A Trigger-based Dynamic Load Balancing Method for WLANs Using Virtualized Network Interfaces

Masahiro Kawada, Morihiko Tamai, Keiichi Yasumoto
Graduate School of Information Science, Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan
Email: {masahiro-ka, morihi-t, yasumoto}@is.naist.jp

***Abstract*—We propose a method for dynamic load balancing in wireless LANs (WLANs), which adapts association topology dynamically based on traffic conditions, while keeping the handoff overhead negligible using virtualized wireless network interfaces (WNICs). In large-scale WLANs, there are many locations that each station (STA) can discover multiple access points (APs). In these locations, the conventional approach to the AP selection in which each station connects to the AP with the strongest Received Signal Strength Indication (RSSI) may suffer from imbalanced load among APs. To address this issue, a number of AP selection schemes have been proposed, which achieve load balancing by changing some STA-AP associations. However, since stations cannot communicate during handoff, frequent changes of STA-AP associations will result in serious deterioration of the communication quality. Therefore, in the existing schemes, we face a problem that it is difficult to decide appropriate timing of association changes. Nevertheless, this problem was not considered as a major concern in the literature. In this paper, we propose a method for trigger-based dynamic load balancing in WLANs. In the proposed method, to minimize the handoff overhead, the WNIC on a station is virtualized and connected to multiple APs simultaneously. Using this approach, we propose a method that continuously monitors changes in traffic conditions and that switches STA-AP associations at appropriate timing based on the monitored results. We evaluate the effectiveness of our method in terms of aggregated throughput and fairness using the ns-3 simulator. Compared with the result in the traditional AP selection method, aggregated throughput is improved by about 11%, while increasing the Jain's fairness index by about 19% in our method.**

I. Introduction

Wi-Fi networks have become increasingly popular and been recognized as one of most important technologies for mobile communications. This trend has led to large-scale deployments of wireless LANs (WLANs) in offices, commercial facilities, public facilities, etc. In large-scale WLAN, many devices such as smartphones, game consoles, tablets etc. share the limited channel resource of each WLAN access point (AP). Therefore, association topology should be carefully decided to mitigate load unfairness among APs.

Load balancing in WLANs has been extensively studied in the literature [1]–[11], and most of existing studies have mainly focused on AP selection whenever a station (STA) joins to a WLAN, or just once at a certain point of time. However, traffic condition of a WLAN network can vary continuously over time depending on various factors, e.g., the number of associated stations, the number of active flows, the location of each station, etc. Therefore, it is desirable for AP selection method to adapt to the current traffic condition by recalculating and reflecting the up-to-date optimal association topology.

The standard 802.11 handoff procedure incurs a significant overhead, which results in the delay of the magnitude of hundreds of milliseconds to several seconds [12] due to exchange of control messages between the station and the APs. For this reason, the existing methods tend to avoid changes in STA-AP association topology and to keep the current association topology as long as possible. Consequently, the appropriate timing of replacing the old topology with the up-to-date one has not been well understood in existing studies.

In this paper, we propose a trigger-based dynamic load balancing method for WLANs. In this method, to minimize the handoff overhead, the wireless network interface (WNIC) on a station is virtualized so as to be connected to multiple APs simultaneously. This virtualized WNIC allows the handoff process to switch between APs with negligible overhead. Using this technique, STA-AP association topology is dynamically updated based on the passive monitoring of WLAN environment.

We evaluated the performance of our method with the ns-3 simulator. Compared with the traditional AP selection method, aggregated throughput is improved by about 11% in our method, while increasing the Jain's fairness index by about 19%.

The rest of the paper is organized as follows. We review the related work in Sect. II. In Sect. III, we discuss handoff overhead and describe how to utilize virtualized NICs to reduce the overhead. In Sect. IV, we propose a trigger-based dynamic load balancing method in WLANs. The performance of our method is evaluated by simulations in Sect. V. Finally, we conclude our paper in Sect. VI.

II. Related Work

RSSI (Received Signal Strength Indication) is widely used as a metric in the traditional AP selection method, in which a station selects the AP with the strongest RSSI among the nearby APs. This is based on the fact that the higher the value of RSSI, the lower the probability of packet loss, resulting in the improvement in the usable data rates. However, this approach may cause imbalanced load among APs due

978-1-4673-5939-9/13/$31.00

to reasons such as non-uniform distribution of clients[1], heterogeneous traffic demand among users, interference among communication links, etc. To overcome this issue, various load balancing schemes have been proposed.

