Enhancements of Active Access-Point Configuration for IEEE 802.11n 2.4GHz Wireless Local-Area Network

March, 2025

Mousumi Saha

Graduate School of
Natural Science and Technology

(Doctor's Course)
Okayama University

Dissertation submitted to
Graduate School of Natural Science and Technology
of
Okayama University
for
partial fulfillment of the requirements
for the degree of
Doctor of Philosophy.

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by
Professor Satoshi Denno
and
Professor Yasuyuki Nogami

Okayama University, March 2025.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Mousumi Saha

Signature of	Seal of
the Supervisor	
	Graduate School of
Prof. Nobuo Funabiki	Natural Science and Technology

Abstract

Nowadays, the *IEEE 802.11 wireless local-area network (WLAN)* is commonly used as a access medium to the Internet. In a WLAN, hosts can access to the Internet through associations with *access points (APs)* using wireless medium.

The *channel bonding (CB)* is one of the emerging technology to enhance the data transmission speed in *IEEE 802.11n protocol*. The CB for IEEE 802.11n can increase the number of sub-carriers for data transmissions using *Orthogonal Frequency-Division Multiplexing (OFDM)* to 108 from 52 in the $20MHz$ channel. It can also reduce the *guard interval* time and enhance the *frame aggregation*, which can further increase the transmission speed. Therefore, the CB has now become common in $11n$ *WLAN* due to the high throughput performance.

Recently, *Raspberry Pi* has gained popularity worldwide as a low cost, small size, and powerful computing device. It is equipped with an *IEEE 802.11n wireless network interface (NIC)*, allowing it to function as a *software access point (AP)* for *WLAN*. *Raspberry Pi AP* only provides the communication link with the *non channel bonding (non-CB)* channel, because the built-in *NIC* adapter in *Raspberry Pi* does not support the *CB* functionality. Previously, the *CB configuration* for the *Raspberry Pi* AP has been studied using an external NIC adapter, and showed that the CB AP improves the throughput performance for the single AP-host communication from the non-CB AP.

In WLAN, multiple APs are often allocated in a wide network field, which may cause interferences and degrade the performance. On the other hand, a limited number of *partially overlapping channels (POCs)* are available on the commonly used 2.4GHz band. Therefore, to enhance the network performance, it is essential to optimize the network configuration according to traffic demands and network environments which will change dynamically. Previously, the *active AP configuration algorithm* has been studied to optimize the number of active APs and host associations, and the AP joint optimization algorithm to optimize the transmission power, frequency channel, and CB or non-CB POC to each AP.

These previous studies disclosed several drawbacks in the active AP configuration. The first drawback is that in order to design an effective WLAN system, the accurate *throughput estimation model* is essential to estimate the performance of the current configuration. The previous throughput estimation model, estimates the *received signal strength (RSS)* at the host by the *log distance path loss model* and converts the RSS into the throughput by the *sigmoid function*. However, this model only considers the single host communication and needs to be modified to consider the CB and non-CB effect of the multiple hosts concurrent communication to make the accurate model.

The second drawback is that the previous throughput estimation model does not consider the 11 POCs and 13 POCs separately. 11 POCs are available at $2.4GHz$ in many countries, whereas 13 POCs can be used in Japan and Europe. Therefore it is important to consider both of the 11 POCs and 13 POCs effect in the throughput estimation for improving the performance of the network configuration.

The third drawback is that the active AP configuration algorithm and the AP joint optimization algorithm does not consider the *AP location optimization*. The allocation of APs in a network significantly affects its performance. The improper placement of APs can lead to weak signals for some hosts, resulting in the low throughput. Also, selecting the promising APs manually takes a lot of efforts and time, which should be minimized as best as possible. On the other hand, in a real network scenario, multiple hosts are frequently connected to a single AP and the AP joint optimization algorithm only considers a single host association with each AP, which needs to be improved.

In this thesis, to solve the above drawbacks, firstly, I propose the modifications of the throughput estimation model for up to three *Raspberry Pi* APs. Next, examine the measurement results for the concurrent communication of up to three *Raspberry Pi* APs and PC hosts for 11 and 13 POCs using different wireless NIC adapters and the *channel distance ChD*. Then, I propose the modifications of the throughput estimation model based on them. In this modification, the *reduction factor* is introduced to consider the throughput drop by the interference among the concurrently communicating links, which is derived from the use of the CB and the *ChD* at the links.

Next, I propose a modification to the throughput estimation model considering the interferences among APs under different numbers of wall obstacles in the presence of concurrent communications. In the modified estimation model, we consider the interferences by the different number of walls between *APs* placed in different rooms within the network. We introduce a *throughput increasing factor* based on the number of average walls, n_{WA}, and the minimum number of walls, n_{Wmin} among *APs* in the network fields. The correctness of the modified throughput estimation model is confirmed through simulations and experiments.

Secondly, in this thesis, I propose the *network configuration optimization algorithm* for 802.11n WLAN with three *Raspberry Pi* APs, by utilizing the modified throughput estimation model. For a given network field, this algorithm selects the CB/non-CB use, the channel assignment, and the associated hosts for each AP, to maximize the throughput performance. The effectiveness of the proposed algorithm has been verified through simulations using the *WIMNET* simulator and testbed experiments in different network topologies.

Thirdly, in this thesis, I propose the *preprocessing stage* for the AP configuration algorithm to reduce the CPU time by limiting the number of APs in advance. Both the exhaustive and heuristic approaches are adopted, where either one should be used depending on the network topology. The effectiveness of the proposal is demonstrated through extensive simulations using the *WIMNET* simulator.

Lastly, in this thesis, I propose the application of the preprocessing stage of the active AP configuration algorithm in the AP joint optimization algorithm to select the promising candidate APs. The pre-processing stage identifies the promising APs locations to reduce the search space. Then, we propose the post-processing stage which refines AP-host associations to further improve the network performance and consider the use of both stages in the AP joint optimization algorithm. The effectiveness of the proposal is verified through simulations using the *WIMNET* simulator in three network scenarios.

In future studies, I will consider further enhancements of the active AP configuration of the elastic WLAN system and their evaluations in various network fields.

To My Daughter and Son

Acknowledgements

This dissertation becomes possible with an enormous amount of support I received during this study. Here, I would like to give my highest appreciation to those who have supported me by giving their valuable time and effort. I am nothing but grateful to receive such blessings.

First, I owe a debt of gratitude to my honorable supervisor, Professor Nobuo Funabiki. His excellent supervision, meaningful suggestions, and persistent encouragements were not only helping my PhD research but also enlightened me to make my life meaningful. Only with his strongest support, I was able to motivate myself to learn a lot of new things, focus on my goal, and achieve it. He taught me how to face the problems, how to analyze them, and how to solve them. The way he manages my study is what I admire the most. His sincerity in guiding all of his students pushed me to publish papers and present them in more intuitive ways. I look up to him as a teacher in many ways, and he is one of the most significant figures in my life.

I am indebted to my Ph.D. co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for taking their valuable time to give me advices, guidance, insightful comments, and proofreading of this thesis.

I want to express my profound gratitude to Professor Minoru Kuribayashi in Tohoku University for his valuable discussions during my research. I would like to thank all my course teachers during my Ph.D. study for sharing their great ideas and knowledge with me. I would also like to give special thanks to Prof. Wen-Chung Kao, Dr. Pradini Puspitaningayu, and Mr. Bin Wu for their advice and support, especially in proofing my papers before submissions.

I would like to convey my appreciations to all the members of FUNABIKI Lab. Especially, I would like to give special thanks to Dr. Nobuya Ishihara, Dr. Rahardhita Widyatra Sudibyo, Dr. Kwenga Ismael Munene, Dr. Khin Khin Zaw, Dr. Kyaw Soe Lwin, Safira Adine Kinari, Lynn Htet Aung, and all members in general. Though it is difficult to mention every name, none of the enjoyable moments we shared would ever slip from my memory. Thank you so much for your great supports and helpfulness in this study.

I especially want to thank my beloved daughter and son who always encourage me throughout my life. Your unconditional love gives me the strength and inspiration to overcome any difficulty in my life.

Last but not least, my utmost gratitude to my beloved parents, and sisters for their unending and unconditional love. You are the reason I am here, and the reason I am living my life.

Mousumi Saha
Okayama University, Japan
March, 2025

List of Publications

Journal Papers

1. **M. Saha**, N. Funabiki, R. W. Sudibyo, S. K. Debnath, M. M. Islam, M. Kuribayashi, and W. -C. Kao, "A network configuration optimization algorithm for wireless local-area network with three Raspberry Pi access-points under concurrent communications," International Journal of Future Computer and Communication, vol. 8, no. 2, pp. 55-62, June 2019.

2. **M. Saha**, R. W. Sudibyo, N. Funabiki, M. Kuribayashi, S. K. Debnath, M. M.Islam, and W. -C. Kao, "Throughput measurement and estimation model for concurrent communications of multiple Raspberry Pi access-points in wireless local-area network," Journal of Communications, vol. 14, no. 11, pp. 1009-1016, November 2019.

3. **M. Saha**, N. Funabiki, B. Wu, and P. Puspitaningayu, "Two extensions of active access-point joint optimization algorithm for wireless local-area network," (conditionally accepted, IAENG International Journal of Computer Science).

International Conference Papers

4. **M. Saha**, N. Funabiki, S. K. Debnath, and W. -C. Kao, "A study of MIMO host location optimization in active access-point configuration algorithm for elastic WLAN system," IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 275-276, May 2018.

5. **M. Saha**, R. W. Sudibyo, N. Funabiki, M. Kuribayashi, S. K. Debnath, M. M. Islam, and W. -C. Kao, "Throughput measurement and estimation model for concurrent communications of multiple Raspberry Pi access-points in wireless local-area network," The 4ht International Conference on Computer and Communication Systems (ICCCS), February 23-25, 2019.

6. **M. Saha**, N. Funabiki, R. W. Sudibyo, M. Kuribayashi, and W. -C. Kao, "A network configuration optimization algorithm for wireless local-area network with three Raspberry Pi access-points under concurrent communications," The 11th International Conference on Future Computer and Communication (ICFCC 2019), February 27-March 1, 2019.

7. **M. Saha**, N. Funabiki, S. K. Debnath, W.-C. Kao, S. Tajima, and T. Higashino, "A proposal of preprocessing stage for active access point configuration algorithm in elastic WLAN system," ICIET, pp. 284-289, March 28–30, 2020.

Other Papers

8. **M. Saha**, N. Funabiki, M. S. A. Mamun, S. K. Debnath, "A minimax approach for access-point allocation in wireless local-area networks," Chugoku-Branch Joint Conv. Inst. Elec. Info. Eng., Oct. 2017.

9. **M. Saha**, N. Funabiki, S. K. Debnath, and K. S. Lwin, "Modification of MIMO host location optimization for active access-point configuration algorithm in elastic WLAN system," IEICE General Conference, March 2018.

10. **M. Saha**, R. W. Sudibyo, N. Funabiki, and M. Kuribayashi, "Modifications of throughput estimation model for concurrent communications of multiple raspberry Pi access-points in wireless local-area network," IEICE Technical Report, SRW2018-9, pp. 1-6, Aug. 2018.

11. **M. Saha**, N. Funabiki, R. W. Sudibyo, M. Kuribayashi, and W. -C. Kao, "A proposal of network configuration optimization algorithm for wireless local-area network with three Raspberry Pi access-points under concurrent communications," IEICE Technical Report, NS2018-176, pp.101-106, Dec. 2018.

12. **M. Saha**, N. Funabiki, S. K. Debnath, and M. S. A. Mamun, "A proposal of preprocessing stage for active access-point configuration algorithm in elastic WLAN system," The 21st IEEE Hiroshima Section Student Symposium, pp. 100 - 103, November 30-December 1, 2019.

List of Figures

List of Tables

List of Abbreviations

WLAN wireless local area network

AP access point

ISM Industrial Scientific and Medica

MAC media access control

RF radio frequency

NIC network interface card

BSS basic service set

IBSS independent basic service set

ESS extended service set

MIMO multiple input multiple output

non-CB non channel bonding

CB channel bonding

OC orthogonal channel

POC partially overlapping channel

RSS^i interfered received signal strength

PC personal computer

chD channel distance

phD physical distance

lkD link distance

tpD throughput drop

DAP dedicated access point

VAP virtual access point

MAP mobile access point

List of Notations

E_1 number of active APs

E_2 minimum average host throughput

E_3 total interfered communication time

tp_{ij} link speed between AP_i and $host_j$

tpD_{nec} throughput drop for *NEC AP*

tpD_{pi} throughput drop for *Raspberry Pi AP*

tpM^{nec} maximum throughput for *NEC AP*

tpM^{pi} maximum throughput for *Raspberry Pi AP*

$Srf(m)$ contention factor at AP_i among the associated hosts to send data by the CSMA/CA

Contents

Chapter 1

Introduction

1.1 Background

Wireless Local Area Network (WLAN) has become popular due to easy installation, low costs, and flexibility. *WLANs* have been deployed worldwide to provide the Internet access in various places, such as companies, educational institutes, airports, shopping malls, hotels, stations, and *IoT* sector [1–3]. Various communication services for IoT, such as smart home networks, smart city networks, smart grid systems, autonomous driving systems, and smart healthcare systems adopt *IEEE 802.11 WLAN* as the common communication technology [4–8].

Raspberry Pi is a small yet powerful computing device capable of effectively performing various tasks [9]. It is cost-effective, energy-efficient, and portable, enabling users to browse the Internet, send emails, and run versatile applications such as home automation systems, zero-powered smartphones, *AI* assistants, motion capture security cameras, and live bots [10][11][12]. *Raspberry Pi* functions as a software *access point (AP)* for *IEEE802.11n WLAN* using *hostapd* [13], as it utilizes a wireless *NIC* (network interface card) for *11n* and *Linux OS*. However, the software *AP* cannot achieve the *CB*, which involves using two independent *20MHz* channels together to create one communication link between an AP and a host. Hardware channel bonding uses a *40MHz* bandwidth for one channel by combining two adjacent *20MHz* channels. Previously, the *CB* configuration for the *Raspberry Pi AP* has been examined using an external *NIC* adapter and investigated the throughput performance and the estimation model for single-link communications. The measurement results demonstrate that the *CB* is superior to the *non-CB* [14].

IEEE 802.11 WLAN operates in two unlicensed frequency bands: 2.4*GHz* and 5*GHz*[15, 16]. The 2.4*GHz* band is typically used in indoor environments due to its wide coverage range and stronger penetration capability [17]. The *IEEE 802.11* standard provides a limited number of frequency channels for communications. In the *2.4 GHz* band, the spectrum of adjacent channels overlaps partially, which is called the *partially overlapping channels (POCs)* [18,19]. In Japan and Europe, 13 *POCs* are available at *2.4GHz*, while in other countries, 11 *POCs* are available. Due to this limited number of available channels, densely located *APs* can interfere with each other, causing network performance degradation due to overlapping frequency signals [20].

The *IEEE 802.11n* and the later standards support the *channel bonding (CB)* by combining two neighboring *20 MHz* channels to create one *40 MHz* channel, which increases the transmission capacity. In [21], *CB* increases the number of sub-carriers for data transmissions using *Orthogonal Frequency Division Multiplexing (OFDM)* to 108 from 52 in the *conventional non-CB channel*. The *OFDM* utilizes multiple narrow-band carriers to transfer data for the higher throughput in this *wide-band CB*. However, the use of *CB* reduces the number of non-interfered channels and can cause further interferences in *dense WLANs*.

The throughput performance of a *WLAN* is influenced by several factors in indoor environments. In real-world scenarios, multiple links communicating simultaneously within the interference range often lead to a decline of the *WLAN* performance. However, the performance of the *WLAN* in the presence of interferences can be improved by properly configuring the *WLAN* network. An accurate *throughput estimation model* is essential for designing an efficient *WLAN* system.

To maximize the throughput performance by reducing interferences among wireless signals, previously, the *orthogonal channel assignment (OC)* assignment under *20 MHz non-CB* in the *active AP configuration algorithm* has been studied for the *elastic WLAN system* [22, 23]. Figure 1.1 shows a simple topology of the elastic WLAN system. The system applies the active AP configuration algorithm to dynamically control the number of active APs according to throughput demands and device conditions, while providing the necessary minimum throughput to every host. The elastic WLAN system controls the number of active APs by deactivating the unnecessary APs. Hence, it can reduce the interference and power consumption. Each active AP is assigned an orthogonal non-CB channel in a way to minimize the overall interference in the network.

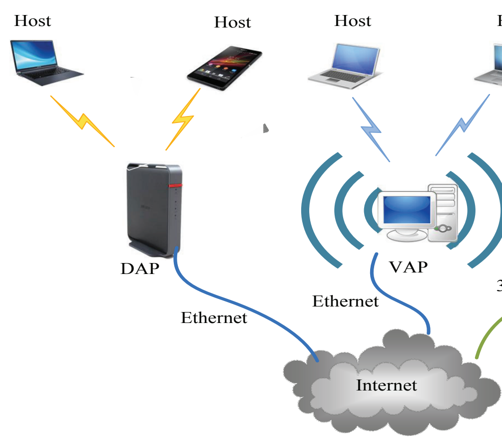


Figure 1.1: Overview of Elastic WLAN.

A *throughput estimation model* has been studied, which estimates the *received signal strength (RSS)* at the host using the *log distance path loss model* and converts it to the throughput using the *sigmoid function* [24]. Unfortunately, this model only considers the *single link communication*, where the concurrent communication of multiple APs is common practically.

In addition, a *throughput drop estimation model* has been studied to estimate the link throughput between an AP and a host under the coexistences of CB and non-CB channels for *concurrently communicating links* in a WLAN. Furthermore, the *AP joint optimization algorithm* has been studied to optimizes the transmission power, the frequency channel, and the channel bonding (CB) or non-CB to each AP in *IEEE 802.11n WLAN* using the *throughput drop estimation model* [25].

However, in the previous studies of the *throughput estimation model*, the *active AP configuration algorithm*, and the *AP joint optimization algorithm*, the following drawbacks need to be solved to reduce the interference for enhancing the throughput performance:

- Firstly, for network configuration optimization, accurate throughput estimation is very important. In a practical scenario, the *concurrent communication* of multiple hosts and multiple APs are common. The *concurrent communication* of multiple APs within the interference range provides different throughput features from the single communication, depending on the adopted NIC adapters and the assigned *partially overlapping channels (POCs).* During the optimization process, the throughput must be estimated accurately for a given configuration when hosts are concurrently communicating with their associated APs in the network field.

- Secondly, to improve the network performance, the optimization of network configuration according to the traffic demands and the network situation in the network field is very essential. In the network configuration, the proper assignment of channels, CB condition, and host associations should be considered together to enhance the throughput performance.

- Thirdly, it has been observed that the allocation of APs in a network field affects its performance. The improper placement of APs can lead to weak signals for some hosts, resulting in low throughput. Also, selecting the promising APs manually takes a lot of effort and time, which needs to be minimized as best as possible.

- Finally, in a real network scenario, multiple hosts are frequently connected to a single AP, and in network optimization, this phenomenon needs to be considered to enhance the performance of network configuration.

1.2 Contributions

In this thesis, I have carried out the following research contributions.

As the first contribution, I propose the *throughput estimation model* [24] for concurrently communicating multiple links under various conditions [26, 27]. Here, both cases are considered: 1) CB, and 2) non-CB under concurrent communications of multiple hosts with upto three APs. First, present the throughput measurement results in the concurrent communication of up to three *Raspberry Pi* APs and PC hosts for 11 POCs. To examine various conditions, I adopt different wireless NIC adapters and change the *channel distance (ChD)* among the APs. Then, modify the existing *throughput estimation model* based on the measurement results [26]. The *reduction factor* is introduced to consider the *throughput drop* due to the interference among the concurrently communicating links, which is derived from the use/non-use of the CB and the *ChD* at the links. Then, I extend these works to 13 POCs to confirm the effectiveness in the wider spectrum for WLAN [27]. Two non-interfered channels for the CB are available theoretically for 13 POCs, which has a potential of greatly increasing the transmission capacity of WLAN. The comparisons of the estimated throughput and the measured one support the correctness of the modification of our throughput estimation model.

Furthermore, I propose the model modification considering the interference effects among APs under different numbers of wall obstacles in the presence of concurrent communications. First present the throughput measurement results in the concurrent communications of up to three *Raspberry Pi* APs and *PC* hosts using 13 *POCs*. In the measurement, I consider multi-room topology where three *APs* are positioned in different rooms at varying physical distances from each other. Then, I found that the total throughput gradually improves as the number of wall obstacles or the physical distance between *APs* increases when three *APs* are placed in different rooms in the network field. Based on these measurement results, I have modified the existing throughput estimation model. In the modified estimation model, we introduce a *throughput increasing factor* based on the number of average walls, n_{WA}, and the minimum number of walls, n_{Wmin} among *APs* in the network fields. The correctness of the modified throughput estimation model is confirmed through simulations and experiments.

As the second contribution, I propose the *network configuration optimization algorithm* for *IEEE 802.11n WLAN* with three *Raspberry Pi* APs, by utilizing the modified throughput estimation model [28, 29]. For a given network field, it selects the CB or non-CB, channel assignment, and associated hosts for each AP, to maximize the throughput performance. The effectiveness of the proposed network configuration optimization algorithm is evaluated through simulations using the *WIMNET simulator* [30] and testbed experiments in different network topologies.

As the third contribution, I propose a *preprocessing stage* for the active AP configuration algorithm to reduce the CPU time by confining the search space by selecting promising candidates for active APs [31]. Both the exhaustive and heuristic approaches are adopted for this stage, where the simulation results found that the better approach among them in terms of the CPU time and the network performance is different depending on the network topology. Thus, the user will need to select either one for each network instance. In both approaches, first, the minimum number of active APs L to satisfy the designed throughput is estimated. Next, the number of candidate APs K is estimated from L. Last, K APs are selected as the promising candidates for active APs by the exhaustive or heuristic approach. This output of the preprocessing stage becomes the input to the active AP configuration algorithm. The effectiveness of the proposal is evaluated through simulations in three network topologies using the WIMNET simulator [30].

Lastly, I study the application of the preprocessing stage of the active AP configuration algorithm and propose two extensions to the *AP* joint optimization algorithm to improve the performance of *WLAN* networks. In the first extension, I adopt the *pre-processing stage* of the *active AP configuration algorithm* in the *AP joint optimization algorithm* to select the promising candidate *APs*. This stage not only improves the network performance but also reduces the search space of the network field. In the second extension, I propose the *post-processing stage* of the algorithm, which refines *AP-host* associations to further improve the overall network performance. The effectiveness is evaluated through simulations in three network topologies using the *WIMNET simulator* [30].

1.3 Thesis Outline

The remaining part of this thesis is organized as follows.

In Chapter 2, I review IEEE 802.11 wireless network technologies related to this thesis, including the IEEE 802.11n protocols, features of IEEE 802.11n protocols, and software tools in the Linux operating system.

In Chapter 3, I review our previous studies related to this thesis.

In Chapter 4, I describe the throughput measurements and estimation model for up to three concurrently communicating Raspberry Pi APs links and their evaluations.

In Chapter 5, I describe the proposed application of throughput estimation model to network configuration optimization algorithm of transmission power, frequency channel, and channel bonding assignment in WLAN.

In Chapter 6, I describe the proposed preprocessing stage for active access-point configuration algorithm and its applications to joint optimization algorithm with a modification of active access-point configuration algorithm.

In Chapter 7, I review relevant works in literature.

Finally, in Chapter 8, I conclude this thesis with some future works.

Chapter 2

Background Technologies

This chapter introduces background technologies for this thesis. First, I give an overview of *IEEE 802.11 WLAN* including advantages, components, types and standards. Then, discuss the *IEEE 802.11n protocol* and its key features. Finally, I outline some Linux tools and commands for WLANs that are used for measurements, and the implementation of elastic WLAN system.

2.1 802.11 WLAN Overview

IEEE 802.11 standards define *physical (PHY)* and *media access control (MAC)* layer specifications for implementing high-speed *wireless local area network (WLAN)* technologies. WLAN is an extension to a wired LAN that enables the user mobility by the wireless connectivity and supports the flexibility in data communications [1]. It can reduce the cabling costs in the home or office environments by sending and receiving data over the air using *radio frequency (RF)* technology. Therefore, WLANs are adopted in many places such as home, school, campus, and offices.

2.1.1 Advantages of WLAN

WLAN provides several benefits over the traditional wired networks in the following [1]:

- *User mobility:*
  Wireless networking allows mobility than wired networking. In wired networking, users need to use wired lines to stay connected to the network. WLAN gives users the ability to move around within a local coverage area and still be connected to the network.

- *Easy and rapid deployment:*
  WLAN can eliminate the requirement of network cables between hosts and connection hubs or APs. Thus, the installation of WLAN can be much easier and quicker than the wired LAN.

- *Cost:*
  The cost of installing and maintaining a traditional wired LAN is normally higher than installing and maintaining WLAN. WLAN reduces the cost of cabling and the works related to installation and reparation. Because WLAN simplifies moving, adding, and changing, the indirect cost of user downtime and administration are decreased.

- *Increased flexibility:*
  WLAN installation eliminates the need to pull cable through walls and ceilings. The network coverage area of WLAN can be easily expanded because the network medium is everywhere.

- *Scalability:*
  WLAN can be configured for a variety of topologies suitable to applications. WLAN can support both peer-to-peer networks suitable for a small number of users and full infrastructure networks of thousands of users. New access points can be added easily to expand the coverage area.

2.1.2 IEEE 802.11 WLAN Components

IEEE 802.11 WLAN consists of four primary components as shown in Figure 2.1 [1]:

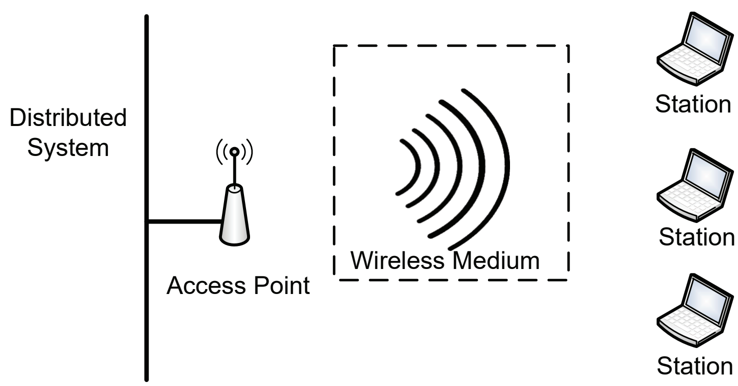


Figure 2.1: Components of IEEE 802.11 WLANs.

- *Stations or hosts:*
  WLAN transfers the data between *stations or hosts*. A *station* in WLAN indicates an electronic device such as a desktop/laptop PC, a smartphone, or a tablet that has the capability of accessing the network over the wireless *network interface card (NIC)*.

- *Access points (APs):*
  An AP acts as the generic base station for WLAN that performs the similar role as a hub/switch in a wired Ethernet LAN. It also provides the bridging function between the wireless and the wired networks with some other tasks.

- *Wireless medium:*
  The IEEE 802.11 standard uses the wireless medium to convey the information/ data from one host to another host in a network.

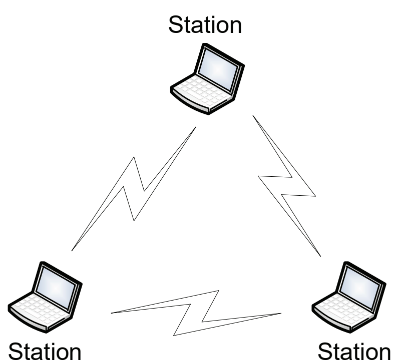
- *Distribution system:*
  When several APs are connected together to make the large coverage area, they must communicate with each other to trace the movements of the hosts. The distribution system is the logical component of WLAN which serves as the backbone connections among APs. It is often referred to as the *backbone network* used to relay data frames between APs. In most cases, *Ethernet* is commonly used as the backbone network technology.

2.1.3 Types of WLANs

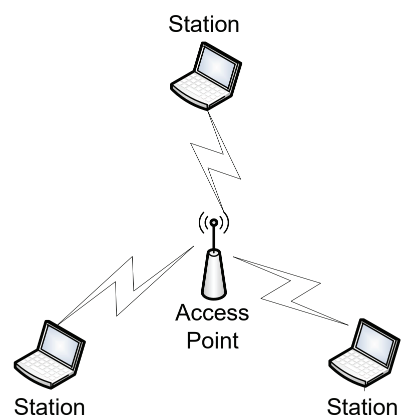
The basic unit of *IEEE 802.11* WLAN is simply a set of hosts that can communicate with each other known as the *basic service set (BSS)*. Based on the types of BSS, the IEEE 802.11 standards support two types of WLAN as illustrated in Figure 2.2.

- *Independent or ad hoc type:*
  In this type, a collection of stations or hosts can send frames directly to each other without an AP. It is also called as an *independent BSS (IBSS)* as shown in Figure 2.2(a). This *ad hoc network* is rarely used for practical networks due to the lack of required performances and security issues.

- *Infrastructure type:*
  In this type, the stations exchange the data through an AP as shown in Figure 2.2(b). A single AP acts as the main controller to all the hosts within its BSS, known as *infrastructure BSS*. In this type, a host must be associated with an AP to obtain network services [32].



(a) Adhoc network.



(b) Infrastructure network.

Figure 2.2: Types of IEEE 802.11 WLAN networks.

To extend WLAN further, multiple BSSes can be connected together with a backbone network to form *extended service set (ESS)* as shown in Figure 2.3.

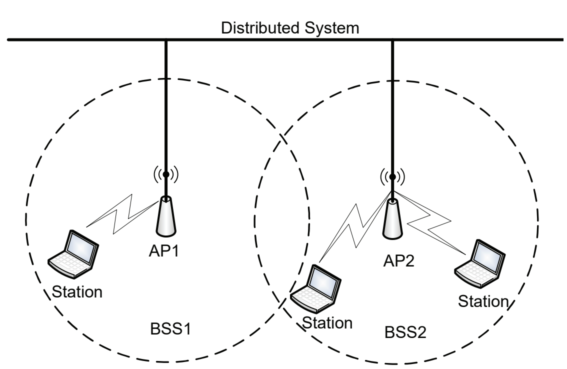


Figure 2.3: Extended service set (ESS).

ESS can form a large size WLAN. Each AP in ESS is given an ID called the *service set identifier (SSID)*, which serves as the “network name” for the users. All the hosts within the same ESS can mutually communicate with each other, even if they are in different basic service areas.

2.1.4 Channel Access Modes in IEEE 802.11 MAC Standard

The IEEE 802.11 MAC standard includes two basic channel access modes: *distributed coordination function (DCF)* and *point coordination function (PCF)*. DCF is the contention-based fundamental MAC mechanism of the IEEE 802.11 WLAN which employs the *CSMA/CA protocol*, and an ordinary *binary exponential backoff (BEB)* algorithm. For contention-free services, PCF is an optionally used. PCF is the polling based protocol limited to the infrastructure based network.

DCF is broadly implemented in most commercial products due to its simplicity and robustness. Therefore, most researchers have given their attention on DCF. Two types of *carrier sensing (CS)* is used in IEEE 802.11 to check if the medium is idle or not. They are the physical carrier sensing at the physical layer and the virtual carrier sensing at the MAC layer. The virtual carrier sensing is provided by *Network Allocation Vector (NAV)*. The station sets NAV to the time where it needs to reserve the medium for its data transmissions [1].

In the IEEE 802.11 MAC protocol, the *inter-frame spacing (IFS)* is used for coordinating the access to the wireless medium. Three basic inter-frame spaces are used: *short IFS (SIFS)*, *point coordination function IFS (PIFS)*, and *distributed coordination function IFS (DIFS)*. The time duration of them is: SIFS < PIFS < DIFS and the relationship of different inter-frame spaces is shown in Figure 2.4.

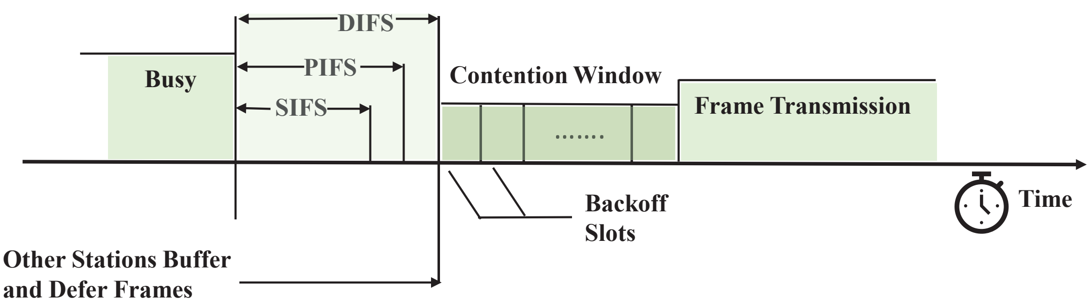


Figure 2.4: Inter-frame spacing relationships.

2.1.5 DCF Operations

DCF has two basic transmission modes to enable the random access to the wireless channel. One is the basic access method using two-way handshakes (DATA + ACK), and another one is the four-way handshakes optionally used by exchanging RTS/CTS messages to reserve the channel before each data transmission.

2.1.5.1 DCF Access Mechanism

In the DCF mechanism, a station wishing to transmit packets first listens to the channel status for more than the DIFS interval. If the channel remains free, this station (sender) sends the packet to the destination (receiver) immediately. The receiver responds to the acknowledgment (ACK) frame after the SIFS interval. If the channel is busy or the idle time is less than DIFS, the sender defers the transmission and uses the binary exponential backoff (BEB) mechanism to avoid the collisions among the contending nodes while providing their fairness. The length of the random backoff time is determined by:

$$T_{bo} = rand(CW) \times T_{slot} \tag{2.1}$$

where T_{bo} represents the backoff period, T_{slot} does the time slot, and CW is the contention window or backoff window in terms of time slots. The contention window is divided into time slots whose length depends on the medium. Stations are activated at the beginning of each slot [1].

2.1.5.2 Binary Exponential Backoff Mechanism

The binary exponential backoff (BEB) algorithm is used in the IEEE 802.11 DCF to schedule retransmissions after collisions in order to avoid collisions again. In the DCF, any station must carry out carrier sensing at the channel for the interval larger than DIFS before attempting any data transmission. If the channel is found busy, the station defers its transmission until the medium is idle for DIFS. Then, the station generates a backoff time with a random value that is uniformly selected from the range [0, CW -1], where CW is the current window size. At the first transmission attempt, CW is initially set to CW_{min}. The backoff timer is decreased each time when the channel is sensed idle, or is frozen when the channel seems busy due to other nodes transmissions, and is restarted again when the channel is sensed free again for more than DIFS. When the backoff timer reaches zero, the station initiates its transmission of the next time slot. If the packet is received successfully, the receiver sends back ACK after the SIFS interval and resets its CW to CW_{min}.

If the sender fails to get ACK within the specified period, the retransmission will be scheduled. It doubles the CW size, under the maximum value of $CW_{max}= 2^m \times CW_{min}$, where m is the maximum backoff stage. Then, it chooses a new random backoff timer, and starts the above process again. When the transmission of a packet fails for a maximum number of times, this packet is dropped.

It is clear that the size of CW increases rapidly with the number of retransmission attempts, and reduces the collision probability. The main reason for the low throughput performance and the poor channel utilization of the IEEE 802.11 MAC protocol is the retransmission. The MAC layer congestion occurs due to the high packet error rate, the limited bandwidth, the large contention, and the long backoff time. When some stations suffer from higher packet loss rates due to higher errors, they will turn to the backoff state more frequently than stations with low error rates, which leads to the backoff window larger for them. On the other hands, stations with lower error rates can get more transmission chances, causing the unfairness problem in WLAN [33].

