An Extension of Active Access-Point Configuration Algorithm to IEEE 802.11n and 11ac Dual Interfaces in Wireless Local-Area Network

Sujan Chandra Roy, Nobuo Funabiki, Kwenga Ismael Munene, Md. Mahbubur Rahman, and Minoru Kuribayashi

***Abstract*—Nowadays, *wireless local-area networks (WLANs)* have been densely deployed in a lot of fields by different providers around the world due to license-free features. To improve the network performance under such environments, we have studied the *active AP configuration algorithm* of optimizing the active *access points (APs)* along with host associations and channel assignments. Unfortunately, it limits the APs adopting one network interface of the *IEEE 802.11n* protocol working at 2.4GHz. Recently, the *11ac* protocol at 5GHz has also become common, which allows the dual interfaces using different frequencies to increase the transmission capacity of an AP. In this paper, we extend the *AP configuration algorithm* to consider dual interfaces of 11n and 11ac at the APs in WLAN. In addition to selecting the active APs, their assigned channels, and associated hosts, the network interface is assigned to each host. The two throughput estimation models are adopted to consider the throughput differences of the two protocols. For evaluating the proposal, we compare the number of active APs and throughput performances against the conventional case using one interface through simulations using the *WIMNET* simulator and throughput measurements using the testbed system that adopts *Raspberry Pi 4B* for APs. The results confirm that the proposal can reduce the number of active APs while increasing the total throughput.**

***Index Terms*—WLAN, access point configuration, algorithm, dual interface, 11n, 11ac, Raspberry Pi.**

I. Introduction

Recently, the IEEE 802.11 *wireless local-area networks (WLANs)* have become popular due to the flexibility, low-cost installation, and high performance. WLANs have been extensively deployed in various places including homes, offices, universities, hotels, shopping centers, and airports. Wireless communications between *access-points (APs)* and hosts make WLAN more extensible and flexible than wired LANs [1], [2].

APs are often densely deployed for service coverage. The network performance of WLAN can be degraded by interferences between neighboring APs. To improve the network performance, the configuration of WLAN should be optimized in dense WLAN environments, according to traffic demands and network conditions. The redundant APs in the network field should be deactivated to reduce interferences and energy consumptions [3], [4]. At the same time, the hosts in WLAN should have the sufficient performances.

To address the issues mentioned above, in [5], [6], we studied the *active AP configuration algorithm* to optimize the number of active APs and the assigned channels to them. It satisfies the minimum host throughput constraint of their associated hosts. Unfortunately, this algorithm limits that any AP uses only one *IEEE 802.11n* network interface at 2.4GHz.

As advancement of devices, currently, the simultaneous use of IEEE 802.11n and 11ac interfaces [7] at one AP is possible. This dual interface using different frequencies can greatly enhance the network performance, while reducing the number of active APs in the network for smaller power consumption and less interference.

In this paper, we extend the *AP configuration algorithm* to consider *dual interfaces* of 11n and 11ac at the AP in WLAN. In addition to selecting the active APs, their assigned channels, and associated hosts in the previous algorithm, the extended one assigns the network interface to each host. The throughput estimation model for 11ac is used together with the model for 11n to consider the throughput differences of the two protocols.

The effectiveness of the proposal is demonstrated through extensive simulation using the *WIMNET* simulator [8] in two network topologies. Besides, the proposed algorithm is implemented on the elastic WLAN system testbed using *Raspberry Pi 4B* [9] for the APs to verify the practicality using the real system. The built-in wireless network interface card (NIC) of this device is used for 11n and *Archer T4U* [10] wireless NIC adapter is for 11ac. The simulation and measurement results show that when compared with the previous algorithm [6], the proposed algorithm reduces the number of active APs in the network and gives the higher network performance.

The rest of the paper is organized as follows: Section II introduces related works in literature. Section III reviews our previous works to this paper. Section IV presents the extension of the active AP configuration algorithm. Sections V and VI evaluate the proposal through simulations and testbed experiments respectively. Finally, Section VII concludes this paper with future works.

II. Related Works

In this section, we briefly review some related works to this paper.

In [11], Tewari and Ghosh addressed the issues of the efficient AP placements and the joint assignment of partially

Manuscript received December 20, 2021; revised March 14, 2022.

S. C. Roy, N. Funabiki, K. I. Munene, M. M. Rahman, and M. Kuribayashi are with the Department of Electrical and Communication Engineering, Okayama University, Okayama, Japan (e-mail: funabiki@okayama-u.ac.jp).

doi: 10.18178/ijfcc.2022.11.2.583

overlapping channels and transmission powers to the APs to maximize the network throughput.

In [12], Heiba *et al*. proposed an active AP configuration algorithm that activates the minimum number of APs according to the number of users and their locations on the network. It finds optimal channel assignments to the active APs from the non-interfering channels to minimize the total interference. It also balances the loads among the active APs by simply switching users to the least loaded AP.

In [13], Tang *et al.* proposed an association control algorithm using the maximum aggregated bandwidth utility (MABU) to optimize the throughput in WLANs. Their solution achieves the load balancing between the APs while considering users' traffic demands.

In [14], Farsi *et al.* presented a joint and separate optimization planning method for both the AP placement and the channel assignment in WLAN to ensure sufficient capacities of users. They adopted a fuzzy C-clustering algorithm for AP deployments.