Fukuda *et al.* proposed the MLT (Maximizing Local Throughput)-based AP selection method [2], in which each client predicts expected throughput for all nearby APs, and associates to the AP with the highest expected throughput. In order to provide additional information from an AP to stations, the probe response frame and the beacon frame are modified to include the current number of associated stations of the AP.

Yen *et al.* proposed an SNMP-based approach [3] to monitor the current traffic condition. In this method, a dedicated server periodically collects information on APs' loads using SNMP protocol, and each client obtains the information from the server to determine the best AP for association. In order to access MIB-II objects, an SNMP agent is needed to run on each client.

Bejerano *et al.* considered the fairness problem in conjunction with load balancing among APs [4], where all stations can achieve almost the same average throughput.

In addition to the studies described above, the AP selection problem in WLAN is studied under the various contexts. For example, in [5], a transmit power control method for load balancing is proposed, which controls AP's coverage in order to stimulate stations to autonomously switch between APs for topology optimization. In [6], a channel assignment method is considered in conjunction with Gibbs sampler-based AP selection algorithm in order to mitigate channel contention and to achieve fair and optimal sharing of bandwidth between clients. In [7], to overcome the flash crowd problem, which causes sudden surge in the number of clients simultaneously attempting to join to the WLAN network, the authors proposed a queue-based association control method for limiting the number of associated stations for each AP. In [8], the authors proposed a distributed fair AP selection method that takes into account the impact of the choice of the data rate. In [9], the authors studied how to predict available throughput in WLAN, and to utilize this result to improve association topology. In [10], the authors focused on the impact of legacy 802.11a/b/g clients in 802.11n WLAN, and formulated the AP association problem with heterogeneous clients as an optimization problem.

Although these previous studies focus on computing the best association topology at a certain point of time, the appropriate timing of recalculation of association topology has not been considered as a primary concern. In contrast, our focus in this paper is on methods for detecting the condition that indicates noticeable changes in the current traffic, and triggering a recalculation of association topology for adaptation.

III. Reducing Handoff Overhead

One approach for adapting to traffic condition changes is to periodically recompute the optimal topology (e.g., every 10 minutes). However, in this approach, it is difficult to specify the appropriate period of recalculation of the optimal topology. For example, if we use a fixed period of time between two re-computations of the optimal topology, there is a possibility that a large change in traffic condition may occur before the time of the next re-computation, and this causes a certain degree of degradation in the network performance.

[1]In this paper, we use the terms client and station interchangeably.

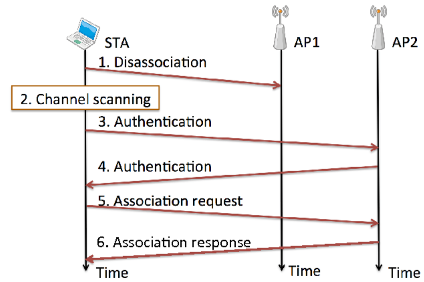

Fig. 1. Handover Process in IEEE 802.11 Networks.

Another solution is to use a shorter period of time between two re-computations. However, this may cause an unreasonably large topology change in order to adapt to a small traffic fluctuation, and thus this approach is not scalable to large WLAN. For example, if a station starts to retrieve the content of a web page, the AP selection algorithm may output the up-to-date association topology that involves changing a large number of associations. To address this problem, we develop a trigger-based method that continuously monitors changes in traffic conditions, and if specific conditions are satisfied, then the re-computation of association topology is executed.

As mentioned in Sect. I, in most of existing AP selection methods, it remains unclear how to adapt to the changes in traffic condition, and they tend to keep the current association topology as long as possible. The main reason why the existing method avoids frequent topology changes is because there is a large overhead in handoff process, which incurs substantial packet loss. Thus, to develop a practical method for dynamic topology adaptation, it is important to adopt a handoff scheme that can switch between APs with negligible overhead. In the following subsections, we briefly review the handoff process in IEEE 802.11 networks, and describe our proposed technique to reduce the handoff overhead.