The BEB mechanism performs well with a small number of active stations. However, when the number of stations increases beyond a certain limit, its throughput performance decreases desperately as shown in Figure 2.5. In this case, the collision rate increases with the number of active stations, and the throughput drops significantly [34].

2.1.6 IEEE 802.11 Standards for WLAN

The IEEE 802.11 working group has improved the existing PHY and MAC layer specifications to realize WLAN at the 2.4-2.5 GHz, 3.6 GHz and 5.725-5.825 GHz unlicensed ISM *(Industrial, Scientific and Medical)* frequency bands defined by the ITU-R. In this working group, several types of IEEE Standard Association Standards are available, where each of them comes with a

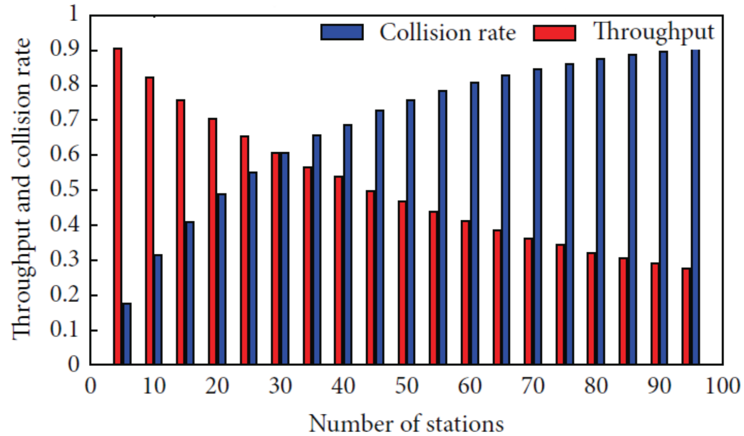


Figure 2.5: Throughput and collision rate of BEB algorithm.

letter suffix, covers from wireless standards, to standards for security aspects, Quality of Service (QoS) and others, shown in Table 2.1 [32, 35–39].

Table 2.1: IEEE 802.11 Standards.

Standard	**Purpose**
802.11a	Wireless network bearer operating in the 5 GHz ISM band, data rate up to 54*Mbps*
802.11b	Operate in the 2.4 GHz ISM band, data rates up to 11*Mbps*
802.11c	Covers bridge operation that links to LANs with a similar or identical MAC protocol
802.11d	Support for additional regulatory differences in various countries
802.11e	QoS and prioritization, an enhancement to the 802.11a and 802.11b WLAN specifications
802.11f	Inter-Access Point Protocol for handover, this standard was withdrawn
802.11g	Operate in 2.4 GHz ISM band, data rates up to 54*Mbps*
802.11h	Dynamic Frequency Selection (DFS) and Transmit Power Control (TPC)
802.11i	Authentication and encryption
802.11j	Standard of WLAN operation in the 4.9 to 5 GHz band to conform to the Japan's rules
802.11k	Measurement reporting and management of the air interface between several APs
802.11l	Reserved standard, to avoid confusion
802.11m	Provides a unified view of the 802.11 base standard through continuous monitoring, management and maintenance
802.11n	Operate in the 2.4 and 5 GHz ISM bands, data rates up to 600*Mbps*
802.11o	Reserved standard, to avoid confusion
802.11p	To provide for wireless access in vehicular environments (WAVE)
802.11r	Fast BSS Transition, supports VoWiFi handoff between access points to enable VoIP roaming on a WiFi network with 802.1X authentication
802.11s	Wireless mesh networking
802.11t	Wireless Performance Prediction (WPP), this standard was cancelled
802.11u	Improvements related to ”hotspots” and 3rd party authorization of clients
802.11v	To enable configuring clients while they are connected to the network
802.11w	Protected Management Frames

Table 2.1: IEEE 802.11 Standards.

Standard	Purpose
802.11x	Reserved standard, to avoid confusion
802.11y	Introduction of the new frequency band, 3.65-3.7GHz in US besides 2.4 and 5 GHz
802.11z	Extensions for Direct Link Setup (DLS)
802.11aa	Specifies enhancements to the IEEE 802.11 MAC for robust audio video (AV) streaming
802.11ac	Wireless network bearer operating below 6 GHz to provide data rates of at least $1Gbps$ for multi-station operation and $500Mbps$ on a single link
802.11ad	Wireless Gigabit Alliance (WiGig), providing very high throughput at frequencies up to 60GHz
802.11ae	Prioritization of management frames
802.11af	WiFi in TV spectrum white spaces (often called White-Fi)
802.11ah	WiFi uses unlicensed spectrum below 1GHz, smart metering
802.11ai	Fast initial link setup (FILS)
802.11aj	Operation in the Chinese Milli-Meter Wave (CMMW) frequency bands
802.11ak	General links
802.11aq	Pre-association discovery
802.11ax	High efficiency WLAN, providing 4x the throughput of 802.11ac
802.11ay	Enhancements for Ultra High Throughput in and around the 60GHz Band
802.11az	Next generation positioning
802.11mc	Maintenance of the IEEE 802.11m standard

Figure 2.6 demonstrates the current and future WiFi standards. Among them, the common and popular ones are IEEE 802.11a, 11b, 11g, 11n, 11ac, and the latest is 11ax. For the physical layer, the 11a/n/ac use *Orthogonal Frequency Division Multiplexing (OFDM)* modulation scheme while the 11b uses the *Direct Sequence Spread Spectrum (DSSS)* technology. 11g supports both technologies. The latest 11ax uses *Orthogonal Frequency Division Multiple Access (OFDMA)* that is a multi-user version of OFDM. Table 2.2 summarizes the features of these common WiFi standards [1, 40–42].

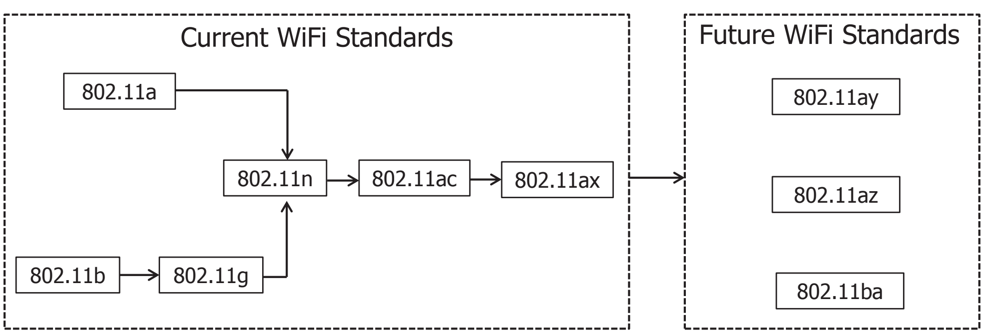


Figure 2.6: Current and future WiFi Standards.

The IEEE 802.11a, b, and g are considered to have medium security because they use the *wired equivalent privacy (WEP)* security mechanism. The WEP encryption uses the RC4 symmetric

Table 2.2: Characteristics of common IEEE802.11 standards.

	IEEE 802.11b	IEEE 802.11a	IEEE 802.11g	IEEE 802.11n	IEEE 802.11ac	IEEE 802.11ax
Release	Sep 1999	Sep 1999	Jun 2003	Oct 2009	Dec 2013	Feb 2021
Frequency Band	2.4 GHz	5 GHz	2.4 GHz	2.4/5 GHz	5 GHz	2.4/5/6 GHz
Max. Data Rate	11 Mbps	54 Mbps	54 Mbps	600 Mbps	1300 Mbps	9608 Mbps
Modulation	CCK[1] modulated with PSK	OFDM	DSSS[2], CCK, OFDM	OFDM	OFDM	OFDMA
Channel Width	20 MHz	20 MHz	20 MHz	20/40 MHz	20/40/80/160 MHz	20/40/80/160 MHz
# of Antennas	1	1	1	4	8	8
security	Medium	Medium	Medium	High	High	High

[1] CCK: Complementary Code Keying
[2] DSSS: Direct Sequence Spread Spectrum

stream cipher with 40-bit and 104-bit encryption keys. The WEP has the following weaknesses; *1) short initializer vectors (IVs) and keys, 2) authentication messages can be easily forged, 3) IV reuse problem which makes stream ciphers vulnerable to analysis, 4) use of cryptographically insecure cyclic redundancy check (CRC) for integrity check, and 5) lack of key-management protocol* [43].

On the other hand, the IEEE 802.11n, ac and ax are considered to have high security because they use the more advanced WPA encryption technology called *temporal key integrity protocol (TKIP)* with message integrity check (MIC). WPA also provides a scheme of mutual authentication using either IEEE 802.1X/extensible authentication protocol (EAP) or pre-shared key (PSK) technology.

- *IEEE 802.11b:* IEEE 802.11b operates at 2.4 GHz band with the maximum data rate up to 11 Mbps. 11b is considered to be robust and has a capacity to compensate the same IEEE 802.11 protocols. Because of the interoperability feature between the products from different vendors, this standard has not only boosted the manufacturing of the products but also motivated the competitions between WLAN vendors. The limitation of this standard is the interference among the products using *industrial, scientific and medical* (ISM) band that uses the same 2.4 GHz band of frequency [44, 45].

- *IEEE 802.11a:* IEEE 802.11a operates at 5 GHz ISM band. It adopts on orthogonal frequency division multiplexing (OFDM) coding scheme that offers a high data rates up to 6, 12, 24, 54 Mbps, and sometimes beyond this speed in comparison to 11b. Two main limitations of 11a are the compatibility issue of the 11a products with the 11b products and the unavailability of 5 GHz band with free of costs in some countries in the world [44, 45].

- *IEEE 802.11g:* IEEE suggested 11g standard over 11a to improve the 2.4 GHz 11b technology. 11g introduces two different modulation techniques including the *packet binary* *convolution code (PBCC)* that supports the data rate up to 33 Mbps and the *orthogonal frequency division multiplexing (OFDM)* that supports up to 54 Mbps data rate. Compatibility issues are also resolved in 11g products with 11b products [44, 45].

- *IEEE 802.11n:* The primary purpose of initiating the 11n standard to improve the usable range and the data rate up to 600 Mbps. 11n supports both of the 2.4 GHz and 5 GHz ISM band *unlicensed national information infrastructure* (UNII) band, and is backward compatible with earlier standards. It introduces new technology features including the use of *channel bonding* and *multiple antennas* to get the better reception of the RF signals to enhance the throughput and coverage range [16, 44].

- *IEEE 802.11ac:* The aim of the 11ac standard to improve the individual link performance and the total network throughput to more than $1Gbps$. Many of the specifications like static and dynamic channel bonding and simultaneous data streams of 11n have been kept and further enhanced for 11ac to reach the gigabit transmission rate. It supports static and dynamic channel bonding up to $160MHz$ and *Multi-User Multiple-Input-Multiple-Output (MU-MIMO)*. 11ac operates only on the $5GHz$ band [46–48].

- *IEEE 802.11ax: IEEE 802.11ax* standard was approved on February 2021, which operates in the frequency bands between $1GHz$ and $7.125GHz$. This standard focuses on enhancing the throughput-per-area or the ratio between the total network throughput and the network area size. The maximum data rate of this standard is $9.6Gbps$. It also adopts channel bonding up $160MHz$. *IEEE 802.11ax* supports *orthogonal frequency division multiple access (OFDMA)* approach that is commonly applied in cellular networks [49–53]. However, only few devices are compatible with this standard now.

2.2 IEEE 802.11n Protocol

In this section, we describe the IEEE 802.11n protocol that has been used for our throughput measurements, proposed models and implementations in this thesis. IEEE 802.11n is an amendment to the IEEE 802.11 2007 wireless networking standard. It adopts several performance enhancement features such *channel bonding (CB)*, *Multiple Input Multiple Output (MIMO)*, frame aggregation, and security improvements over the previous 11a, 11b, and 11g standards. Table 2.3 summarizes the key features of *IEEE 802.11n* standard.

Table 2.3: IEEE 802.11n specification.

Specification	**IEEE 802.11n**	
Frequency Band	2.4 GHz	5 GHz
Simultaneous Uninterrupted Channel	2 ch	9 ch
Available Channel	13 ch	19 ch
Max. Speed	$600Mbps$	
Max. Bandwidth	40 MHz	
Max. Spatial Streams	4	
Subcarrier Modulation Scheme	64 QAM	
Release Date	Sept 2009	

The IEEE 802.11n supports both on 2.4GHz and 5GHz bands. Currently, 2.4GHz is most popular. This frequency band has become crowded with lots of WiFi signals using the same channel or partially overlapping channels. As a result, these WiFi signals with adjacent channels will suffer from interferences between them, and end up with throughput performance drops [38, 40, 54, 55]. For 2.4GHz band, there is a limited number of non-interfered channels, which are Channel 3 and Channel 11 in the 40MHz bandwidth. While for the 20MHz bandwidth, Channel 1, Channel 6, and Channel 11 are basically free from interferences among them. In overall, the wider bandwidth will reduce the number of orthogonal channels. Figure 2.7 [39] shows the WiFi channels for IEEE 802.11n 2.4 GHz band.

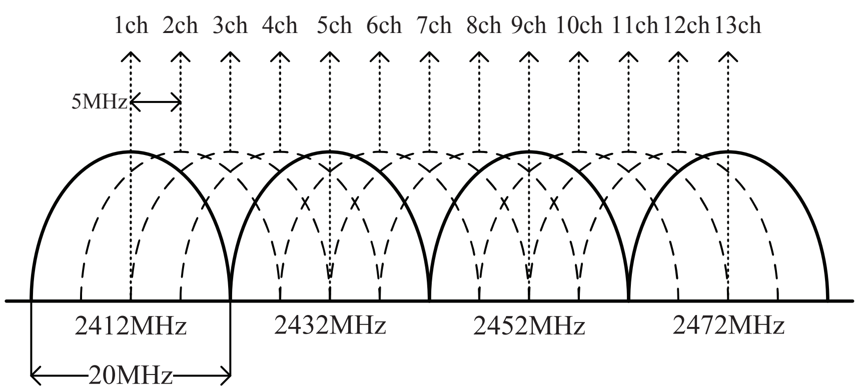


Figure 2.7: WiFi channels in 2.4 GHz band.

In the 5 GHz band of IEEE 802.11n protocol, it has 19 uninterrupted channels available with the 20 MHz bandwidth. In the 40 MHz bandwidth, which doubles the channel width from the 20 MHz, there are nine channels. For the 80 MHz bandwidth, there are four of them. Figure 2.8 shows the channels for the 5GHz band [56].

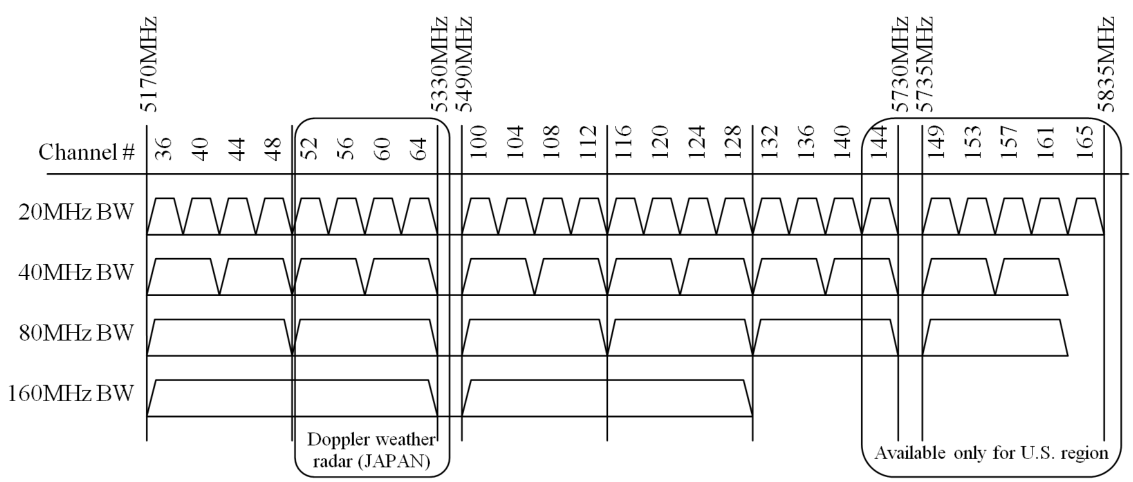


Figure 2.8: WiFi channels in 5 GHz band.

2.3 Features of IEEE 802.11n Protocol

IEEE 802.11n protocol incorporates several new technologies to boost up its performance. The standard uses the multiple antennas technology, channel bonding, frame aggregation, and security improvements mechanism to improve the throughput. In this section, we describe these features of the protocol.

2.3.1 Channel Bonding

The *IEEE 802.11n* protocol supports channel bonding where each channel can operate with the 40 MHz bandwidth by using two adjacent 20 MHz channels together to double its physical data rate [57] as shown in Figure 2.9. However, the usage of the channel bonding will reduce the available non-interfered channels for other devices as there are only two non-interfered bonded channels available at 2.4 GHz band. Table 2.4 summarizes the channel bonding for the 13 *20 MHz* channels at *2.4 GHz* band [58].

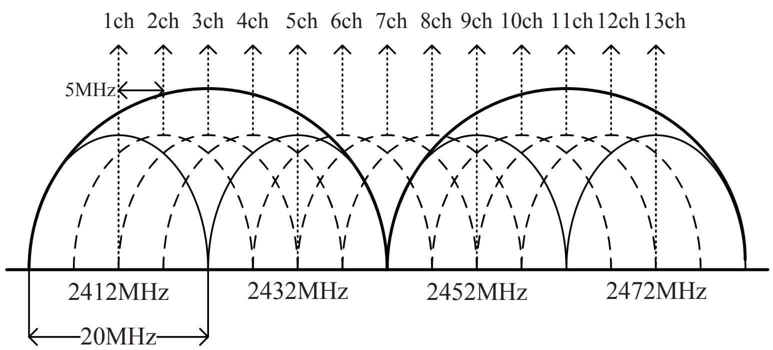


Figure 2.9: IEEE 802.11n channel bonding concept.

Table 2.4: Channel bonding in IEEE 802.1n.

20MHz		40MHz	
center frequency of primary channel	center frequency of secondary channel	bonded channel	center frequency of bonded channel
1	5	1+5	3
2	6	2+6	4
3	7	3+7	5
4	8	4+8	6
5	9	5+9	7
6	10	6+10	8
7	11	7+11	9
8	12	8+12	10
9	13	9+13	11

Table 2.5 shows the usage of different channel bandwidths and spatial streams towards the throughput of *IEEE 802.11n*.

Table 2.5: Effects of channel bandwidth and spatial stream's selection towards IEEE 802.11n's throughput.

Stream number	Bandwidth	
	20 MHz	40 MHz
1 Stream	$72.2 Mbps$	$150 Mbps$
2 Streams	$144.4 Mbps$	$300 Mbps$
3 Streams	$216.7 Mbps$	$450 Mbps$
4 Streams	$288.9 Mbps$	$600 Mbps$

2.3.2 Partially Overlapping Channels

In *IEEE 802.11*, at 2.4 GHz band, each channel has 20 MHz width, and two adjacent channel bands are 5 MHz apart. Thus, all the adjacent channels are partially overlapped with each other. That is to say, each channel is partially overlapped with at least three neighbour channels.

2.3.3 MIMO (Multiple Input Multiple Output)

In MIMO, the throughput can be linearly increased to the number of transmitting (T_X) and receiving (R_X) antennas up to four times, without the additional bandwidth or transmission power. The coverage area can be enhanced over the single antenna technology in *Single-Input Single-Output* (SISO). The multiple antenna configurations in *MIMO* can overcome the detrimental effects of multi-path and fading, trying to achieve high data throughput in limited bandwidth channels. For example, in the 4×4 MIMO, four independent data streams can be multiplexed and transmitted simultaneously with the *spatial division multiplexing* (*SDM*), to speed up the transmission capacity by quadruple as shown in Figure 2.10.

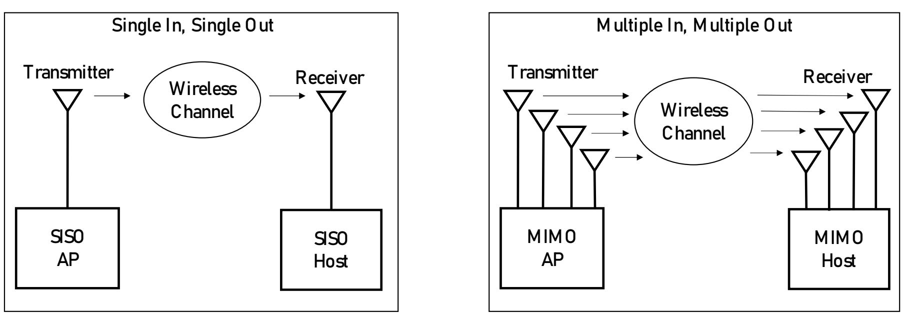


Figure 2.10: Comparison between SISO and 4×4 MIMO technology.

When the *space-time block coding* (*STBC*) is adopted in the 4×4 MIMO link, the sender can transmit four copies of the data stream over four antennas to improve the reliability and the effective range of data transmissions.

2.3.4 MAC Layer Enhancements

Besides the introduction of channel bonding and MIMO, IEEE 802.11n also provides performance improvements through the *frame aggregation* and the proper selection of the *modulation and coding scheme (MCS)*.

- *Frame Aggregation:*
  The *IEEE 802.11n* provides extra performance improvement through the frame aggregation in the MAC layer, besides MIMO. The frame aggregation can transmit multiple frames by one big frame with a single pre-ample and header information to reduce the overhead by them. IEEE 802.11n introduces the *Aggregation of MAC Service Data Units (A-MSDUs)* and *Aggregation of MAC Protocol Data Units (A-MPDUs)*. Frame aggregation is a process of packing multiple A-MSDUs and A-MPDUs together to reduce the overheads and average them over multiple frames, thereby increasing the user level data rate [59].

- *Modulation and Coding Scheme:*
  Various modulation, error-correcting codes are used in the IEEE 802.11n, represented by a Modulation and Coding Scheme (MCS) index value, or *mode*. IEEE 802.11n defines 31 different modes and provides the greater immunity against selective fading by using the Orthogonal Frequency Division Multiplexing (OFDM). This standard increases the number of OFDM sub-carriers of 56 (52 usable) in *High Throughput (HT)* with $20MHz$ channel width and 114 (108 usable) in HT with $40MHz$. Each of these sub-carriers is modulated with BPSK, QPSK, 16-QAM or 64-QAM, and Forward Error Correction (FEC) coding rate of 1/2, 2/3, 3/4 or 5/6 [60].

2.4 Security Technologies of IEEE 802.11

The *IEEE 802.11* standards, commonly referred to as *Wi-Fi*, have significantly evolved in terms of security protocols to address vulnerabilities in wireless communication. In this section, I discuss three key security protocols: Wired Equivalent Privacy *(WEP)*, Wi-Fi Protected Access *(WPA)*, and Wi-Fi Protected Access II *(WPA2)*.

2.4.1 Wired Equivalent Privacy

The wired equivalent privacy *(WEP)* was introduced in 1997 as part of the original *IEEE 802.11* standard to provide security that is comparable to that of wired networks. This protocol utilizes the *RC4* stream cipher for encryption. Initially, WEP supported 40-bit keys, but later it was extended to support 104-bit keys. It employs 24-bit IVs to help randomize the encryption keys. However, *WEP* has significant drawbacks. The use of short *IVs* allows attackers to collect enough packets to perform statistical analysis and potentially break the encryption. Additionally, the use of fixed keys creates vulnerabilities; once a key is compromised, all communications encrypted with that key can be decrypted. *WEP* also lacks robust integrity checks, which makes it possible for packets to be modified without detection.

2.4.2 Wi-Fi Protected Access

The Wi-Fi protected access *(WPA)* was introduced in 2003 as a more secure alternative to the older *WEP* protocol. It employs *TKIP (temporal key integrity protocol)* for dynamic key encryption and includes a message integrity check *(MIC)* to prevent packet forgery. Although *WPA* addresses some of the vulnerabilities of WEP, such as key management, it still has weaknesses, including susceptibility to attacks like the michael attack and potential legacy issues due to its backward compatibility with *WEP*. Consequently, while *WPA* improves security, it is considered outdated and not suitable for modern wireless environments.

2.4.3 Wi-Fi Protected Access II

The Wi-Fi protected access II *(WPA2)* is a secure wireless networking protocol introduced in 2004 to provide enhanced security compared to its predecessor, *WPA*. It employs *AES* (advanced encryption standard) for strong encryption and utilizes the *CCMP* (counter mode with cipher block chaining message authentication code protocol) protocol for improved data integrity and confidentiality. While it offers a robust security framework compliant with the *IEEE 802.11i* standard, *WPA2* is still vulnerable to certain attacks, such as the *KRACK* attack and brute-force methods exploiting weak passwords. Therefore, it is essential for users to implement strong passwords and stay informed about potential security risks.

2.5 Commercial AP

In WLAN, the *access point (AP)* is the networking device that acts as the central trans-receiver or base station. It allows one WiFi device to connect with each other within the network and serves as the interconnection point between WLAN and the wired network. Each AP can support the connection of multiple wireless devices.

A lot of commercial APs from different vendors are now available for WLAN. In Japan, APs from Buffalo, I-O data, TP-link, and NEC are popular [61–63]. They can operate in both IEEE 802.11n and 11ac protocols. According to our measurements for 802.11n, the AP from NEC usually has the largest throughput of around 210 $Mbps$ for a PC with 2×2 MIMO in the outdoor environment. However, this peak throughput varies significantly depending on the propagation environment including the obstacles, the channel interference, the number of antennas, the transmission power of AP, the Wi-Fi adapter, and the placement height and orientation angle of AP.

2.6 Software AP using Raspberry Pi

Recently, the *software enabled AP (SoftAP)* is popular in IEEE 802.11n WLAN. A laptop/PC user can be used as an AP by installing the software of AP functions using hostapd software [64]. Along with this trend, *Raspberry Pi* device can be a good choice for the device to run as a software AP.

2.6.1 Raspberry Pi

Raspberry Pi is a series of small single-board computers that have been developed by Raspberry Pi Foundation [65]. The original purpose of this device is to promote teaching the basic computer science in schools and developing countries. It uses the *Broadcom system on a chip (SoC)* with the integrated ARM compatible central processing unit (CPU) and the on-chip graphics processing unit (GPU) for the graphics. A monitor and a USB keyboard/mouse can be used as peripheral devices. In this paper, we adopt *Raspberry Pi 3* [66] using *Raspbian OS*, *Broadcom BCM2837 SOC* with the *1.2Ghz 64-bit quad-core ARM Cortex-A53 CPU*, *LPDDR2-900MHz 1GB SDRAM*, *10/100Mbps Ethernet*, *IEEE 802.11b/g/n wireless NIC*, *Bluetooth 4.1 classics/low energy*, and *400MHz video core IV GPU* [67][65].

2.6.2 Configuration of Raspberry Pi as Software AP

Raspberry Pi basically adopts Raspbian, a Debian based Linux distribution, as the OS and has the built-in wireless network interface card (NIC) which has only one antenna. This built-in NIC in Raspberry Pi supports only $20MHz$ channel for data transmission since its internal chip does not permit $40MHz$ channel bonding functionality. According to our experiment using Raspberry Pi with built-in NIC, the peak throughput becomes about $40Mbps$. To improve the performance, an external wireless NIC adapter can be used in Raspberry Pi which can realize hardware channel bonding.

2.6.3 Channel Bonding (CB) Configuration of Raspberry Pi AP

In this section, we introduce the *channel bonding (CB)* configuration for *Raspberry Pi* using an external wireless NIC adapter with USB3.0. The following procedure describes how to configure the CB [68]:

1. Download *hostapd* and *bridge-utils* from *Raspbian Repository* and install them by using the command:
    - $sudo apt-get install hostapd bridge-utils.
2. Rebuild *hostapd* to support the channel bonding by the following steps:
    (1) Activate *deb-src* in */etc/apt/sources.list* and update it using the command:
    - $sudo apt-get update.

    (2) Obtain the source code of *hostapd* and built it using the two commands:
    - $sudo apt-get source hostapd
    - $sudo apt-get build-dep hostapd.

    (3) Modify */src/ap/hw_features.c* as shown in Listing 1, to create a new patch and rebuilt it.

Listing 2.1: hw_features.c

```
@@ -535,6 +535,7 @@
    oper40 = ieee80211n_check_40mhz_2g4
             (iface, scan_res);
  wpa_scan_results_free(scan_res);

+#if 0
  if (!oper40) {
    wpa_printf(MSG_INFO,
    "20/40 MHz operation not permitted on"
    "channel pri=%d sec=%d based on
    overlapping BSSes",
@@ -544,6 +545,12 @@
  iface->conf->secondary_channel = 0;
  iface->conf->ht_capab &=
       ~HT_CAP_INFO_SUPP_CHANNEL_WIDTH_SET;
  }
```

```
+#endif
+    wpa_printf(MSG_INFO,
     "Force 20/40 MHz operation on "
+    "channel pri=%d sec=%d even there are
     overlapping BSSes",
+      iface->conf->channel,
+      iface->conf->channel +
+      iface->conf->secondary_channel * 4);
res =
ieee80211n_allowed_ht40_channel_pair(iface);
hostapd_setup_interface_complete(iface,!res)
```

(4) Compile the packages using the command:
   – $fakeroot debian/rules binary.

(5) Reinstall*hostapd* using *dpkg* command.

3. Create the configuration file in *hostapd.conf* with the following configurations:
   - bridge=br0; hw_mode=g; channel=(1-11)
   - ieee80211n=1; obss_interval=1;
   - ht_capab=[HT40+][SHORT-GI-20][SHORT-GI-40] [DSSS_CCK-40] [MAX-AMSDU-3839].

4. Edit the interface to create the bridge between the wireless adapter and the Ethernet adapter in */etc/network/interfaces* file and set a static IP address.

5. Enable *hostapd* using the command:
   - $sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf.

Based on the *hostapd* configuration for the channel bonding, after the channel number is chosen, it becomes the primary channel. The secondary channel is automatically chosen based on the configuration.

2.7 Linux Tools for Wireless Networking

As an open-source operating system, *Linux* has been used as a platform to implement new algorithms, protocols, methods, and devices for advancements of wireless networks [69]. In this section, we give the overview of the Linux tools and software used for the measurement performed throughout the thesis and the implementation of the *elastic WLAN system*.

2.7.1 'hostapd' - to Make AP-mode Linux-PC

hostapd is a Linux tool to realize the AP and the authentication server. It implements IEEE 802.11 AP managements, along with other IEEE 802.1X protocols and security applications. In *Linux*, *hostapd* can be installed by downloading the source code from [70] or using the following command:

```
$ sudo apt-get install hostapd
```

After installing this tool in a Linux PC that contains WLAN driver that supports the AP mode, it can be configured to create a command-line based AP in the Linux-PC. The *hostapd* can be started or stopped by the following commands:

```
$ sudo /etc/init.d/hostapd start
$ sudo /etc/init.d/hostapd stop
```

2.7.2 'ssh' - to Remotely Execute Command

ssh is an abbreviation of *Secure Shell* that is a cryptographic network protocol to securely initiate a shell session on a remote machine [71, 72]. It is operated in two parts: *SSH client* and *SSH server*, and establishes a secure channel between them over an insecure network. The open source version of *ssh* is *OpenSSH* [73] that can be installed using the following command [74]:

```
$ sudo apt-get install openssh-server openssh-client
```

The following list shows an example to remotely execute *nm-tool* on a remote host through the network using *ssh* [71, 72, 75]:

```
$ ssh username@192.168.1.31 'nm-tool'
username@192.168.1.31's password:
```

Here, 192.168.1.31 represents the IP address of the remote host.

2.7.3 'iperf' - to Measure Link Speed

iperf [76] is a software to measure the available throughput or bandwidth on IP networks. It supports both TCP and UDP protocols along with tuning various parameters related to timing and buffers, and reports the bandwidth, the loss, and other parameters. *iperf* is usually installed by default in the Ubuntu distribution. It can also be installed manually using the following command:

```
$ sudo apt-get install iperf
```

To measure the TCP throughput between two devices using *iperf*, one of them uses the server-mode and the other one uses the client-mode, where packets are transmitted from the client to the server. The *iperf* output contains the time-stamped report of the transmitted data amount and the measured throughput. The following list shows the typical use of *iperf* on the server and client side for throughput measurement:

```
$ iperf -s //server side
$ iperf -c 172.24.4.1 //client side
```

Here, 172.24.4.1 represents the IP address of the server. In this thesis, we use *iperf* to measure the throughput between an AP and a host through the *IEEE802.11n* protocol.

2.8 Summary

In this chapter, I presented various wireless network technologies, the key features of IEEE 802.11n protocols, and Linux tools which are adopted in this thesis for the implementations in this study. In the next chapter, I will review the previous studies related to this thesis.

Chapter 3

Review of Previous Studies

In this chapter, I overview the previous studies related to this thesis. Firstly, I review the *elastic WLAN system*. Secondly, I review the study of throughput estimation model for single link communication under no interference, for interfered link under CB, for interfered link under Non-CB, and for interfered link under coexistence of CB and Non-CB respectively. Thirdly, I review the study of the active AP configuration algorithm with orthogonal channel assignment and AP joint optimization algorithm for the elastic WLAN system. Finally, I review the implementation details of the elastic WLAN system testbed using *Raspberry Pi* APs.

3.1 Elastic WLAN System

In this section, I review the *elastic WLAN system* that has been studied in our group. It dynamically optimizes the network configuration by activating/deactivating APs, according to traffic demands and network conditions. Thus, it can reduce energy consumptions and interferences while improving the throughput.

3.1.1 Overview

Nowadays, WLANs have been deployed in several areas including educational institutions and public places like buses, airplanes, or trains. Unplanned or independently controlled APs can lead to causing problems of throughput drops and/or wastages of energy. In one hand, WLANs can suffer from over-allocation problems with redundant APs that have overlapped coverage areas. On another hand, WLANs can be overloaded with hosts suffering from low performances. Therefore, WLANs should be adaptive according to the network traffic demands and conditions by changing the number of active APs and the hosts associated to them. To realize this feature, previously, the elastic WLAN system has been studied.

The motivations behind the elastic WLAN system study are summarized as follows:

1. Operational cost and energy consumption reduction:

    - Companies, educational institutions, and offices may allocate a high number of APs to provide high WLAN performances at peak times and may activate these APs for the entire days. However, only a small number of APs are used during off-peak hours or holidays. The elastic WLAN system can resolve this problem by minimizing the number of APs by traffic demands and can reduce energy consumptions.

- Most developing countries can suffer from volatile Internet connections due to electricity supply discontinuities. The elastic WLAN system can improve the network performance by optimizing the power usage.

2. WLAN performance improvement:

   - When the current active APs cannot cover the users, new APs should be added to ensure the WLAN performance according to the required traffic demands.
   - When the WLAN performance becomes low due to shortages of *internet service provider (ISP)* connections or the supplied power, it activates the cellular networks using *mobile APs* to maintain the required WLAN performance.
   - In a dense WLAN, as the number of APs increases, interferences due to the frequency signal overlaps can increase causing throughput drop. The elastic WLAN system can dynamically change the assigned orthogonal channels of APs, so it can reduce the interferences among them and enhance the WLAN performance.

3.1.2 Design and Operational Flow

Figure 3.1 demonstrates an example topology of the elastic WLAN system. It dynamically controls the number of active APs in the network by activating or deactivating the allocated APs according to traffic demands and network conditions.

