example of cell breathing method. First, the three APs in Figure 1(a) are associated with 1, 7, and 1 users, respectively. Through adjusting the power levels of APs, some users of AP b shift to adjacent APs, and then the APs are all associated with three users in Figure 1(b). In the cell breathing method, an access controller can obtain the load of APs, but it has no information of users' positions. In this case, the controller can only adjust the beacon power of the AP with the highest load step by step until the load of APs is balanced. To achieve load balancing, in [12], Bejerano et al. proposed an algorithm to determine globally optimal inter-AP fairness. However, both [11] and [12] adjust the AP power level based on limited information (e.g., number of users).

In this paper, we propose an adaptive load balancing scheme which can further improve performance of cell breathing by considering the flexibility of SDN. Through SDN, we can use the global vision to collect the load of AP and users' RSSIs accurately. Furthermore, the users' positions can be derived by users' RSSI according to [13].

III. Proposed Scheme

A. System Model

Figure 2 shows our system architecture. The notations used in our scheme are shown in Table I. We consider an IEEE 802.11 WLAN with a set of APs, which is denoted by A, and $|A|$ indicates the number of APs. All APs are attached to a wired infrastructure and an OpenFlow controller.

In our scheme, we focus on the transmission power of AP beacon messages since beacon messages are used for AP association. We denote transmission power of AP a as P_a. According to the configuration and ability, each AP has its maximum and minimum transmission power, denoted as P_{max} and P_{min} respectively. Each AP also provides several transmission power levels from 0 to N, denoted by $P^*_{a(lv.n)}$. We denote $P^*_{a(lv.0)} = P^*_{a,min}$ as the minimum power level of AP, and similarly we denote $P^*_{a(lv.N)} = P^*_{a,max}$ as the maximum power level. For normal commercial AP products, the transmission power level configuration follow that

TABLE I
NOTATIONS

Symbol	Semantics
A	the set of all access points (APs)
U	the set of all users
U_a	the set of users associated with AP a, $a \in \{1, 2, ..., \|A\|\}$
L_a	the load of AP a
$L_{a,u}$	the load contribution of user u to AP a
C	the set of overcrowding APs
P_a	transmission power of AP a
$P_{a,max}$	the maximum transmission power of AP a
$P_{a,min}$	the minimun transmission power of AP a
P^*_a	the transmission power level of AP a
N	the number of transmission power level
$P^*_{a(lv.n)}$	the transmission power level of AP a at level n, $n \in [0, N]$
S_v, S_n, S_o	three different load level set of APs, A = S_v + S_n + S_o
b_{vn}, b_{no}	boundaries between three load state
R_u	the RSSI value of user u
S^k	the network states during a load-balancing process

$$P^*_a = log_\gamma(P_a) \quad ,\gamma = \sqrt[N]{\frac{P_{max}}{P_{min}}} \tag{1}$$

Then $P^*_{a(lv.n)}$ is a geometric series which can be expressed as

$$P^*_{a(lv.k)} = P^*_{a(lv.k-1)} * \gamma \tag{2}$$

$$P^*_{a(lv.k)} = P^*_{a,min} * \gamma^k, k \in [0, N]. \tag{3}$$

In our scheme we have some assumptions. First, we assume that the interference between adjacent cells can be ignored. Second, we assume that the AP deployment ensures complete overlaps between the ranges of all APs. In other words, if all APs are configured to the minimum transmission power P_{min}, every user in network coverage area can be still covered.

When a mobile device joins a WLAN, it listens every channel for all beacon messages from APs. Then, it associates with an AP which has the strongest RSSI, which is determined by the beacon transmission power and the distance between AP and user. In our system, we use all APs to collect the RSSIs from users (Figure 2). Each AP reports the RSSIs to the controller by using OpenFlow protocol. Once in a while, each AP collects its load L_a and the load contribution of its