In [15], Lima *et al*. divided the WLAN design into two stages. The first stage solves the problem of the AP location using a non-dominated sorting genetic algorithm that aims minimizing the number of APs on the network. The second stage solves the problem of the channel assignment by modifying the DSATUR algorithm to take into account the load of each AP.

In [16], Fang and Lin presented an algorithm for the AP setup optimization in an indoor environment. Their algorithm aims choosing a proper set of AP locations so that the signal at a host is maximized and the noise is minimized simultaneously.

In [17], Eisenblatter *et al*. proposed a joint optimization algorithm for the AP placement and the channel assignment using the integer linear programming model. The objective of the AP placement is to maximize the total throughput by minimizing the interference between the APs.

In our survey, all the research papers study the network using a single *IEEE 802.11n* network interface at the AP. None of them consider dual interfaces at the AP to improve network performance.

III. Review of Previous Works

In this section, we review our previous works related to this paper.

A. Throughput Estimation Model

The throughput estimation model estimates the throughput of a wireless communication link from a source node to a destination node in WLAN. First, it estimates the received signal strength (RSS) at the destination node using log-distance path loss model. Then, it converts the RSS to throughput using the sigmoid function.

1) Received signal strength estimation

First, the Euclidean distance *d(m)* is calculated for each link (AP/host pair) by:

$$d = \sqrt{\left(AP_x - H_x\right)^2 + \left(AP_y - H_y\right)^2} \quad (1)$$

where AP_x, AP_y and H_x, H_y does the x and y coordinates for the AP and the host respectively. Then, *RSS* is calculated by:

$$P_d = P_1 - 10\alpha \log_{10} d - \sum_k n_k W_k \quad (2)$$

where P_d represents *RSS* (-dBm) at the host, P_1 does *RSS* at the 1m distance from the AP when no obstacle exists, α does the path loss exponent, n_k does the number of *type_k* obstacles along the path from the AP to the host, and W_k does the signal attenuation factor (dBm) for *type_k* obstacle. P_1, α, and W_k are parameters in Table I.

2) Throughput estimation from received signal strength

From *received signal strength* P_d, the throughput/link speed is calculated by using the *sigmoid function* [18],

$$tp = \frac{a}{1+e^{-\left(\frac{(120+P_d)-b}{c}\right)}} \quad (3)$$

where *tp* represents the estimated throughput (Mbps), and a, b, and *c* are constant parameters obtained through measurements and optimized by *parameter optimization tool* in Table I.

3) Model parameter optimization

The throughput estimation model has several parameters whose value determines the estimation accuracy. These values are optimized by using the *parameter optimization tool*. It uses a local search algorithm that combines the tabu table and the hill climbing procedure to avoid a local minimum convergence [19].

Table I summarizes the parameter optimization results of the throughput estimation models for 11n and 11ac that are used in this paper for two fields, *Field #1* and *Field #2* [18]. *Field #1* represents the *3rd floor of Engineering Building #2* in Fig. 2(b) and *Field #2* does the *2nd floor of Graduate School Building* at Okayama University in Fig. 2(c).

TABLE I: Parameter Optimization Results for 11n and 11ac

Parameter	Field #1 in Figure 2(b)		Field #2 in Figure 2(c)	
	11n	11ac	11n	11ac
P1	-28.1	-27.8	-27.1	-27
α	2.2	2.4	2.2	2.25
W1	7.5	7.1	4.5	2.4
W2	6	8	3	3.6
W3	4	4	2	2
W4	2.5	2	1.9	1
W5	2.4	2.2	1.8	2
W6	2	2.4	1.2	1.3
a	42	84	43.75	85
b	57	56.5	56.8	57
c	6.5	6.5	7	6.8

B. Elastic WLAN System

The *elastic WLAN system* has been designed to dynamically optimize the network topology and configuration according to the network conditions. Fig. 1 illustrates an example topology of the system.

The implementation of the elastic WLAN system adopts the *management server* to manage and control the APs and the hosts by running the *active AP configuration algorithm* [6]. This server not only has administrative access to all network devices, but also controls the overall system through the following three steps:

1) The server explores all network devices and collects the required information for the active AP configuration algorithm.

2) The server executes the active AP configuration algorithm using the inputs derived in the previous step. The output of the algorithm contains the list of the active APs, the host associations, and the assigned channels.

3) The server applies this output to the network by activating or deactivating the specified APs, changing the specified host associations, and assigning the channels.

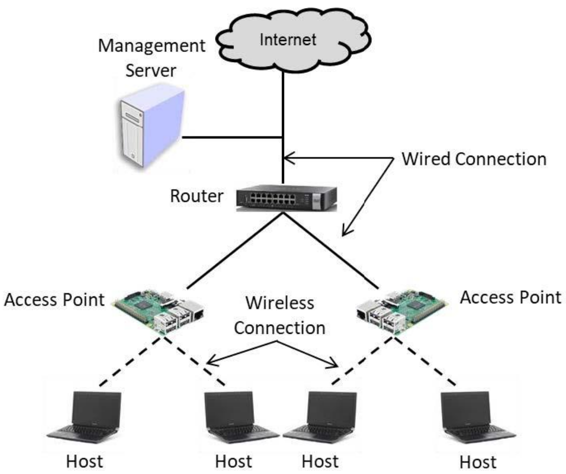


Fig. 1. Example topology of elastic WLAN system.