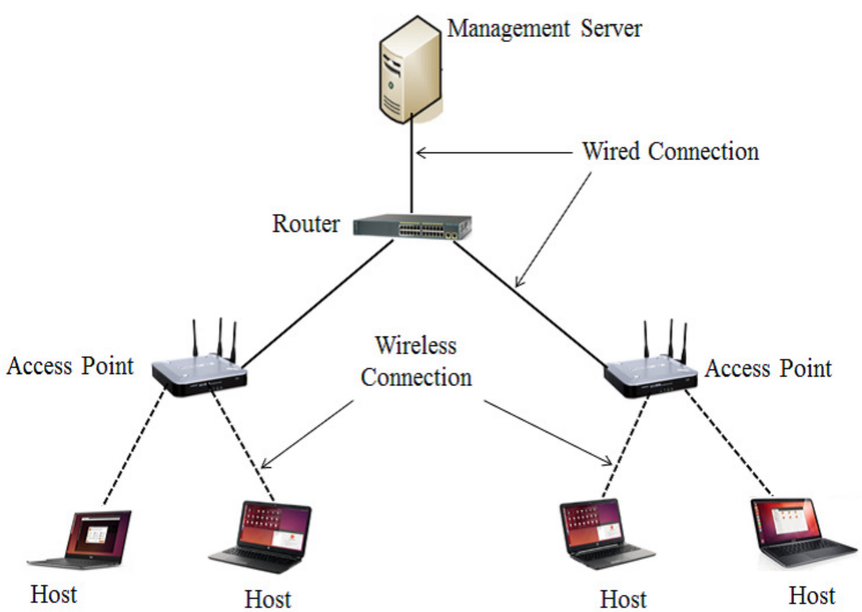


Figure 3.1: Design of Elastic WLAN system.

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

1. The server explores all the devices in the network and collects the requisite 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.

3.2 Definitions of Three Distances

In this section, the *channel distance*, the *physical distance*, and the *link distance* are defined to describe the throughput estimation model under POCs.

1. *Channel distance (chD)* represents the minimum channel difference between the channels of the two links. For example, when both links are assigned the same channel, *chD* is 0, where they will be fully overlapped. On the other hand, when one link is assigned *channel 1* and another link *channel 3*, *chD* is 2.

2. *Physical distance (phD)* represents the Euclidean distance between the two APs of the links. By increasing the *phD* between the links, the interfered signal fades due to the path loss and the absorption by obstacles.

3. *Link distance (lkD)* represents the Euclidean distance between the transmitter and receiver of the link. Since the signal is propagated from the transmitter to the receiver, the longer *lkD* reduces the RSS at the receiver and can degrade the throughput.

3.3 Throughput Estimation Model for Single Link Communication under No-interference

The throughput between an AP and a host can be changed by several factors such as the modulation and coding scheme, the transmission power, the transmission distance, and transmission obstacles. Therefore, the theoretical computation of the accurate link speed can be challenging [77, 78]. In this section, we review the *throughput estimation model* for IEEE 802.11n link in WLAN.

3.3.1 Overview

The *throughput estimation model* estimates the link speed or throughput of an IEEE 802.11 link in WLAN. It has two main steps to estimate the throughput between a source node (AP) and a destination node (host).

In the first step, it estimates the *receiving signal strength* (*RSS*) at the host by using the *log distance path loss model* [78]. In the second step, it converts the estimated *RSS* into the corresponding throughput by the *sigmoid function* [24]. Both functions have several configuration parameters that can affect the estimation accuracy, which depends on link specifications and network field environments.

3.3.2 Receiving Signal Strength Estimation

The signal strength is estimated by the log-distance path loss model in [78]. First, the Euclidean distance *d (m)* is calculated for each link (AP/host pair) by:

$$d = \sqrt{(AP_x - H_x)^2 + (AP_y - H_y)^2} \tag{3.1}$$

where AP_x, AP_y and H_x, H_y does the x and y coordinates for the AP and the host respectively.

Then, the *RSS*, RSS_d (dBm) at the host is estimated using the *log distance path loss model* by considering the distance and the obstacles loss between end nodes [78]:

$$RSS_d = P_1 - 10\alpha \log_{10} d - \sum_k n_k W_k \tag{3.2}$$

where P_1 represents the signal strength at $1m$ from the AP (source) for no obstacles, α is the path loss exponent, d (m) does the link distance from the AP, n_k does the number of the type-k walls along the path between the AP and the host, and W_k does the signal attenuation factor (dBm) for the type-k wall in the environment. The estimation accuracy of *RSS* relies on the parameter values, which depend on the propagation environment [78].

3.3.3 Throughput Estimation by Sigmoid Function

From the estimated *RSS* RSS_d, the throughput tp_{ij} ($Mbps$) of the link is estimated using the *sigmoid function* as follows:

$$tp_{ij} = \frac{a}{1 + e^{-(\frac{(RSS_d+120)-b}{c})}} \tag{3.3}$$

where a, b, and c are the constant parameters of the sigmoid function that should be optimized by parameter optimization tool [79]. The assumption of the sigmoid function for the throughput estimation is based on our real-world measurement results which clearly reflects the relationship between the RSSs and the estimated throughput. Figure 3.2 demonstrates the sigmoid function curve using $a = 140$, $b = 54$ and $c = 8$.

3.4 Throughput Estimation Model for Interfered Link under CB

In [54, 80], presented the *throughput drop estimation model* under interfered *CB* links. For two interfered links, this model adopts the *logarithm function* of the RSS, RSS^i and the *chD* from the interfered link in Eq. (3.4).

$$tpD(RSS^i, chD) = p(chD) \times \ln(q(chD) + RSS^i) + r(chD) \tag{3.4}$$

where $tpD(RSS^i, chD)$ indicates the estimated throughput drop ($Mbps$), and $p(chD)$, $q(chD)$, and $r(chD)$ represent the constants determined by the channel distance (chD). The *physical distance* (phD) between the two APs is closely related with the RSS (RSS^i) of the interfered signal at the AP. When phD increases, the corresponding RSS^i decreases, as shown in Eq. (3.2) where RSS_d represents RSS^i and d does phD. The values of the three constant parameters in Eq. (3.4), p, q,

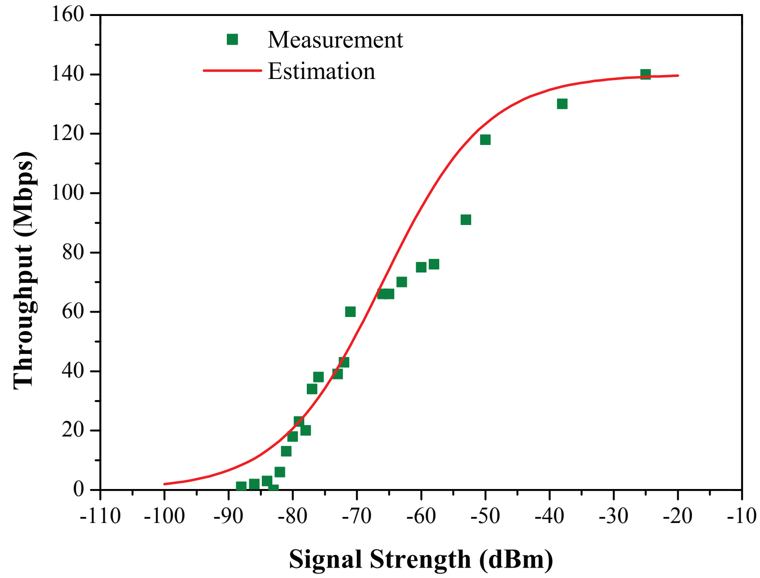


Figure 3.2: Sigmoid function for throughput estimation from signal strength.

and r, in [80] are computed from the throughput drop measurement results for each *chD* by running *Origin Pro8* software [81].

The interfered link causing the highest drop is considered first. Then, the interfered link causing the second highest drop is considered, which further reduces the throughput by increasing the contention. The following procedure is applied;

1. Estimate the throughput of the target link using the model under non-interference.

2. Estimate the throughput drop *tpD* from each interfered link using Eq. (3.4).

3. Sort the links in descending order of the drop magnitude. Here, the two interfered links are considered to the target link, where the drops are given by tpD^{1st} and tpD^{2nd}.

4. For the largest interfered link, adjust tpD^{1st} by the maximum speed of the AP of the target link, because the different APs may have the different throughput performances. The largest interfered link is defined as the interfered link that causes the largest throughput drop (*tpD*) at the target link.

$$tpD^{1st}_{adj} = tpD^{1st} \times \frac{tpM^{AP}}{140} \tag{3.5}$$

where tpD^{1st}_{adj} represents the adjusted throughput drop by the largest interfered link, tpM^{AP} does the maximum throughput for the AP of the target link, and 140 does the maximum throughput (Mbps) under channel bonding (CB) for NEC AP adopted in the model.

Then, the throughput tp^{1st}_{ij} of the target link is estimated after considering the drop by the first interfered link by:

$$tp^{1st}_{ij} = tp_{ij} - tpD^{1st}_{adj}. \tag{3.6}$$

5. For the second interfered link, adjust the tpD^{2nd} by:

$$tpD_{adj}^{2nd} = tpD^{2nd} \times \frac{tpM^{AP} - tpD_{adj}^{1st}}{140}. \tag{3.7}$$

   The throughput tp_{ij}^{2nd} of target link is estimated after considering the drop by the second interfered link by:

$$tp_{ij}^{2nd} = tp_{ij}^{1st} - tpD_{adj}^{2nd}. \tag{3.8}$$

6. If more interfered links exist, repeat the same procedure.

The throughput drop's actual value depends on the device's throughput range at the target link. Eq. (3.4) was introduced to estimate the throughput drop, where the maximum link-speed of the target link is 140 Mbps for the *NEC WG2600HP* AP device with CB used in the experiments. For other devices whose maximum speed is different from 140 Mbps, this value needs to be adjusted linearly by its maximum speed, as confirmed by experiments.

3.5 Throughput Estimation Model for Interfered Link under Non-CB

In [55], presented the *throughput drop estimation model* under interfered *non-CB* links. According to the measurement results, the *natural logarithm function* in Eq. (3.4) is again used to estimate the *throughput drop* (tpD) for *non-CB* links. The parameter values are tuned from measurement results of each *chD* [25].

For three or more interfered links, the interfered links are explored sequentially in descending order of their throughput drops as for *CB* links. First, the throughput drop in Eq. (3.5) and Eq. (3.7) is adjusted to consider the difference of the maximum throughput of the APs for *non-CB* ($75Mbps$) and *CB* ($140Mbps$) as follows:

$$tpD_{adj}^{1st} = tpD^{1st} \times \frac{tpM^{AP}}{75} \tag{3.9}$$

$$tpD_{adj}^{2nd} = tpD^{2nd} \times \frac{tpM^{AP} - tpD_{adj}^{1st}}{75} \tag{3.10}$$

Then, the dropped throughput under the interferences for the target link is obtained by sequentially subtracting the tpD_{adj}^{1st} and tpD_{adj}^{2nd} from tp_{ij}, as in Eq. (3.6) and Eq. (3.8).

3.6 Throughput Estimation Model for Interfered Link under Coexistence of CB and Non-CB

In [25], presented throughput drop measurements under coexistence of *CB* and *non-CB* links.

3.6.1 Throughput Drop Estimation Model for Two Interfered Links

According to the throughput measurement results, in [25] estimated the *throughput drop (tpD)* from the interfered *received signal strength (RSS^i)* and the *channel distance (chD)*, as in Eq. (3.4). The parameter values are tuned from measurement results under coexistence of *CB* and *non-CB* as in [25] by running *Origin Pro8* software.

3.6.2 Throughput Drop Estimation Model for Multiple Interfered Links

Again for the multiple interfered links, the interfered links are explored sequentially in descending order of their throughput drops. Here only two interfered links are described.

1. When both of the interfering APs adopt *CB*, estimate the throughput drop under *CB* in [54].

2. When both of the interfering APs adopt *non-CB*, estimate the throughput drop under *non-CB* in [55].

3. When one AP adopts *CB* and another does *non-CB*, the following procedure is applied:

   (a) Estimate the single link throughput for each host by Eq. (3.2) and Eq. (3.3).

   (b) Estimate the sum of the throughput drops for the two APs by Eq. (3.4) using the parameters in [25].

   (c) Sort the links in descending order of the throughput drops that are given by tpD^{1st} and tpD^{2nd}.

   (d) For the largest interfered link, adjust tpD^{1st} with the maximum speed of the target AP by Eq. (3.11) and Eq. (3.12).

$$tpD_{adj}^{1st} = tpD^{1st} \times \beta \times \frac{tpM^{AP}}{140}. \tag{3.11}$$

$$tpD_{adj}^{1st} = tpD^{1st} \times \beta \times \frac{tpM^{AP}}{75}. \tag{3.12}$$

   It is assumed that one AP represents the target AP and the other does interfering AP. Eq. (3.11) is applied if the target AP uses CB, and Eq. (3.12) is applied otherwise. β represents the throughput drop normalization factor of 0.635 for *CB* and 0.365 for *non-CB*.

   (e) For the second interfered link, adjust the tpD^{2nd} by Eq. (3.13) and Eq. (3.14) for the target AP with CB and the AP with *non-CB* respectively.

$$tpD_{adj}^{2nd} = tpD^{2nd} \times \beta \times \frac{tpM^{AP} - tpD_{adj}^{1st}}{140}. \tag{3.13}$$

$$tpD_{adj}^{2nd} = tpD^{2nd} \times \beta \times \frac{tpM^{AP} - tpD_{adj}^{1st}}{75}. \tag{3.14}$$

   Then, the dropped throughput under the interferences for the target link is obtained by sequentially subtracting tpD_{adj}^{1st} and tpD_{adj}^{2nd} from tp_{ij}, as in Eq. (3.6) and Eq. (3.8).

   (f) If more interfered links exist, repeat the same procedure.

3.7 Active AP Configuration Algorithm

In this section, I review the *active AP configuration algorithm* for the elastic WLAN system that optimizes the number of active APs and the host associations [22, 23].

3.7.1 Problem Formulation

The active AP configuration problem for this algorithm is formulated as follows:

1. Inputs:
   - Number of hosts: H
   - Number of APs: $N = N^D + N^V + N^M$ where N^D, N^V, and N^M represent the number of DAPs, VAPs, and MAPs respectively.
   - 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 by the model in [24].
   - Minimum link speed for association: tp
   - 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.

   Let, E_1 represents the number of active APs (DAPs, VAPs, and MAPs) in the network:

$$E_1 = E_1^D + E_1^V + E_1^M \tag{3.15}$$

where E_1^D represents the number of active DAPs, E_1^V does the number of active VAPs, and E_1^M does the number of active MAPs respectively.
The transmission delay of the jth AP can be defined by:

$$T_j = \sum_{k \in \mathcal{P}_j} \frac{D_k}{tp_{jk}} \tag{3.16}$$

where D_k represents the traffic of the kth host, tp_{jk} does the link speed between the jth AP to the kth host, and $\mathcal{P}_j$ does the set of the hosts associated with the jth AP. Then, the average throughput TH_{ij} of the ith host associated with the jth AP can be estimated by:

$$TH_{ij} = \frac{D_i}{T_j} = \frac{D_i}{\sum\limits_{k \in \mathcal{P}_j} \frac{D_k}{tp_{jk}}} \tag{3.17}$$

Since the real traffic of each host is unpredictable, we assume the identical traffic for every host, which can be represented by the unit traffic for the sake of simplicity. Then, the average host throughput for AP_j is given by:

$$TH_j = \frac{1}{\sum\limits_{k \in \mathcal{P}_j} \frac{1}{tp_{jk}}} \tag{3.18}$$

If $TH_j \geq G$, the minimum host throughput constraint is satisfied, where G represents the threshold for this constraint. Then, the second objective function E_2 is defined to maximize the *minimum average host throughput* for the bottleneck AP, which is given by:

$$E_2 = \min_j \left[TH_j \right] \tag{3.19}$$

E_3 signifies the total interfered communication time:

$$E_3 = \sum_{i=1}^{N} [IT_i] = \sum_{i=1}^{N} \left[\sum_{\substack{k \in I_i \\ c_k = c_i}} T_k \right] \tag{3.20}$$

where IT_i represents the *interfered communication time* for AP_i, T_i does 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. They are given by follows:

- T_k is given by the sum of the time that is required to transmit one bit data between AP_k and its each associated host.
- I_i represents the set of the indices of the APs that are interfered with AP_i if they are assigned the same channel.
- c_k signifies the channel assigned to AP_k by the active AP configuration algorithm.
- IT_i is given by the sum of the total communication time for the APs that are interfered with 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.

3.7.2 Algorithm Procedure

The active AP configuration algorithm consists of three phases: the *active AP and associated host selection* phase, the *channel assignment* phase, and the *channel load averaging* phase.

3.7.2.1 Active AP and Associated Host Selection Phase

In this phase, the set of the active APs and their associated hosts are selected. This phase comprises following eight steps:

1. **Preprocessing**
   The link speed for each possible pair of an AP and a host is estimated with the measurement or the throughput estimation model [24]. Then, this step initializes the variables for the following steps:

   (a) For each AP, make a list of hosts that can be associated with this AP, where the throughput of the link between a host and an AP is S or larger, it can be associated. This list is called the *associable host list for APs*.

   (b) Initialize each AP as the *non-active* AP. Initially, only the DAPs are selected as *candidate APs*.

2. **Initial Solution Generation**
   An initial solution to the cost function E_1 is derived using a greedy algorithm [82], which repeats the following procedures:

   (a) Select the AP that can be associated with the maximum number of non-associated hosts.

   (b) Activate this AP and increment E_1 by one.

   (c) Update the number of non-associated hosts in the host list for APs.

3. **Host Association Improvement**
   The cost function E_2 is calculated for the greedy solution using Eq. (6.5). Then, this solution is improved by repeating the following procedure:

   (a) Find the AP that gives the lowest host throughput in Eq. (6.5).

   (b) Select one host randomly from the associated hosts with this AP, and associate it with another active AP that is selected randomly. Then, calculate E_2.

   (c) Keep the new association only if this new E_2 is higher than the previous E_2. Otherwise, return to the previous association.

4. **AP Selection Optimization**
   The cost functions E_1 and E_2 are further jointly improved in this step under the constraints mentioned before by the *local search* [83]. This local search repeats the following three procedures:

   (a) If the current solution satisfies the *minimum host throughput constraint*, it seeks to reduce the number of active APs E_1 by deactivating an active AP. In the implementation, it repeats to 1) randomly select an active AP and deactivate it, 2) apply *Host Association Improvement*, and 3) check the feasibility of this deactivation.

   (b) Otherwise, it seeks to improve the *minimum average host throughput* E_2 with the same number of active APs by changing the active AP. In the implementation, it repeats to 1) randomly select a non-active AP and activate it, 2) apply *Host Association Improvement*, and 3) check the possibility of deactivating another active AP.

(c) If (b). is not achieved, it seeks to satisfy the *minimum host throughput constraint* by increasing the number of active APs while improving the *minimum average host throughput*. In the implementation, it 1) randomly selects a non-active AP and 2) applies *Host Association Improvement*.

5. **Link Speed Normalization**
The fairness criterion will be applied when the total expected bandwidth exceeds B^a. Generally, the link speed is normalized as:

   (a) Calculate the expected total bandwidth B^e by the summation of the throughputs of all the APs.

   (b) If $B^e > B^a$, adjust each AP-host link speed as:

$$\hat{tp}_{ij} = tp_{ij} \times \frac{B^a}{B^e} \tag{3.21}$$

   where $\hat{tp}_{ij}$ is the normalized link speed.

6. **Termination Check**
The algorithm is terminated when either of the following conditions is satisfied:

   (a) The *minimum host throughput constraint* is satisfied.

   (b) All the APs in the network have been activated.

7. **Additional VAP Activation**
If all the DAPs become active but the *minimum host throughput constraint* is still not satisfied, VAPs are newly selected as candidate APs. A VAP is slower than a DAP, but faster than a MAP. The locations of hosts are considered as the locations of the candidate VAPs, because user hosts may be used for VAPs. Then, it returns to *AP Selection Optimization* step.

8. **Additional MAP Activation**
If all the DAPs and VAPs become active but the *minimum host throughput constraint* is still not satisfied, MAPs are newly selected as candidate APs. A MAP is the slowest among the three AP types. The locations of hosts are considered as the locations of the candidate MAPs, because users may have MAPs. Then, it returns to *AP Selection Optimization* step.

3.7.2.2 Orthogonal Channel Assignment Phase

In this phase, the channels to assign to the active APs are selected and it has the following four steps:

1. **Preprocessing**
The interference and delay conditions of the network are represented by a graph.

   (a) Construct the *interference graph*, $G = (V, E)$, from the APs and the hosts, where the vertex V represents the set of APs and the edge E presents the existence of the interference between two APs. $e(i, j) \in E$ if AP_i is interfered with AP_j in the network.

(b) Calculate the *communication time* for each AP. The communication time T_i for AP_i is defined as the total time when the AP transmits 1-bit to all the associated hosts. It is given by:

$$T_i = \sum_j \frac{1}{tp_{ij}} \tag{3.22}$$

where tp_{ij} represents the link speed between AP_i and $host_j$.

(c) Calculate the *neighbor interfered communication time* for each AP. The neighbor interfered communication time NT_i for AP_i is given by:

$$NT_i = \sum_{e(i,k)=1} T_k \tag{3.23}$$

2. **Interfered AP Set Generation**
The set of APs that are interfering with each other is found for each AP.

(a) Sort the APs in descending order of NT_i, where the tie-break is resolved by T_i.

(b) Find the interfered AP set for each AP by repeating the following steps:

i. Initialize the interfered AP set by $I_i = \{i\}$ for AP_i.

ii. Expand I_i by examining the APs in sorted order in a) whether the AP is interfered with each AP in I_i. If so, include this AP, AP_j, into I_i, i.e. $I_i = I_i \cup \{j\}$.

(c) Calculate the total interfered communication time AT_i for AP_i, which is given by:

$$AT_i = \sum_{k \in I_i} T_k \tag{3.24}$$

3. **Initial Solution Construction**
Then, an initial solution is derived with a greedy algorithm.

(a) Sort the APs in descending order of the total interfered communication time AT_i, where the tie-break is resolved by NT_i.

(b) Assign a channel c to AP_i such that the interfered communication time IT_i is minimized if it is assigned. IT_i is given by:

$$IT_i = \sum_{\substack{k \in I_i \\ c_k = c_i}} T_k \tag{3.25}$$

where c_k represents the assigned channel to AP_k.

(c) Repeat 2) until each active AP is assigned to one channel.

(d) Calculate the cost function E_3 using Eq. (3.20) and save this initial solution as the best solution E_3^{best}.

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

(a) Randomly select one AP and one new channel for the channel change trial.

(b) Calculate the interfered communication time IT_i after assigning this new channel by:

(c) Calculate E_3^{new} using Eq. (3.20) for the new channel assignment, and $\Delta E_3 = E_3^{new} - E_3$.

(d) If $\Delta E_3 \leq 0$, accept the new channel assignment, and address this new solution as the best one.

(e) Otherwise, generate a 0-1 random number, *rand*, and if $rand \leq \frac{-\Delta E_3}{T^{SA}}$, then accept the new channel assignment.

3.7.2.3 Channel Load Averaging Phase

After the channel assignment using the limited number of channels, the total loads may be imbalanced depending on different channels that are assigned to the APs. In this phase, the channel load is averaged by changing associated APs for hosts. It has four steps as follows:

1. **Preprocessing**
   The *AP flag* for each AP is initialized with OFF to avoid processing the same AP.

2. **AP Selection**
   One AP is selected to move its associated host to a different AP that is assigned a different channel.

   (a) Terminate the procedure if each AP has ON AP flag.

   (b) Initialize the host flag by OFF for each host.

   (c) Select one AP, say AP_i, that satisfies the two conditions:

      i. The AP flag is OFF.

      ii. The interfered communication time IT_i is largest among the OFF APs.

   (d) Set the AP flag ON.

3. **Host Selection**
   Then, one host associated with AP_i is selected for the associated AP movement.

   (a) Select one host, say H_j, that satisfies the four conditions:

      i. The host flag is OFF.

      ii. The host is associated with AP_i.

      iii. The host can be associated with another AP assigned a different channel from AP_i, or is located out of the interference range of AP_i.

      iv. The link speed with AP_i is the smallest among the hosts satisfying (a)–(c).

   (b) If one host is selected, set the host flag ON.

   (c) Otherwise, return to *AP Selection* for the new AP selection.

4. **Association Change Application**
   Finally, the new associated AP is selected for H_j.

   (a) Select the AP that has the largest link speed among the APs that are assigned to a different channel from AP_i and can be associated with H_j.

   (b) Calculate the new cost function E_3^{new} with Eq. (3.20) if H_j is associated with this AP.

(c) If E_3^{new} is equal to or smaller than the previous E_3, accept the new association, and return to *Host Selection*.
(d) Otherwise, select another AP that has the next largest link speed, and return to 3).
(e) If no such AP exists, return to *Host Selection* for the new host selection.

3.8 AP Joint Optimization Algorithm

In this section, I review the AP joint optimization algorithm for the elastic WLAN system that optimizes the transmission power, the frequency channel, and the channel bonding of each AP.

3.8.1 Formulation of AP Joint Optimization Algorithm

The AP joint optimization algorithm is formulated as follows [25].

1. Input and Output: In the algorithm input, the number of channels C_{CB} for *CB POC*s and C_{non} for *non-CB POC*s, and the center frequency of each channel are given.

   In the algorithm output, the *CB* or *non-CB POC*, and the *maximum* or *minimum* transmission power is assigned to each active AP, instead of *non-CB*, *orthogonal channels (OC)*, and the *maximum* transmission power.

2. Objective: In the algorithm objective, the cost function E_{ch} in Eq. (3.26) maximize the total throughput of the links in WLAN.

$$E_{ch} = \sum_{i=1}^{N} TP_i^{POC} \tag{3.26}$$

   where TP_i^{POC} represents the total throughput of the links associated with AP_i and N is total number of APs.

   TP_i^{POC} is calculated by:

$$TP_i^{poc} = \sum_{j}^{m} \left(tp_{ij}^{poc} \times Srf(m) \right) \tag{3.27}$$

   where m represents the number of hosts associated with AP_i, tp_{ij}^{POC} does the estimated throughput of the link between AP_i and $host_j$ by the proposed model, and $Srf(m)$ does the contention factor at AP_i for associated hosts to send data through CSMA/CA. $Srf(m)$ is calculated by:

$$Srf(m) = \left(\frac{1}{m + \frac{0.1(m-1)}{4}} \right) \times 1 - (0.1 \times m - 1). \tag{3.28}$$

   The constants in Eq. (3.28) are obtained from measurements by increasing the number of associated hosts to a single AP one by one under no interference.

3.8.2 Procedure of Joint Optimization Algorithm

Initially, only *CB* channels with *maximum transmission power* are assigned by the greedy procedure, since they can maximize the throughput in general. Then, this assignment is improved by optimizing the selection of *CB*, *non-CB*, the maximum power, and the minimum power by the following steps [25]:

1. Randomly select one AP with the maximum transmission power for the change trial.
2. Randomly select a different CB channel from the current one to this selected AP.
3. Run the throughput estimation model. If the estimated throughput improves E_{ch} in Eq. (3.26), this channel change is accepted. To avoid the local optimum, the hill-climbing procedure is applied where if 0-1 random number is smaller than exp(DeltaE_{ch}/Temp), where DeltaE_{ch} is difference between old and new E_{ch} and Temp is given algorithm parameter for temperature, this channel change is accepted.
4. Go back to Step 1, when the new channel is accepted. Otherwise, go to the next step.
5. Change the transmission power to the minimum and run Step 3.
6. Go back to Step 1, when the new power is accepted. Otherwise, go to the next step.
7. Change the selected channel to *non-CB* and run Step 3.
8. Go back to Step 1, when the new non-CB is accepted. Otherwise, go to the next step.
9. Change the transmission power to the maximum and run Step 3.
10. Go back to Step 1.

The maximum transmission power for *CB* and *non-CB* channels is $-20dB$ and $-28.2dB$, respectively, and the minimum for both is $-33.2dB$.

3.9 Implementation of WLAN Testbed System Using Raspberry Pi

In this section, I describe the testbed implementation of the elastic WLAN system using *Raspberry Pi* and Linux PCs. *Raspberry Pi* is a small-size low-cost computer, and has become popular in academics and industries around the world. Therefore, the use of *Raspberry Pi* in the elastic WLAN system is significant for its disseminations in developing countries.

3.9.1 Implementation Environment/Platform

As the initial implementation platform of our elastic wireless LAN system, we choose the Linux environment that has been used as a platform to implement new algorithms, protocols, methods, and devices for advancements of wireless networks, because of being an open-source operating system. Linux environment has a lot of open source tools to use. Most of them are easily configurable and have flexibility to use and integrate with other tools [69]. On the other hand, while searching for the network configuration and management tools in Windows operating system, we found most of them are less flexible and less configurable, and not open source. Currently, we are using *Ubuntu* for our implementation platform as the popular distribution of Linux environment for general-purpose users. The Ubuntu LTS 14.04 version of the operating system was adopted in *elastic WLAN system testbed* for compatibility and similarity of throughput performance in measurements. Implementations of the elastic WLAN system on various platforms and latest version of Ubuntu OS will be in our future studies.

The device environments and software in Table 3.1 are used for the testbed implementation of the system. The IEEE 802.11n protocol is used for any communication link with the channel bonding.

Table 3.1: Device environment and software in testbed.

devices and software		
server PC	OS	Ubuntu LTS 14.04
	model	Lesance W255HU
	Processor	Intel(R), Core(TM)-i3
client PC	OS	Ubuntu LTS 14.04
	Model	Fujitsu Lifebook S761/C/SSD
	Processor	Intel(R), Core-i5
access point	OS	Raspbian
	Model	Raspberry Pi 3
	Processor	1.2 GHz
software/tools	openssh	to access remote PC and AP
	hostapd	to prepare and configure AP
	nmcli	for association change
	nm-tool	to measure signal strength
	arp-scan	to discover active network devices

3.9.2 System Topology

Figure 3.3 shows the simple network topology of the elastic WLAN system. *Raspberry Pi* is used for the AP and a *Linux laptop PC* is for the server and the host. The server can manage and control all the APs and the hosts by using the administrative access to them. The APs are connected to the server through wired connections. The hosts and the APs are connected through wireless connections.

3.9.3 AP Configuration of Raspberry Pi

This section explains how to configure *Raspberry Pi* for AP using *hostapd* daemon [64, 84].

1. Install the *hostapd* using the following command:

   ```
   $ sudo apt-get install hostapd
   ```

2. Modify the configuration file */etc/hostapd/hostapd.conf* with the desired SSID and PASSWORD. A simple example of the configuration file is given below:

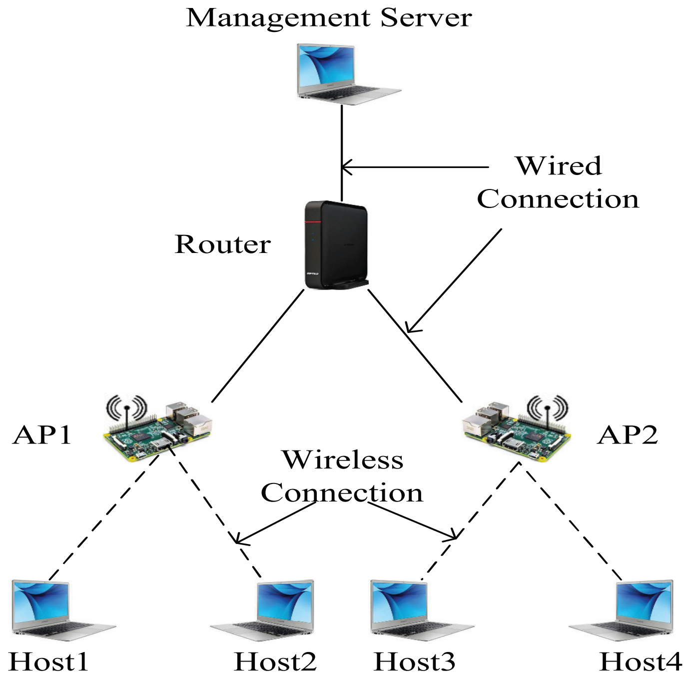


Figure 3.3: Elastic WLAN system topology.

```
interface=wlan0
ssid=SSID
channel=1
wpa_passphrase=PASSWORD
```

3. 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"
   ```

4. Setup the *wlan0* interface to have a static IP address in the network interface configuration file */etc/network/interfaces*. An example of the interface file is given below:

   ```
   auto wlan0
   iface wlan0 inet static
   address 192.168.1.11
   netmask 255.255.255.0
   network 192.168.1.0
   ```

5. Finally, install the DHCP server for assigning the dynamic IP addresses to the hosts.

3.9.4 Execution Flow of Elastic WLAN

Figure 3.4 shows the execution flow of the elastic WLAN system testbed implementation.

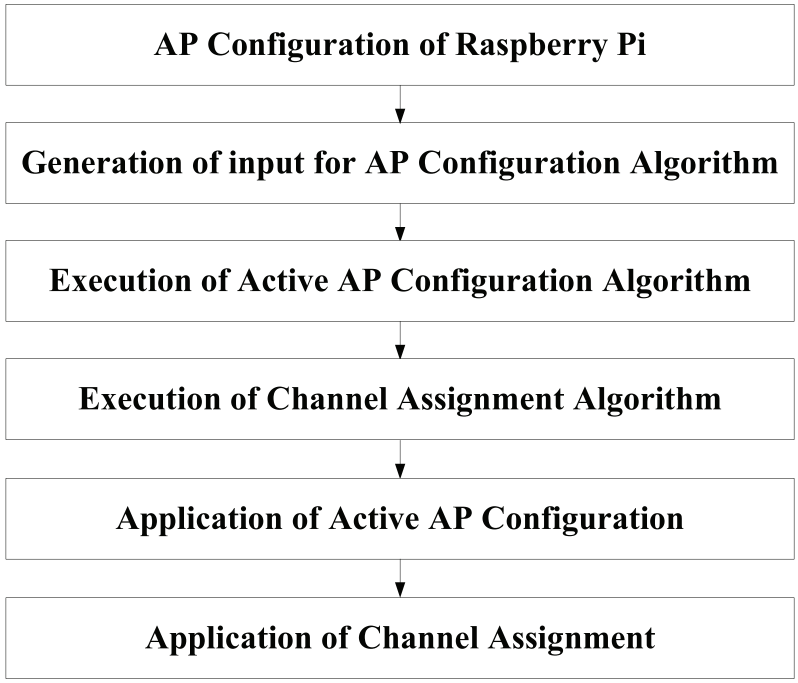


Figure 3.4: Execution Flow of Elastic WLAN system.

3.9.4.1 Generation of input for Active AP Configuration Algorithm

In this step, the server explores all the connected device to the network and generates the input for the active AP configuration algorithm using the following procedure:

1. The server explores all the connected device to the network using *arp-scan* [85]. The command is given below:

   ```
   $ sudoarp-scan --interface=eth0 192.168.11.0/24
   ```

   Here, *–interface=eth0* represents the interface and 192.168.11.0/24 is the network IP range to scan. The output consists of the IP and MAC addresses of the hosts and the APs that are available in the network. A simple C program is developed to identify the hosts and APs in this system using the MAC addresses of the devices. After this, the server generates the list of permitted APs and the list of permitted hosts.

2. The following command finds the receiving signal strength of each host from each AP using *nm-tool* [86, 87]. *ssh* protocol is used to execute the command remotely in each host [71, 72, 75].

   ```
   $ sudo nm-tool
   ```

3. After this, the server converts the receiving signal strength to the estimated link speed using the sigmoid function in [88], and generates the input for the active AP configuration algorithm.

3.9.4.2 Execution of Active AP Configuration Algorithm

The active AP configuration algorithm is executed in this step. The following commands compile the program for the active AP configuration algorithm and execute it respectively. The *minimum host throughput constraint* and the *bandwidth limitation constraint* are specified by the user.