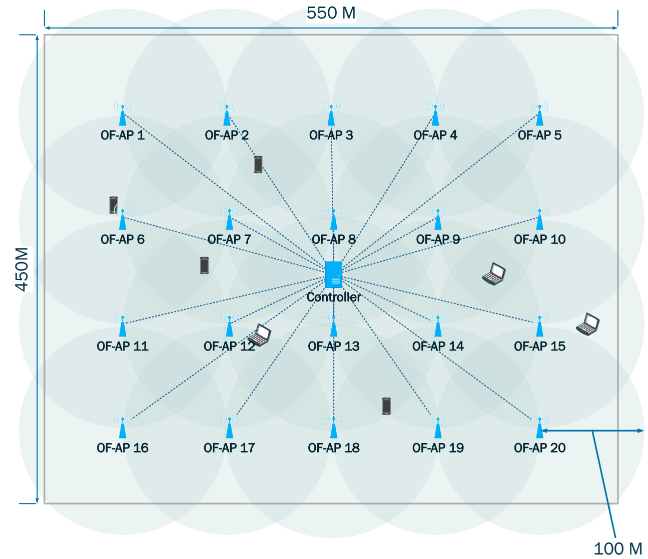


Fig. 2. System Architecture

users $L_{a,u}$ to controller. Based on the information above, the controller knows the association relationships between users and APs. In the same way, the load of APs is also known by controller. In our system, the controller has a global view of the whole network.

We use U to denote the set of all users in the network coverage area and $|U|$ is total number of users in U. Each user associates with at most one AP at any time, and we denote U_a as the set of users associated with AP a. In normal cases, if a large amount of user association requests are sent to an AP in a short time, the AP become overloaded seriously. To stress the coordination ability of APs, we consider the group arrival cases. We assume that all users follow a group mobility model in our system. The users usually move together from one place to another in several groups.

Owing to the global vision of controller, we design an adaptive load balancing model using association allocation. Our model can coordinate the load of all APs and take the users' load contributions into consideration. For example, the busy users, who need whole allocated bandwidth to upload or download data frequently, are considered. The controller records the load contribution of user u to AP a, $L_{a,u}$ by considering the load report from APs. Each AP can provide different bandwidths to its associated users in our case.

B. Arrival Event Detection

In this subsection, we present the initial stage of adaptive load balancing. We design the procedures for APs to report its user association events and load in real time. Figure 3 shows the flow chart of AP reporting mechanism, and the steps are described below.

Step 1. Each AP records the user RSSI, user association list and its load periodically. All APs receive RSSIs in every authentication frames from their users.

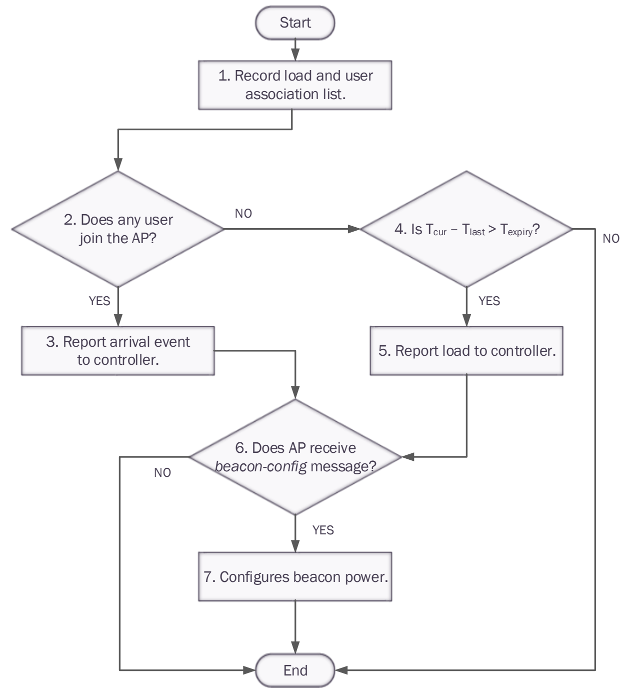


Fig. 3. Flow Chart for AP Reporting Mechanism

Step 2. If there is a new user associating with the AP, go to **Step 3**. Otherwise, go to **Step 4**.

Step 3. The AP sends the user RSSI to the controller. Then go to **Step 6**.

Step 4 and 5. If there is no new arrival in **Step 2**, the AP reports its load to controller periodically.

Step 6. After sending report messages to controller, the AP would receive a Beacon-Config message from the controller.

Step 7. The AP configures its beacon transmission power.

The flow chart of the controller management mechanism is shown in Figure 4, and following is the descriptions of steps.

Step 1. The controller receives arrival events and load messages from APs.

Step 2. The number of new arrival events is added into the arrival counting table. Then the controller calculates event increase at the latest time interval.

Step 3. The controller compares the number of current arrival events with the past. When an AP exceeds the threshold of arrival events, it has a high risk of overcrowding. At this time, the controller gives the AP a predicted load value and then go to **Step 6**.

Step 4. The controller classifies all APs into three load levels: vacant, normal and overcrowding. According to the levels, the APs are added into (S_v, S_n and S_o) respectively. The b_{vn} and b_{no} denote the boundaries between these levels.