C. Formulation of Active AP Configuration Algorithm

The active AP configuration algorithm is formulated as follows:

1) Inputs:

- Number of hosts: H
- Number of APs: N
- Link speed between AP_i and $host_j$ for i= 1 to N, j= 1 to H: tp_{ij}, where the link speed can be estimated using Eq. (3).
- Minimum link speed for association: S.
- Number of non-interfered channels: C.

2) Outputs:

- Set of active APs.
- Set of hosts associated with each active AP.
- Channel assigned to each active AP.

3) Objectives:

- To minimize E_1.
- Holding the first objective, to maximize E_2.
- Holding the two objectives, to minimize E_3 for channel assignments.

E_1 represents the number of active APs in the network and E_2 indicates the *minimum average host throughput* for the bottleneck AP:

$$E_2 = \min_j \left[TH_j \right] \tag{4}$$

where TH_j represents the average host throughput for AP_j that is given by:

$$TH_j = \frac{1}{\sum_k \frac{1}{tp_{jk}}} \tag{5}$$

where tp_{jk} represents the link speed between $node_j$ and $node_k$ ($link_{jk}$).
E_3 signifies the total interfered communication time:

$$E_3 = \sum_{i=1}^{N} \left[\sum_{\substack{k \in I_i \\ c_k = c_i}} T_k \right] \tag{6}$$

where T_i represents the total communication time for AP_i, I_i does the set of the interfered APs with AP_i, and c_i does the assigned channel to AP_i.

4) Constraints:

- *Minimum host throughput constraint*: each host in the network must enjoy the given threshold G on average when all the hosts are communicating simultaneously.
- *Bandwidth limit constraint:* the bandwidth of the wired network to the Internet must be less than or equal to the total available bandwidth of the network B^a.
- *Channel assignment constraint:* each AP must be assigned one channel between 1 and C.

D. Procedure of Active AP Configuration Algorithm

The procedure of the active AP configuration algorithm is as follows:

1) First Phase: The first phase of the algorithm selects the active APs and the host associations to minimize E_1 and E_2.

a) Preprocessing: At first, the locations of the APs and the hosts are used as inputs of the algorithm. It estimates the link speed between every possible pair of AP and host using *throughput estimation model* described in Section III.

b) Initial Solution Generation: An initial solution E_1 is derived using a greedy algorithm [20].

c) Host Association Improvement: The overall throughput and the minimum host throughput are improved by randomly changing the association of a host by the procedure in [6].

d) AP Selection Optimization: The cost functions E_1 and E_2 are further jointly optimized in this phase via the *local search* [5][21].

e) Link Speed Normalization: The fairness criterion will be adopted when the total expected bandwidth exceeds B^a. Then, the link speed is normalized.

f) Termination Check: When either of the following conditions is satisfied, move to the second phase in 2:

i. The *minimum host throughput constraint* is satisfied.
ii. All the APs in the network have been activated.

2) *Second Phase:* The second phase assigns a channel to each active AP to minimize E_3.

a) Preprocessing: The interference and delay conditions of the network are represented by a graph.

b) Interfered AP Set Generation: The set of APs that are interfering with each other is found for each AP.

c) Initial Solution Construction: An initial solution is derived using a greedy algorithm.

d) Solution Improvement by Simulated Annealing: The initial solution is improved by the simulated annealing (SA) procedure with the constant *SA temperature* T^{SA} for the *SA repeating times* R^{SA}, where T^{SA} and R^{SA} are given algorithm parameters.

3) *Third Phase:* The third phase averages the loads among the different channels to minimize E_3.

a) Initialization: The AP *flag* is initialized by 0 (= OFF) for every AP. This flag is used to avoid processing the same AP again.

b) AP Selection: One OFF flag AP is selected to move its associated host to a different AP that is assigned a different channel.

c) Host Selection: One host associated with the selected AP is selected for the AP movement.

d) Change Application: Finally, the new associated AP is chosen for this selected host.

IV. Extension of Active AP Configuration Algorithm

In this section, we present the extension of the active AP configuration algorithm to consider dual interfaces of each AP.

A. Problem Formulation in Extension

The problem formulation for the algorithm is extended to handle dual interfaces at the AP. The modifications of the algorithm for the extension is specified as follows:

1) Inputs

- Number of APs: N where each AP has the dual interfaces, one for 11n and another for 11 ac.
- Link speed between AP_i using each interface and $host_j$ for i= 1 to N, j= 1 to H: tp_{ij}, where the link speed can be estimated using Eq. (3) with corresponding parameter values in Table I.
- Number of orthogonal channels (OCs) for each interface: C.

2) Outputs

- Set of active APs with dual interfaces.
- Set of hosts associated with each interface of every active AP.
- Channel assigned to each interface of every active AP.

3) Objectives

- To minimize the number of active APs E1 with dual interfaces while satisfying the average throughput constraint of each host.

4) Constraints

- *Minimum host throughput constraint:* every host connected to an AP interface must enjoy the given threshold G on average when all the hosts are communicating simultaneously.
- *Channel assignment constraint:* each interface of an AP must be assigned a channel.

B. Preprocessing in Algorithm Extension

The link speed for each possible pair of an AP interface and a host is estimated by the *throughput estimation model* in Section III. Besides, the 11n interface of an AP is selected as the initial candidate one for any host.

C. AP Selection Optimization in Algorithm Extension

The *AP Selection Optimization* phase optimizes the selection of the active APs with dual interfaces and the AP-host associations to further minimize both E_1 and E_2 by the local search method [21].