A. Handoff Overhead in IEEE 802.11 Networks

In traditional 802.11 networks, as shown in Fig. 1, the handoff process requires the following control messages to be exchanged. First, a station disassociates from the current AP. Next, the station starts a channel scanning process, which broadcasts a probe request frame for each channel and waits for probe response frames from APs. In WLANs, since multiple channels are available, this process incurs a large overhead due to message exchange on all available channels. After that, the station connects to a new AP with the message exchange for authentication and association, which also incurs

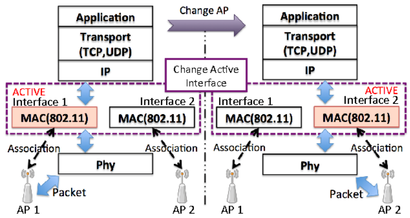

Fig. 2. Virtualization of WNIC at a client side.

a substantial degree of overhead.

According to the study in 802.11b networks by Mishra *et al.*, this handoff process requires 500 milliseconds or more depending on the environment [12]. To reduce this overhead, fast handoff methods (e.g., [13], [14]) can be used. However, it is still difficult to reduce the overhead required by authentication and re-association processes.

On the other hand, there exists another approach for fast handoff that uses multiple network interfaces. For example, Brik *et al.* proposed an approach, called *MultiScan*, which uses two WNICs on a station and connects these WNICs to different APs. In MultiScan, one WNIC is used as primary, which is associated with an AP and is used for communication, while the other (secondary) WNIC is performing other tasks such as channel scanning and association to a new AP. When the method determined that it would be beneficial to switch to the new AP, the roles of primary and secondary WNICs are switched, and thus the former secondary WNIC becomes primary and is used for communication. This approach completely eliminates handoff overhead, because the message exchange for handoff process is not required.

B. Virtualizing WNIC

As described above, it is possible to eliminate handoff overhead by connecting to multiple APs using multiple physical WNICs. However, equipping a mobile device with multiple WNICs may not always be practical, as it is costly, consumes much energy, and generates interference between antennas. For these reasons, we focus on virtualization technique of WNIC, which allows a physical WNIC to simultaneously connect to multiple APs.

Recently, there have appeared several studies [15]–[17] on virtualized WNIC. Chandra *et al.* proposed *VirtualWiFi* [15], [18], which is available on Windows 7. VirtualWiFi is capable of making a physical WNIC to be used as not only a station but also an AP. This capability can be used to bridge communication between an AP and a station (say STA_1) that is out of the AP's transmission range by making another station (say STA_2) that is in the AP's transmission range to act as an AP for STA_1. Kandula *et al.* proposed a method that aggregates the backhaul bandwidth available at nearby APs by letting a station simultaneously connects to the multiple APs using virtualized WNICs [16].

In this study, by using similar virtualization technique as previous ones, we allow each station to simultaneously connect to multiple APs through virtualized WNIC. Fig. 2 illustrates how the virtualization is achieved in the network stack of a client side. WNIC is virtualized at MAC layer level by modifying an 802.11 driver, so that the modified driver allocates and keeps multiple 802.11 state machines for each associated AP. Since the physical layer is shared by multiple MAC layers, an exclusive access control is taken by switching *active* interface among multiple virtualized WNICs, and only the active interface can send/receive packets to/from the AP to which the active interface is associated.

IV. Trigger-based Dynamic Load Balancing

A. Control Messages for Monitoring of WLAN and Updating Association Topology

Topology management in WLAN can be categorized into two types: *centralized* and *distributed*. Since topology management in centralized scheme is relatively easy to avoid uncoordinated behavior among stations such as ping-pong effect [13], which refers to the situation where stations needlessly and repeatedly change its association from one AP to another, we adopt a centralized scheme for topology management.

In the proposed method, we assume that a WLAN network consists of multiple APs, multiple clients, and a centralized server (computation server), which monitors the WLAN environment and calculates association topology. The following messages are introduced to monitor the environment:

- **Client Status Announcement Message:** This message contains the following information about a client: list of MAC addresses of APs to which virtualized WNICs are associated, list of the RSSI values of beacons or probe responses of all associated APs, and the MAC addresses of all virtualized WNICs.
- **AP Status Announcement Message:** This message contains the following information about an AP and associated stations: list of received client status announcement messages, list of traffic rates and data rates for all clients, and MAC address of the AP.