```
$ g++ -o apc APConfigurationAlgorithm.cpp
$ ./apc input.txt min_host_throughput bw_limit
```

Here, *input.txt* presents the input file generated in the previous step, *min_host_throughput* does the minimum host throughput constraint, and *bw_limit* does the bandwidth limitation constraint. After this, the list of active APs and their associations with the hosts are obtained.

3.9.4.3 Execution of Channel Assignment Algorithm

The following commands compile the program for the channel assignment extension and execute it respectively.

```
$ g++ -o ca ChannelAssignment.cpp
$ ./ca HostAPassociation.txt num_of_channels
```

HostAPassociation.txt presents the input file to the channel assignment extension that contains the list of active APs and their associations with the hosts, and *num_of_channels* does the number of available channels.

3.9.4.4 Application of Active AP Configuration

The management server applies the output of the two algorithm.

1. The server adjusts the number of active APs according to the algorithm output by activating or deactivating APs in the network. The two commands given below is used to activate and deactivate the Raspberry Pi AP respectively.

   ```
   $ sudo /etc/init.d/hostapd start
   $ sudo /etc/init.d/hostapd stop
   ```

2. The following command connects a host to a new AP using *nmcli* [89, 90]. *NewSSID* represents the new AP for the host and *PASSWORD* does the security key of the AP. The server modifies the AP-host association according to the algorithm output using this command.

   ```
   $ sudo -s nmcli dev wifi connect NewSSID password PASSWORD
   ```

3.9.4.5 Application of Channel Assignment

The server uses the following commands to assign the new channel to the Raspberry Pi AP using *sed* [91]. For this, the server modifies the configuration file */etc/hostapd/hostapd.conf* with the channel number.