The procedure in the extension is described as follows:

1) (a) Initialize E_1^{best} and E_2^{best} by the current algorithm output after the *Host Association Improvement* phase in [6].

(b) Initialize the *selection flag* for every AP interface by OFF, and

(c) Initialize the number of current active AP interfaces P by 0 (P= 0).

Here, the *selection flag* is introduced to avoid continuously processing the same AP interface.

2) Repeat the following procedure 18 × N × H times:

(a) (i) Randomly select the one active 11ac AP interface with the OFF selection flag and deactivate it. If it is found, decrease P by 1.
(ii) If the active 11ac interface is not found for deactivation in (i), randomly select one active 11n AP interface with the OFF selection flag and deactivate it. If it is found, decrease P by 1.
(iii) Set the selection flag ON for this deactivated interface.

(b) (i) Find a new associable active AP interface for each host that is currently associated with the AP interface that is deactivated in (a), and
(ii) Associate the host with this new interface. If multiple active AP interfaces can be associated with this host, select the one that is associated with the largest number of hosts among them.

(c) If some hosts cannot be associated with any new AP interface, repeat the following steps until every host can be associated with an active interface.
(i) Find a non-active 11n interface with the OFF selection flag that can be associated with the host associated with the deactivated interface.
(ii) If there is no such interface, find a non-active 11n interface with the ON selection flag.
(iii) If multiple interfaces are found in both cases, select one interface that provides the maximum link speeds to the host for tie break.
(iv) If one interface is found, increase P by 1, activate this interface, and set the selection flag ON.

(d) If the total number of the interfaces of the current active APs, which is given by $E_1 \times 2$, is smaller than P,

apply the following procedure until it satisfies $E_1 \times 2 = P$:
(i) Randomly select the non-active 11ac interface with the OFF selection flag of the active AP that has the active 11n interface and activate the interface.
(ii) If there is no such 11ac interface in (i), randomly select one 11ac interface with the ON selection flag of the active AP and activate the interface.
(iii) If there is no such 11ac interface in (ii), randomly select one non-active 11n interface with the OFF selection flag of the non-active AP and activate the AP.
(iv) If there is no such 11n interface in (iii), randomly select one non-active 11n interface with the ON selection flag of the non-active AP and activate the AP.
(v) If a non-active interface is selected, activate this interface, and increase P by 1.

(e) Apply the following steps:
(i) Set the selection flag to OFF for the newly activated interface of the AP in Step (2c).
(ii) If the number of the active APs ($=E_1$) is smaller than E_1^{best}, apply the *Host Association Improvement* phase, calculate (E_2), and update E_1^{best}, E_2^{best}.
(iii) Otherwise, if $E_1 = E_1^{best}$, apply the *Host Association Improvement* phase, and calculate E_2. If E_2 is smaller than E_2^{best}, update E_1^{best}.

(f) If at least one interface of an active AP has the OFF selection flag, go to Step (2a).

3) Apply the following steps to satisfy the average throughput constraint with $18 \times N$ times.

(a) If $1/E_2^{best} \geq G$ (minimum throughput constraint) is satisfied, mark the solution as feasible, and go to Step (4).

(b) Otherwise, $1/ E_2^{best} \geq G$ (minimum throughput constraint) is not satisfied and if $P = 0$, apply the following steps:
(i) Set $P=E_1^{best} +1$.
(ii) Randomly select one 11ac interface of the active AP. If not found, randomly select one 11n interface of an AP, and run the *Host Association Improvement* phase. Update E_1^{best} and E_2^{best}.
(iii) Go back to Step (2).

4) Output the best found result with E_1^{best} and E_2^{best}.

D. Termination Check in Algorithm Extension

For every active AP, if either of the two interfaces is not activated, activate the interface, and apply the *Host Association Improvement* phase. Then, if the minimum host throughput constraint is satisfied, terminate the algorithm. Otherwise, go to *AP Selection Optimization phase*.

V. Evaluations by Simulations

In this section, we evaluate the extended active AP configuration algorithm for the APs with dual interfaces through simulation using the *WIMNET* simulator.

A. Simulation Platform

Tables II and III show the hardware and software specifications and the parameter values in the simulations using the *WIMNET* simulator.

TABLE II: Simulation Environment

simulator	WIMNET simulator
interface	IEEE 802.11n, IEEE 802.11ac
CPU	Intel Core i7
memory	8GB
OS	Ubuntu LTS 14.04

TABLE III: Simulation Parameters in WIMNET Simulator

Parameter	**Values**
packet size	1500 bytes
max. transmission rate	150 Mbit/s
propagation model	log distance path loss model
rate adaptation algorithm	link speed estimation model [18]
carrier sense threshold	-85 dBm
transmission power	19 dBm
collision threshold	10
RTS/CTS	yes

B. Results for Large Topology

First, we evaluate the proposal in a large-size network topology in Fig. 2(a) that models the expanded *3rd floor of Engineering Building #2* at Okayama University. 14 APs with dual interfaces and 40 hosts are distributed in the four rooms.

Table IV shows the simulation results using the AP configuration found by the proposal for the dual network interfaces and that by the previous algorithm for the single network interface in [6]. It compares the number of active APs and the overall throughput by the *WIMNET* simulator when the minimum host throughput constraint is changed from 5Mbps to 15 Mbps. The results in this table confirm that the proposal outperforms the previous algorithm. It reduces the number of active APs to satisfy the constraint and increases the overall throughput in the network.