By using these messages, the computation server continuously corrects the current condition of the deployed WLAN, and stores the information to the *monitoring database*. When the server determined that it would be beneficial to update the current topology, the following messages are exchanged between the server and the client that is required to switch AP:

- **AP Switching Request Message:** This message contains the MAC address of the AP to which the virtualized WNIC needs to switch.
- **AP Switching Response Message:** This message indicates a response to the AP switching request message.

B. Trigger for Re-computation of Association Topology

The goal of our method is to dynamically balance load by using *trigger*, performing fine-grained update of the associa-

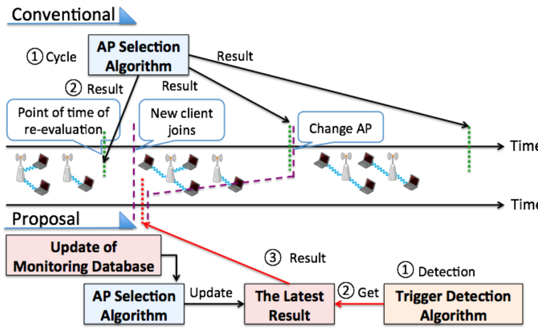

Fig. 3. Difference in updating process of association topology between conventional approach (periodic re-evaluation) and our method.

Require: Let $M (M > 1)$ denote the number of APs.
while true **do**
  if database is updated **then**
    for $j = 1$ to M **do**
      Calculate $U_{StaRate}(j)$ and $U_{MinRate}(j)$
      if $UPrev_{MinRate}(j) < U_{StaRate}(j)$ **or** $UPrev_{StaRate}(j) > U_{MinRate}(j)$ **then**
        $UPrev_{StaRate}(j) \Leftarrow U_{StaRate}(j)$,
        $UPrev_{MinRate}(j) \Leftarrow U_{MinRate}(j)$
        Get the latest result.
        Send AP Switching Request Message to each client.
        break
      end if
    end for
  end if
end while

Fig. 4. Trigger detection algorithm for updating the current association topology.

tion topology. As shown in the upper half of Fig. 3, in most of existing studies, topology adaptation will be delayed since the re-evaluation of traffic condition is performed periodically at a certain period of time. However, the variation of traffic condition is not periodic. Thereby, it is difficult for the periodic approach to achieve quick and flexible topology adaptation due to the unpredictable variation of traffic condition.

In existing studies, up-to-date association topology is computed at the same timing as re-evaluation of the current traffic condition. On the other hand, our method separates the process of the trigger detection from the AP selection algorithm as shown in the lower half of Fig. 3. If a re-computation of association topology is requested by the trigger detection algorithm, our method receives the up-to-date topology, which is called *latest result* hereafter, and sends AP Switching Request Message to each client in order to replace the old topology with the up-to-date one. Here, the latest result is the best association topology computed by an AP selection algorithm and it is continuously updated based on the information stored in the monitoring database, as described in the following subsection.

The trigger detection algorithm utilizes the present bandwidth usage of the deployed WLAN to capture noticeable changes in the current traffic. When N is the number of clients that are associated to an AP, let $TrafficRate_i$ and $TxRate_i$ be the bandwidth which is used by client i and the transmission rate of client i, respectively. Let $TxRate_{min}$ denote the minimum transmission rate that is used in an AP. We compute the following two scores for each AP j:

$$U_{StaRate}(j) = \sum_{i=1}^{N} \frac{TrafficRate_i}{TxRate_i} \quad (1)$$

$$U_{MinRate}(j) = \sum_{i=1}^{N} \frac{TrafficRate_i}{TxRate_{min}} \quad (2)$$

Equation (1) represents the sum of bandwidth usage ratio of all clients in AP j. Bandwidth usage ratio of each client is different from each other since actual transmission rate differs among clients in multi-rate environment even when

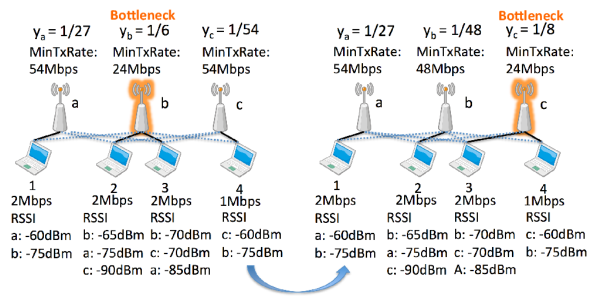

Fig. 5. Example scenario for our AP selection algorithm that detects the bottleneck AP and switches a client from the bottleneck AP to other AP.