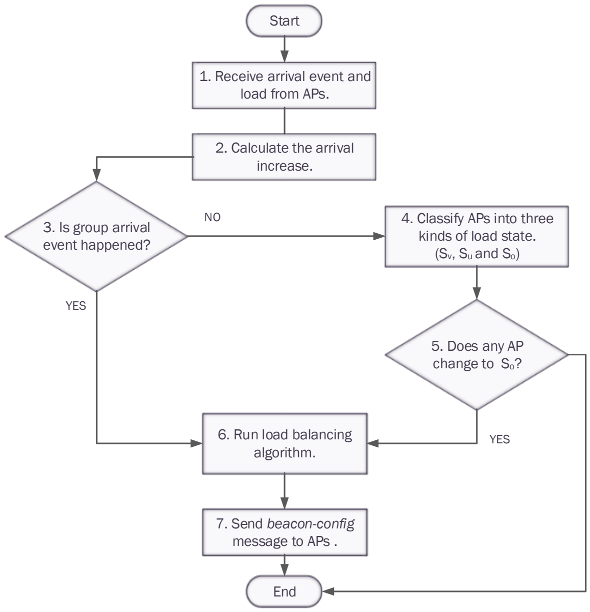


Fig. 4. Flow Chart for Controller Management Mechanism

Algorithm 1: Adaptive_Load_Balancing (A, U)

```
Input: A means the set of all APs. U means the set of all users.
Initialize beacon power level P*_a = MAX_POWER_LEVEL, a ∈ {1,2, ..., |A|};
Initialize sum of APs utilization as state S^index, index = 0;
Initialize overcrowding set C = NULL;
Initialize stack Adjustment_Log;
Let L_a denotes the load of AP a;
Let End_Flag = FALSE;
Let max_load = 0, max_load_index = 0;
{U_a, L_{a,u}, P_a} ← Association_Relationship_Update(A, U)
while End_Flag = FALSE do
    Find L_a = max AP load in A , a ∈ {1,2, ..., |A|};
    Add AP a into set C;
    P*_a ← P*_a − 1;
    Adjustment_Log.push(a);
    S^{index+1} ← Association_Relationship_Update(A, U);
    if (S^{index+1} > max_load) then
        max_load_index = index + 1;
    end if
    if (Exist x ∈ C and P*_x = 0) or ( C = A ) then
        End_Flag = TRUE;
        Optimization_Load(Adjustment_Log, max_load_index);
    end if
end while
```

Fig. 5. Adaptive Load-Balancing Algorithm

$$L_{l,a} = \left\{ \begin{array}{c} v, L_a \le b_{vn} \\ n, L_a \le b_{no} \\ o, L_a < b_o \end{array} \right\} \quad (4)$$

$$S_v = a|L_{l,a} = v \quad (5)$$

$$S_n = a|L_{l,a} = n \quad (6)$$

$$S_o = a|L_{l,a} = o \quad (7)$$

Step 5. This procedure ends when there is no AP changing to overcrowding state S_o.

Step 6. The controller calculates the adjustments of APs according to the load balancing algorithm which is described in next subsection.

Step 7. The controller send a beacon-config messages to each AP.

C. Adaptive Load Balancing

In this subsection, we present the adaptive load balancing mechanism. Recall what we described in subsection III-A and III-B, the association relationship, RSSIs of users and load of APs are all collected by the controller. Based on the information above, the controller has a global vision of whole wireless network. We combine SDN and Cell-Breathing method into an Adaptive Load Balancing mechanism.

Figure 5 shows the pseudo code of our mechanism for controller. In this algorithm, the beacon power level is first initialized to the maximal power level. The controller collects and calculates the summations of all AP utilizations, and the network state is initialized as S^0. First, the algorithm iteratively finds the most overcrowding AP. Then, the most overcrowding AP is added into overcrowding set C and sets its power one level down. In the meanwhile, the adjustment is recorded in a stack structure. Every time the most overcrowding AP sets down beacon power level, the algorithm updates the AP states by simulating the association relationship between APs and users. Each state is compared with previous states to find the optimal state. Until the iteration ends, the algorithm sets all AP beacon power level according to the optimal state.

This algorithm terminates in two conditions. The first condition is that the overcrowding set C is equal to AP set A. When these two sets are equal, every AP is adjusted. At this time, the results are the same, even though the iteration continues. The second condition happens when any AP beacon power level is zero. If the iteration continues, some APs can not trun down their beacon power anymore, thus they become overcrowding again.