C. Results for Small Topology

Next, we evaluate the proposal in two network fields that models the real floors at the *3rd floor of Engineering Building #2 and the 2nd floor of Graduate School Building* at Okayama University in Figs. 2(b) and 2(c) for small topologies. In the two fields, six different network topologies are considered together. Table V shows the device locations in each topology in them.

Table VI shows the results for each topology by the proposal and by the previous. It is noted that each host satisfies the minimum host throughput constraint. The results in the table again confirm that our algorithm with dual interfaces at the AP can minimize the number of activate APs and provide the higher throughput.

VI. Evaluations by Testbed Experiments

In this section, we evaluate the proposal through experiments using the elastic WLAN system testbed with *Raspberry Pi* for the APs.

A. Experiment Setup

For the experiments, the *Raspberry Pi 4B* [9] is used for the APs running *hostapd* [22]. The built-in wireless NIC is used for 11n interface and *Archer T4U* [10] wireless NIC adapter is for 11ac interface. The dual interfaces are configured by following the commands in Appendix A.

TABLE IV: PERFORMANCE COMPARISON BETWEEN PROPOSAL AND PREVIOUS FOR LARGE TOPOLOGY

Min. host throughput constraint G (Mbps)	Total Number of hosts (H)	Proposed algorithm			Algorithm [6]		
		Total number of Active APs out of 14	Ave. min. host throughput (Mbps)	Ave. overall throughput (Mbps)	Total number of Active APs out of 14	Ave. min. host throughput (Mbps)	Ave. overall throughput (Mbps)
5	40	2	5.23	206.11	5	5.17	185.25
10		4	11.71	468.26	10	10.43	410.25
15		6	16.58	667.77	14	14.34	575.3

TABLE V: AP AND HOST LOCATION FOR SMALL TOPOLOGY

Exp. Field	Topology	AP position					Host position								
		AP1	AP2	AP3	AP4	AP5	H1	H2	H3	H4	H5	H6	H7	H8	H9
Field #1	1	P1	P2	P3	P4		A1	A2	B2	B4	B3	C3	D4	D1	
	2						front D308	A2	B1	B4	front D305	C3	D4	D2	
	3	P1	P2	P3	P4	P5	A1	A2	B1	B4	C4	C1	front D301	D4	D1
	4						front D308	A2	front D307	B4	C4	front D305	front D301	D4	D2
Field #2	5	P1	P2	P3	P4	P5	H1	H2	H3	H4	L1	A3	B2	D2	E3
	6						front #G	H2	H3	H4	L1	front #A	C2	D2	F1

TABLE VI: COMPARISON BETWEEN THE PROPOSED ALGORITHM AND THE ALGORITHM IN [6] FOR SMALL TOPOLOGY

Exp. Field	Topology	Total number of APs	Total number of hosts (H)	Min. host throughput constraint G (Mbps)	Proposed algorithm		Algorithm [6]	
					Total number of Active APs	Ave. min. host throughput (Mbps)	Total number of Active APs	Ave. min. host throughput (Mbps)
Field #1	1	4	8	5	1	9.52	2	7.32
				15	2	19.02	4	14.96
	2			5	1	8.40	2	7.87
				15	2	17.79	4	15.33
	3	5	9	5	1	8.81	2	6.44
				15	3	21.29	5	16.72
	4			5	1	8.35	2	6.38
				15	2	17.13	5	16.24
Field #2	5	5	9	5	1	8.77	2	6.46
				15	2	18.40	5	17.83
	6			5	1	9.61	2	6.19
				15	2	19.95	5	16.91

Linux laptop PCs are used for the server and the client hosts. The *20MHz* non-bonded channel is used at both interfaces.

To measure the throughput, *iperf* 2.0.5 [23] is used, which generates TCP traffics with the 477KB TCP window and the 8KB buffer. Table VII shows the specifications of the devices and software in the experiments.

Figure 3 illustrates the example network topology of the elastic WLAN system testbed for experiments that uses two APs with dual interfaces and four hosts. The management server controls the APs and the hosts by using the administer-ative access rights to them. The APs are connected to the server through wired connections. The hosts and the APs are connected through wireless connections.

B. Experiment Fields

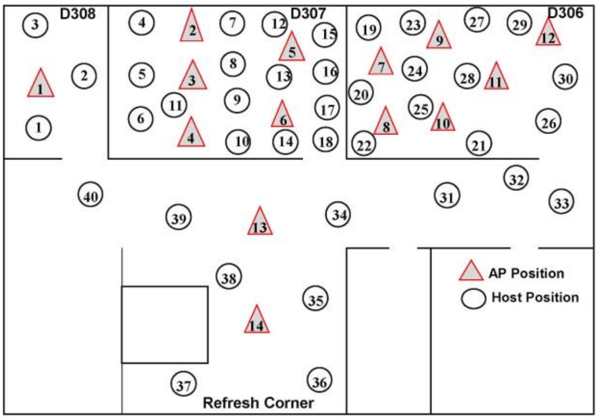
The measurements were conducted in two indoor environments. Figures 2(b) and 2(c) illustrate the layouts of the experiment sites. Table V shows the devices locations for each topology.