the traffic amounts are the same. Moreover, a client with low transmission rate has a negative impact on throughput of other clients with higher transmission rate [19]. We use this score as an impact degree on AP load. In addition, since the bandwidth provided by AP j is reduced by the lowest transmission rate, we define the bandwidth usage ratio by the minimum transmission rate by Equation (2). Let $UPrev_{StaRate}(j)$ and $UPrev_{MinRate}(j)$ be the previously measured values of $U_{StaRate}(j)$ and $U_{MinRate}(j)$, respectively. If $UPrev_{MinRate}(j) < U_{StaRate}(j)$, the impact degree on AP load is over the previous AP load, so we regard that the load increases. On the other hand, we regard that the load decreases, if $UPrev_{StaRate}(j) > U_{MinRate}(j)$. We show this algorithm in Fig. 4.

C. AP Selection Algorithm

A variety of AP selection algorithms can be used in our method. As a typical example, we introduce an algorithm which minimizes the highest AP usage among all APs. Let $y_j = U_{MinRate}(j)$ denote the usage of each AP. This algorithm selects a bottleneck AP (say AP_1) with the lowest usage y, and picks a client receiving the highest RSSI from another AP (say AP_2) as a candidate for switching from AP_1 to AP_2. Then, the AP_2 usage is re-calculated, assuming that the

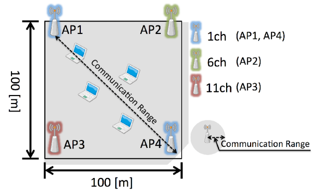

Fig. 6. Deployment of APs and clients in the experiments.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Simulation time	300 sec
Transport layer	UDP
MAC layer	IEEE 802.11g
Offered load	4 Mbps
Packet size	1024 byte
Number of clients	40

selected client switches to AP_2. If the usage of bottleneck AP (i.e., AP_1) becomes lower than the previous one, this algorithm continues to run. Otherwise, this algorithm is stopped, and previous solution is output as the latest result of AP selection.

Our method continuously executes this algorithm so that the latest result is updated whenever the database in the server is updated. Fig. 5 shows an example of this algorithm. Since the usage of AP b is $y_b = 1/6$, AP b is the bottleneck AP. Here, we notice that client 2 and client 3 are both associated to AP b. In this case, client 3 is determined as the candidate, because client 3 receives the highest RSSI (-70 dBm) from another AP (c). Thus, the algorithm selects client 3 and switches it to AP c. Thereafter, AP c becomes the bottleneck AP since usage of AP c changes to $y_c = 1/8$. Then, the algorithm is continued since the usage of bottleneck AP becomes lower than previous one ($1/6$). However, after next switching of a client, bottleneck AP becomes AP b and the usage of bottleneck AP becomes higher than the previous. Hence, the algorithm is stopped, and the result shown in the right side of Fig. 5 is output as the latest result.

V. EXPERIMENTAL EVALUATIONS

In this section, we evaluate our proposed method via simulations. For evaluation, we compare its results against the traditional AP selection method in which each client connects to the AP with the strongest RSSI value, which is called *legacy method* hereafter.

A. Simulation Setup

We implemented our proposed method using the ns-3 [20] simulator. In this experiment, we only simulate the basic 802.11 MAC mechanism, and do not take into account the influence of security protocols such as WEP, WPA, and IEEE 802.1X, because they are not essential to our evaluation.

In the experiments, a WLAN network is deployed in a field of 100×100 square meters, where four APs are placed at corners (see Fig. 6), respectively. The channels of AP1, AP2, AP3, and AP4 are set to 1, 6, 11, and 1, respectively. In this setup, the communication range of each AP covers the entire field. 40 client nodes are randomly distributed over the field, and each client does not change its location during simulation. We suppose that each AP is connected to the wired network, and each client generates downlink traffic from the content server, which is located at the wired network, to WLAN. We also suppose that our method is executed on the computation server, which is located at the wired network. Simulation parameters used in the experiments are shown in Table I. The simulation time is set to 300 seconds following the warm-up period of 60 seconds. APs and clients operate in 802.11g mode. For each downlink flow, offered load is set to 4Mbps.