Figure 6 illustrates the message flow of association control between users, APs and controller. We assume that user is in the coverage of AP 1 and 2, and AP 1 is closer to user than AP 2. We assume that the user has passwords of AP 1 and AP 2. Both these two APs connect to controller via secure channel.

Step 1. AP 1 and AP 2 send Beacon Message to the user periodly.

Step 2. AP 1 and AP 2 report their load (utilization) and association events to controller. In OpenFlow protocol, association event can be transmit in Packet_In message and the load of AP can be transmitted in Port_Status message.

Step 3. Controller periodly checks all of the AP load and

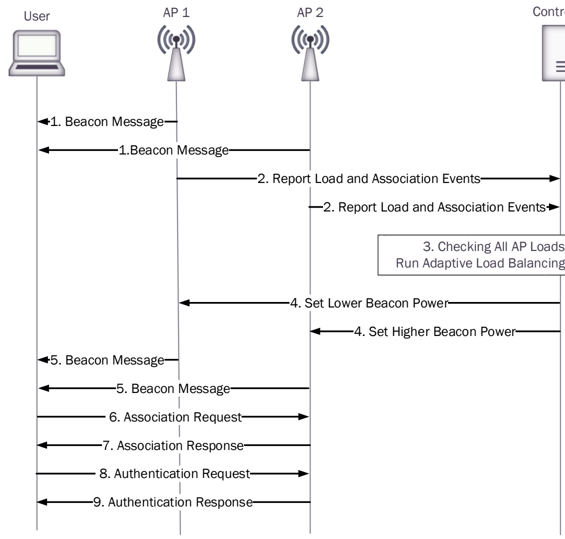


Fig. 6. Message Flow of Association Control

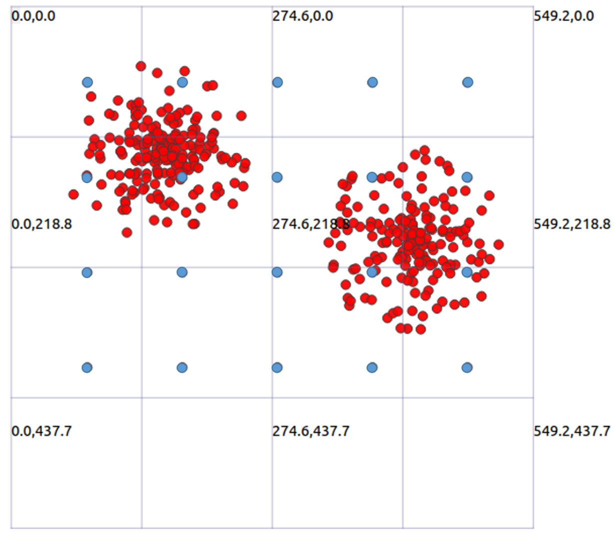


Fig. 7. A WLAN Scenario with 20 APs in $550 * 450m^2$ for NS3 Simulations (The 400 users follow the Reference Position Group Mobility model.)

integrates all association event reports of APs. Then controller inputs these information to Adaptive Load Balancing program.

Step 4. Based on the result of **Step 3**, controller sends Beacon-Config messages (i.e. SetConfig message in OpenFlow) to APs. In this case, controller decides to reduce load of AP 1, thus controller sets the AP 1 beacon power lower than AP 2 beacon power.

Step 5. The same as **Step 1**, AP 1 and AP 2 send beacon message to the user.

Step 7-9. The user sends association request to AP 2, which has the highest RSSI. After receiving an association response from AP 2, user sends an authentication request to AP 2. When AP 2 sends back authentication response, the connection between user and AP 2 is established.

IV. Performance Evaluation

In this section, we verify whether our scheme balances the load more significantly than other existing methods including Strongest-Signal-First (SSF), Least-Load-First (LLF) and Cell-Breathing.

A. Simulation Model

The details of our simulation setup is shown in Table II. Most parameters are configured according to existing Cell-Breathing method in literature [12]. We use NS3 as our simulation tool, and we set simulation time to 10000s. We totally set 20 APs, which are all support IEEE 802.11g standard. Each AP is equipped with 54 megabits per second backhaul link. These APs are located in $550 * 450m^2$ area and they are arranged into four lines for five APs per line. The distance between two adjacent APs is set to 100 meters, and those APs are 75 meters far from the simulated area border. In order to ensure that all users are in the coverage of APs, we set the minimum transmission distance is 75 meters. To determine the transmission bit rate between users and APs, we list the relationship between SNR and bit rate in Table III.