C. Experiment Results

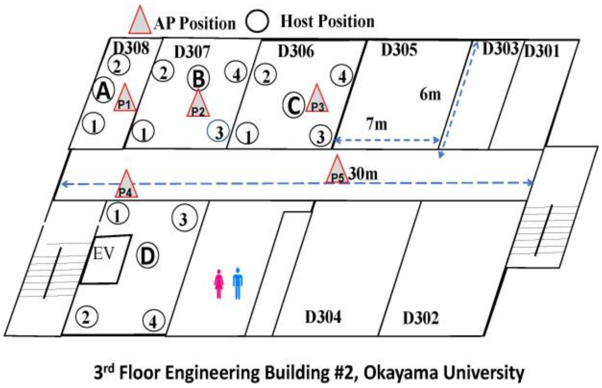
In this section, we discuss the experiment results in two network fields.

1) Evaluation in engineering building #2:

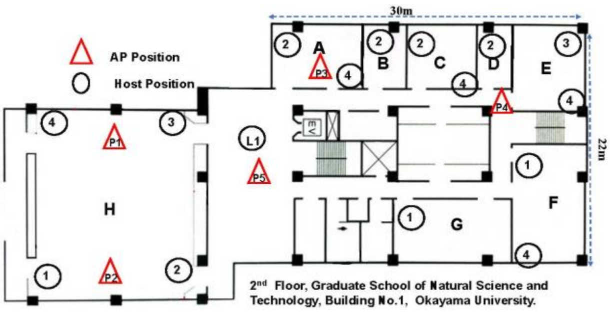
Fig. 4 shows the overall throughput in the network field of Engineering Building #2. The minimum host throughput constraint is applied with G=5Mbps and G=15Mbps. For each case, the proposed algorithm provides the higher overall throughput than the previous one. Besides, when the measured throughput and the simulation throughput are compared, they are matched well in any case, which confirms the accuracy of the proposed algorithm.



(a) Solution for large topology with 40 hosts and 14 APs



(b) Field #1 in engineering building #2



(c) Field #2 in Graduate School Building
Fig. 2. Simulation and measurement topology.

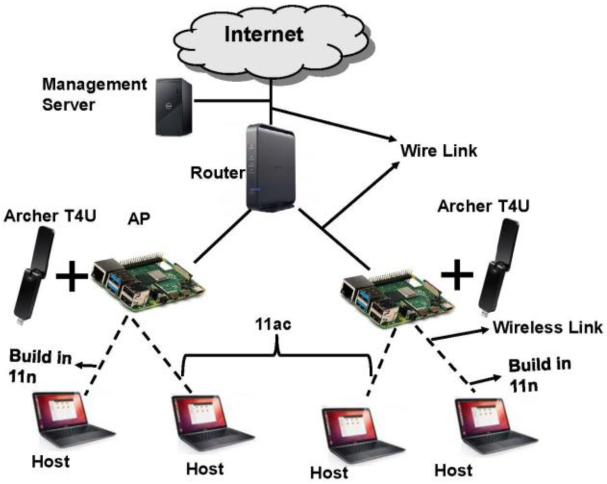


Fig. 3. Example network topology of elastic WLAN system testbed.

TABLE VII: DEVICE AND SOFTWARE SPECIFICATIONS

access point	
Model	Raspberry Pi 4B
CPU	Broadcom BCM2711 @1.5Ghz
RAM	8GB LPDDR4-3200 SDRAM
Operating System	Linux Raspbian
Software	hostapd
External NIC	Archer T4U V3.0 AC1300
server PC	
Model	Fujitsu Lifebook S761/C
CPU	Intel Core i5-2520M @2.5Ghz
RAM	4GB DDR3 1333MHz
Operating System	Linux Ubuntu 14.04 LTS
Software	Iperf 2.0.5
host PC	
Model	1. Toshiba Dynabook R731/B 2. Toshiba Dynabook R734/K 3. Fujitsu Lifebook S761/C
CPU	1. Intel Core i5-2520M @2.5Ghz 2. Intel Core i5-4300M @2.6Ghz 3. Intel Core i5-2520M @2.5Ghz
RAM	4GB DDR3 1333MHz
Operating System	Linux Ubuntu 14.04 LTS
Software	Iperf 2.0.5

2) Evaluation in graduate school building:

Fig. 5 shows the overall throughput in the network field of Graduate School Building. The same minimum host throughput constraints are considered. Again, for each case, the proposal provides the higher overall throughput than the previous. Besides, the measured throughput and the simulation throughput are matched well. Thus, the accuracy and effectiveness of the proposed algorithm are confirmed.

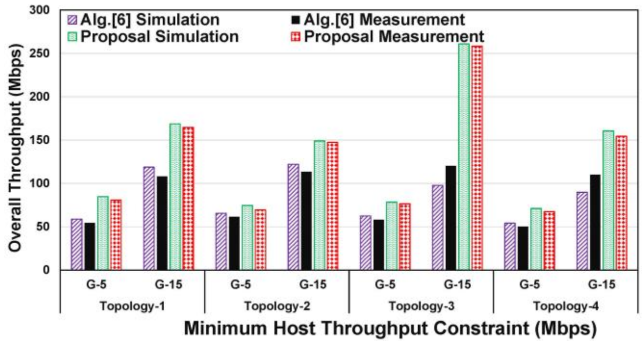


Fig. 4. Throughput results in Field #1

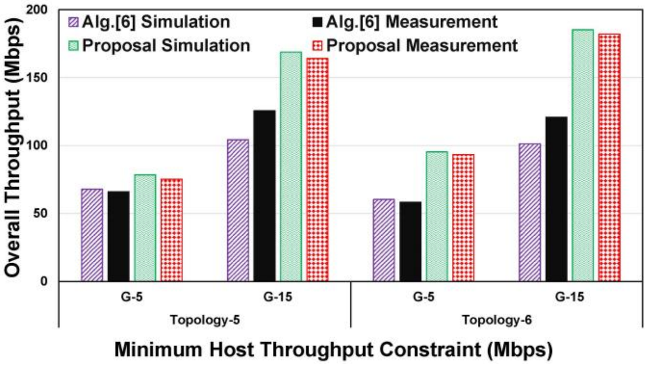


Fig. 5. Throughput results in Field #2.