In addition to the clients that generate UDP-based CBR (Constant Bitrate) flow, we deployed on/off flow clients that repeatedly alters *on state* and *off state*. During the on state, UDP-based CBR traffic with offered load of 4Mbps is generated, while no traffic is generated during the off state. The duration of each of these states is set to 5 seconds. The on/off flow is intended to model the application traffic such as web browsing, which generates traffic intermittently.

B. Experimental Results

1) Variation of aggregated throughput for each AP over time: We first investigate the behavior of our method under the environment where the traffic condition varies over time by observing aggregated throughput for each AP, and compare the result to that of legacy method. In this experiment, all of the clients generate an on/off flow with offered load of 4Mbps.

Fig. 7 shows the result, and we can see that in the legacy method, the aggregated throughput for each AP periodically decreases significantly. This is because the traffic load of each AP fluctuates over time, while the legacy method does not change the association topology, resulting in the unbalanced load among APs. On the other hand, in the proposed method, the load of APs is relatively balanced over time. For example, in the results for AP2 and AP3, the load of AP in the legacy method periodically decreases from more than 10Mbps to below 5Mbps, while the load is relatively constant in the proposed method. From this result, we see that the proposed method can effectively utilize channel resources by changing association topology dynamically based on the current traffic condition.

2) Performance comparison with the Legacy method: Next, we investigate how our method improves the aggregate throughput and the fairness index [21] compared with the legacy method. Here, the fairness index f is defined as:

$$f = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \quad (1 \le i \le n), \tag{3}$$

where x_i is the aggregated throughput of AP$_i$.The value of f ranges from 0 to 1, and becomes 1 when all APs have the same aggregated throughput (i.e., perfectly fair).

In the experiment, we measured the aggregated throughput and the fairness index by increasing the ratio of on/off

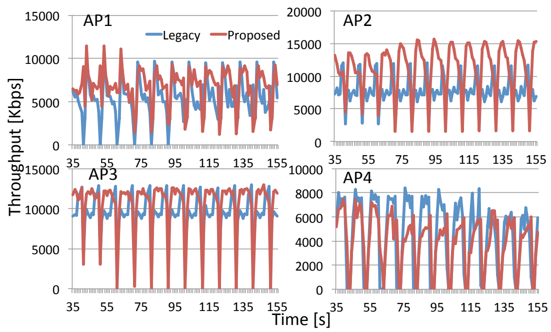

Fig. 7. Aggregated throughput for each AP over time. To focus on the short time behavior, the data in the range [35:155] is extracted from the entire data acquired by the simulation for 300 seconds.

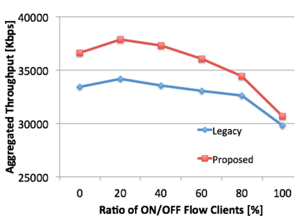

Fig. 8. Comparison of the aggregated throughput.

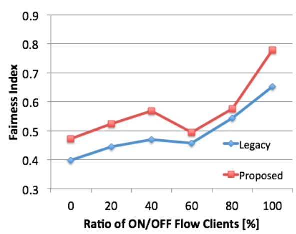

Fig. 9. Comparison of the Fairness index.

clients to CBR clients from 0 to 100%. Fig. 8 and Fig. 9 show the result of aggregated throughput and fairness index, respectively. In all cases, we can see that the proposed method achieves higher aggregated throughput than the legacy method, while increasing the fairness index. This is because in the proposed method, the load among APs is balanced, in spite of the changes in the traffic condition by changing association topology at a finer granularity. More specifically, aggregated throughput is improved by about 11% in our method, while increasing the Jain's fairness index by about 19%, compared with the legacy method.

VI. Conclusion

This paper presented a trigger-based dynamic load balancing method for WLANs, which uses virtualized WNICs to alleviate the handoff overhead, and updates the association topology as soon as possible when a certain change in the current traffic condition is detected. By separating the trigger detection process from the AP selection algorithm, the computation algorithm of the optimal topology can be easily extended and replaced with other algorithms based on the requirements on communication quality of a deployed WLAN. We also proposed a simple algorithm for topology optimization, which reduces the usage rate of the bottleneck AP.