```
$ sed -i -e ‘s/.*channel.*/channel=’$NewChannel‘/’
/etc/hostapd/ hostapd.conf
$ sudo /etc/init.d/hostapd restart
```

Here, *‘s’* represents the substitution command and *NewChannel* does the channel to be assigned in the *hostapd.conf* file of the AP. After the assignment of the new channel, the server restarts it to make the change take effect.

3.10 Summary

In this chapter, firstly, I reviewed the *elastic WLAN system*. Secondly, I reviewed the study of the *throughput estimation model* for single link communication under no interference, the one for interfered link under CB, the one for interfered link under Non-CB, and the one for interfered link under coexistence of CB and Non-CB, respectively. Thirdly, I reviewed the study of the active AP configuration algorithm with the orthogonal channel assignment and the AP joint optimization algorithm for the elastic WLAN system. Finally, I reviewed the implementation details of the elastic WLAN system testbed. In the next chapter, I will present the throughput measurements and estimation models for up to three Raspberry Pi APs.

Chapter 4

Throughput Measurements and Estimation Models for up to Three Raspberry Pi APs

In this chapter, I present the *throughput measurements* under various conditions for concurrent communications using up to three *Raspberry Pi* APs and the modifications of the *throughput estimation model*.

Firstly, I discuss the throughput measurement results and modify the existing *throughput estimation model* under concurrent communications with 11 POCs. Secondly, I verify the model estimation accuracy for 11 POCs. Thirdly, I discuss the throughput measurement results and modify the existing *throughput estimation model* under concurrent communications with 13 POCs. Fourthly, I verify the model accuracy for 13 POCs. Finally, I present the extension of the throughput estimation model for handling 13 POCs under various number of walls interference and verify the model estimation accuracy.

4.1 Introduction

The high-speed Internet access has always been an issue in WLAN systems. The user in WLAN may perform various high-load jobs such as downloading bulk files and watching live videos simultaneously [92]. To meet these demands, the analysis and the estimation of the throughput performance of a WLAN are very important for improving and/or developing protocols or algorithms for a WLAN system.

To design an efficient WLAN system, the accurate throughput estimation model is essential to evaluate the performance of the current configuration of the system. However, previous experimental data using a real WLAN indicated that these models are largely inaccurate [93, 94].

In Chapter 3, the *throughput estimation model* is introduced. It estimates the *received signal strength (RSS)* at the host using the *log distance path loss model* and converts it to the throughput using the *sigmoid function* in Section 3.3. Then, the *throughput drop estimation model* has been studied in Section 3.4, 3.5, 3.6 for interfered link under CB, Non-CB, and both coexistence of CB and Non-CB respectively

An empirical model, in contrast to a physical model, is based on observations from actual network environments. As a result, the empirical model for interference in WLAN is expected to be more descriptive and accurate than a physical model. Recently, the Raspberry Pi has gained global popularity due to its low cost, compact size, and powerful computing capabilities. It is equipped with an IEEE 802.11n wireless *network interface card (NIC)*, allowing it to function as a software access point (AP) for WLAN.

Through experiments, it was observed that concurrent links with Raspberry Pi AP for both CB and non-CB AP exhibit different throughput characteristics compared to single-link communication. We also found that as the number of walls or the physical distance between links increases, the interfering signal weakens, leading to the improved throughput. Because of this, in this chapter, I present the modifications of the throughput estimation model under both 11 and 13 POCs links [26, 27]. Furthermore, I extend the 13 POCs model under the interference of wall effects.

4.2 Throughput Measurement

In this section, I present the throughput measurement setup and the network topologies for concurrently communicating links that are adopted in this thesis under up to three *Raspberry Pi* APs for 11 and 13 POCs.

4.2.1 Measurement Setups

Table 4.1 shows the adopted devices and software specifications in the measurement. Table 4.2 reveals the specifications of the adopted wireless NIC adapters in the measurement. One combination of *Raspberry Pi* and the NIC adapter is called the *AP type*. AP types 1 and 2 use the external NIC for the CB, and AP type 3 uses the built-in NIC for the non-CB.

Table 4.1: Devices and software specifications for measurements.

PC AP	
model	Raspberry Pi 3
CPU	Broadcom BCM2837 @1.2Ghz
memory	1Gb LPDDR2 900Mhz
OS	Raspbian
AP	hostapd
PC server	
model	Fujitsu Lifebook S761/C
CPU	Intel Core i5-2520M @2.5Ghz
memory	4Gb DDR3 1333Mhz
OS	Windows 7
TCP	iperf 2.0.5
PC host	
model	Toshiba Dynabook R731/B
CPU	Intel Core i5-2520M @2.5Ghz
memory	4Gb DDR3 1333Mhz
OS	Windows 7
TCP	iperf 2.0.5

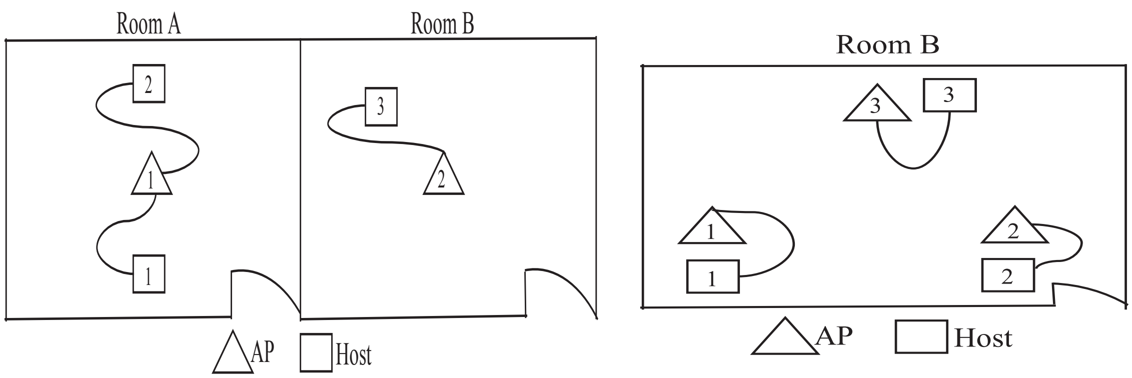
4.2.2 Network Topologies

Figure 4.1 demonstrates the network topologies for measurements on the 3^{rd} floor of Engineering Building #2 in Okayama University. A triangle represents an AP and a rectangle does a host.

Table 4.2: Wireless NIC adapters.

AP type	NIC CB	model	wireless chipset	operating mode	channel width
1	USB CB	IO-Data WN-AC433UA	Realtek RTL8811AU	2.4GHz	20MHz, 40MHz
2	USB CB	TP-LINK TL-WN722N	Atheros AR9002U	2.4GHz	20MHz, 40MHz
3	Built-in, non-CB	Raspberry Pi 3	Broadcom BCM43438	2.4GHz	20MHz

The rooms A and B have the same size of $7m \times 6m$. In any case, all the hosts are concurrently communicating with the associated APs simultaneously.



(a) Network topology for two APs case.

(b) Network topology for three APs case.

Figure 4.1: Network topology

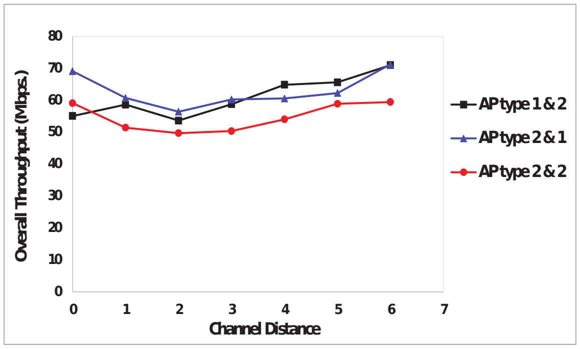
4.3 Modification of Throughput Estimation Model for 11 POCs

In this Section, I present the measurement results under 11 POCs, then modify the model in Section 3.3 based on the measurement results, and finally evaluate the accuracy of the modified model.

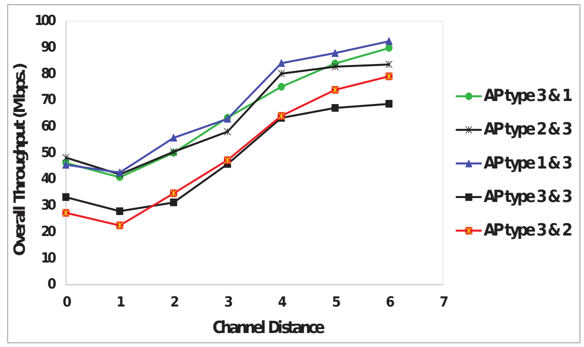
4.3.1 Measurement Results for Two APs Case

Figure 4.2 shows the overall throughput results in the two APs case for the use of only CB APs and the use of a non-CB AP respectively. In Figure 4.2 (a), the overall throughput is mostly constant regardless of the *ChD* due to the large interference among the two APs. In Figure 4.2 (b), the overall throughput is improved as the *ChD* increases, since the interference becomes lesser. The overall throughput is highest in the latter case at *APtype* = 1&3 and *ChD* = 6.

Table 4.3 exhibits the individual throughput of each AP for the four combinations of AP types. For example, the two APs in the first combination offer different throughput results for the most *ChD*, although they have the same AP type. In WLAN, this can often happen because of the unfairness of transmission opportunities between interfered links. Therefore, in this paper, we use the overall throughput of the APs in the network for evaluations.



(a) Results of CB AP only.



(b) Results of CB AP + non-CB AP.

Figure 4.2: Throughput measurement results of two APs case.

Table 4.3: Measurement results for two APs case.

AP	channel distance (*ChD*)						
type	0	1	2	3	4	5	6
2	25.70	15.40	31.74	41.34	19.2	19.99	32.14
2	26.07	35.16	21.3	11.0	38.2	40.65	26.77
2	13.84	7.75	13.03	8.2	10.8	25.4	50.76
1	55.26	52.95	43.38	52.05	49.75	36.85	20.47
2	35.3	27.21	36.65	42.55	42.18	47.25	48.46
3	12.9	14.65	13.75	15.5	37.95	35.5	35.15
3	26.61	25.29	24.11	26.88	34.72	33.09	33.97
2	18.24	18.0	22.27	31.36	43.26	50.89	50.58

4.3.2 Measurement Results for Three APs Case

Table 4.4 shows the overall throughput results in the three APs case. It indicates that no. 6 provides the best total throughput, where two non-CB APs are assigned channels 1 and 11, and one CB AP is channel 1+5 in Figure 4.3 (a). In this combination, the non-CB AP with channel 11 and the CB AP with 1+5 are not interfered.

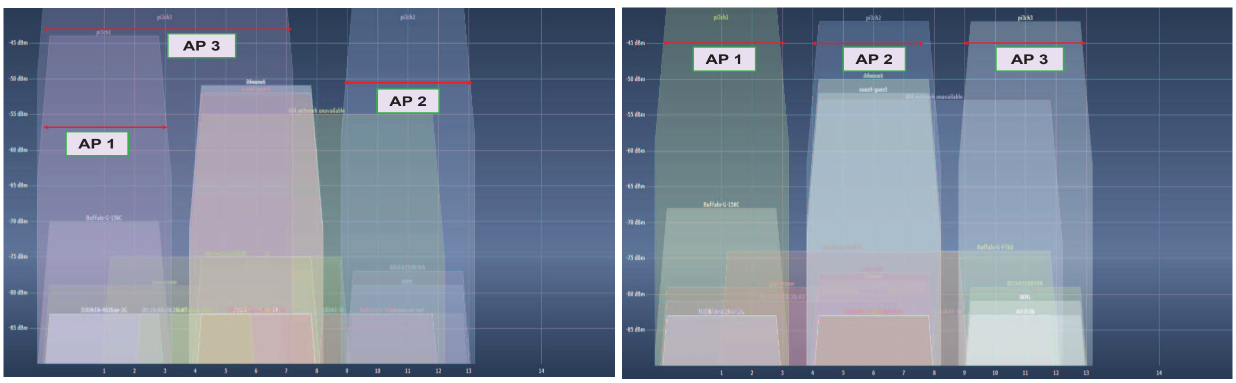
As well, no. 9 provides the similar total throughput, where three non-CB APs are assigned channels 1, 6, and 11 as the non-interfered channels in Figure 4.3 (b). Depending on the host distribution in the network field, either of no. 6 or no. 9 should be adopted. When an area in the field is congested with many hosts, no. 6 should be adopted, because the CB AP can offer higher throughput for the congested hosts than the non-CB AP. When hosts are distributed evenly, no. 9 should be adopted. Table 4.4 shows that no. 7 does the worst throughput, where three non-CB APs are assigned the same channel.

4.3.3 Model Modification for Two APs Case

In this section, we present the modifications of the throughput estimation model for the concurrent communication of two APs, based on the results in Section 4.3.1.

Table 4.4: Throughput measurement results for three APs case.

no.	CB (AP1, AP2, AP3)	channel (AP1, AP2, AP3)	throughput (Mbps)			
			AP1	AP2	AP3	total
1	all CB	1+5, 1+5, 1+5	10.58	12.35	34.85	57.78
2	all CB	1+5, 7+11, 1+5	10.49	13.5	31.63	55.62
3	non-CB, CB, CB	11, 4+8, 1+5	28.9	2.95	59.1	90.95
4	non-CB, CB, CB	11, 7+11, 1+5	38.49	12.21	46.14	96.84
5	non-CB, non-CB, CB	8, 11, 1+5	5.6	26.9	51.2	83.7
6	non-CB, non-CB, CB	1, 11, 1+5	23.62	33.87	41.4	98.90
7	all non-CB	11, 11, 11	12.2	8.94	9.43	30.58
8	all non-CB	1, 11, 11	37.15	11.13	27.78	76.06
9	all non-CB	1, 6, 11	32.6	32.9	31.5	97.00



(a) Channel condition in *no.* 6.

(b) Channel condition in *no.* 9.

Figure 4.3: Observed channel conditions at 2.4 GHz by Homedale network monitoring tool of three APs case.

4.3.3.1 CB AP Only

When only the CB APs are used, the total throughput is mostly constant at any ChD as in Figure 4.2 (a). Thus, the throughput (transmission speed) of any link is reduced by the following constant factor $F(ChD)$, called the *reduction factor*, from the one estimated by the original model.

$$F(ChD) = 0.6 \tag{4.1}$$

4.3.3.2 CB AP and Non-CB AP

When the CB AP and the non-CB AP are used, the total throughput increases as the ChD increases, as shown in Figure 4.2 (b). Thus, the throughput of any link is reduced by the reduction factor $F(ChD)$ in Eq. (4.2), from the one estimated by the original model.

$$F(ChD) = \begin{cases} 0.46, & \text{if } ChD \le 2 \\ 0.115 \times ChD + 0.26, & \text{otherwise} \end{cases} \tag{4.2}$$

4.3.4 Model Modification for Three APs Case

Next, we examine the three APs case. From the results in Table 4.4, the throughput features in the three APs case can be classified into the following four cases:

1. All the APs are CB.
2. All the APs are non-CB and assigned the same channel.
3. All the APs are non-CB and assigned the non-interfered three channels, namely 1, 6 and 11.
4. At least one AP is non-CB and assigned the non-interfered two channels.

In the first and second cases, all the APs are interfering with each other. Thus, the total throughput can be reduced by a large constant factor.

In the third case, all the APs do not interfere with each other, and the total throughput can be reduced by a small constant factor.

In the fourth case, the total throughput of the two APs that have not interfered with each other, can be reduced by a small constant factor, and the throughput of the last AP that is interfered by either of the other two APs, can be reduced by a large constant factor. It is noted that one non-CB AP and one CB/non-CB AP are not interfering with each other, if they are assigned channels 1 and 7+11/11.

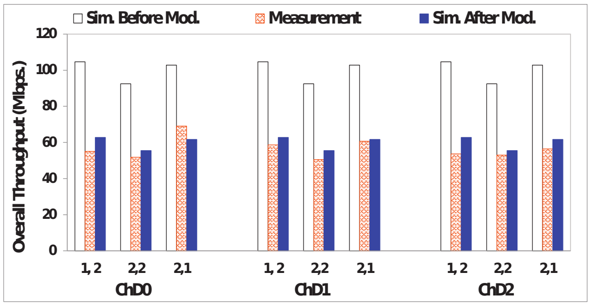
Then, by examining several values for the constant reduction factors, the following equation is introduced to approximate the total throughput using the throughput estimation model:

$$TS_{11} = \begin{cases} (x+y+z)\times\beta^2, & \text{for 1)} \\ (x+y+z)\times\gamma^2, & \text{for 2)} \\ (x+y+z)\times\alpha^2, & \text{for 3)} \\ (x+y)\times\alpha^3 + z\times\gamma, & \text{for 4)} \end{cases} \tag{4.3}$$

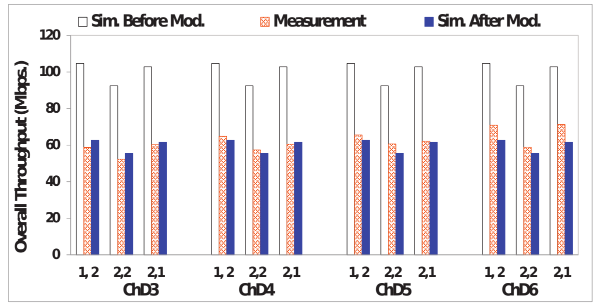
where x, y, and z represents the estimated throughput of each AP using the original throughput estimation model for the single communication. Specifically, x and y does the throughput for the two APs that have not interfered with each other, and z does the throughput for the remaining AP. α, β, and γ represents the constant reduction factors, where $\alpha = 0.95$, $\beta = 0.6$, and $\gamma = 0.46$ are used in this paper. $\beta = 0.6$ comes from Eq. (4.1) for CB AP only in the two APs case. $\alpha = 0.95$ and $\gamma = 0.46$ come from Eq. (4.2) for CB AP + non-CB AP in the two APs case with $ChD = 6$ $(0.115 \times 6 + 0.26 = 0.95)$ and with $ChD = 0$, respectively.

4.3.5 Model Evaluations for Two APs Case

Figures 4.4 and 4.5 compare the three throughput results from the original model, the measurement, and the modified model in the two APs case with CB AP only and with CB AP + non-CB AP, respectively. Table 4.5 summarizes the average and the standard deviation (SD) of the throughput estimation errors of the original model and the modified model for them. It has been proved that the modified model greatly improves the estimation accuracy in the two APs case.

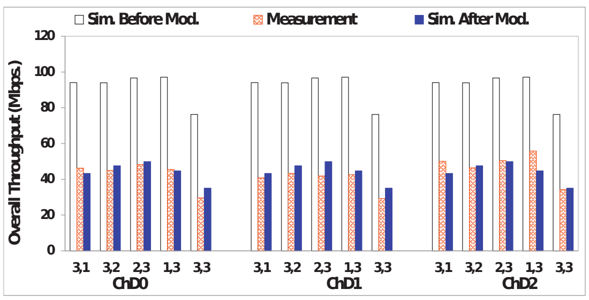


(a) Results of *ChD*=0, 1, 2.

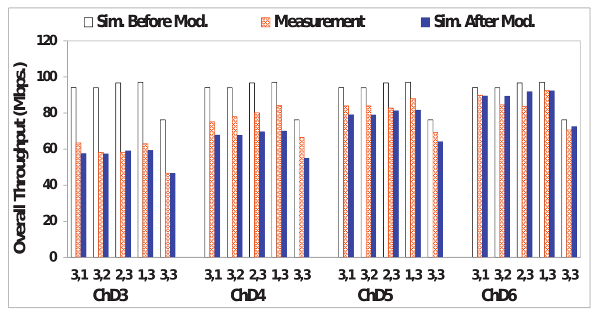


(b) Results of *ChD*=3, 4, 5, 6.

Figure 4.4: Throughput comparisons in two APs case with CB AP only.



(a) Results of *ChD*=0, 1, 2.



(b) Results of *ChD*=3, 4, 5, 6.

Figure 4.5: Throughput comparisons in two APs case with CB AP + non-CB AP.

Table 4.5: Summary of throughput estimation errors.

case		original model		modified model	
		ave. (Mbps)	SD (Mbps)	ave. (Mbps)	SD (Mbps)
2APs CB only	1, 2	43.57	6.28	5.46	2.85
	2, 2	37.52	3.96	3.53	1.19
	2, 1	39.91	5.29	3.76	3.65
2APs CB+non-CB	3, 1	29.88	19.32	4.41	2.52
	3, 2	21.29	18.53	8.88	4.85
	2, 3	21.39	18.05	8.95	5.68
	1, 3	29.79	20.74	5.49	5.35
	3, 3	26.82	19.03	4.41	3.94
3APs		54.32	26.84	6.81	5.34

4.3.6 Model Evaluations for Three APs Case

Figure 4.6 compares the three throughput results in the three APs case. In any result, the modified model can estimate the measured throughput accurately, which confirms the effectiveness of our proposal. Table 4.5 also shows the average and the standard deviation (SD) of the throughput estimation errors in this case, which indicates that the modified model greatly reduces the estimation error in the three APs case.

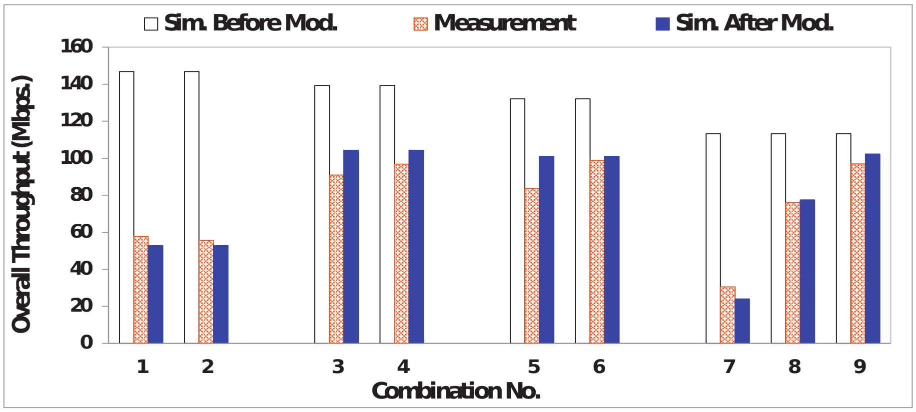


Figure 4.6: Throughput comparisons of three APs case.

4.4 Modification of Throughput Estimation Model for 13 POCs

In this section, I present the measurement results under 13 POCs, then modify the model in Section 4.3 based on the measurement results, and evaluate the accuracy of the modified model.

4.4.1 Measurement Results for Two APs Case

First, I discuss the results in the two APs case. Figure 4.7 shows the overall throughput measurement results in the two APs case with 13 POCs. When only the CB APs are used, the overall throughput is almost constant up to $ChD = 6$. For $ChD = 7$ and 8, the throughput increases because the interference between the two APs is decreased. Especially, for $ChD = 8$, the throughput is greatly improved because of the non-interference theoretically. When the CB AP and the non-CB AP are used, the total throughput increases as ChD increases. However, this increase rate becomes small after $ChD = 7$.

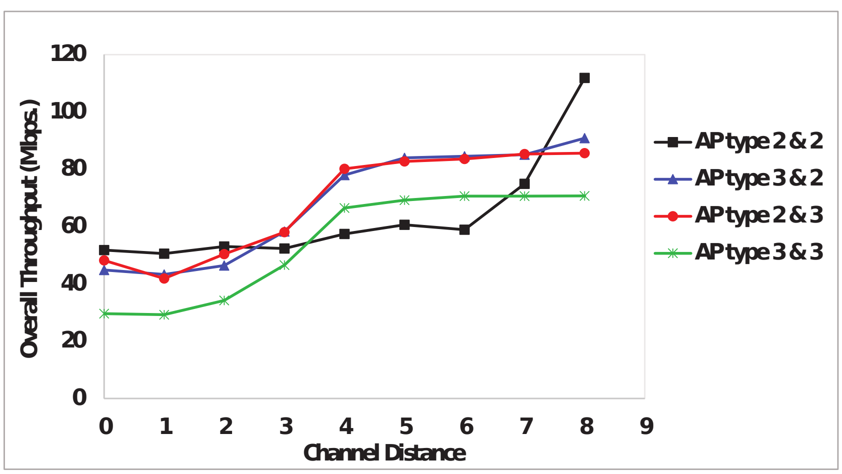


Figure 4.7: Throughput measurement results in two APs case with 13 POCs.

4.4.2 Measurement Results for Three APs Case

Next, I discuss the results in the three APs case. Table 4.6 shows the overall throughput results in the three APs case. It indicates that no. 2 provides the best total throughput, where two non-CB APs are assigned 1 and 5 for the channel, and one CB AP is 9 + 13. In this combination, all the APs use the non-interfered channels in Figure 4.8.

Table 4.6: Throughput measurement results in three APs case with 13 POCs.

no.	CB (AP1, AP2, AP3)	channel (AP1, AP2, AP3)	throughput (Mbps)			
			AP1	AP2	AP3	total
1	all non-CB	1, 9, 13	34.4	37.7	36.7	108.8
2	non-CB, non-CB, CB	5, 1, 9+13	26.6	31	61.2	118.8
3	non-CB, CB, CB	13, 9+13, 1+5	24.25	14.39	67.65	106.29

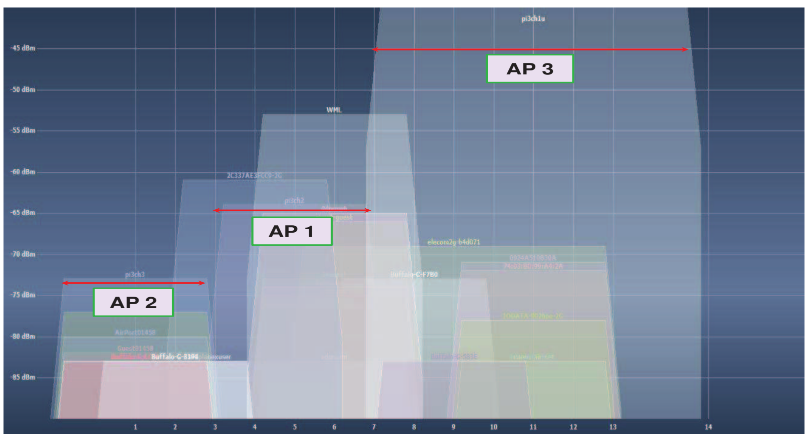


Figure 4.8: Observed channel conditions at 2.4 GHz by Homedale network monitoring tool for no. 2 combination.

4.4.3 Model Modification for Two APs Case

In this section, I present the modifications of the throughput estimation model for the concurrent communication of two APs based on the results in Section 4.4.1 under 13 POCs.

4.4.3.1 CB AP Only

When only the CB APs are used, the total throughput is almost constant up to $ChD = 6$. For $ChD = 7$ and 8, the throughput increases because the interference between the two APs is decreased. Especially, for $ChD = 8$, the throughput is greatly improved because of the non-interference theoretically. Then, the reduction factor $F(ChD)$ to estimate the transmission speed for the two CB APs is modified as follows:

$$F(ChD) = \begin{cases} 0.6, & \text{if } ChD \leq 6 \\ 0.19\times\mathrm{ChD}-0.5533, & \text{if ChD} > 6 \end{cases} \tag{4.4}$$

4.4.3.2 CB AP and Non-CB AP

When the CB AP and the non-CB AP are used, the total throughput increases as *ChD* increases. Thus, the throughput of any link is reduced by the reduction factor $F(ChD)$ in Eq. (4.5) by extending Eq. (4.2), from the one estimated by the original model.

$$F(ChD) = \begin{cases} 0.46, & \text{if } ChD \leq 2 \\ 0.115\times\text{ChD}+0.26, & \text{if } 3 \leq ChD \leq 6 \\ 0.005\times\text{ChD}+0.92, & \text{if ChD} \geq 7 \end{cases} \tag{4.5}$$

4.4.4 Model Modification for Three APs Case

Finally, I examine the measurement results for the three APs case. From the results in Table 4.6, the throughput features can be classified into the following three cases:

1. All the APs are non-CB, and are assigned the non-interfered three channels.
2. One APs is CB and two APs are non-CB, and are assigned the non-interfered three channels.
3. At least one AP is non-CB, and the APs are assigned the non-interfered two channels.

Then, the following equation is introduced to approximate the total throughput using the throughput estimation model:

$$TS_{13} = \begin{cases} (x + y + z) \times \alpha^2, & \text{for case 1) \& 2)} \\ (x + y) \times \alpha^3 + z \times \gamma, & \text{for case 3)} \end{cases} \tag{4.6}$$

where x, y, and z represents the estimated throughput of each AP using the throughput estimation model for the single communication. Specifically, for case 3), x and y does the throughput for the two APs that are not interfered with each other, and z does the throughput for the remaining AP. α and γ represent the constant reduction factors, where $\alpha = 0.96$, and $\gamma = 0.46$.

4.4.5 Model Evaluations for Two APs Case

Figure 4.9 compares the total throughput results by the original model, the measurement, and the proposed model, for the CB AP only, and for the CB AP and the non-CB AP, in the two APs case with 13 POCs, respectively. It indicates that our modified throughput estimation model can estimate the throughput with the high accuracy.

4.4.6 Model Evaluations for Three APs Case

Figure 4.10 compares the total throughput results by the original model, the measurement, and the modified model for the CB AP only, and for the CB AP and the non-CB AP, in the three APs case with 13 POCs, respectively. Again, it indicates that our modified throughput estimation model can estimate the throughput with the high accuracy.

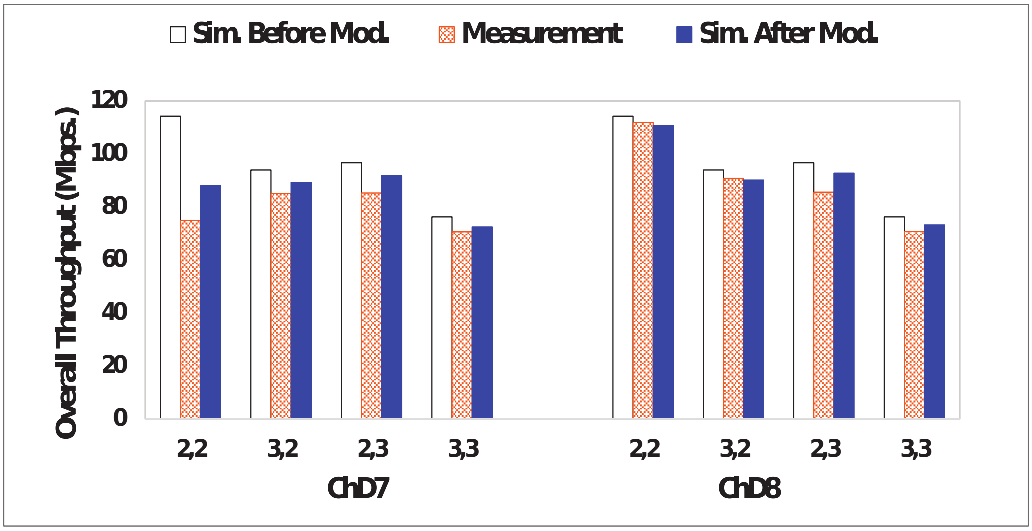


Figure 4.9: Throughput comparisons at *ChD* = 7 and 8 for two APs case with 13 POCs.

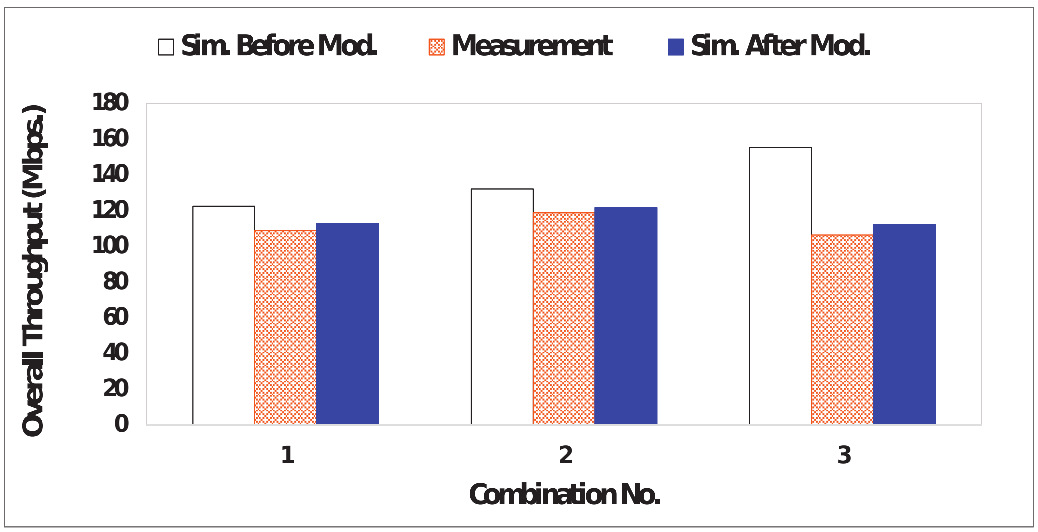


Figure 4.10: Throughput comparisons of three APs case with 13 POCs.

4.5 Extension of Throughput Estimation Model for 13 POCs

In this section, I present the extension of the throughput estimation model under wall interference effect for 13 POCs. Firstly, we discuss the throughput measurement results under various numbers of wall interference. Next, I present the model extension under various numbers of wall interference. Lastly, I verify the extended model estimation accuracy.

4.5.1 Motivation for Model Extension

In Section 4.4, previous *throughput estimation model* for concurrent communications of three *Raspberry Pi APs* is based on the preliminary throughput measurement results which were conducted in a single room simple network topology. In this topology, I placed three *APs* at some fixed physical distances between them (Euclidean Distance between the two links) where there is no wall obstacle among *APs*. I conducted experiments using various channel distance among APs.

Unfortunately, this estimation model did not consider interference effects due to the various number of wall obstacles among *APs* in a complex network topology. In a practical network scenario, *APs* are often located in different rooms in the network field. When the number of wall obstacles between *APs* rise it reduces the interference among *APs* due to the larger physical distance among them. By increasing the number of walls or the physical distance between the links, the interfering signal between them weakens due to the *path loss* and the absorption by wall obstacles on the path. These effects must be considered in the throughput estimation model for accurate throughput estimation of multiple links in practical network scenarios.

4.5.2 Throughput Measurement under Various Number of Walls Interference

In this section, I present the experiment results to verify the modifications of the throughput estimation model for three *Raspberry Pi APs* by considering the interference effects under various number of wall obstacles and physical distance among *APs*.

4.5.2.1 Measurement Setup

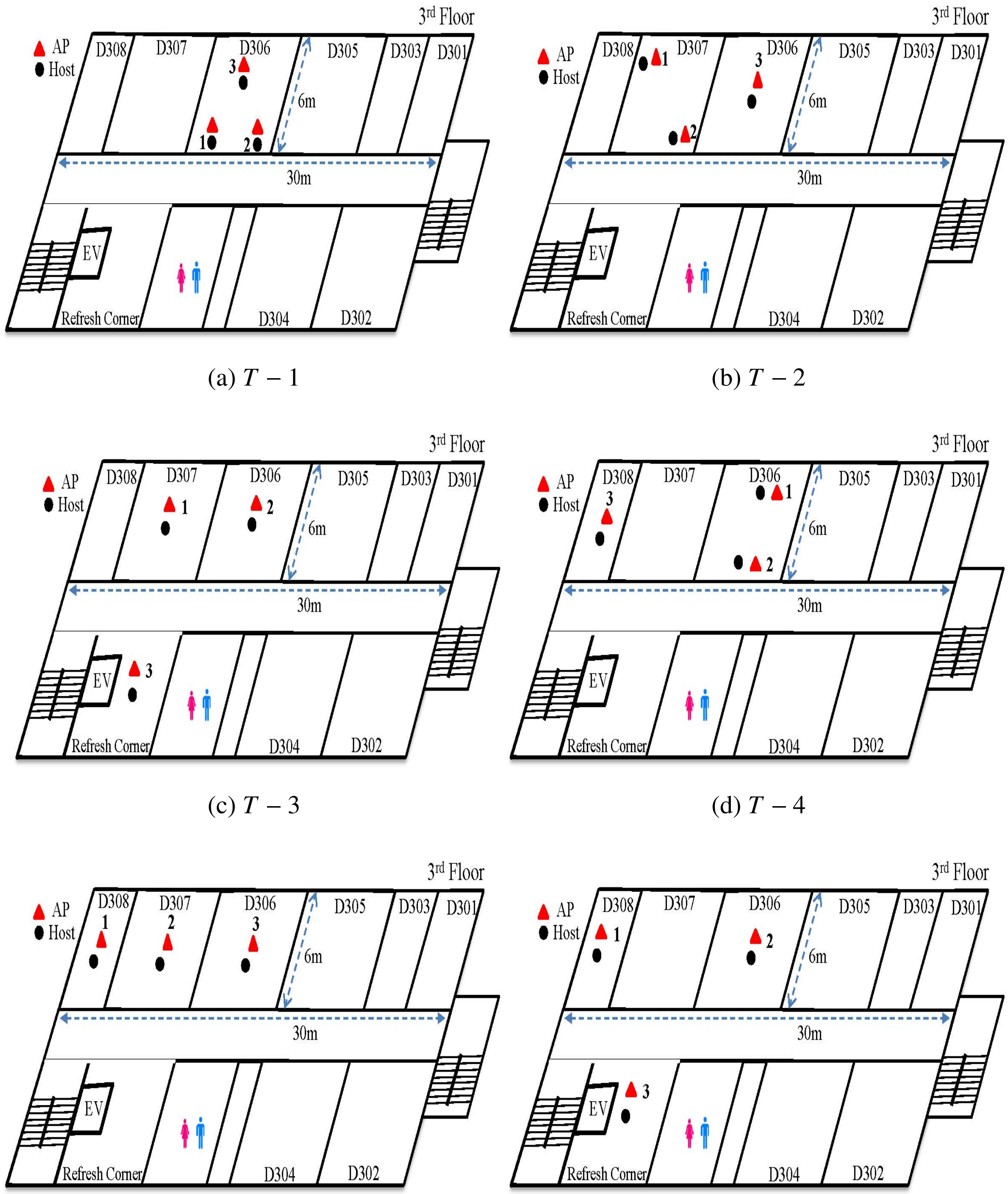
The experiments were carried out in two different indoor fields. *Field* #1 represents the 3^{rd} floor of *Engineering Building* #2 and field #2 does the 2^{nd} floor of Graduate School of Natural Science and Technology Building in Okayama University. Each network field consists of several rooms portioned by different type of walls, which can affect the throughput through the multi-path effect. The throughputs are measured when we change the physical distance among the three APs by varying the number of walls among them. In field #1, we adopt 6 topologies ($T-1$ to $T-6$) and in field #2, we consider 9 topologies *($T-1$ to $T-9$)*.

Figure 4.11 and 4.12 reveals all the topologies the field #1 and field #2 that we consider for experiments. Figure 4.13 reveals the channel configuration for measurements, where the best two channel configurations (case 1 and case 2) are adopted. For case 1, two *APs* use *non-CB* channel *(channel* 1 *and channel* 5*)* and the third *AP* uses *CB* channel *(channel* 9 + 13*)*.

Table 4.7 indicates the necessary devices and software used in the experiments. It is noticed that if different devices are used, the parameter values of the model will vary accordingly.

4.5.2.2 Throughput Measurement Results

In the experiments, I assigned the *non-interfered* or less interfered channels to the most nearest two *APs* (smallest wall count between two *APs*), and assigned the other channel to the rest *AP*. Figures 4.14 and 4.15 show the overall throughput measurement results of various topologies in network *field*#1 and *field*#2 under concurrent communications by three *Raspberry Pi APs* for both case 1 and case 2, respectively. From the measurement results, we observed that the *T*-1 topologies in field#1 and field#2 provide the lowest throughput for both cases. Because in this case, there is no wall obstacles between *APs* which causes larger interference among APs due to the smallest physical distance among them. The total throughput is gradually improved as the number of walls obstacles or physical distance among *APs* is increased by placing three *APs* in different rooms in the network field. From our measurement results, it is also observed that the maximum throughput is achieved for *T*-6 topology in field#1 and *T*-9 topology in field#2 due to the largest number of



(a) $T-1$ (b) $T-2$

(c) $T-3$ (d) $T-4$

(e) $T-5$ (f) $T-6$

Figure 4.11: Topologies in network field#1.

walls in terms of the minimum number of walls and the average number of walls or the largest physical distances among the *APs*, which make less channel interference among them.

4.5.3 Extension of Throughput Estimation Model for 13 POCs

In this section, I describe the extension of the throughput estimation model for 13 POCs. The extended model estimates the throughput by considering the following procedure:

1. Estimate the throughput of each link between an *AP* and a host using the estimation model for single link in section 3.3.

Table 4.7: Devices and software specifications for experiments.

PC AP	
model	Raspberry Pi 3
CPU	Broadcom BCM2837 @1.2Ghz
memory	1Gb LPDDR2 900Mhz
OS	Raspbian
AP	hostapd
NIC	TP-LINKTL-WN722N
PC server	
model	Fujitsu Lifebook S761/C
CPU	Intel Core i5-2520M @2.5Ghz
memory	4Gb DDR3 1333Mhz
OS	Windows 7
TCP	iperf 2.0.5
PC host	
model	Toshiba Dynabook R731/B
CPU	Intel Core i5-2520M @2.5Ghz
memory	4Gb DDR3 1333Mhz
OS	Windows 7
TCP	iperf 2.0.5

2. Calculate the number of walls (n_W) for each pair of *APs (AP1, AP2), (AP1, AP3), (AP3, AP1)* among three *APs* and find the minimum number of walls, (n_{Wmin}) in the network field. Then, we estimate the total number of walls n_{wT}, as

$$n_{wT} = \sum\nolimits_{i=0,i\neq j}^{2} n_{wij} \tag{4.7}$$

where $n_{w_{ij}}$ represents the number of walls between the ith and jth *AP* pair.

3. Estimate the average number of walls n_{wA} for each pair of *APs* by using the following equation:

$$n_{wA} = \begin{cases} \frac{1}{3} n_{wT}, \text{for case 1 } n_{Wmin.} \geq 0 \\ \frac{1}{3} n_{wT}, \text{for case 2 } n_{Wmin.} < 2 \\ 1 + \frac{1}{3} n_{wT}, \text{for case 2 } n_{Wmin.} = 2 \end{cases} \tag{4.8}$$

where n_{wT} is the total number of walls among all pair of *APs* in the network field.

4. Based on the average number of walls between *AP* pairs, estimate the throughput increasing factor, $f(n_{wA})$ as

$$f(n_{wA}) = \begin{cases} (0.153 \times n_{wA}) - 0.1271, & \text{for case 1} \\ (0.129 \times n_{wA}) - 0.1213, & \text{for case 2} \end{cases} \tag{4.9}$$

5. Estimate the total throughput for concurrent communications of three *Raspberry Pi APs* by considering the interference effects due to the number of walls between them as

$$TS_{13}^{new} = \{TS_{13} \times (1 + f(n_{wA})) \tag{4.10}$$

where TS_{13} represents the total throughput estimated by the previous for 13 POCs model and TS_{13}^{new} does the estimated throughput by the modified model for 13 POCs.

4.5.4 Evaluations in Field#1

Figure 4.16 compares the three throughput results by the previous model, the measurement, and the modified model for both cases (case 1, case 2) in field#1. Table 4.8 summarizes the average and the *standard deviation (SD)* of the throughput estimation errors of the previous model and the modified model for them. It has been proved that the modified model greatly improves the estimation accuracy in both two cases as compared to the previous model.

Table 4.8: Summary of throughput estimation errors.

Case	Previous model		Modified model	
	Avg. error	SD	Avg. error	SD
1	8.05	6.12	5.73	3.84
2	7.99	7.42	5.34	4.47

4.5.5 Evaluations in Field#2

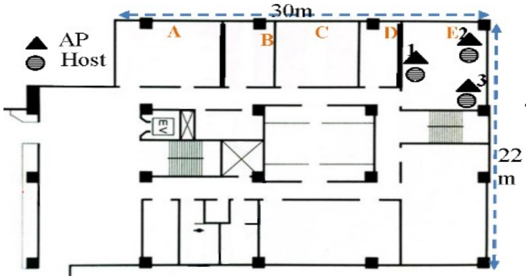
Figure 4.17 compare the three throughput results by the previous model, the measurement, and the modified model for both cases in field#2. Table 4.9 summarizes the average and the *standard deviation (SD)* of the throughput estimation errors of the previous model and the modified model for them. Similarly, it has been proved from table 4.9 that the modified model greatly improves the estimation accuracy in both two cases.

Table 4.9: Summary of throughput estimation errors.

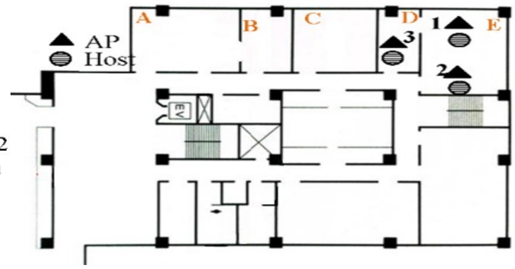
Case	Previous model		Modified model	
	Avg. error	SD	Avg. error	SD
1	30.53	14.93	7.37	4.98
2	15.89	19.72	5.95	4.67

4.6 Summary

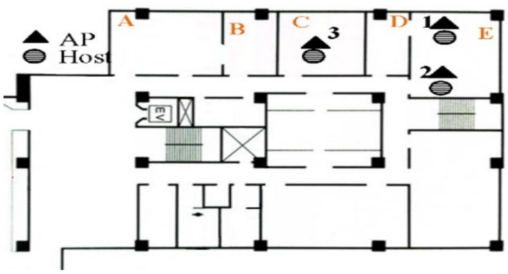
In this chapter, I presented the modified *throughput estimation models* under various conditions for concurrent communications of up to three *Raspberry Pi* APs. Firstly, I discussed the throughput measurement results and modified the existing *throughput estimation model* under concurrent communications with 11 POCs. Secondly, I verified the model estimation accuracy for 11 POCs. Thirdly, I discussed the throughput measurement results and modified the existing *throughput estimation model* under concurrent communications with 13 POCs. Fourthly, I verified the model estimation accuracy for 13 POCs. Finally, I presented the extension of the throughput estimation model for 13 POCs under various number of walls interference and verified the model estimation accuracy. In the next chapter, I will present the model applications to network configuration optimization algorithm.



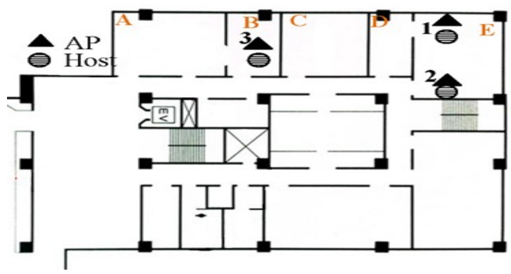
(a) $T-1$



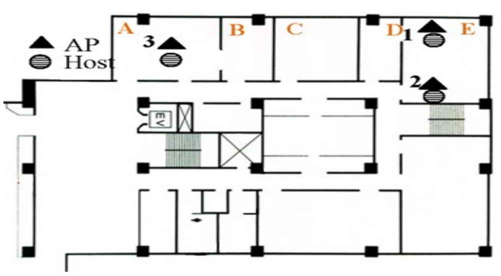
(b) $T-2$



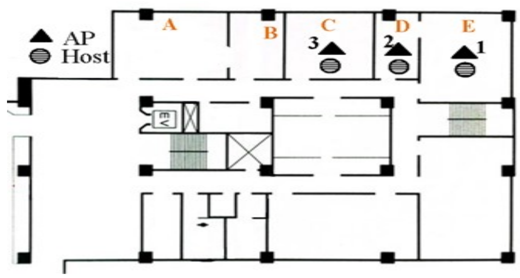
(c) $T-3$



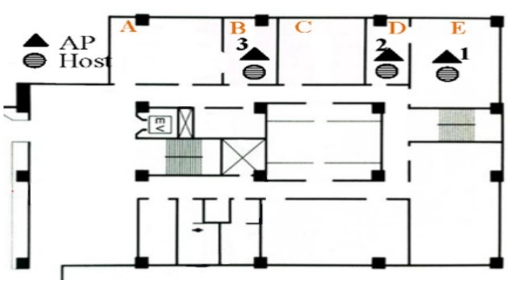
(d) $T-4$



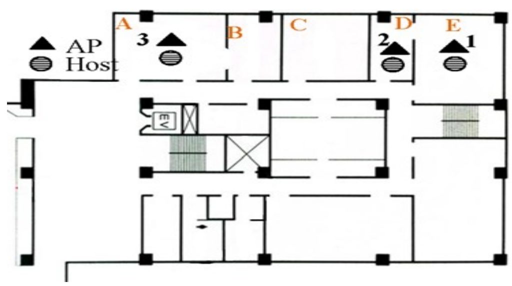
(e) $T-5$



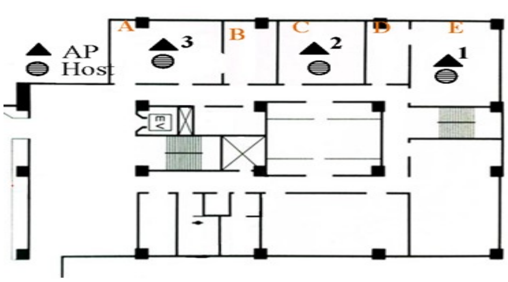
(f) $T-6$



(g) $T-7$

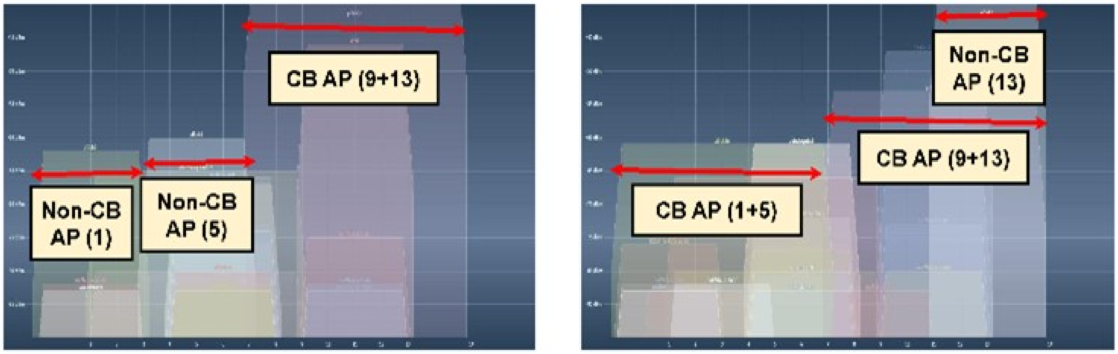


(h) $T-8$



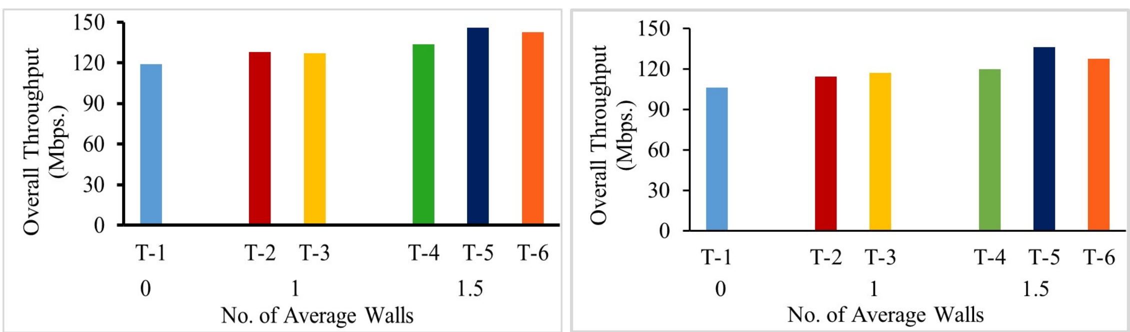
(i) $T-9$

Figure 4.12: Topologies in network field#2.



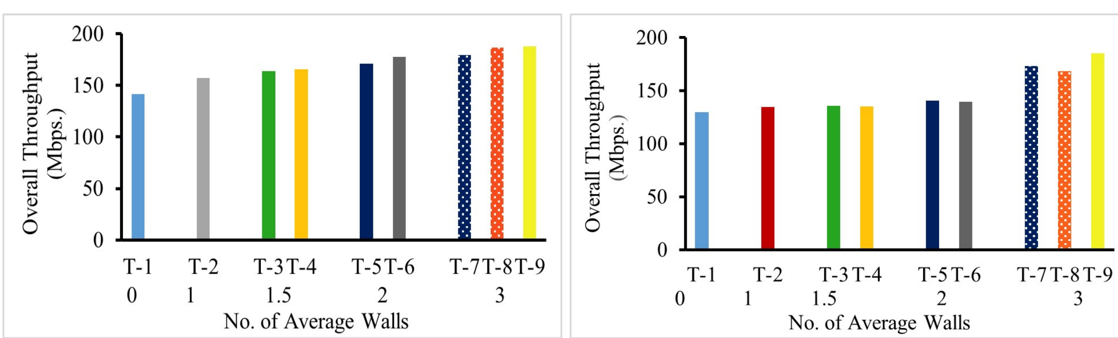
(a) Case 1 (b) Case 2

Figure 4.13: Channel configuration in experiments



(a) Case 1 (b) Case 2

Figure 4.14: Average number of walls vs. total throughput for field#1.



(a) Case 1 (b) Case 2

Figure 4.15: Average number of walls vs. total throughput for field#2.

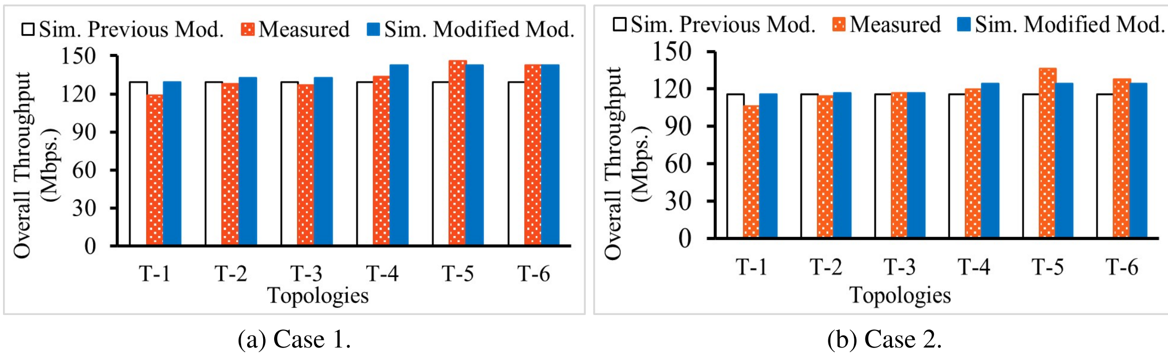


Figure 4.16: Throughput comparisons in field#1.

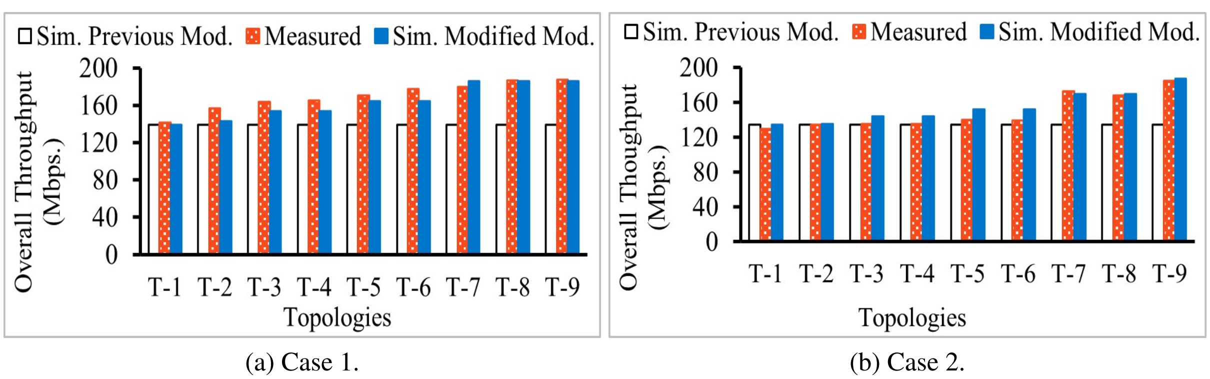


Figure 4.17: Throughput comparisons in field#2.

Chapter 5

Model Applications to Network Configuration Optimization Algorithm

In this chapter, I present the *network configuration optimization algorithm* based on the modified throughput estimation model for concurrent communications of multiple hosts. Firstly, I formulate the network configuration optimization problem to design the algorithm. Secondly, I describe the procedure for the network configuration optimization algorithm for the WLAN system. Thirdly, I evaluate the application of *modified throughput estimation model* for *concurrent communications* with upto three *Raspberry Pi* APs to the *network configuration optimization algorithm* through simulations and testbed experiments for both 11 POCs and 13 POCs. Finally, I evaluate the application of *extended throughput estimation model* for 13 POCs to the *network configuration optimization algorithm* through simulations and testbed experiments.

5.1 Introduction

In Sections 4.3, 4.4 and 4.5, I presented the modifications of the *throughput estimation model* for *concurrent communications* using different wireless NIC adapters and assigned channels with 11 POCs and 13 POCs respectively, based on the measurement results. To improve the network performance, the optimal configuration of a network is very essential. In WLAN, the network demands always change. Under concurrent communications, it is very challenging to configure the network.

However, according to the experiment results, it has been noticed that the two and three configurations for 11 and 13 POCs give the best throughput results in Sections 4.3 and 4.4, respectively. To select the best network configuration, I propose the network configuration optimization algorithm considering CB or non-CB, channel assignments, and associated hosts for each AP.

In this chapter, I present the *network configuration optimization algorithm* for IEEE 802.11n WLAN with three Raspberry Pi APs by utilizing the modified throughput estimation model. For a given network field, it selects the CB or non-CB, channel assignment, and associated hosts for each AP, to maximize the throughput performance. Next, I evaluate the application of the modified and extended *throughput estimation model* for *concurrent communications* to the *network configuration optimization algorithm* through simulations and testbed experiments, respectively.

5.2 Problem Formulation

The network configuration optimization algorithm for concurrent communications with three APs is formulated as follows:

1. Inputs:
   - coordinates of three APs in the field
   - coordinates of H hosts in the field
   - coordinates of walls in the field
   - parameters of the throughput estimation model

2. Outputs:
   - CB/no-CB assignment to each AP.
   - channel assignment to each AP.
   - set of the associated hosts with each AP.

3. Objectives
   - to maximize the cost function E:

$$E = E_{21} \times E_{22} \tag{5.1}$$

where E_{21} is the minimum AP average host throughput and E_{22} is the total AP throughput. E_{21} and E_{22} can be calculated as follows:

$$E_{21} = \min \left[TH_j \right] \tag{5.2}$$

$$TH_j = k \times \frac{1}{\sum \frac{1}{TH_{ij}}} \tag{5.3}$$

where TH_j represents the estimated average throughput of a host associated with the j-th AP, TH_{ij} does the estimated throughput of the i-th associated host with the AP, and k does the reduction factor (α^2, α^3, or γ) in the model.

$$E_{22} = \sum k \times \frac{H_j}{\sum \frac{1}{TH_{ij}}} \tag{5.4}$$

where H_j represents the number of associated hosts with the j-th AP.

5.3 Procedure for Network Configuration

According to the algorithm formulation, in this section, I present the procedure of the network configuration optimization algorithm for concurrent communications with three APs.

5.3.1 Initialization

The throughput of the single link communication for each pair of an AP and a host with both CB and non-CB is estimated using the throughput estimation model in Section 3.3. Then, for each host, the AP whose link has the largest throughput will be selected for the initially associated AP for both CB and non-CB. After that, the APs are sorted in descending order of the number of initially associated hosts, where the tie is resolved in ascending order of the average host throughput.

5.3.2 Assignment for Configuration 1)

For 11 POCs, by the sorted order of the APs, the CB/non-CB and the channel are assigned to each AP for configuration 1):

- 1st AP: CB & channel 1 + 5
- 2nd AP: non-CB & channel 11.
- 3rd AP: non-CB & channel 1.

Then, for 13 POCs, by the sorted order of the APs, the CB/non-CB and the channel are assigned to each AP for configuration 1):

- 1st AP: CB & channel 9 + 13
- 2nd AP: non-CB & channel 1.
- 3rd AP: non-CB & channel 5.

5.3.3 Assignment for Configuration 2)

For 11 POCs, by the sorted order of the APs, the CB/non-CB and the channel are assigned to each AP for configuration 2):

- 1st AP: non-CB & channel 1
- 2nd AP: non-CB & channel 6.
- 3rd AP: non-CB & channel 11.

Then, for 13 POCs, by the sorted order of the APs, the CB/non-CB and the channel are assigned to each AP for configuration 2):

- 1st AP: CB & channel 1 + 5
- 2nd AP: CB & channel 9 + 13.
- 3rd AP: non-CB & channel 13.

5.3.4 Assignment for Configuration 3)

Additionally, for 13 POCs, by the sorted order of the APs, the CB/non-CB and the channel are assigned to each AP for the configuration 3):

- 1st AP: non-CB & channel 1
- 2nd AP: non-CB & channel 13.
- 3rd AP: non-CB & channel 8.

5.3.5 Configuration Improvement

For each configuration, the assignment for each AP is improved by repeating the following procedure in the given times:

1. Randomly select one host.
2. Randomly select one objective function among E_{21}, E_{22} and E.
3. Find the AP such that the selected function is maximized if the host is associated with it.
4. Associate the host with this AP if E with this association is not decreased.

5.3.6 Solution Selection

From the two configuration results, the one that has the larger E is selected as the final solution.

5.4 Evaluations by Simulations

In this section, I evaluate the *network configuration optimization algorithm* considering the modified model through simulations using the *WIMNET simulator* [30].

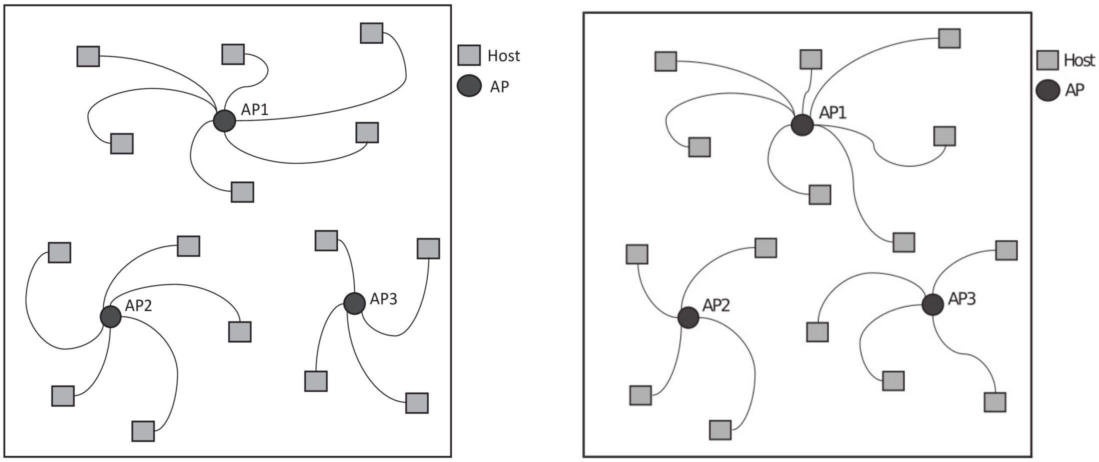
5.4.1 Evaluation for One Room Field

First, the network field of one 100m $\times$ 100m room with three APs and 15 hosts are adopted in simulations.

5.4.1.1 Results for Uniform Topology

Figure 5.1 shows the *one-room uniform topology*. The 15 hosts and the three APs are *evenly distributed*, where the circle represents the AP and the square does the host. Figure (a) and Figure (b) also suggests the associations between APs and hosts found by the algorithm for this topology with 11 POCs and 13 POCs respectively.

Table 5.1 shows the cost function E for each configuration found by the algorithm and the overall throughput by the simulator for this topology. For comparisons, it also shows the results for the configurations with the same channel assignments but the nearest host associations. It indicates that for the one-room uniform network topology, the configuration with three non-CB APs for 11 POCs and that with one CB AP ($AP1$) and two non-CB APs for 13 POCs by the algorithm provides the highest overall throughput respectively.



(a) Simulations with 11 POCs. (b) Simulations with 13 POCs.

Figure 5.1: One-room uniform topology for simulations.

Table 5.1: Simulation results for one-room uniform topology.

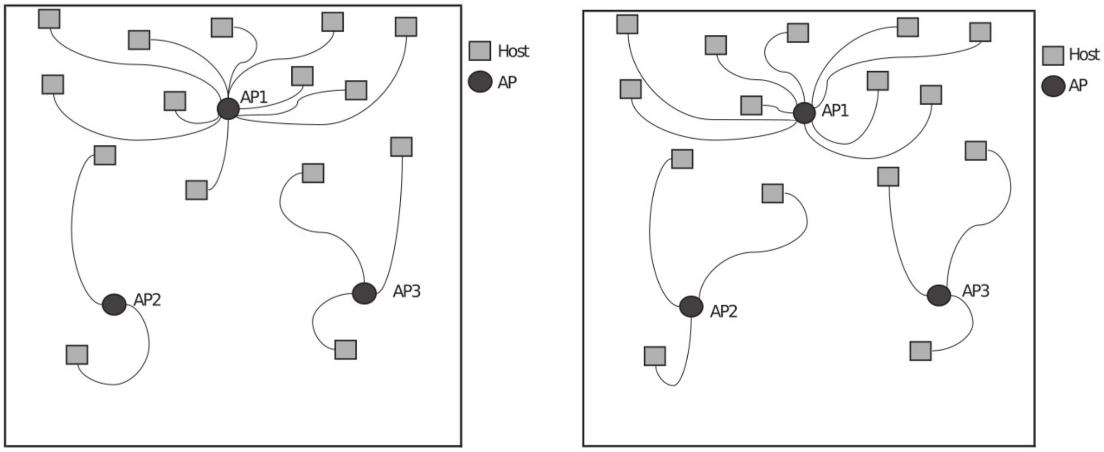
case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	471.61	460.33	94.10	90.99
	CB + 2 non-CB (1+5, 11, 1)	380.27	371.0	91.69	90.17
13 POCs	CB + 2 non-CB (9+13, 5, 1)	767.92	767.58	112.69	110.77
	2CB + 1 non-CB (1+5, 13, 9+13)	605.42	462.42	110.85	109.74
	3 non-CB (1, 8, 13)	511.33	491.32	95.19	93.55

5.4.1.2 Results for Non-Uniform Topology

Figure 5.2 shows the *one-room non-uniform network topology*, where the hosts are *disproportionately* distributed. Table 5.2 shows the simulation results for this topology. For this one-room non-uniform topology, the configuration with one CB AP and two non-CB APs provides the highest overall throughput for both 11 and 13 POCs.

5.4.1.3 Results for Two-Crowded APs Topology

Figure 5.3 shows the *one-room two-crowded APs network topology*. The hosts are also *disproportionately* distributed such that two APs can be more crowded with their associated hosts than the remaining AP. Table 5.3 shows the simulation results for this topology. For this topology, the configuration with one CB AP ($AP1$) and two non-CB APs for 11 POCs and that with one non-CB AP ($AP3$) and two CB APs for 13 POCs by the algorithm provides the highest overall throughput respectively.

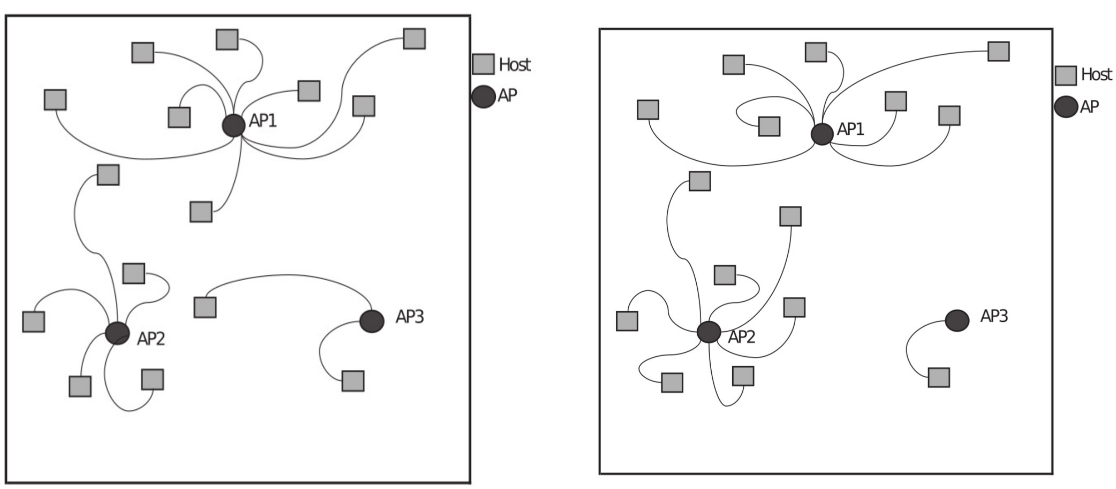


(a) Simulations with 11 POCs. (b) Simulations with 13 POCs.

Figure 5.2: One-room non-uniform topology for simulations.

Table 5.2: Simulation results for one-room non-uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	296.37	245.22	85.15	84.91
	CB + 2 non-CB (1+5, 1, 11)	393.79	359.18	89.28	87.52
13 POCs	CB + 2 non-CB (9+13, 1, 5)	573.59	475.07	108.54	106.49
	2CB + 1 non-CB (1+5, 13, 9+13)	541.62	449.18	107.93	104.98
	3 non-CB (1, 13, 8)	309.04	255.71	91.53	90.04



(a) Simulations with 11 POCs. (b) Simulations with 13 POCs.

Figure 5.3: One-room two-crowded APs topology for simulations.

Table 5.3: Simulation results for one-room two-crowded APs topology.

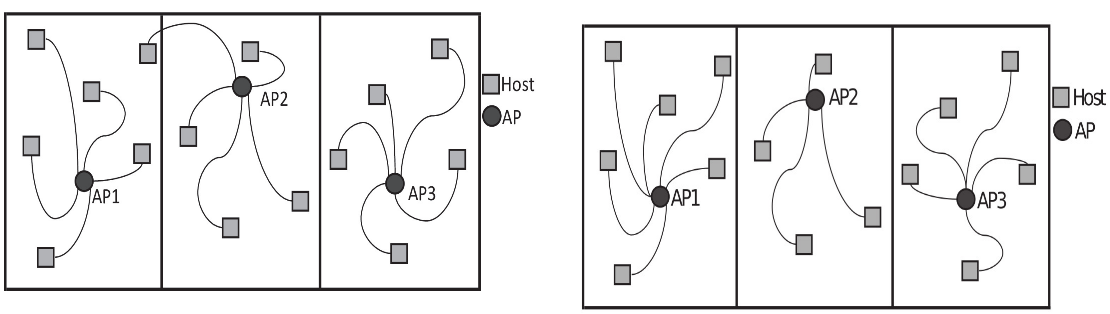
case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	421.18	373.13	93.50	92.23
	CB + 2 non-CB (1+5, 11, 1)	522.86	490.83	94.16	92.51
13 POCs	CB + 2 non-CB (9+13, 1, 5)	688.73	655.29	112.37	108.75
	2CB + 1 non-CB (1+5, 9+13, 13)	755.82	670.38	112.92	109.83
	3 non-CB (1, 13, 8)	439.19	389.09	98.65	97.81

5.4.2 Evaluation for Three Room Field

Next, the network field of three 40m × 60m rooms with three APs and 15 hosts are adopted in simulations.

5.4.2.1 Results for Uniform Topology

Figure 5.4 shows the *three-room uniform network topology*. Table 5.4 shows the simulation results. For the three-room uniform topology, the configuration with three non-CB APs for 11 POCs and that with one CB AP and two non-CB APs for 13 POCs provides the highest overall throughput respectively.



(a) Simulations with 11 POCs.

(b) Simulations with 13 POCs.

Figure 5.4: Three-room uniform topology for simulations.

Table 5.4: Simulation results for three-room uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	587.89	494.78	94.11	91.15
	CB + 2 non-CB (1+5, 1, 11)	380.59	369.59	92.53	91.03
13 POCs	CB + 2 non-CB (9+13, 5, 1)	754.95	754.95	114.51	114.51
	2CB + 1 non-CB (1+5, 13, 9+13)	609.05	461.69	110.82	108.60
	3 non-CB (1, 8, 13)	515.94	515.94	96.27	96.27

5.4.2.2 Results for Non-Uniform Topology

Figure 5.5 shows the *three-room non-uniform network topology*. Table 5.5 shows the simulation results. For the three-room non-uniform topology, the configuration with one CB AP and two non-CB APs achieves the height throughput for both 11 and 13 POCs.

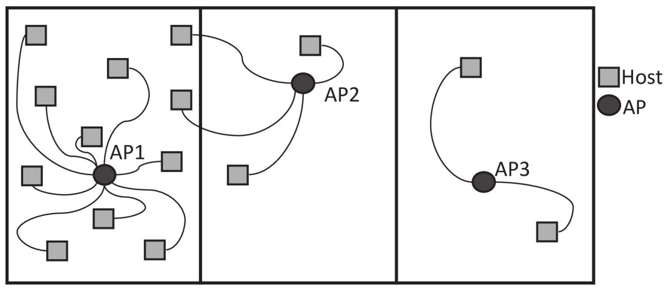


Figure 5.5: Three-room non-uniform topology for simulations with 11 POCs and 13 POCs.

Table 5.5: Simulation results for three-room non-uniform topology.

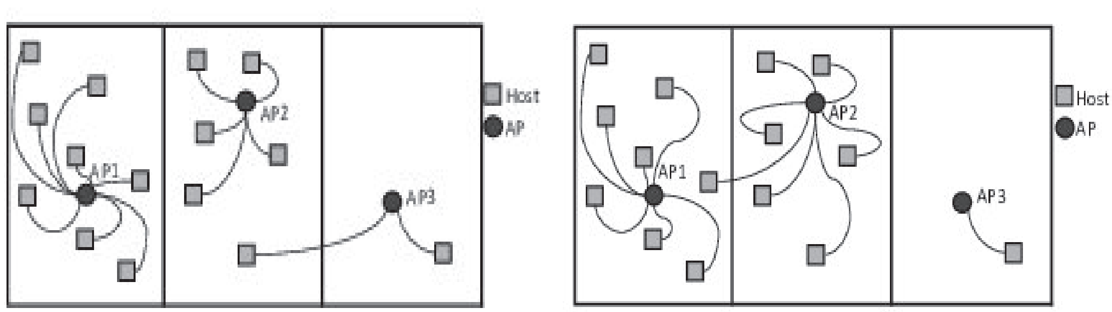
case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	315.80	313.14	92.66	91.63
	CB + 2 non-CB (1+5, 11, 1)	503.69	456.86	95.58	94.33
13 POCs	CB + 2 non-CB (9+13, 1, 5)	667.87	605.04	115.22	113.50
	2CB + 1 non-CB (1+5, 9+13, 13)	612.54	557.78	109.97	107.59
	3 non-CB (1, 13, 8)	402.15	329.31	92.69	91.44

5.4.2.3 Results for Two-Crowded APs Topology

Figure 5.6 shows the *three-room two-crowded APs network topology*, where two APs can be more crowded. Table 5.6 shows the simulation results. For the three-room two-crowded APs topology, the configuration with one CB AP and two non-CB APs for 11 POCs and that with one non-CB AP and two CB APs for 13 POCs by the algorithm provides the highest overall throughput respectively.

Table 5.6: Simulation results for three-room two-crowded APs topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E		over. through. (Mbps)	
		algo.	near.	algo.	near.
11 POCs	3 non-CB (1, 6, 11)	426.28	406.67	90.53	90.01
	CB + 2 non-CB (1+5, 11, 1)	561.94	503.39	91.57	90.85
13 POCs	CB + 2 non-CB (9+13, 1, 5)	719.82	668.01	110.48	109.66
	2CB + 1 non-CB (1+5, 9+13, 13)	771.59	724.98	113.06	111.75
	3 non-CB (1, 13, 8)	444.52	424.06	98.48	94.06



(a) Simulations with 11 POCs.

(b) Simulations with 13 POCs.

Figure 5.6: Three-room two-crowded APs topology for simulations.

5.5 Evaluations by Testbed Experiments

In this section, I evaluate the network configuration optimization algorithm through experiments using the testbed in two network fields in different buildings at Okayama University, with the uniform, non-uniform, and two-crowded APs topologies.

5.5.1 Hardware and Software in Testbed

Table 5.7 shows the specifications of the hardware and software in the testbed.

5.5.2 Experiments in Building A Field

First, the network field of two 7m × 6m closed rooms and one open room in Engineering Building 2 (Building A) is adopted in experiments.

5.5.2.1 Results for Uniform Topology

Figure 5.7 shows the *Building-A uniform topology*. Two hosts are located in D306 and D307 respectively, one host is in Refresh Corner. Each host is associated with the AP in the same room by both the algorithm and the nearest association.

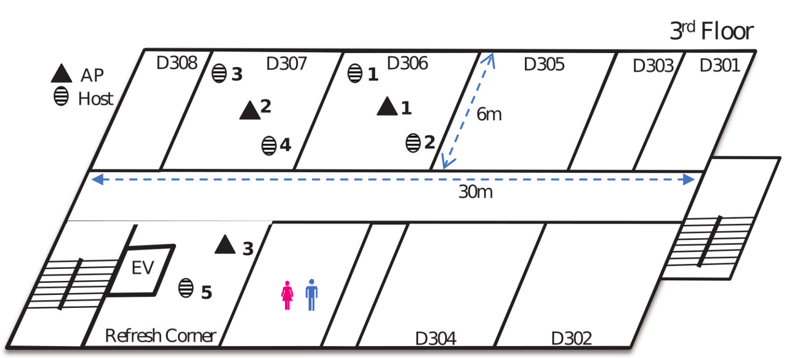


Figure 5.7: Building A uniform topology for experiments.

Table 5.8 shows the cost function by the algorithm, and the overall throughput by the simulation and by the experiment for each configuration. It indicates that for this uniform topology in

Table 5.7: Hardware and software specifications in testbed.

AP	model	Raspberry Pi 3
	CPU	Broadcom BCM2837 @1.2Ghz
	memory	1Gb LPDDR2 900Mhz
	OS	Raspbian
	AP	hostapd
	external NIC	IO-Data WN-AC433UA
PC server	model	Fujitsu Lifebook S761/C
	CPU	Intel Core i5-2520M @2.5Ghz
	memory	4GB DDR3 1333Mhz
	OS	Windows 7
	TCP	iperf 2.0.5
PC host type1	model	Toshiba Dynabook R731/B
	CPU	Intel Core i5-2520M @2.5Ghz
	memory	4GB DDR3 1333Mhz
	OS	Windows 7/10
	TCP	iperf 2.0.5
PC host type2	model	Lesance W255HU
	CPU	Intel(R) Core i5 2450M @2.5 GHz
	memory	4GB
	OS	Windows 7/10
	TCP	iperf 2.0.5

Building-A, the configuration with three non-CB APs for 11 POCs and that with one CB AP and two non-CB APs for 13 POCs by the algorithm provides the highest overall throughput in experiments using the testbed. However, the throughput by the simulation is much larger than that by the experiment, and the difference between the two configurations becomes smaller in the simulation. The consideration of the interference may be insufficient in the simulation, which should be improved in future works.

Table 5.8: Experiment results for Building-A uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
11 POCs	3 non-CB (1, 6, 11)	1713.81	101.12	90.9
	CB + 2 non-CB (11, 1+5, 1)	1606.34	100.22	85.47
13 POCs	CB + 2 non-CB (9+13, 1, 5)	2108.95	121.87	99.16
	2CB + 1 non-CB (9+13, 1+5, 13)	2062.74	118.91	85.6
	3 non-CB (1, 13, 8)	1787.12	103.25	83.83

5.5.2.2 Results for Non-Uniform Topology

Figure 5.8 shows the Building-A non-uniform topology. Three hosts are located in D306, and one host is in D307 and refresh corner. Every host is associated with the AP in the same room. Table 5.9 shows the results. For the non-uniform network topology in Building-A, the configuration with one CB AP and two non-CB APs for both 11 and 13 POCs provides the highest overall throughput.

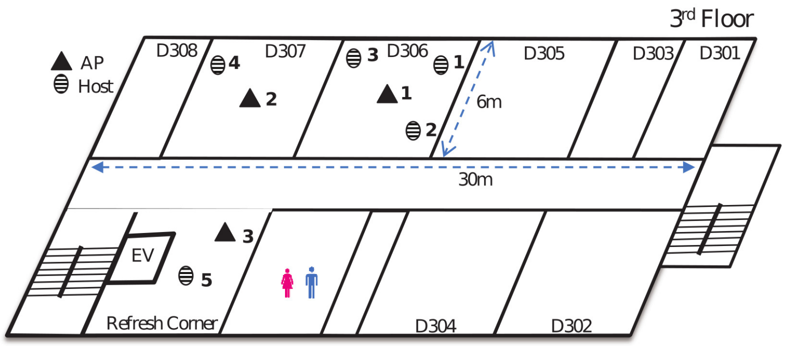


Figure 5.8: Building A non-uniform topology for experiments.

Table 5.9: Experiment results for Building-A non-uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
11 POCs	3 non-CB (1, 6, 11)	1144.68	100.75	86.84
	CB + 2 non-CB (1+5, 11, 1)	1640.59	100.86	90.3
13 POCs	CB + 2 non-CB (9+13, 1, 5)	2169.72	121.62	91.35
	2CB + 1 non-CB (1+5, 9+13, 13)	2033.39	118.59	84.48
	3 non-CB (1, 13, 8)	1193.65	102.89	80.22

5.5.2.3 Results for Two-Crowded APs Topology

Figure 5.9 shows the *Building-A two-crowded APs topology*. Three hosts are located in D306 and D307 respectively, one host is in Refresh Corner, and each host is associated with the AP in the same room. Table 5.10 shows the results. For the two-crowded APs network topology in Building-A, the configuration with two CB APs and one non-CB AP for 13 POCs provides the highest overall throughput both in the simulation and experiment, as expected.

5.5.3 Experiments in Building B Field

Next, the network field of a 9.5m × 6.5m room, a 4.0m× 6.5m room, and a 6.5m × 6.5m room in Graduate School Building (Building B) is adopted in experiments.

5.5.3.1 Results for Uniform Topology

Figure 5.10 shows the *Building-B uniform topology*. Three hosts are located in room A, and two hosts are in room B and room C respectively, and one AP is located in each room. Each host is

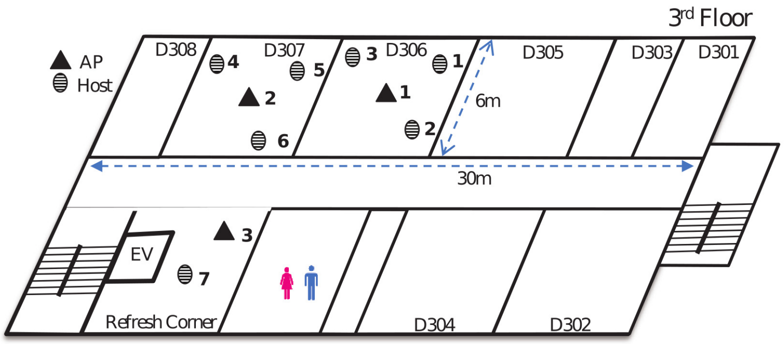


Figure 5.9: Building-A two-crowded APs topology for simulations.

Table 5.10: Experiment results for Building-A two-crowded APs topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
13 POCs	CB + 2 non-CB (9+13, 1, 5)	1387.89	121.42	100.94
	2CB + 1 non-CB (9+13, 1+5, 13)	2010.67	122.55	106.4
	3 non-CB (1, 13, 8)	1175.38	102.74	94.65

associated with the AP in the same room. Table 5.11 shows the results. For the uniform network topology in Building-B, the configuration with three non-CB APs for 11 POCs and with one CB AP and two non-CB APs for 13 POCs provides the highest overall throughput.

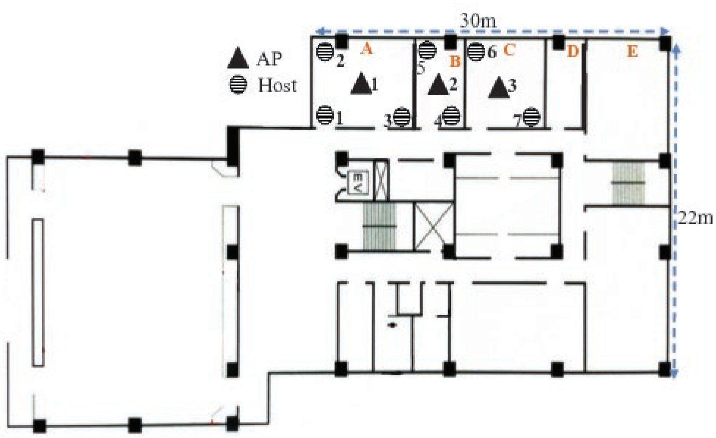


Figure 5.10: Building B uniform topology for experiments.

5.5.3.2 Results for Non-Uniform Topology

Figure 5.11 shows the *Building-B non-uniform topology*. Four hosts are located in room A, one host is in room B, and two hosts are in room C. Likewise, each host is associated with the AP in the same room. Table 5.12 shows the results. For the non-uniform network topology in Building-B, the configuration with one CB AP and two non-CB APs for both 11 and 13 POCs provides the highest overall throughput.

Table 5.11: Experiment results for Building-B uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
11 POCs	3 non-CB (1, 6, 11)	1136.51	100.91	117.77
	CB + 2 non-CB (11, 1+5, 1)	858.15	98.35	104.1
13 POCs	CB + 2 non-CB (9+13, 1, 5)	2114.18	121.69	128.68
	2CB + 1 non-CB (1+5, 13, 9+13)	1831.34	114.66	121.9
	3 non-CB (1, 13, 8)	1185.11	103.05	113.68

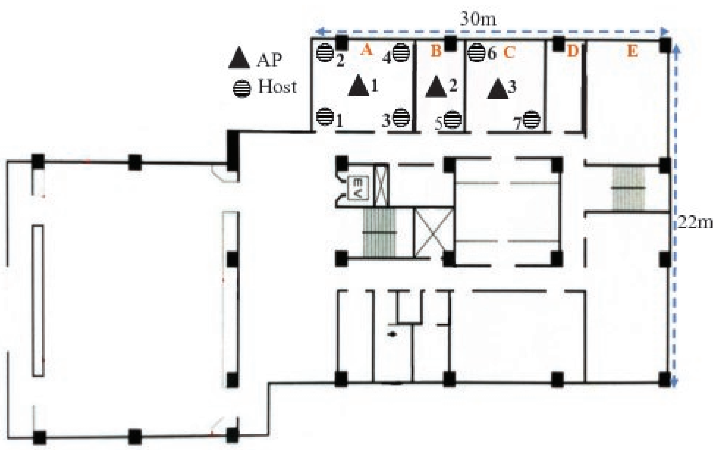


Figure 5.11: Building B non-uniform topology for experiments.

5.5.3.3 Results for Two-Crowded APs Topology

Figure 5.12 shows the *Building-B two-crowded APs topology*. Four hosts are located in room A and room C respectively, and one host is in room B. Likewise, each host is associated with the AP in the same room. Table 5.13 shows the results. For the two-crowded APs topology in Building-B, the configuration with two CB APs and one non-CB AP for 13 POCs provides the highest overall throughput.

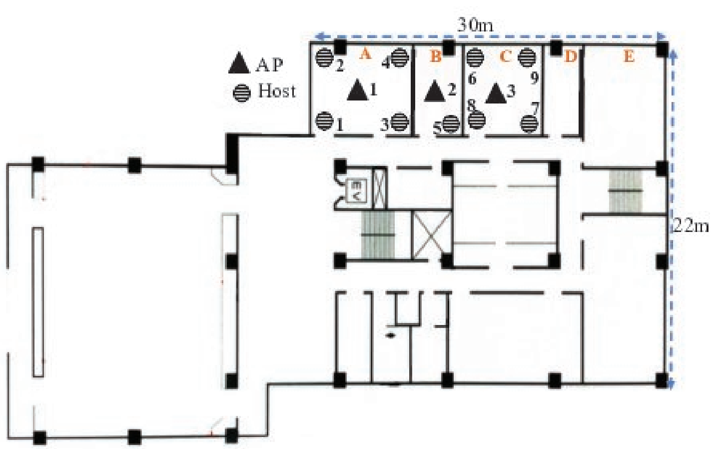


Figure 5.12: Building-B two-crowded APs topology for simulations.

Table 5.12: Experiment results for Building-B non-uniform topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
11 POCs	3 non-CB (1, 6, 11)	854.96	97.58	109.23
	CB + 2 non-CB (1+5, 11, 1)	1226.69	100.07	116.47
13 POCs	CB + 2 non-CB (9+13, 5, 1)	1623.17	121.82	148.48
	2CB + 1 non-CB (1+5, 13, 9+13)	1520.24	118.73	128.96
	3 non-CB (1, 8, 13)	891.53	103.09	124.71

Table 5.13: Experiment results for Building-B two-crowded APs topology.

case	CB/non-CB (channel) (AP1, AP2, AP3)	cost function E	over. through. (Mbps)	