VII. CONCLUSIONS

This paper presented the extension of the *active AP configuration algorithm* to the dual network interfaces of the APs that adopt IEEE 802.11n and 11ac dual interfaces at the same AP in WLAN. The effectiveness of the proposal is confirmed through simulations using the *WIMNET* simulator and real testbed experiments using *Raspberry Pi* for the APs. The number of active APs was reduced and the total throughput of the network was increased, while the minimum host throughput constraint was satisfied. In future works, we will further enhance the AP configuration algorithm for network interfaces using more advanced protocols such as 11ax and will apply the algorithm in various network fields and topologies.

APPENDIX A-SETUP PROCEDURE OF DUAL INTERFACES AT RASPBERRY PI

1) Install the hostapd and dnsmasq using the following command [24].

$ sudo apt-get install hostapd
$ sudo apt-get install dnsmasq

2) Uncomment and set *DAEMON_CONF* to the absolute path of a hostapd configuration file to start hostapd during system boot:

DAEMON_CONF="/etc/hostapd/hostapd.conf"
DAEMON_ CONF="/etc/hostapd/hostapd1.conf"

3) Configure a static IP for the wlan0 and wlan1 interface by modifying the dhcpd configuration file:

$ sudo nano /etc/dhcpcd.conf

At the end of the dhcpd file, add the following example content and then save.

interface wlan1
static ip_address=192.168.2.1/24
interface wlan0
static ip_address=192.168.3.1/24

4) Configure the DHCP server (dnsmasq) for network configuration parameters, such as IP addresses, for interfaces and services. As for the dhcp rules, rename the default configuration file for backup and edit a new one:

$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
$ sudo nano /etc/dnsmasq.conf

Add the following example content in dnsmasq file, then save:

interface=wlan1
dhcp-range=192.168.2.2,192.168.2.40,255.255.255.0,24h
interface=wlan0
dhcp-range=192.168.3.2,192.168.3.40,255.255.255.0,24h

5) To build a bridge between the wlan0, wlan1, and eth0 interfaces, install a package using the following command:

$ sudo apt-get install bridge-utils

Then, add a new bridge (called br0) using the following command:

$ sudo brctl addbr br0

Next, connect the eth0 interface to bridge using the following command:

$ sudo brctl addif br0 eth0

6) Setup the *wlan0, wlan1* interfaces in the network interface configuration file */etc/network/interfaces.*

Use the following command for the interface file:

$ sudo nano /etc/network/interfaces

Add the following lines at the end of the file:

auto br0
iface br0 inet manual
bridge_ports eth0 wlan0 wlan1

7) Modify the configuration file */etc/hostapd/hostapd.conf* for wlan0 with the desired SSID and PASSWORD.

A simple example of the configuration file is given below:

interface=wlan0
bridge=br0
hw_mode=g
channel=1
ssid=SSID
wpa_passphrase=PASSWORD

8) Add the another one configuration file */etc/hostapd-/hostapd5g.conf* for wlan1 with the desired SSID and PASSWORD.

A simple example of the configuration file is given below:

interface=wlan1
bridge=br0
hw_mode=a
channel=36
ssid=SSID5G
wpa_passphrase=PASSWORD

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

S. C. Roy designed the algorithm, implemented the system, conducted experiments, and wrote the paper; N. Funabiki supervised the research and revised the paper; K. I. Munene and M. M. Rahman conducted experiments; M. Kuribayashi co-supervised the research; all authors had approved the final version.

REFERENCES

[1] M. S. Gast, "802.11 n: A survival guide," *O'Reilly Media*, Inc, 2012.

[2] K. J. Runsera, J. M. Gorceb, and S. Ubedab, "Mono-and multiobjective formulations for the indoor wireless LAN planning problem," *Comput. Oper. Res.*, vol. 35, no. 12, pp. 3885-3901, Dec. 2008.

[3] S. L. Kang, G. Y. Chen, and J. Rogers, "Wireless LAN access point location planning," in *Proc. Inst. Ind. Eng. Asian Conf.*, Springer, Singapore, pp. 907-914, Jul. 2013.

[4] K. I. Munene, N. Funabiki, H. Briantoro, M. M. Rahman, F. Akhter, M. Kuribayashi, and W. C. Kao, "A throughput drop estimation model for concurrent communications under partially overlapping channels without channel bonding and its application to channel assignment in IEEE 802.11n WLAN," *IEICE Trans. Inf. Syst.*, vol. 104, no. 5, pp. 585-596, May 2021.

[5] M. S. A. Mamun, M. E. Islam, and N. Funabiki, "An active access-point configuration algorithm for elastic wireless local-area network system using heterogeneous devices," *Int. J. Netw. Comput.*, vol. 6, no. 2, pp. 395-419, Jul. 2016.