In order to build the mobility of our scenario, we use the BonnMotion tool to generate user traces. We use the Reference Position Group Mobility model (RPGM) to generate three cases of user number, which are 100, 250 and 400. Figure 7 shows the scenario with 400 users in our simulated area. These users are divided into two groups and follow the group mobility. Each group brings heavy connection demand when the group moves into coverage of an AP.

TABLE II
Simulation Setup

Parameters	Assumption
Simulated Time	10000s
Simulated Area	550 * 450 m^2
MAC Protocol	IEEE 802.11b
Transmission Range	[75, 150] m
Adjacent AP distance	100 m
AP Number	20
User Number	100, 250,400
Mobility Model	Reference Position Group,Mobility (RPGM)
Moving Speed	[0.5, 1.5] m/s

TABLE III
Traffic Bit Rate

Bit Rate (Mbps)	SNR(dB)	Distance (m)
11	$\geqq 9$	50
5.5	$\geqq 5$	80
2	$\geqq 3$	120
1	$\geqq 1$	150

B. The Performance of APs

To measure the performance of APs, we define per second load of AP , L_a, to repesent the sum of associated user throughput, $L_{a,u}$). $L_{a,u}$ is the product of user bandwidth and user data rate. We assume the user bandwidth is the time interval that each AP fairly allocates the transmission time slot to its users. Users are able to transmit their data in the time slot. The user data rate can be derived from user SNR or distance with AP in Table III.

$$L_a = \sum_{u \in U_a} L_{a,u} \tag{8}$$

$$L_{a,u} = b_{a,u} * r_{a,u} \tag{9}$$

$$b_{a,u} = \frac{1}{|U_a|}, |U_a| = number\ of\ users \tag{10}$$

Figure 8, 9 and 10 show the average load of all APs with 100, 250 and 400 users respectively. The X-axis represents the index of APs, and the APs are sorted by their average load in increasing order. In the case of 100 users, we generate two groups which have average 50 users. The users are less than 50 meters from their group center. In Figure 8, we notice that the curves of our scheme, LLF and Cell-Breathing are more gently than the curve of SSF. In SSF, some APs are heavily congested and some APs are vacant. Although the curve of Cell-Breathing is the most gently one, the sum of its average load is the least (in Table IV). As Table IV shows, the load of our scheme performs 11% better than SSF and 347% times better than Cell-Breathing. The difference between our scheme and Cell-Breathing is that we consider both the imbalance load distribution and the optimal state of all APs. Limited to the knowledge of network situation, Cell-Breathing is not able to adjust AP beacon power levels to the optimal result.

Figure 9 and 10 show the curves of our scheme, LLF and Cell-Breathing are more gently than the curve of SSF. In the case of 250 users, we generate five groups which have average 50 users and all users are less than 50 meters near from their group center. In the case of 400 users, we generate two groups which have average 200 users. The users of each group move around the center in 100 meters, and they bring a large amount of connection request to nearby APs. Figure 10 illustrates that the gap between SSF and Cell-Breathing is smaller than the gap in Figure 9, and our scheme outperforms these two method.

We calculate the average load of above three cases and compare the performance in Table IV. As the population distribution becomes more crowded and imbalance (i.e. the number of users increases), the gap between our scheme and SSF is bigger. The load of our scheme performs 11 $\sim$ 28% better than SSF. Though the performance of Cell-Breathing becomes better in crowded and imbalance case, our scheme still performs 26% better than Cell-Breathing. Figure 11 shows the total AP load during simulation time.

Figure 12 illustrates the average user number of all APs in four methods. The X-axis represents the index of APs,

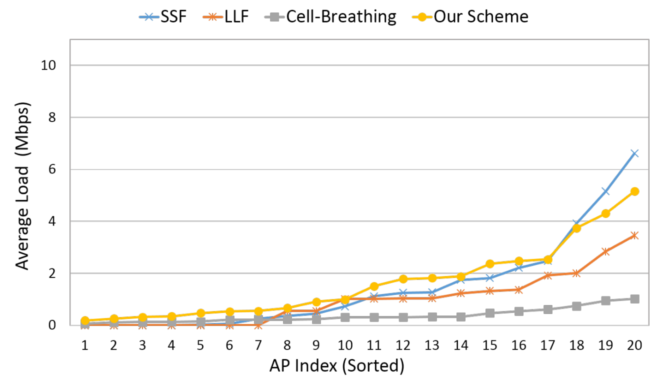


Fig. 8. The Average Load of all APs (100 Users)