			simul.	measure.
13 POCs	CB + 2 non-CB (9+13, 5, 1)	1054.17	121.62	136.83
	2CB + 1 non-CB (1+5, 13, 9+13)	1518.92	122.93	168.41
	3 non-CB (1, 8, 13)	890.66	102.95	133.13

5.6 Extended Model Applications to Network Configuration

In this section, first I modified the *network configuration optimization algorithm* to apply the *extended throughput estimation model* for 13 POCs. Then, I evaluate the application of *extended throughput estimation model* for 13 POCs to *network configuration optimization algorithm* through simulations and testbed experiments.

5.6.1 Modification of Network Configuration Algorithm

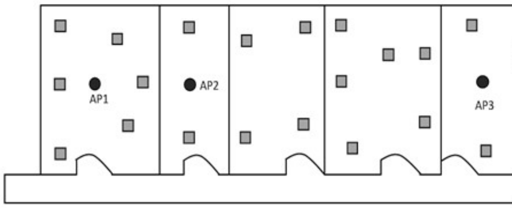
The two objective functions of the network configuration optimization algorithm are modified to consider the interference effects among *APs* due to the average number of walls between them. The minimum average host throughput E_{21} and the total throughput E_{22} are modified as:

$$E_{21} = \min\left[k \times r \times \frac{1}{\sum \frac{1}{TH_{ij}}}\right] \tag{5.5}$$

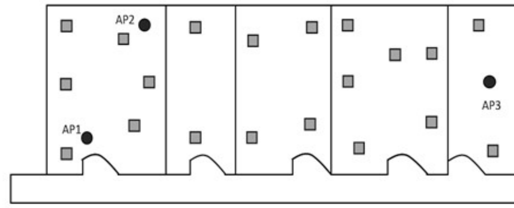
$$E_{22} = \sum k \times r \times \frac{H_j}{\sum \frac{1}{TH_{ij}}} \tag{5.6}$$

where TH_{ij} represents the estimated throughput of the link between the i-th host and the j-th AP, k does the reduction factor (α^2, α^3, or γ) in the previous throughput estimation model [27], r does the throughput increasing factor in the modified model, and H_j does the number of associated hosts with the j-th *AP*.

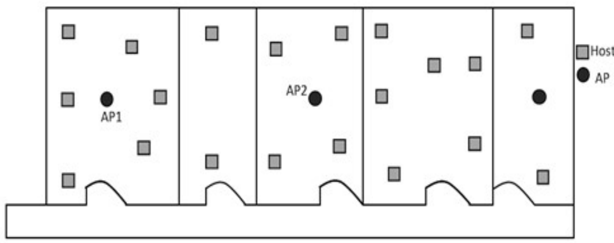
The modified throughput estimation model considers interference effects between three *APs* due to wall obstacles among them. In this model, it assumes that the large value of the minimum number of walls and the average number of walls between the *APs* make larger physical distances among them, which will cause less interference and can increase the overall throughput of the network. This can be supported by the evaluation results in Section 4.5.2.2, where the highest



(a) AP position A.



(b) AP position B.



(c) AP position C.

Figure 5.13: Five-room uniform topology for simulations.

number of walls causes the large throughput and the smallest number of walls causes the lower throughput.

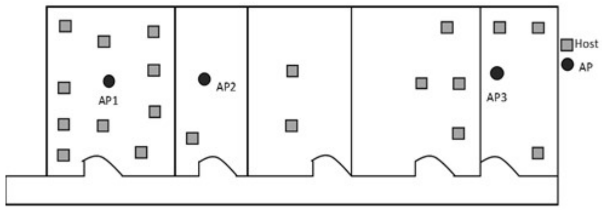
5.6.2 Evaluations by Simulations

In this section, I evaluate the application of *extended throughput estimation model* for 13 POCs to the *network configuration optimization algorithm* through simulations using the *WIMNET simulator* [30].

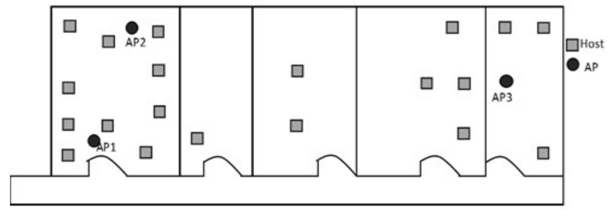
5.6.2.1 Results for Five-room Topology

Firstly, we apply the modified model in network configuration optimization algorithm by considering *five-room uniform* and *non-uniform topologies*. Each topology is composed of two 50m × 60m, two 30m × 60m and one 40m × 60m size rooms where 20 hosts and 3 APs are distributed. *APs* are placed in different rooms in simulations to consider various physical distances among *APs* as shown in Figures 5.13 and 5.14. The circles and squares represent the *APs* and hosts respectively.

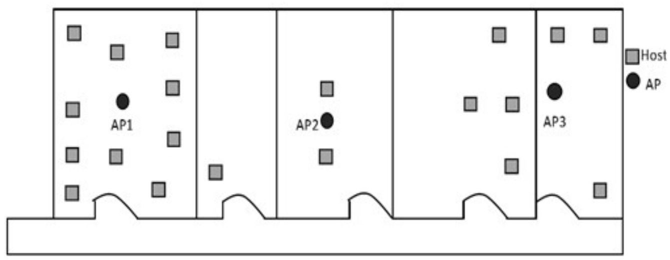
Figure 5.13 shows the five-room uniform network topology with three different *AP* positions (A, B, and C). Table 5.14 shows the changes of the cost function E for each topology and the overall throughput by the *WIMNET* simulator for each association by the network configuration algorithm and for that by the *NAP*. It indicates that for the five-room uniform topology, the configuration of case 1 with one *CB AP* and two *non-CB APs* provides the highest overall throughput for all *AP positions A*, *B*, and *C* respectively. Especially, the AP position C provides the highest overall throughput results as compare to positions A and B. In this case, the minimum number of walls, (n_{Wmin}) among APs is larger than the other cases, although all three topologies *(AP position A, B, and C)* have the same number of average walls, n_{wA}. By locating each *AP* in a separate room with the maximum value of n_{Wmin}, the interference between APs can be much reduced, which can improve the overall throughput of the network. From Table 5.14, it is also observed that the cost



(a) AP position A.



(b) AP position B.



(c) AP position C.

Figure 5.14: Five-room non-uniform topology for simulations.

function E and the overall throughput by the proposal is better than that by the comparison at most cases. For case 1 with AP position C, the performance by our approach and the *NAP* are same due to the same *AP-host* association.

Table 5.14: Simulation results for five-room uniform topology.

Case	AP Position and channel of each AP (AP1, AP2, AP3)	Cost Function E		Overall Through. (Mbps.)	
		Algo.	Near.	Algo.	Near.
1	A (1, 5, 9+13)	554.89	553.53	118.42	117.24
	B (5, 1, 9+13)	279.02	276.52	100.32	98.69
	C (9+13, 1, 5)	787.08	787.08	138.02	138.02
2	A (13, 1+5, 9+13)	436.33	364.26	100.43	97.26
	B (13, 1+5, 9+13)	174.48	172.54	66.92	66.68
	C (1+5, 9+13, 13)	603.17	574.97	131.99	131.52

The five-room non-uniform network topology with *AP* positions *A*, *B*, and *C* is shown in Figure 5.14. Table 5.15 shows the simulation results. For this topology, the configuration of *case*1 with one *CB AP* and two *non-CB APs* provide the highest overall throughput for *AP* positions *A*, *B*, and the configuration of case 2 with two *CB APs* and one *non-CB AP* provides the highest throughput for *AP* positions *C*. Since APs are more isolated in the network field, the case 2 with two *CB APs* are less interfered each other and therefore the throughput is little improved. Again, for both cases, *AP positions C* provides the higher throughput results than *AP position A* and *B*. Table 5.15 indicates that the performance by the proposal algorithm is better than that that by the comparison in most cases. For *case*1 with *AP position C*, our approach and the *NAP* gives similar performance due to the same AP-host association.

Table 5.15: Simulation results for five-room non-uniform topology.

Case	AP Position and channel of each AP (AP1, AP2, AP3)	Cost Function E		Overall Through. (Mbps.)	
		Algo.	Near.	Algo.	Near.
1	A (9+13, 5, 1)	972.83	955.68	154.78	151.98
	B (9+13, 5, 1)	599.55	559.89	135.47	132.59
	C (9+13, 5, 1)	971.52	971.52	162.18	162.18
2	A (1+5, 13, 9+13)	799.74	789.27	145.25	138.69
	B (1+5, 9+13, 13)	290.17	288.64	100.69	100.04
	C (1+5, 13, 9+13)	1124.76	1048.97	163.18	161.63

5.6.2.2 Results for Six-room Topology

Next, we consider six-room uniform and non-uniform topologies. This topology consists of a network field of two 60m × 60m, two 30m × 60m, and two 50m× 60m size rooms where 20 hosts and three APs are adopted in simulations as shown in Figures 5.15 and 5.16.

Figure 5.15 shows the six-room uniform network topology with three different *AP* positions (A, B, and C). Table 5.16 shows the simulation results for this topology. For the six-room uniform topology, the configuration of case 1 with one *CB AP* and two *non-CB APs* provides the highest overall throughput for all *AP* positions A, B, and C respectively. In this case, the performance by our algorithm is better than the *NAP* approach.

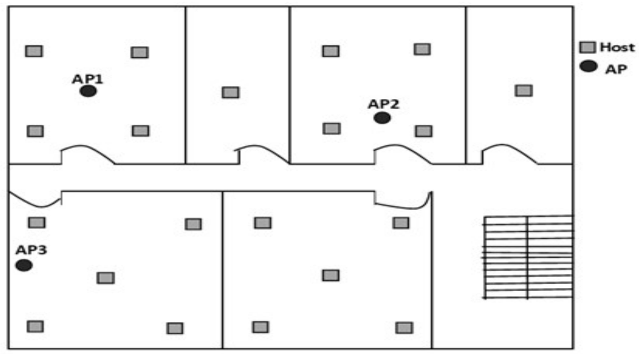
Table 5.16: Simulation results for six-room uniform topology.

Case	AP Position and channel of each AP (AP1, AP2, AP3)	Cost Function E		Overall Through. (Mbps.)	
		Algo.	Near.	Algo.	Near.
1	A (1, 5, 9+13)	348.84	333.99	102.09	97.92
	B (1, 5, 9+13)	275.36	172.42	105.07	93.94
	C (9+13, 1, 5)	596.36	596.36	113.94	113.94
2	A (13, 9+13, 1+5)	314.00	307.02	89.52	87.65
	B (9+13, 13, 1+5)	126.44	112.15	74.61	66.35
	C (1+5, 9+13, 13)	367.24	351.79	104.8	104.70

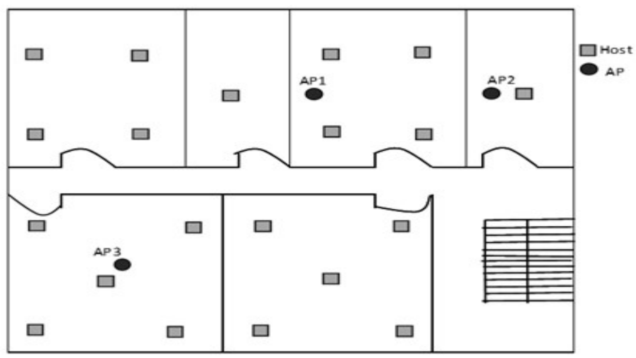
The six-room non-uniform network topology with three different *AP* positions (A, B, and C) is shown in Figure 5.16. Table 5.17 shows the simulation results for this topology. In this case, the configuration of case 1 with one *CB AP* and two *non-CB APs* provides the highest overall throughput for all *AP* positions A, B, and C respectively. The performance by our algorithm is better than that by the *NAP approach*.

Table 5.17: Simulation results for six-room non-uniform topology.

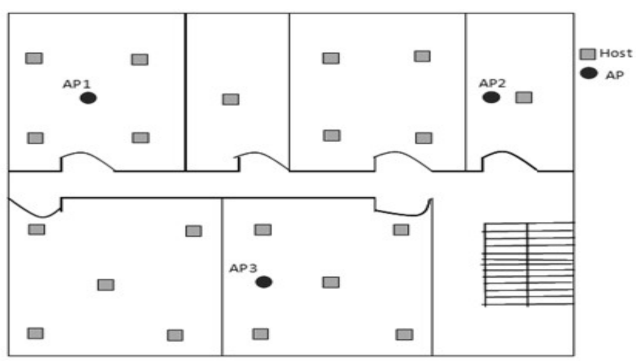
Case	AP Position and channel of each AP (AP1, AP2, AP3)	Cost Function E		Overall Through. (Mbps.)	
		Algo.	Near.	Algo.	Near.
1	A (5, 1, 9+13)	309.43	309.43	108.15	108.15
	B (1, 5, 9+13)	379.24	369.24	115.77	111.33
	C (5, 1, 9+13)	264.02	252.74	97.42	93.51
2	A (13, 9+13, 1+5)	302.99	302.99	93.61	93.61
	B (1+5, 13, 9+13)	194.13	194.13	79.63	79.63
	C (13, 9+13, 1+5)	247.37	244.77	89.92	88.97



(a) AP position A.



(b) AP position B.



(c) AP position C.

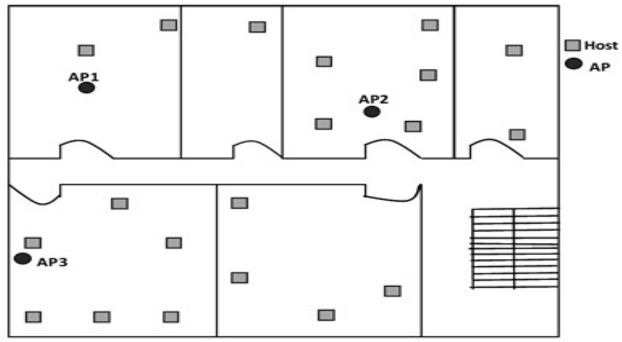
Figure 5.15: Six-room uniform topology for simulations.

5.6.3 Evaluations by Testbed Experiments

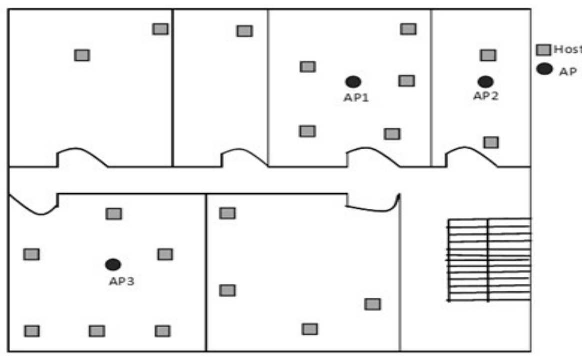
In this section, I evaluate the application of *extended throughput estimation model* for 13 POCs to *network configuration optimization algorithm* through testbed experiments.

The test-bed experiments were carried out in two fields same as 5.5, namely field A and field B. The third floor of Engineering Building #2 (as field A) and the 2*nd* floor of Graduate School of Natural Science and Technology Building (as field B) at *Okayama University* in Figure 5.17 and 5.18 respectively, are used.

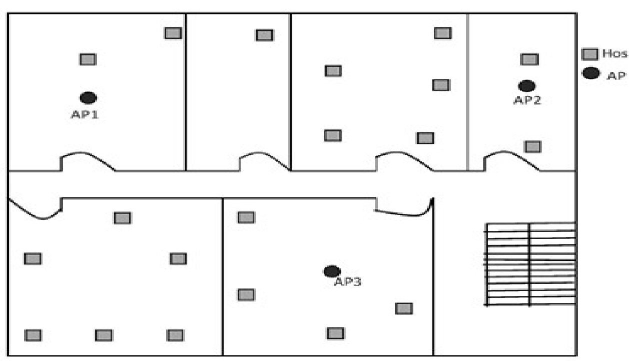
Table 5.18 reveals the specifications of the adopted wireless NIC adapters in the test-bed experiment. To illustrate it, one combination of *Raspberry Pi* and the NIC adapter is called the *AP*



(a) AP position A.



(b) AP position B.



(c) AP position C.

Figure 5.16: Six-room non-uniform topology for simulations.

type. In *field A AP* type 2 is adopted and in *field B AP* type 1 is adopted.

5.6.3.1 Results for Engineering Building

First, field A is adopted in experiments where one scenario is considered using a different number of rooms. *AP* and hosts are distributed in rooms *D*308, *D*306, and refresh corner respectively as shown in Figure 5.17.

Table 5.19 compares the overall throughput and error rate between the previous and modified models for both cases. The results confirms that the modified model greatly improves the throughput estimation accuracy in both cases compared to the previous model where the error rate

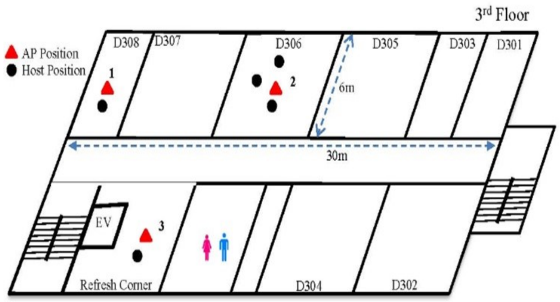
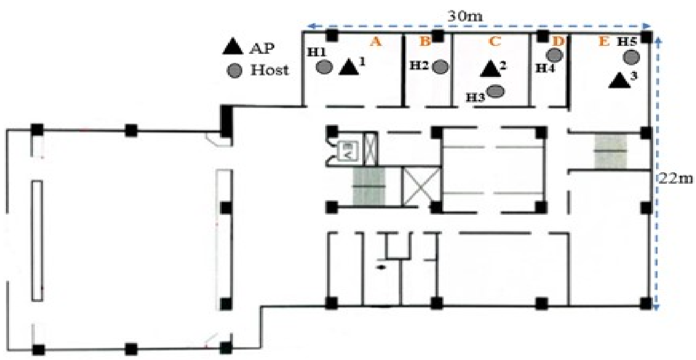
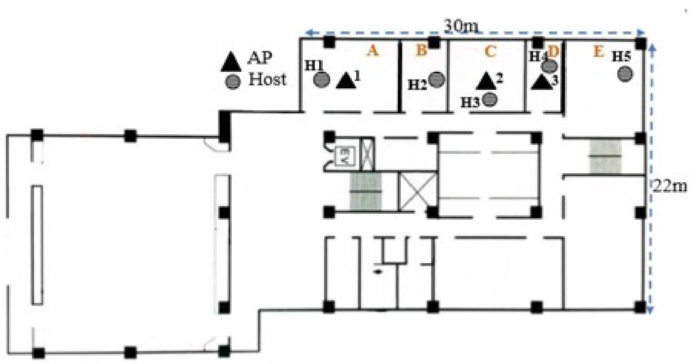


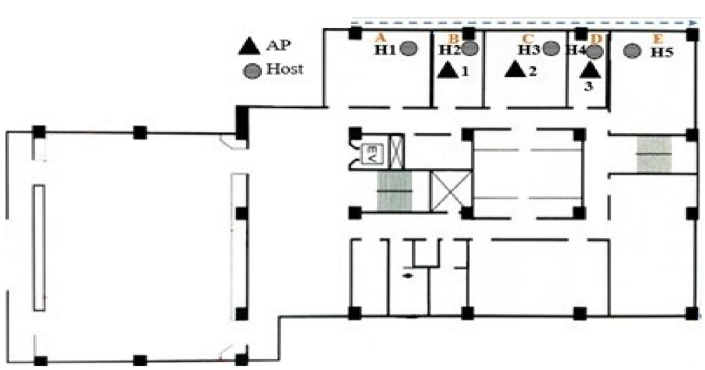
Figure 5.17: Test-bed topology in field A.



(a) Topology 1.



(b) Topology 2.



(c) Topology 3.

Figure 5.18: Test-bed topology in field B.

Table 5.18: Wireless NIC adapters.

AP type	model	wireless chipset	operating mode	channel width
1	IO-Data WN-AC433UA	Realtek RTL8811AU	2.4GHz	20MHz, 40MHz
2	TP-LINK TL-WN722N	Atheros AR9002U	2.4GHz	20MHz, 40MHz

is greatly reduced by the proposed model.

Table 5.19: Test-bed experiment results for field A.

Case	Overall Through. (Mbps.)			Error Rate(%)	
	Sim. Pre. Model	Measured	Sim. Mod. Model.	Pre. Model.	Mod. Model.
1	129.41	140.41	142.68	8.5	1.59
2	115.85	124.6	124.23	7.55	0.29

5.6.3.2 Results for Graduate School Building

Finally, field B is adopted in experiments where three scenarios are considered using a different number of rooms as shown in Figure 5.18. *Access points (AP)* and hosts are distributed across different rooms, from room *A* to room *E*. The average number of walls decreases gradually from *Topology 1* to *Topology 3*, depending on the *AP* location.

Table 5.20 compares the overall throughput and error rate between the previous and modified models for both cases. In both cases, the overall throughput for both simulations and experiments gradually decreases from *Topology 1* to *Topology 3* due to increased interferences and the reduction in the average number of walls among *APs*. In all topologies, the results confirm that the modified model significantly enhances throughput estimation accuracy in both cases compared to the previous model, resulting in a greatly reduced error rate.

Table 5.20: Test-bed experiment results for field B.

Topology	Case	Overall Through. (Mbps.)			Error Rate(%)	
		Sim. Pre. Model	Measured	Sim. Mod. Model.	Pre. Model.	Mod. Model.
1	1	114.22	139	148.60	21.69	6.46
	2	109.94	129.28	146.31	17.59	11.63
2	1	109.63	124.84	129.25	13.87	3.41
	2	102.81	119.23	116.86	15.97	2.03
3	1	114.89	120.24	123.72	4.68	2.81
	2	109.75	117.67	115.32	7.22	2.04

5.7 Summary

In this chapter, I presented the *network configuration optimization algorithm* for WLAN. First, I formulated the *network configuration optimization problem* as a combinatorial optimization problem to design the algorithm. Next, I described the procedure for the proposed algorithm. Third, I evaluated the effectiveness of the *network configuration optimization algorithm with the modified model* through simulations and test-bed experiments. Fourth, I modified the *network configuration optimization algorithm* for the *extended throughput estimation model*. Finally, I evaluated the application of *extended throughput estimation model* for 13 POCs to the *network configuration optimization algorithm* through simulations and testbed experiments. In the next chapter, I will present the preprocessing stage for the AP configuration algorithm and its application to joint optimization algorithm in WLAN.

Chapter 6

Preprocessing Stage for Active Access-Point Configuration Algorithm

In this chapter, I present the *preprocessing stage* for the *active AP configuration algorithm* to reduce the CPU time by limiting the number of APs in advance [23]. Both the exhaustive and heuristic approaches are adopted here, where either one should be used depending on the network topology. Firstly, I present the *preprocessing stage* for the active AP configuration algorithm in [23]. Secondly, I evaluate the proposal of *preprocessing stage* through simulations. Thirdly, I present the application of *preprocessing stage* to *active AP configuration algorithm* in WLAN. Lastly, I evaluate the results through simulations.

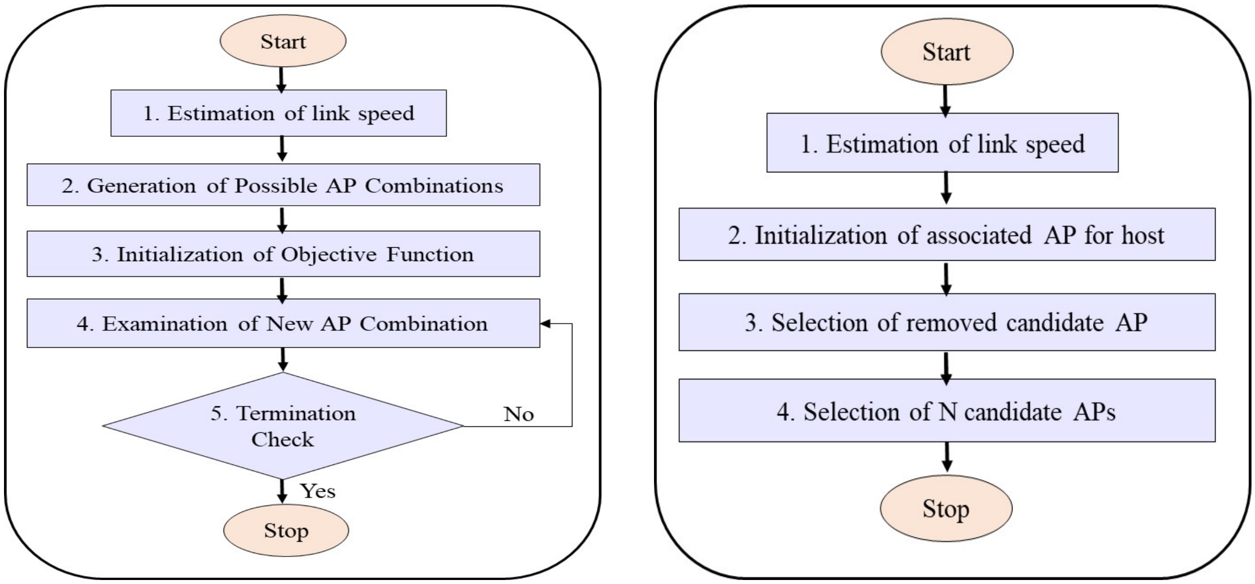
6.1 Introduction

In WLAN, the proper allocation of APs in the network field has the great impact in determining the performance of the network. The improper allocation of APs may cause undesired situations where some hosts cannot receive a strong signal from any AP and may suffer from low throughputs. However, to identify the optimal allocation of the APs is a challenging issue due to unpredictable propagation characteristics of the wireless medium, particularly in the indoor environment [95]. That is, the distribution of user hosts appears to be non-uniform and tends to change dynamically, and traffic demands have been fluctuating all the time [20].

To solve the above mentioned problem, in Section 3.7, the *active AP configuration algorithm* have studied for the *elastic WLAN system*. The elastic WLAN system can dynamically optimize the network topology by activating or deactivating allocated APs and changing the host associations and channel assignments according to the network conditions [22]. Unfortunately, this algorithm may take a long CPU time and may not provide an optimal solution when a large number of APs are deployed in the field due to the large search space. It is important to reduce the search space by limiting APs under consideration in advance before applying the algorithm.

In this chapter, I present the *preprocessing stage* for the *active AP configuration algorithm* to reduce the CPU time by confining the search space by selecting promising candidates for active APs. Both the exhaustive and heuristic approaches are adopted for this stage, where the simulation results found that the better approach among them in terms of the CPU time and the network performance is different depending on the network topology. Thus, the user will need to select either one for each network instance.

Next, I present the application of *preprocessing stage* to the *joint optimization algorithm* in



(a) Exhaustive approach (b) Heuristic approach

Figure 6.1: Preprocessing stage of the active AP configuration algorithm.

WLAN. Here I propose two extensions to improve the performance of *WLAN* networks. In the first extension, I adopt the *pre-processing stage* of the *active AP configuration algorithm* in the *AP joint optimization algorithm* to select the promising candidate *APs*. This stage not only improves the network performance but also reduces the search space of the network field. In the second extension, I propose the *post-processing stage* of the algorithm, which refines *AP-host* associations to further improve the overall network performance. The effectiveness of the proposal is evaluated through simulations using the *WIMNET simulator* [30].

6.2 Procedure of Preprocessing Stage

In this section, I present the *preprocessing stage* for the active AP configuration algorithm including both exhaustive and heuristic approaches. Figure 6.1 shows the flow chart for the preprocessing stage of active AP configuration algorithm. Figure 6.1 (a) illustrates the exhaustive approach and Figures 6.1 (b) illustrates the heuristic approach of the pre-processing stage.

6.2.1 Problem Formulation for Preprocessing Stage

The pre-processing stage of the active AP configuration algorithm is formulated as follows:

1. **Inputs:**

   - Locations of M candidate APs, where a *candidate AP* represents the possible location of an AP
   - Locations of H hosts
   - Locations of walls
   - Number of selected APs, N ($N \leq M$)

2. **Outputs:**
   - N promising candidate APs for the active AP configuration algorithm
3. **Objectives:**
   - Maximize the summation of the bottleneck host throughputs for N APs: E. A *bottleneck host* represents the host for each AP such that the throughput is minimum among the associated hosts with this AP.
   - Holding the first objective, to maximize E_2.

6.2.2 Estimation of Candidate AP Number

In the preprocessing stage, only promising candidates for active APs are selected for the input to the active AP configuration algorithm. To estimate the number of such candidates N properly, the minimum number of active APs L that are necessary to satisfy the host throughput constraint is estimated first. L can be estimated from the minimum host throughput threshold G, the number of hosts H, and the maximum possible link speed TP_{max} in the network field by:

$$L < \frac{G \times H}{TP_{max}} \tag{6.1}$$

where TP_{max} is generally $100Mbps$ for an *IEEE 802.11n* link, which has been observed in our experiments. Therefore, L is estimated by:

$$L = \frac{G \times H}{100} \tag{6.2}$$

Then, N is estimated from L by:

$$N = \beta \times L \tag{6.3}$$

where β represents a given constant parameter. Through simulations, it is found that the proper value of β is different depending on the minimum host throughput threshold G: $\beta = 4.3$ for $G \leq 10$, and $\beta = 3.1$ otherwise.

6.2.3 Procedure of Exhaustive Approach

The procedure of the exhaustive search approach for the pre-processing stage consists of the following steps:

1. **Estimation of link speed:** The link speed (throughput) for each link between each of the given M candidate APs and each of the given H hosts is assessed using the throughput estimation model in Section 3.3.

2. **Generation of Possible AP Combinations:** The total of ${}_MC_N$ possible combinations of N APs are generated by selecting N APs from the M candidates.

3. **Initialization of Objective Function:** The best-found objective functions E^{best} and E_2^{best} are initialized by 0, where E_2^{best} represents the best-found value of the average host throughput and E^{best} does the best-found value of the total bottleneck host throughput for N APs.

4. **Examination of New AP Combination:** For each combination of N APs, the following procedure is applied:

   (a) The candidate AP that has the largest link speed is selected for the associated AP of each host.

   (b) The best-found objective functions E^{best} and E_2^{best} are updated, and the current AP selection is saved in memory, if at least one of the two best found objective functions, E^{best} or E_2^{best}, is updated and another one remains the same by the current AP selection and the host associations.

5. **Termination check:** If every combination of N APs is examined, the procedure is terminated, and the selected N APs is used for the input to the active AP configuration algorithm. Otherwise, move to step (4) to analyze another new combination.

6.2.4 Procedure of Heuristic Approach

The procedure of the heuristic search approach for pre-processing stage consists of the following steps:

1. **Estimation of link speed:** The throughput of the link between each of M candidate APs and each of H hosts is calculated using the throughput estimation model 3.3.

2. **Initialization of associated AP for host:** The initial associated candidate AP is found for each host, and the non-associated candidate APs are excluded with four steps as below:

   (a) Select the candidate AP for each host such that the throughput is maximized.

   (b) Calculate the expected number of associated hosts per AP, n as follows:

   $$n = AP_{max}/S \tag{6.4}$$

   Here, AP_{max} is the maximum throughput for each AP and S does the minimum required throughput for the association.

   (c) For each candidate AP, sort all the hosts in descending order of the throughput with it. Then, for each candidate AP, select n hosts from the top of the sorted list, and calculate the total throughput of the links with the n hosts.

   (d) Sort all the candidate APs in descending order of the total throughput in iii). From the top of the sorted AP list, select the candidate APs, until each host in the network is included in the set of hosts selected in iii) for the candidate APs selected in iv). The remaining candidate APs are excluded.

3. **Selection of removed candidate AP:** Find one candidate AP to be removed such that at least one of the two best found objective functions, E^{best} or E_2^{best}, can be updated, and another one remains the same after removing that AP for the current AP number.

   (a) For each remaining candidate AP, discover the other remaining candidate AP to associate each associated host with this AP such that the throughput is maximized, and identify the minimum throughput (bottleneck host throughput) among the links to this AP after the host re-associations.

(b) Calculate the objective function E^{best} or E_2^{best} for the new associations.

(c) Remove the candidate AP if at least one of the two best found objective functions, E^{best} or E_2^{best} are becomes maximum and another one remains the same after applying i) and ii) to every remaining candidate AP, and apply the re-association for each associated host with this AP.

4. **Selection of** N **candidate APs:** Repeat Step 3 until the number of remaining candidate APs becomes N.

6.3 Evaluations by Simulations

In this section, I evaluate the proposal of *preprocessing stage* through simulations.

6.3.1 Results for Topology I

As the first topology, *Topology I* in Figure 6.2 is adopted where that 25 hosts and 30 candidate APs are allocated in two $50m \times 50m$ rooms. The circles and squares represent the APs and hosts respectively.

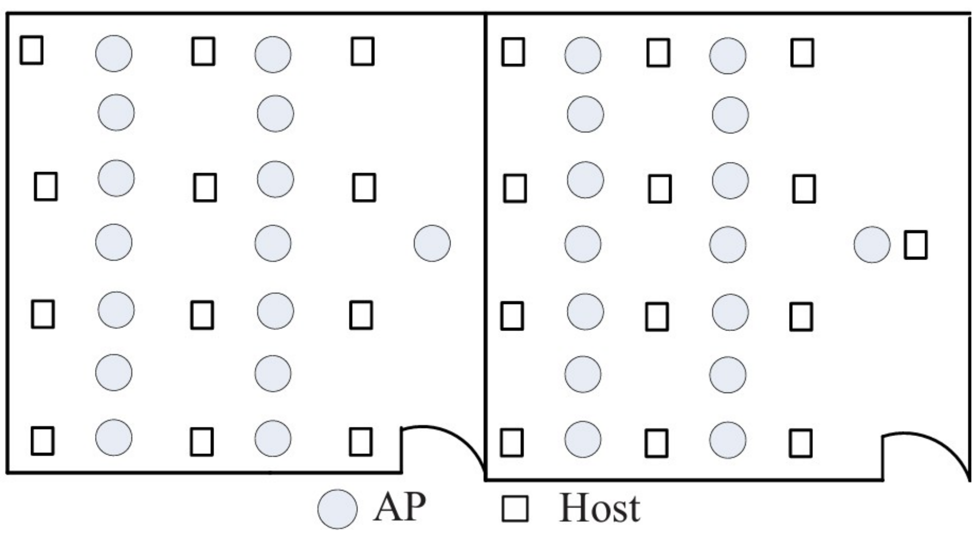


Figure 6.2: Topology I.

Table 6.1 demonstrates simulation results for *Topology I*. In any case, the exhaustive approach provides the better configuration than the original algorithm with the smaller CPU time by 53.98% on average. The heuristic approach further decreases the CPU time by 91.09% on average, where the total throughput is similar to that of the original.

6.3.2 Results for Topology II

As the second topology, *Topology II* in Figure 6.3 is adopted where 40 hosts and 33 candidate APs are allocated in two $60m \times 45m$ rooms, and six $30m \times 45m$ rooms.

Table 6.2 reveals simulation results for *Topology II*. Again, the exhaustive approach exhibits the better configuration than the original algorithm. Nevertheless, it increases the CPU time by

Table 6.1: Simulation results for topology I.

G (Mbps)	# of active APs	without preprocessing			with preprocessing						
						exhaustive search			heuristic search		
		min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	K promising candidate APs	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)
5	2	6.03	149.87	51.83	9	6.14	152.60	15.38	6.10	151.29	1.04
10	4	11.86	299.23	221.25	13	11.87	299.20	150.52	11.86	298.54	12.61
15	5	15.30	381.73	333.33	13	15.58	393.27	157.23	15.34	383.95	20.44
20	7	21.07	535.05	621.25	16	21.13	536.78	261.39	21.14	536.97	75.26

14.30% on average due to the large number of combinations of selecting APs. The heuristic approach decreases the CPU time by 69.23% on average, where the total throughput is similar to that of the original.