[6] M. S. A. Mamun, N. Funabiki, K. S. Lwin, M. E. Islam, and W. C. Kao, "A channel assignment extension of active access-point configuration algorithm for elastic WLAN system and its implementation using Raspberry Pi," *Int. J. Netw. Comput.*, vol. 7, no. 2, pp. 248-270, Jul. 2017.

[7] M. S. Gast, "802.11ac: A survival guide," *1st ed., O'Reilly*, 2013.

[8] N. Funabiki. Wireless mesh networks. InTech-Open Access Pub. [Online]. Available: https://www.intechopen.com/books/26

[9] "Raspberry Pi," [Online]. Available: http://raspberrypi.org, (Accessed 14 Jun. 2021).
"Archer T4U," [Online]. Available: https://www.tp-link.com/jp/home-networking/adapter/archer-t4u/

[10] B. P. Tewari and S. C. Ghosh, "Efficient AP placement through power control and partially overlapping channel assignment," *Wireless Personal Commun.*, vol. 110, no. 1, pp. 223-244, Jan. 2020.

[11] N. Heiba, M. Alghoniemy, M. Elwekeil, A. Mokhtar, and M. R. M. Rizk, "An adaptive algorithm for channel assignment and load balancing in elastic IEEE 802.11 WLANs," in *Proc. Int. Comp. Eng. Conf. (ICENCO)*, pp. 343-347, Dec. 2017.

[12] H. Tang, L. Yang, J. Dong, Z. Ou, Y. Cu, and J. Wu, "Throughput optimization via association control in wireless LANs," *Mobi. Netw. Appl.*, vo. 21, no. 3, pp. 453-466, Sep. 2016.

[13] A. Farsi, N. Achir, and K. Boussetta, "WLAN planning: Separate and joint optimization of both access point placement and channel assignment," *Ann. Telecommun.*, vol. 70, no. 5, pp. 263-274, Jun. 2015.

[14] M. P. Lima, E. G. Carrano, and R. H. C. Takahashi, "Multiobjective planning of wireless local area networks (WLAN) using genetic algorithms," in *Proc. IEEE Cong. Evol. Comp.*, pp. 1-8, Jun. 2012.

[15] S. H. Fang and T. N. Lin, "A novel access point placement approach for WLAN-based location systems," in *Proc. Wireless Commun. Netw. Conf.*, Sydney, Australia, pp. 1-4, Apr. 2010.

[16] A. Eisenblatter, H. F. Geerdes, and I. Siomina, "Integrated access point placement and channel assignment for wireless LANs in an indoor office environment," in *Proc. Int. Symp. World. Wirel. Mob. Multi. Netw. (WoWMoM)*, pp. 1-10, Jun. 2007.

[17] K. S. Lwin, N. Funabiki, C. Taniguchi, K. K. Zaw, M. S. A. Mamun, M. Kuribayashi, and W. C. Kao, "A minimax approach for access point setup optimization in IEEE 802.11 n wireless networks," *Int. J. Netw. Comput.*, vol. 7, no. 2, pp. 187-207, Jul. 2017.

[18] N. Funabiki, C. Taniguchi, K. S. Lwin, K. K. Zaw, and W. C. Kao, "A parameter optimization tool and its application to throughput estimation model for wireless LAN," in *Proc. Int. Works. Virtual Environ. Net. -Ori. Appli. (VENOA)*, pp. 701-710, Jul. 2017.

[19] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol.2, no. 4, pp. 385-393, Dec. 1982.

[20] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*, Cambridge Univ. Press, Apr. 2011.

[21] "Hostapd," [Online]. Available: https://w1.fi/hostapd/

[22] "Iperf - the TCP, UDP and SCTP network bandwidth measurement tool," [Online]. Available: https://iperf.fr/

[23] Setting up a Raspberry Pi as a wireless access point. [Online]. Available: https://thepi.io/how-to-use-your-raspberry-pi-as-a-wireless-access-point/

Sujan Chandra Roy received the B.S. and the M.S. degrees in information and communication engineering from University of Rajshahi, Bangladesh, in 2016 and 2017, respectively. He is currently a Ph.D. student in Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include wireless communication and computer network. He is a student member of IEICE.

Nobuo Funabiki received the B.S. and the Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, and educational technology. He is a member of IEEE, IEICE, and IPSJ.

Kwenga Ismael Munene received the B.S in information sciences from Moi University in Kenya in 2014 and the master of engineering from Okayama University in 2019. He is currently a Ph.D. student in Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include wireless communication and networking. He is a student member of IEICE.

Md. Mahbubur Rahman received the B.S. and M.S. in information and communication engineering from University of Rajshahi, Bangladesh, in 2010 and 2011, respectively. Currently, he is a Ph.D. student in Graduate School of Natural Science and Technology at Okayama University, Japan. In 2016, he joined the Department of Electrical and Electronic Engineering at Jatiya Kabi Kazi Nazrul Islam University, Bangladesh, as a lecturer and became an assistant professor in 2018. His research interest includes wireless communication and networking. He is a student member of IEICE.

Minoru Kuribayashi received the B.E., M.E., and D.E. degrees from Kobe University, Japan, in 1999, 2001, and 2004, respectively. In 2002, he joined the Department of Electrical and Electronic Engineering, Kobe University, as an assistant professor. Since 2015, he has been an associate professor in Graduate School of Natural Science and Technology, Okayama University. His research interests include digital watermarking, information security, cryptography, and coding theory. He is a senior member of IEEE and IEICE.