Our experiments using the ns-3 simulator showed that the proposed method can select the appropriate topology based on the changes in traffic condition, which results in a improvement in the network utilization. Compared with the result in the traditional AP selection method, aggregated throughput is improved by about 11% in our method, while increasing the Jain's fairness index by about 19%.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 23700099.

References

[1] L-H. Yen, T-T. Yeh, and K-H. Chi, "Load Balancing in IEEE 802.11 Networks," *IEEE Internet Computing*, Vol. 13, No. 1, pp. 56-64, 2009.

[2] Y. Fukuda, T. Abe, and Y. Oie "Decentralized Access Point Selection Architecture for Wireless Lans - Deployability and Robustness," *Proc. IEEE VTC*, Vol. 2, pp.1103-1107, 2004.

[3] L-H. Yen, and T-T, Yeh, "SNMP-Based Approach to Load Distribution in IEEE802.11 Networks," *Proc. IEEE VTC*, Vol. 3, pp. 1196-1200, 2006.

[4] Y. Bejerano, S-J. Han, and L. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control," *Proc, ACM Mobicom*, pp. 315-329, 2004.

[5] Y. Bejerano and S. J. Han, "Cell Breathing Technique for Load Balancing in Wireless LANs," *Proc. IEEE Trans. Mobile Computing*, Vol. 8, No. 6, pp.735-749, 2009.

[6] B. Kauffmann, F. Baccelli, A. Chainteau, V. Mhatre, K.Papagiannaki, and C. Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks," *Proc. IEEE Infocom*, pp. 1451-1459, 2007.

[7] A. P. Jardosh, K.Mittal, K. N. Ramachandran, E. M. Belding, and K. C.Almeroth, "IQU: Practical Queue-Based User Association Management for WLANs," *Proc. MobiCom*, pp. 158-169, 2006.

[8] H. Gong, K. Nahm and J. W. Kim, "Distributed Fair Access Point Selection for Multi-Rate IEEE 802.11 WLANs," *Proc. CCNC*, pp. 528-532, 2008.

[9] K. Sundaresan and K. Papagiannaki, "The Need for Cross-Layer Information in Access Point Selection Algorithms," *Proc. ACM IMC*, pp. 257-262, 2006.

[10] D. Gong and Y. Yang, "AP Association in 802.11n WLANs with Heterogeneous Clients," *Proc. IEEE Infocom*, pp. 1440-1448, 2012.

[11] T. M. Oliveira, M. W. R. da Silva, K. V. Cardoso, and J. F. de Rezende, "Virtualization for Load Balancing on IEEE 802.11 Networks," *Proc. Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Vol.73, pp. 237-248, 2010.

[12] A. Mishra, M. Shin and W. A, Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *Proc. ACM Sigcomm Computer Communication Review*, Vol. 33, No.2, pp. 93-102, 2003.

[13] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," *Proc. Infocom*, Vol.1, pp. 675-684, 2005.

[14] H. Wu, K. Tan, Y. Zhang, and Q. Zhang. "Proactive Scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN," *Proc. Infocom*, pp. 749-757, 2007.

[15] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: Connecting to Multiple IEEE802.11 Networks Using a Single Wireless Card," *Proc. IEEE Infocom*, Vol. 2, pp. 882-893, 2004.

[16] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi, "Fatvap: Aggregating AP Backhaul Bandwidth," *Proc. 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 89-104, 2008.

[17] X. Xing, S. Mishra and X. Liu, "ARBOR: Hang Together Rather than Hang Separately in 802.11 Wifi Networks," *Proc. IEEE Infocom*, pp. 1-9, 2010.

[18] Virtual Wi-Fi, http://research.microsoft.com/en-us/um/redmond/projects/virtualwifi/

[19] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *Proc. IEEE Infocom*, 2003.

[20] Network Simulator ns-3, http://www.nsnam.org/

[21] D. -M. Chiu and R. Jain, "Analysis of The Increase and Decrease Algorithms for Congestion Avoidance in Comupter Networks," *Computer Networks and ISDN Systems* vol. 17, No. 1, pp. 1-14, 1989.