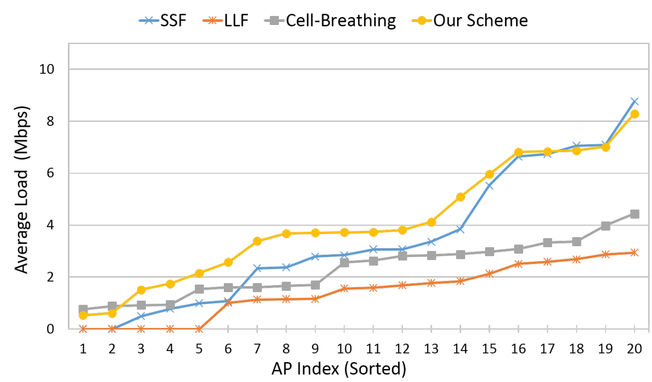


Fig. 9. The Average Load of all APs (250 Users)

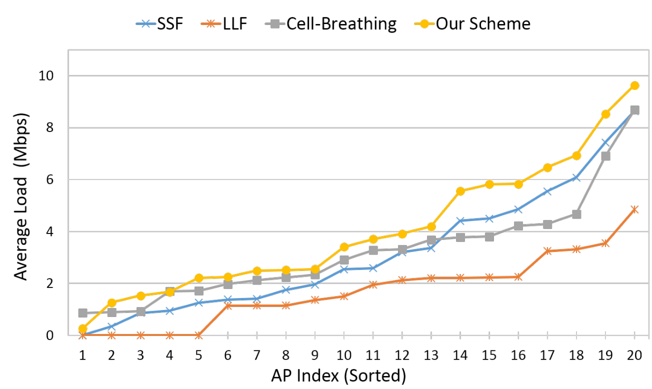


Fig. 10. The Average Load of all APs (400 Users)

TABLE IV
SUMMARY OF THE AP LOAD

# of Users	100	250	400
SSF	29.45	68.78	63.08
LLF	19.32	28.58	34.24
Cell Breathing	7.35	46.48	64.32
Our Scheme	32.77	82.11	80.80
Improvement of Our Scheme Over SSF	11%	19%	28%
Improvement of Our Scheme Over LLF	70%	187%	136%
Improvement of Our Scheme Over Cell Breathing	346%	77%	26%

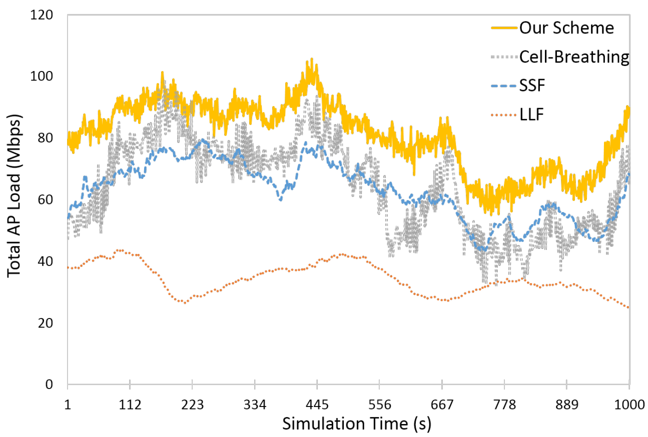


Fig. 11. The Total AP Load (number of Users = 400)

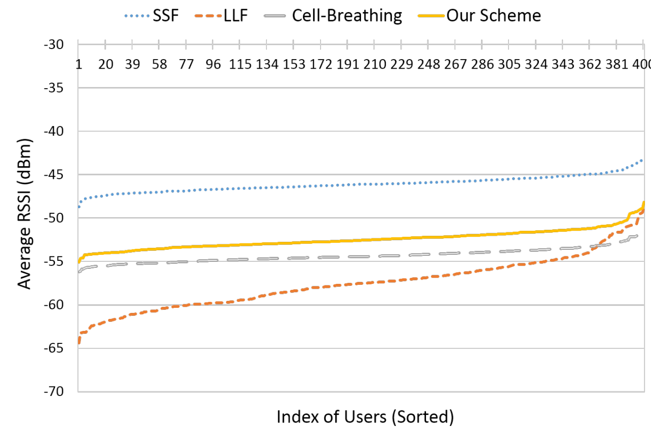


Fig. 13. The Average RSSI of Users (number of Users = 400)

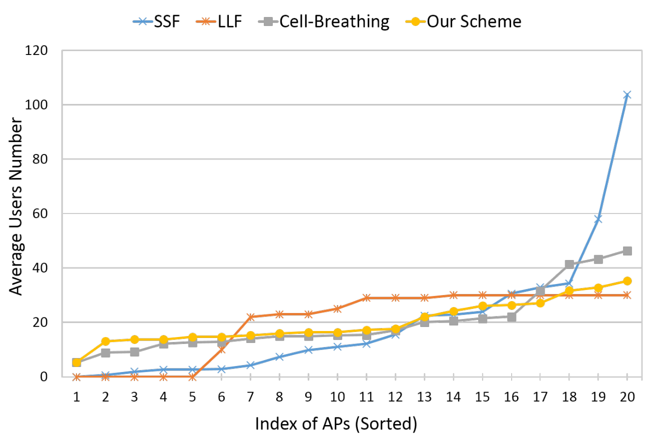


Fig. 12. The Average User Number of APs (number of Users = 400)

and the APs are sorted by their average user number in increasing order. Our scheme and Cell-Breathing both balance the user number of all APs. SSF is the most imbalance method, because all users prefer to connect with the AP which has the strongest signal (i.e. the nearest AP). If there are too much users connecting to an AP at the same time, these users have poor bandwidth to transmit data.

C. Impact of methods to Users

In this subsection, we investigate the average user RSSI and average throughput with four methods. To measure the performance of user RSSI, we use the following formula and set Antenna gain 5 dBi.

$$RSSI = Signal - Pathloss + Antenna_Gain; \qquad (11)$$

In our simulation, we assume that the interference between adjacent APs can be ignored. We borrow the FSPL (Free Space Path Loss) equation of TP-Link to derive the path loss value.

$$Pathloss = 20log10(d) + 20log10(f) + K; \qquad (12)$$

K is a constant number that depends on the unit used of distance d and frequency f, and we set K to 32.44 in our simulation.

Figure 13 illustrates the average signal strength which user received from AP. The X-axis represents the index of users, and the users are sorted by their average RSSI in increasing order. Each average RSSI value is obtained from 10000 seconds. As a result, the users receive the highest average RSSI in SSF and the worst average RSSI in LLF. Since the users in SSF choose the highest RSSI AP to associate, and in LLF, the users intend to associate the lightest load AP. In Figure 13, the users in our scheme receive higher RSSI than the users in Cell-Breathing.

Figure 14 shows the throughput of users. In our simulation, we assume that all users are greedy to use all the resource allocated from APs. However, the user usage is limited by not only users data rate but users bandwidth. In Figure 14, the users in our scheme have higher throughput than other methods. Due to our scheme shifts users from heavy load AP to light load AP, users can have better bandwidth and higher throughput.

We calculate the average throughput of three cases and compare the performance in Table V. The load of our scheme performs 16 $\sim$ 26% better than SSF and performs 23 $\sim$ 377% better than Cell-Breathing. The performance of Cell-Breathing is very low when the population distribution is not too crowded. Overall, our scheme performs better than the other three methods in imbalanced and overcrowding environment.

Figure 15 shows the impact of the power levels of our scheme. In the case of 400 users, these results show that the cases of more power levels perform better. Though the cases of more power levels take more time to compute the optimal solution, the computing time is much shorter than the people moving speed. In the figure 15, the curve of 30 levels is almost overlapped with the curve of 50 levels, so we think that the power level ranging between 30 and 50 is suitable.

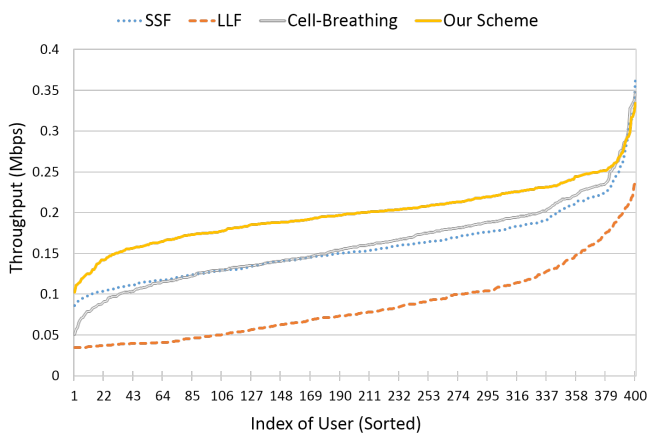


Fig. 14. The throughput of Users (number of Users = 400)

TABLE V
SUMMARY OF THE USER THROUGHPUT

# of Users	100	250	400
SSF	0.29	0.28	0.16
LLF	0.19	0.11	0.09
Cell Breathing	0.07	0.19	0.16
Our Scheme	0.34	0.33	0.20
Improvement of Our Scheme Over SSF	16%	21%	26%
Improvement of Our Scheme Over LLF	76%	192%	133%
Improvement of Our Scheme Over Cell Breathing	377%	75%	23%

V. CONCLUSIONS

In this paper, we proposed an adaptive load balancing scheme through association control in wireless SDN. The proposed scheme consists of two parts: arrival event detection and adaptive load balancing. Through SDN-based technique, APs report their load and connection situation to controller in real time. Based on the global vision of controller, we presented an algorithm on controller to derive the optimal association relationship between APs. We compared the performance of the proposed scheme with SSF, LLF and Cell-Breathing methods through simulation experiments. The simulation results show that the average AP load of the proposed scheme is around 11 $\sim$ 28 % higher than SSF and 26 $\sim$ 346% higher than Cell-Breathing. The proposed scheme checks AP load in real time and then shifts users from heavy-loaded APs to light-loaded APs, so that the total throughput is higher than that in other methods. Simulation results also show that the average user throughput of the proposed scheme is 16 $\sim$ 26% higher than SSF and 23 $\sim$ 377% higher than Cell-Breathing.

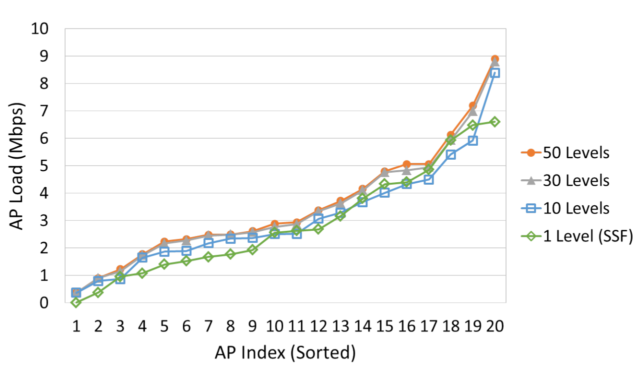


Fig. 15. Impact of the Power Levels of Our Scheme

ACKNOWLEDGMENT

This work was sponsored in part by Ministry of Science and Technology (MOST), Taiwan, under the contract number MOST 105-2221-E-006-186- and MOST 105-2815-C-006-108-E.

REFERENCES

[1] Institute for Information Industry, *Consumer Behavior Research of the first half of* 2014, http://www.iii.org.tw/Press/NewsDtl.aspx?fm_sqno=14&nsp_sqno=1367.

[2] National Development Council, *Personal and Household Digital Opportunity Survey in* 2014, http://www.ndc.gov.tw/m1.aspx?sNo=0028380#.VactLfmqpBc.

[3] iPass Inc., *iPass Wi-Fi Surveys and Reports*, http://www.ipass.com/mobile-resources-surveys/.

[4] 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*, vol. 38, no. 2, pp. 69–74, 2008.

[5] I. S. Association *et al.*, *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, 2001.

[6] J. Teng, C. Xu, W. Jia, and D. Xuan, "D-scan: Enabling fast and smooth handoffs in ap-dense 802.11 wireless networks," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2616–2620.

[7] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: Fast handoff with smart triggers for 802.11 wireless lan," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 749–757.

[8] I. Papanikos and M. Logothetis, "A study on dynamic load balance for ieee 802.11 b wireless lan," in *Proc. COMCON*, vol. 2001. Citeseer, 2001.

[9] A. Balachandran, P. Bahl, and G. M. Voelker, "Hot-spot congestion relief in public-area wireless networks," in *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*. IEEE, 2002, pp. 70–80.

[10] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless lans using association control," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 315–329.

[11] P. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless lans: Algorithms and evaluation," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 2, pp. 164–178, 2007.

[12] Y. Bejerano and S.-J. Han, "Cell breathing techniques for load balancing in wireless lans," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 6, pp. 735–749, 2009.

[13] G. V. Zàruba, M. Huber, F. Kamangar, and I. Chlamtac, "Indoor location tracking using rssi readings from a single wi-fi access point," *Wireless networks*, vol. 13, no. 2, pp. 221–235, 2007.