Table 6.2: Simulation results for different value of G in topology II.

G (Mbps)	# of active APs	without preprocessing			with preprocessing						
						exhaustive search			heuristic search		
		min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	K promising candidate APs	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)
5	4	6.88	274.57	376.16	9	6.90	275.85	65.29	6.88	274.46	6.69
10	6	10.63	423.98	800.91	18	10.67	424.38	2324.5	10.65	423.99	131.10
15	9	15.98	640.77	1707.98	19	16.02	640.90	2213.4	16.02	640.99	350.11
20	10	20.60	822.18	2074.68	25	20.64	823.39	1065.8	20.62	822.73	1037.9

6.3.3 Results for Topology III

As the third topology, *Topology III* in Figure 6.4 is adopted where 40 hosts and 35 candidate APs are allocated in six rooms of two different sizes of $7m \times 6m$, and $3.5m \times 6m$.

Table 6.3 indicates simulation results for *Topology III*. The exhaustive approach provides the better configuration than the original algorithm. Unfortunately, it increases the CPU time by

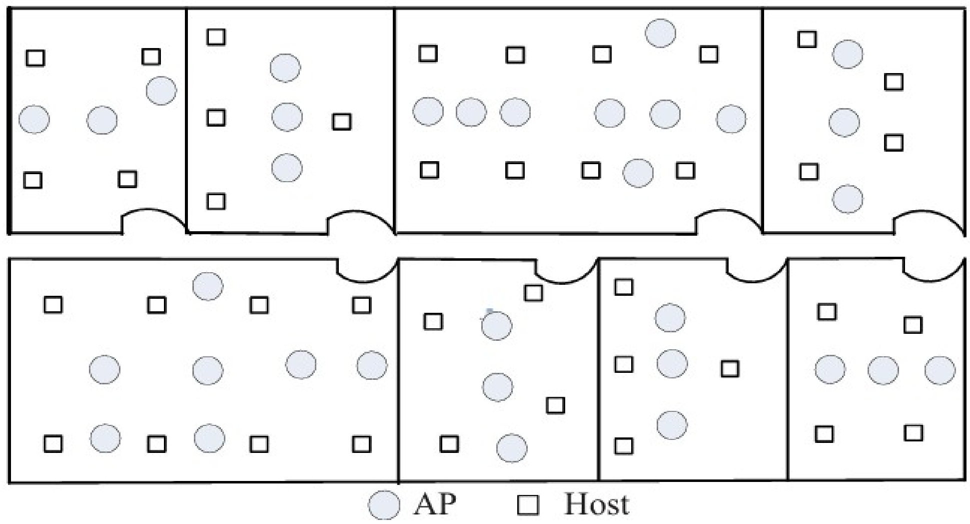


Figure 6.3: Topology II.

207.36% on average due to the large number of combinations of selecting APs. The heuristic approach further decreases the CPU time by 69.24% on average, where the total throughput is similar to that of the original.

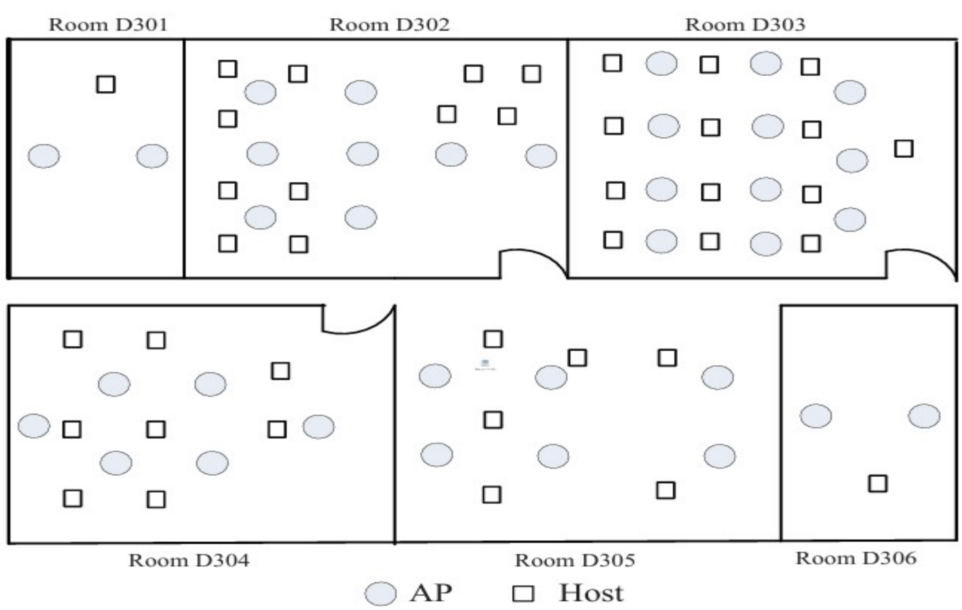


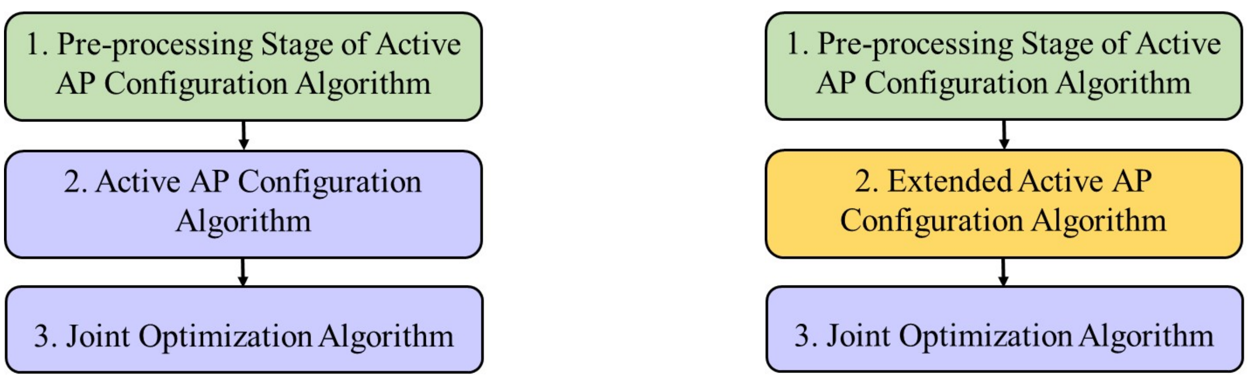
Figure 6.4: Topology III.

6.4 Applications to Joint Optimization Algorithm

In this section, I present the application of *preprocessing stage* to joint optimization algorithm in WLAN. Figure 6.5 shows the overall execution flow for the two extensions of the *AP* joint optimization algorithm. Figure 6.5 (a) illustrates the first extension of the *AP* joint optimization algorithm and Figures 6.5 (b) illustrates the second extension of the *AP* joint optimization algorithm. Firstly, I propose two extensions of the *AP* joint optimization algorithm. Secondly, I evaluate the

Table 6.3: Simulation results for topology III.

G (Mbps)	# of active APs	without preprocessing			with preprocessing						
						exhaustive search			heuristic search		
		min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	K promising candidate APs	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)	min. host through. (Mbps)	overall through. (Mbps)	CPU time (S)
5	4	7.03	280.48	402.71	9	7.05	281.18	93.05	7.04	280.34	7.09
10	6	11.08	443.79	868.27	18	11.10	444.59	8019.29	11.09	443.65	128.80
15	9	16.71	676.57	1740.47	19	16.73	681.10	7434.16	16.71	676.54	333.12
20	11	20.71	834.09	2580.45	25	20.74	840.92	1641.14	20.72	833.27	1250.5



(a) First extension (b) Second extension

Figure 6.5: Overall execution flow of the proposal.

proposal through simulations. Lastly, I evaluate the performance analysis of the proposal with random AP selection method and greedy approach method.

6.4.1 Modifications of Joint Optimization Algorithm

In this section, I present the two extensions of the *AP* joint optimization algorithm.

6.4.1.1 Formulation of Joint Optimization

This extended algorithm is formulated as follows:

1. **Inputs:**

    - Locations of M candidate APs

- Locations of H hosts
- Locations of walls

2. **Outputs:**

- Set of active APs
- Set of hosts associated with each active AP
- CB or non-CB POC for each AP
- Maximum or minimum transmission power for each AP

3. **Objectives:**

- To maximize E_T. E_T is the total throughput of the WLAN network, which is calculated by the sum of the throughputs of all the active APs and their associated hosts
- Holding the first objective, to maximize E_2

E_2 indicates the *minimum average host throughput* for the bottleneck AP:

$$E_2 = \min_j \left[TP_j \right] \tag{6.5}$$

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

$$TP_j = \frac{1}{\sum_k \frac{1}{tp_{jk}}} \tag{6.6}$$

where tp_{jk} represents the link speed between $node_j$ and $node_k$ ($link_{jk}$).

4. **Constraints:**

(a) Minimum host throughput constraint: each host in the network must enjoy the given threshold G on average when all the hosts are communicating simultaneously.

(b) 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.

(c) Transmission power constraint: the transmission power of each AP must be within a predefined maximum or minimum range.

(d) Frequency channel constraint: the frequency channel of each AP must be one of the available channels at $2.4GHz$ band.

(e) Channel bonding constraint: the channel bonding of each AP must be either $20MHz$ or $40MHz$.

6.4.1.2 Procedure of First Extension: Pre-processing Stage

The procedure of the network configuration considering the pre-processing stage of the active AP configuration algorithm in the AP joint optimization algorithm is given as follows:

1. Select the promising candidate APs at the pre-processing stage. Here, it only considers the exhaustive search approach in Section 6.2.3.

2. Optimize the AP and host associations by the algorithm in Section 3.7 and optimize the channel, channel type and power of each AP by the algorithm in Section 3.8.

Table 6.4: Simulation environment.

PC Model	Lenovo ThinkPad-L560
Processor	Inter(R), Core(TM) i3-6006U (2.00 GHz x 4)
Memory (RAM)	4 GB
OS	Ubuntu LTS 14.04, 64 bit
Simulator	WIMNET simulator [?]
Interface	IEEE 802.11n

6.4.1.3 Procedure of Second Extension: Post-processing Stage

In the second extension, I present the post-processing stage of the AP joint optimization algorithm.

The active AP configuration algorithm consists of the following eight steps in Section 3.7. In this paper, I present a new Step 9, namely, *Post Host Association Optimization*, as the post-processing stage. The procedure of the post host association optimization is given as follows:

1. Select one host from the current active host list.
2. Randomly select another host whose associated AP is different from the first one.
   (a) Calculate the cost function (G) for the current association.
   (b) Exchange the associations of the two hosts if the cost function is improved. Otherwise, roll back to the previous association.
3. Repeat Step 2 until all hosts are checked.
4. Terminate the algorithm if all active hosts are checked otherwise go back to Step 1.

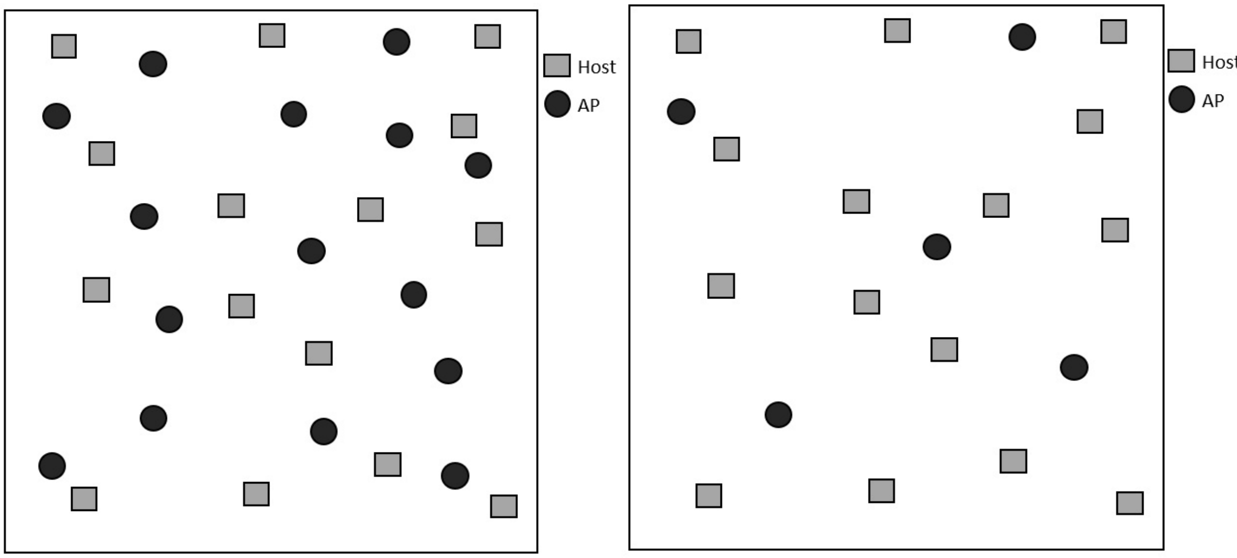
6.4.2 Evaluations by Simulations

In this section, I evaluate the proposed two extensions of the AP joint optimization algorithm through simulations using the *WIMNET* simulator [30]. Table 6.4 shows the adopted hardware and software specifications of the *PC* to evaluate the proposal in the simulations.

6.4.2.1 Results for Random Topology

Firstly, I evaluate the proposal using a simple network topology where *APs* and hosts are randomly allocated in one room. Figure 6.6 (a) illustrates this random topology. 15 hosts and 15 *APs* are allocated in one 50m $\times$ 50m room. The circle represents the *AP* and the square does the host. Figures 6.6 (b) suggests the output of the pre-processing stage.

1. **Results of First Extension:** Tables 6.5 and 6.6 show the first extension results and the simulation results respectively. Table 6.5 shows the number of active *APs*, *CB or non-CB POC* for each *AP*, and the maximum or minimum transmission power for each *AP* for different value of G. Table 6.6 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the total throughput respectively. From the simulation results, by the *pre-processing stage* in the first extension, the throughput performance is improved compared to the original method.



(a) Input topology with 15 candidate APs

(b) Output topology of pre-processing stage with 5 promising APs

Figure 6.6: Random topology.

Table 6.5: Comparison of network configuration with first extension for random topology.

G Mbps	No. of active APs (Orig./Prepro.)	Channel (Power) AP1, AP2, AP3, AP4, AP5	
		Original	Only preprocessing
10	2/2	1+5(max), 9+13(max)	9+13(max), 1+5 (max)
20	3/3	7(max), 13(max), 1(max)	1(max), 1+5(max), 9+13(max)
30	4/4	7(max), 13(max), 1(max), 1(max)	9+13(min), 1(min), 1+5(min), 13(min)
35	5/5	1+5(max), 13(min), 9+13(max), 9+13(max), 9+13(max)	13(min), 1+5(max), 1(min), 1+5(max), 9+13(min)

However, due to the 1 or 2 host associations, in a few cases, some results for $G = 10$ and for $G = 20$ gives lower throughput than the original method.

2. **Results of Second Extension:** Tables 6.7 and 6.8 show the second extension results and the simulation results respectively. Table 6.7 shows the number of active *APs*,*CB* or *non-CB POC* for each AP, and the maximum or minimum transmission power for each *AP* for the different value of *G*. Table 6.8 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the total throughput respectively. By considering both the *pre-processing* and *post-processing stages* in the second extension, the throughput performance is improved for all cases compared to the original method.

   As shown in Table 6.6, the first extension results show that in some cases, such as $G = 20$, there is the lower throughput compared to the original method. However, after the second extension, improvements can be seen in Table 6.8.

   To assess the effectiveness of the second extension, we compare the pre-processing and post-

Table 6.6: First extension simulation results for random topology.

G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	Original	2.27	2.59	3.39	4.09	50.87
	Only pre.	2.27	2.55	3.33	4.09	49.86
20	Original	3.86	4.32	6.49	8.23	97.44
	Only pre.	2.96	4.71	6.71	7.88	100.68
30	Original	3.25	6.75	8.61	10.57	129.14
	Only pre.	5.81	9.73	11.06	13.29	165.84
35	Original	4.42	10.06	11.53	13.16	173.01
	Only pre.	7.26	10.32	12.38	13.88	185.73

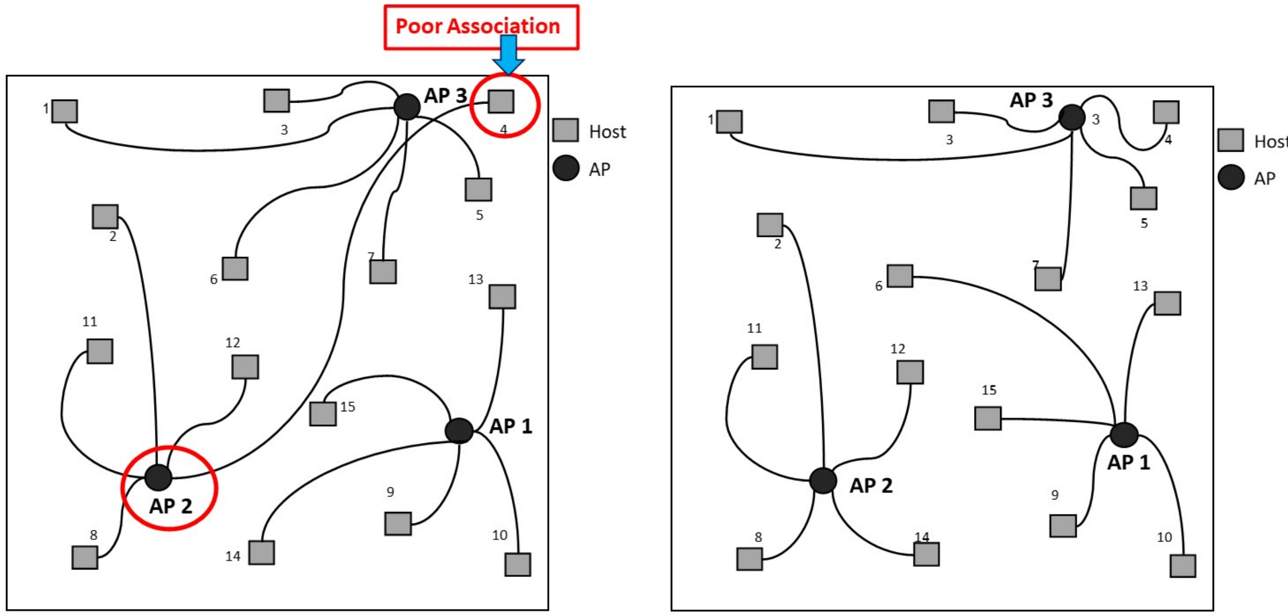
Table 6.7: Comparison of network configuration with second extension for random topology.

G mbps	No. of active APs (Ori./ Pre.& post.)	Channel (Power) AP1, AP2, AP3, AP4, AP5	
		Original	With preprocessing & postprocessing
10	2/2	1+5 (max), 9+13(max)	1+5(max), 9+13 (max)
20	3/3	7(max), 13(max), 1(max)	1(max), 1+5(max), 9+13(max)
30	4/4	7(max), 13(max), 1(max), 1(max)	9+13(min), 1(min), 1+5(min), 13(min)
35	5/5	1+5(max), 13(min), 9+13(max), 9+13(max), 9+13(max)	13(min), 1+5(max), 1(min), 1+5(max), 9+13(min)

Table 6.8: Second extension simulation results for random topology.

G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	Original	2.27	2.59	3.39	4.09	50.87
	With pre. & post.	2.30	2.7	3.93	4.09	50.87
20	Original	3.86	4.32	6.49	8.23	97.44
	With pre. & post.	3.01	5.47	7.14	7.87	107.04
30	Original	3.25	6.75	8.61	10.57	129.14
	With pre. & post.	5.81	9.73	12.01	13.29	165.84
35	Original	4.42	10.06	11.53	13.16	173.01
	With pre. & post.	7.26	10.32	12.38	13.91	185.73

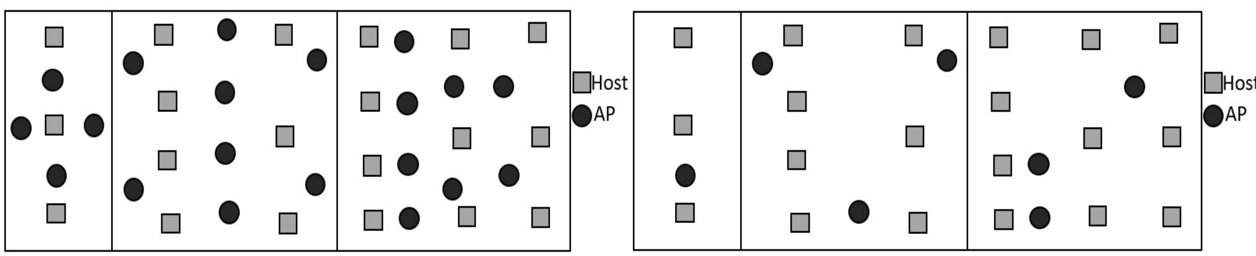
processing results for $G = 20$. Figure 6.7 illustrates the random network topology with the AP host associations for $G = 20$. In Figure 6.7 (a), only the pre-processing is considered, showing that a few hosts suffer from poor associations, leading to the lower throughput and the network performance degradation, as represented by the red marks. In Figure 6.7



(a) First extension association

(b) Second extension association

Figure 6.7: AP host associations for G=20 in random topology.



(a) Input topology with 20 candidate APs

(b) Output topology of preprocessing stage with 7 promising APs

Figure 6.8: Regular topology 1.

(b), the second extension is illustrated, showing improved associations by refining *AP-host* associations, leading to the overall network performance improvement.

6.4.2.2 Results for Regular Topology 1

Secondly, I evaluate the proposal using a three room regular network topology. Figure 6.8 (a) illustrates the regular topology with three rooms, where 20 hosts and 20 APs are regularly allocated in two 20m × 30m rooms, one 10m × 30m room. Figures 6.8 (b) suggests the output of the preprocessing stage.

1. **Results of First Extension:** Tables 6.9 and 6.10 show the first extension results and the simulation results respectively. Table 6.9 shows the number of active *APs*, *CB or non-CB POC* for each *AP*, and the maximum or minimum transmission power for each *AP* for different value of *G*. Table 6.10 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the to-

Table 6.9: Comparison of network configuration with first extension for regular topology 1.

G Mbps	No. of active APs (Orig./ Prepro.)	Channel (Power) AP1, AP2, AP3, AP4, AP5, AP6, AP7	
		Original	Only preprocessing
5	2/2	9+13(max), 1+5(max)	1+5(max), 9+13(max)
10	4/4	9+13(min), 1(max), 13(max), 7(max)	1+5(min), 13(max), 7(max), 1(max)
15	6/6	5+9(min), 13(min), 13(max), 13(max), 1(max), 7(max)	9+13(min), 1(max), 1(max), 9+13(min), 13 (max), 1+5 (min)
20	7/7	1(max), 13(min), 7(min), 13(min), 13(min) , 9+13(min), 1+5(min)	1+5(min), 9+13 (max), 9+13(min), 9+13(min), 1+5(min), 13(min), 9+13(max)

Table 6.10: First extension simulation results for regular topology 1.

G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
5	Original	0.58	0.72	0.96	1.11	19.25
	Only pre.	0.69	0.82	1.07	1.25	21.53
10	Original	3.6	5.49	7.43	9.22	148.71
	Only pre.	4.98	5.98	7.58	9.19	151.70
15	Original	1.29	7.72	10.26	12.26	205.36
	Only pre.	6.22	10.45	12.26	14.27	245.21
20	Original	7.03	11.49	12.76	14.09	255.20
	Only pre.	5.09	15.07	15.33	17.97	306.73

tal throughput respectively. By considering the *pre-processing stage* in the first extension, the throughput performance is improved compared to the original method.

2. **Results of Second Extension:** Tables 6.11 and 6.12 show the second extension results and the simulation results respectively. Table 6.11 shows the number of active *APs*, *CB or non-CB POC* for each *AP*, and the maximum or minimum transmission power for each *AP* for different value of *G*. Table 6.12 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the total throughput respectively. By considering both the *pre-processing* and *post-processing stages* in the second extension, the throughput performance is improved for all cases compared to the original method.

   After implementing the second extension, significant throughput improvements are observed in most cases in Table 6.12 when compared to the results from the first extension in Table 6.10. To evaluate the effectiveness of the second extension, we compare the *pre-processing and post-processing* results for $G = 15$. Figure 6.9 depicts the regular network topology with *AP* host associations for $G = 15$. In Figure 6.9 (a), only the pre-processing is considered, revealing that a few hosts experience poor associations, resulting in the lower throughput and the network performance degradation, indicated by the red marks. In Figure 6.9 (b), the second extension is illustrated, demonstrating improved associations through refining *AP-host associations*, which leads to improvements of the overall network performance.

Table 6.11: Comparison of network configuration with second extension for regular topology 1.

G Mbps	No. of active APs (Orig./ Pre. & post.)	Channel (Power) AP1, AP2, AP3, AP4, AP5, AP6, AP7	
		Original	With preprocessing & postprocessing
5	2/2	9+13(max), 1+5(max)	1+5(max), 9+13(max)
10	4/4	9+13(min), 1(max), 13(max), 7(max)	1+5(min), 13(max), 7(max), 1(max)
15	6/6	5+9(min), 13(min), 13(max), 13(max), 1(max), 7(max)	9+13(min), 1(max), 1(max), 13(min), 9+13(min), 1+5 (min)
20	7/7	1(max), 13(min), 7(min), 13(min), 13(min) , 9+13(min), 1+5(min)	9+13(min), 1+5(min), 1(min), 13(min), 1(max), 7(min), 1(max)

Table 6.12: Second extension simulation results for regular topology 1.

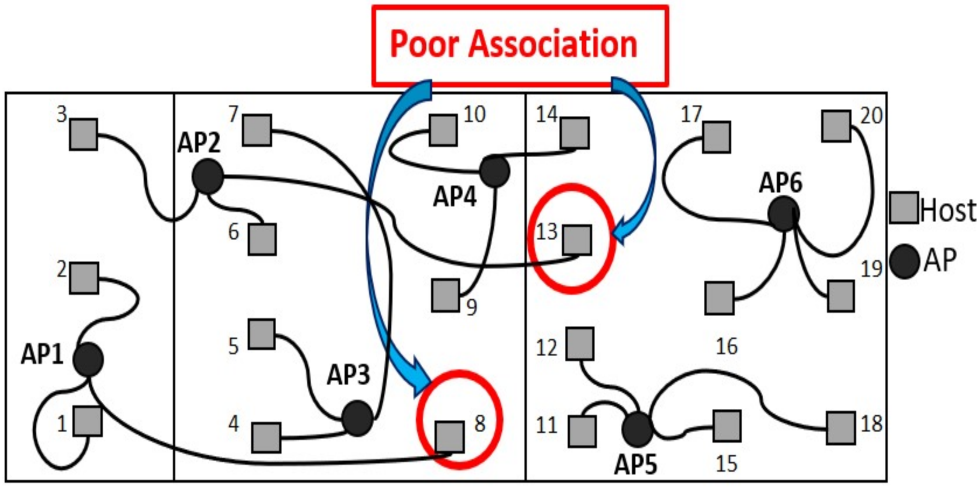
G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
5	Original	0.58	0.72	0.96	1.11	19.25
	With pre. & post.	0.69	0.82	1.07	1.25	21.53
10	Original	3.6	5.49	7.43	9.22	148.71
	With pre. & post.	4.98	5.98	7.58	9.19	151.7
15	Original	1.29	7.72	10.26	12.26	205.36
	With pre. & post.	4.07	8.33	12.54	15.18	250.90
20	Original	7.03	11.49	12.76	14.09	255.20
	With pre. & post.	8.61	15.67	15.81	18.26	316.12

6.4.2.3 Results for Regular Topology 2

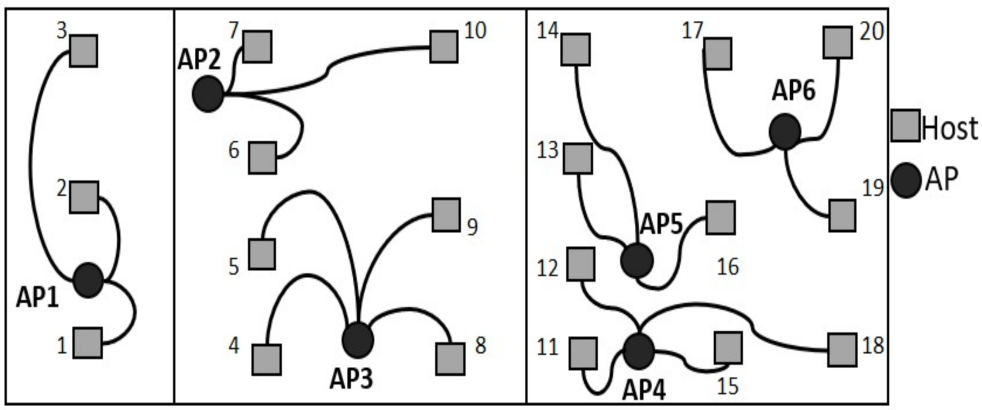
Finally, evaluate the proposal using a five room regular network topology 2. Figure 6.10 (a) illustrates the regular topology 2 with five rooms, where 30 hosts and 25 APs are regularly allocated in two 20m $\times$ 50m rooms, two 10m $\times$ 50m rooms, and one 40m $\times$ 50m room. Figures 6.10 (b) suggests the output of the pre-processing stage.

1. **Results of First Extension:** Tables 6.13 and 6.14 show the first extension results and the simulation results respectively. Table 6.13 shows the number of active *APs*, *CB or non-CB POC* for each *AP*, and the maximum or minimum transmission power for each *AP* for different value of *G*. Table 6.14 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the total throughput respectively. By considering the *pre-processing stage* in the first extension, the throughput performance is improved compared to the original method.

2. **Results of Second Extension:** Tables 6.15 and 6.16 show the second extension results and the simulation results respectively. Table 6.15 shows the number of active *APs*, *CB or non-CB POC* for each *AP*, and the maximum or minimum transmission power for each *AP* for different value of *G*. Table 6.16 shows the minimum host throughput, the average minimum host throughput, the average host throughput, the average maximum host throughput, and the total throughput respectively. By considering both the *pre-processing* and *post-processing*

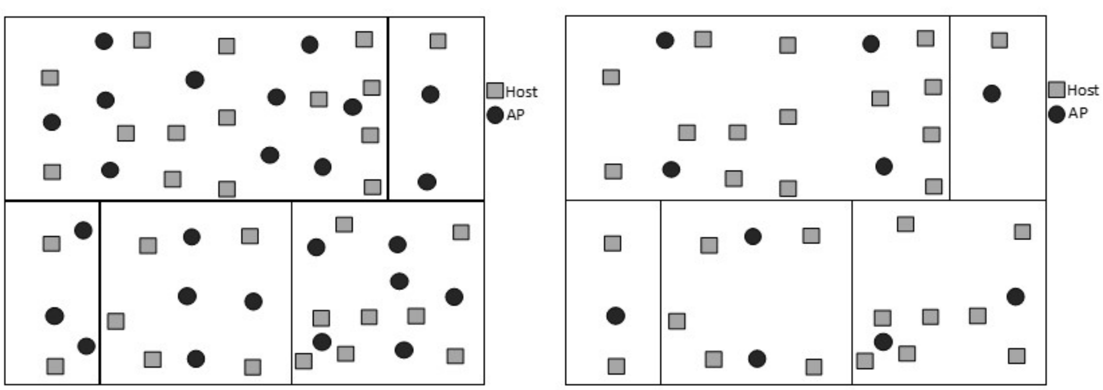


(a) First extension association



(b) Second extension association

Figure 6.9: AP host associations for G=15 in regular topology 1.



(a) Input topology with 25 candidate APs

(b) Output topology of preprocessing stage with 10 promising APs

Figure 6.10: Regular topology 2.

stages in the second extension, the throughput performance is improved for all cases com-

Table 6.13: Comparison of network configuration with first extension for regular topology 2.

G Mbps	No. of active APs (Orig./ Prepro.)	Channel (Power) AP1, AP2, AP3, AP4, AP5, AP6, AP7, AP8	
		Original	Only preprocessing
10	3/3	1(min), 9+13 (max), 1+5(max)	9+13(max), 1+5(max), 1(max)
15	4/4	1+5(max), 9+13(min), 13(min), 1(max)	1(max), 1+5(max), 13(max), 9+13(max)
20	5/6	7(min), 1+5(min), 9+13(min), 13(max), 1(max)	1+5(max), 5+9(min), 1(min), 13(max), 9+13 (min), 1+5 (min)
25	7/7	13(max), 1(min), 5+9(min), 9+13(min), 1+5(min) , 13(min), 9+13(min)	9+13 (min), 13 (max), 1(max), 1+5(min), 1(max), 9+13(min), 5+9(min)
30	8/8	13(max), 13 (max), 1(max), 9+13(min), 1+5(min), 13 (max), 7(min), 1+5(min)	9+13 (min), 1 (max), 1 (min), 9+13(min), 9+13 (min), 1+5 (min), 1+5 (min), 13 (min)

Table 6.14: First extension simulation results for regular topology 2.

G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	Original	0.04	0.23	0.58	0.79	17.54
	Only pre.	0.11	0.38	0.67	0.89	19.99
15	Original	0.51	1.01	2.73	4.08	82.19
	Only pre.	1.41	2.14	3.00	3.74	89.99
20	Original	0.67	1.86	4.47	6.15	134.23
	Only pre.	0.95	3.79	7.03	9.37	210.75
25	Original	1.05	3.59	8.04	12.26	241.45
	Only pre.	1.5	4.68	8.42	10.61	252.49
30	Original	2.41	7.11	10.58	13.85	317.64
	Only pre.	3.57	7.71	10.63	13.83	318.92

Table 6.15: Comparison of network configuration with second extension for regular topology 2.

G Mbps	No. of active APs (Orig./ Pre.& post.)	Channel (Power) AP1, AP2, AP3, AP4, AP5, AP6, AP7, AP8	
		Original	With preprocessing & postprocessing
10	3/3	1(min), 9+13 (max), 1+5(max)	1+5(max), 9+13(max), 13(max)
15	4/4	1+5(max), 9+13(min), 13(min), 1(max)	1(max), 1+5(max), 13(max), 9+13(max)
20	5/6	7(min), 1+5(min), 9+13(min), 13(max), 1(max)	1+5(max), 5+9(min), 1(min), 13(max), 9+13(min), 1+5(min)
25	7/7	13(max), 1(min), 5+9(min), 9+13(min), 1+5(min) , 13(min), 9+13(min)	5+9(max), 13(max), 1(max), 7(max), 7 (min) , 1+5 (min), 9+13 (min)
30	8/8	13(max), 13 (max), 1(max), 9+13(min), 1+5(min), 13 (max), 7(min), 1+5(min)	9+13(min), 1(max), 1(min), 9+13(min), 9+13(min), 1+5(min), 1+5(min), 13(min)

pared to the original method.

Table 6.16: Second extension simulation results for regular topology 2.

G mbps	Method	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	Original	0.04	0.23	0.58	0.79	17.54
	With pre. & post.	0.14	0.3	0.68	0.89	20.29
15	Original	0.51	1.01	2.73	4.08	82.19
	With pre. & post.	1.41	2.14	3.00	3.74	89.99
20	Original	0.67	1.86	4.47	6.15	134.23
	With pre. & post.	3.16	4.76	7.29	9.56	218.69
25	Original	1.05	3.59	8.04	12.26	241.45
	With pre. & post.	2.54	6.32	9.11	10.81	273.31
30	Original	2.41	7.11	10.58	13.85	317.64
	With pre. & post.	4.53	7.79	10.93	14.18	327.93

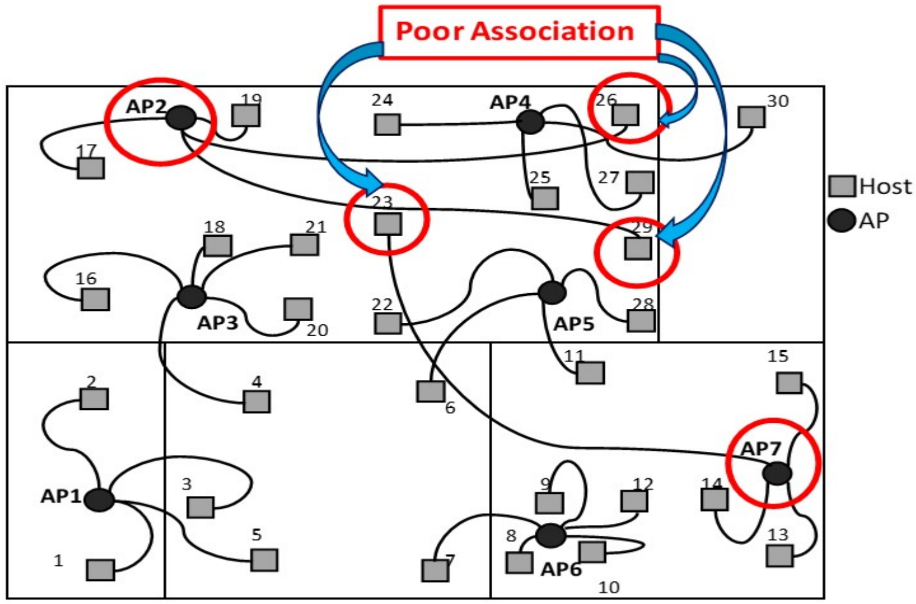
After implementing the second extension, significant throughput improvements are observed in most cases in Table 6.16 when compared to the results from the first extension in Table 6.14. To evaluate the effectiveness of the second extension, we compare the *pre-processing and post-processing* results for $G = 25$. Figure 6.11 depicts the regular network topology 2 with *AP* host associations for $G = 25$. In Figure 6.11 (a), only the pre-processing is considered, revealing that a few hosts experience poor associations, resulting in the lower throughput and the network performance degradation, indicated by the red marks. In Figure 6.11 (b), the second extension is illustrated, demonstrating improved associations through refining *AP-host associations*, which leads to improvements in the overall network performance.

6.4.3 Performance Analysis with Random AP Selection

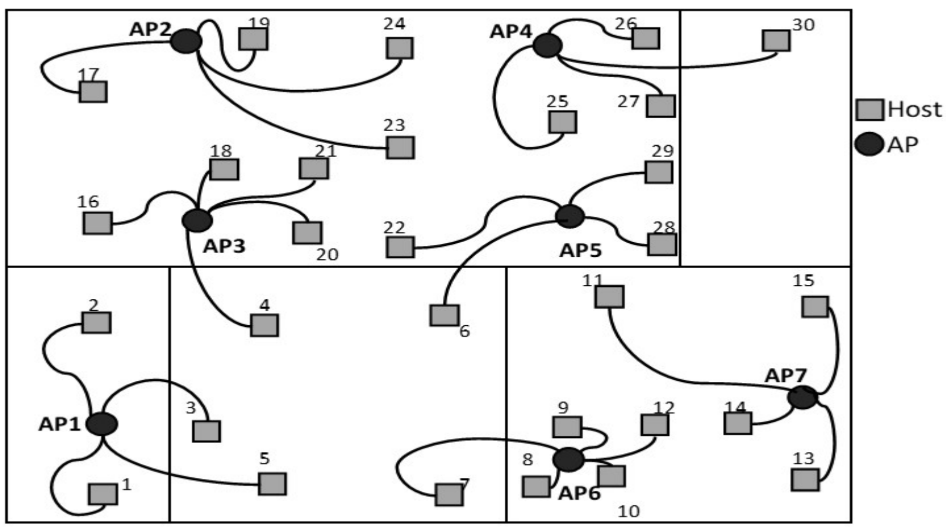
In the first extension, the *pre-processing stage* is considered, which mainly select the *promising candidate APs* depending on the network conditions. The first extension simulation results can be found in Tables 6.6, 6.10 and 6.14. These show that the results for random and regular topologies on the pre-processing stage of the AP joint optimization algorithm improve throughput performance compared to the original method. Furthermore, the pre-processing stage also reduced the *CPU* time by reducing the search space due to the selection of limited promising *APs*, which was already confirmed in [31].

If the promising *APs* are manually selected, gathering the necessary information takes a lot of effort and can be time-consuming. On the other hand, if they are randomly selected, it may carry the risk of poor selection. To evaluate the effectiveness of the pre-processing stage in the first extension, we compare the results with the results for randomly selecting candidate *APs*. Figure 6.12 illustrates the overall execution flow for the joint optimization algorithm with the *random AP selection method*.

Figure 6.13 shows the network topology where candidate *APs* are selected randomly. Figure 6.13 (a) illustrates the random topology with 5 randomly selected *APs*, Figure 6.13 (b) illustrates the regular topology 1 with 7 randomly selected *APs*, and Figure 6.13 (c) illustrates the regular topology 2 with 10 randomly selected *APs*.



(a) First extension association



(b) Second extension association

Figure 6.11: AP host associations for G=25 in regular topology 2.

Figure 6.14 shows the throughput comparison results between the random *AP* selection and the pre-processing methods for the random topology. Figures 6.14 (a) and 6.14 (b) show the minimum host throughput, the average minimum host throughput, the average host throughput, and the total throughput, respectively. The simulation results demonstrate that the random *AP* selection method provides lower throughput performance compared to the pre-processing method for all the values of G. For $G = 35$, although the *random AP selection method* yields the slightly higher average minimum host throughput, the minimum host throughput and the total throughput are lower than those of the *pre-processing method*.

Figure 6.15 shows the throughput comparison results between the *random AP selection* and

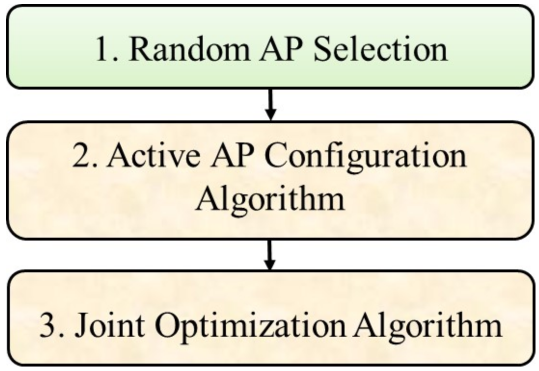
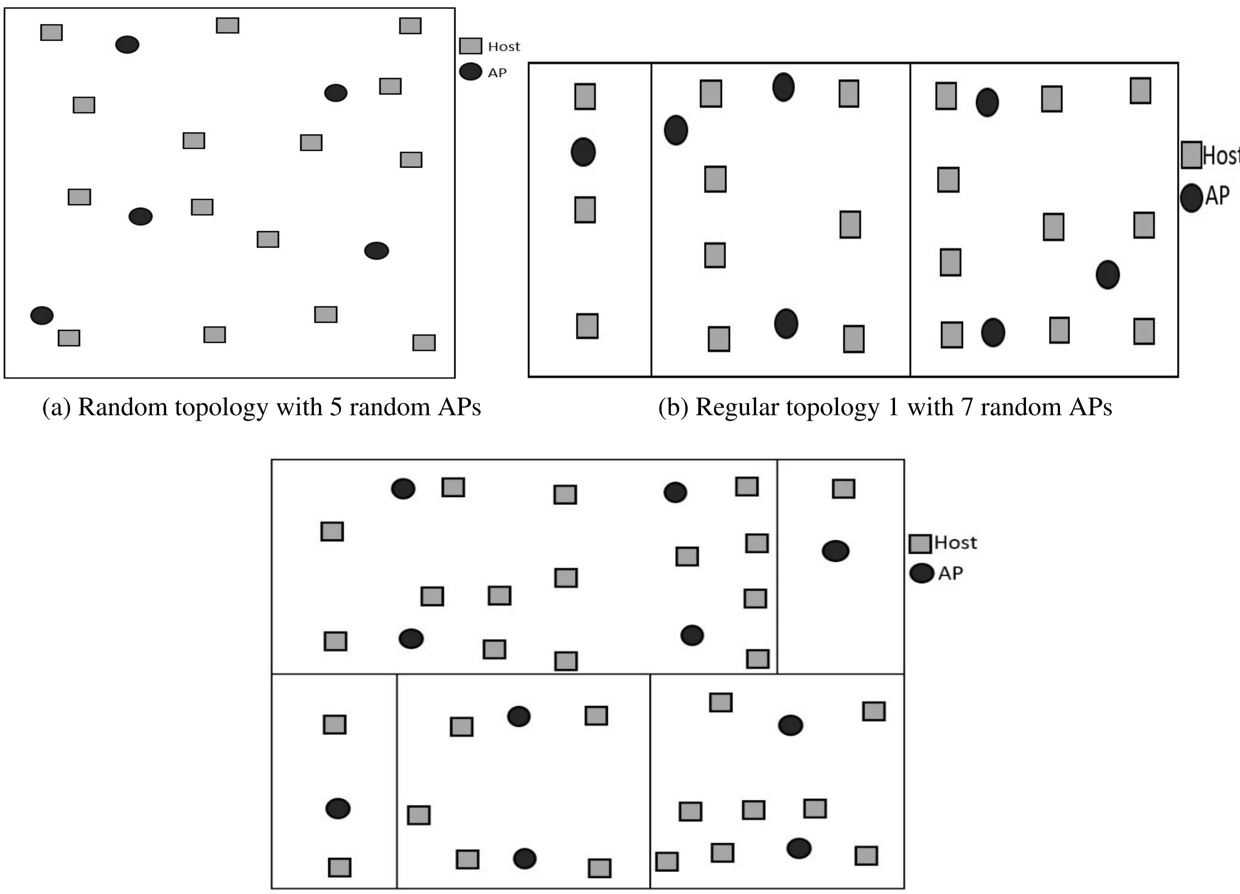


Figure 6.12: Flow of joint optimization algorithm with random AP selection method.



(a) Random topology with 5 random APs

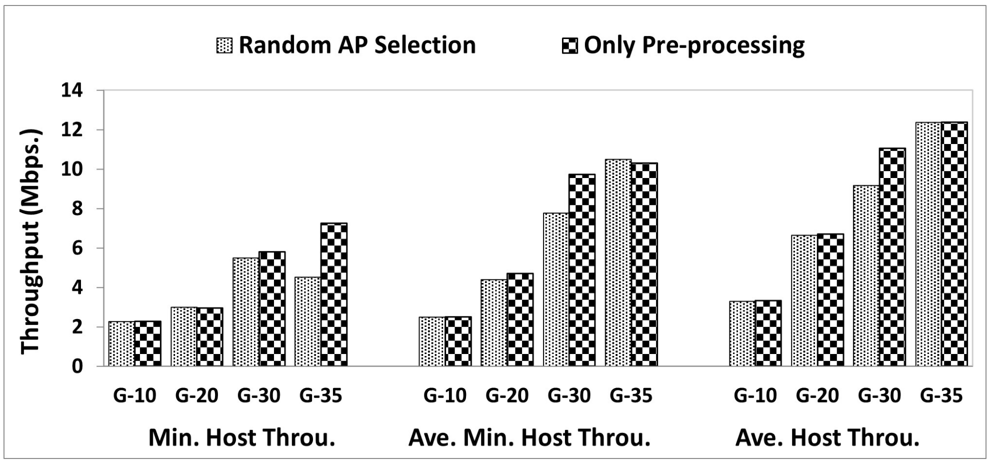
(b) Regular topology 1 with 7 random APs

(c) Regular topology 2 with 10 random APs

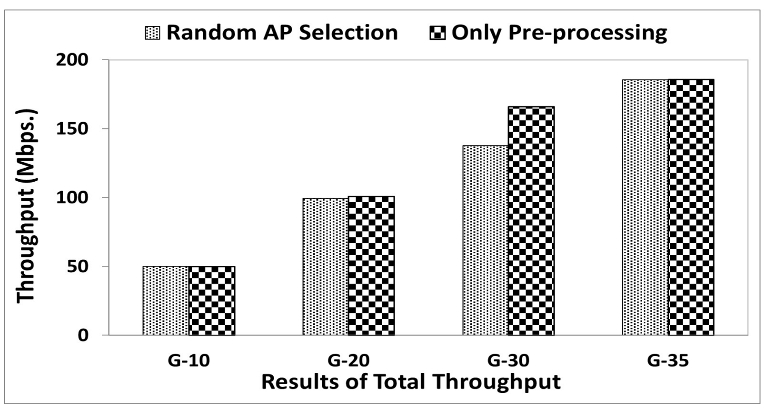
Figure 6.13: Topologies for random AP selection method.

the *pre-processing methods* for the regular topology 1. Figures 6.15 (a) and 6.15 (b) show the minimum host throughput, the average minimum host throughput, the average host throughput, and the total throughput respectively. The simulations demonstrate that the random *AP* selection method yields the lower throughput performance than the pre-processing method for all the *G* values.

Figure 6.16 shows the throughput comparison results between the *random AP selection* and the *pre-processing methods* for the regular topology 2. Figures 6.16 (a) and 6.16 (b) show the minimum host throughput, the average minimum host throughput, the average host throughput,

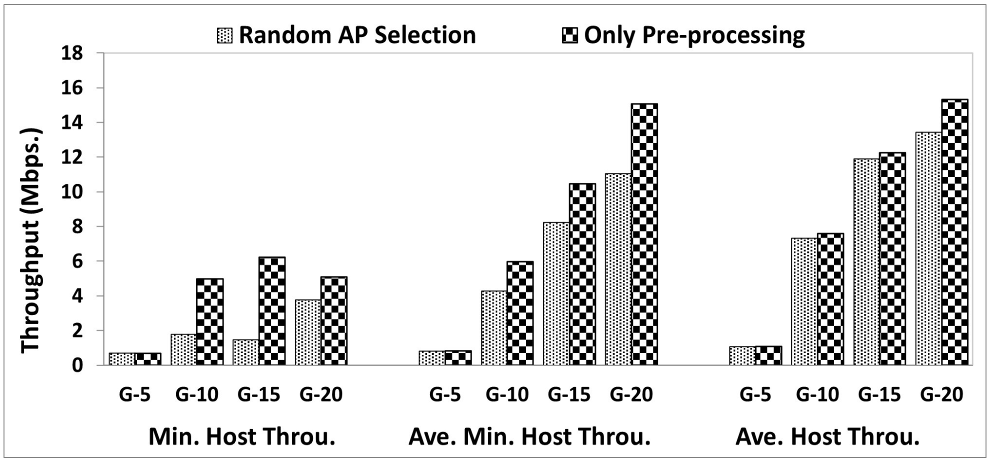


(a) Min. host, ave. min. host, and ave. host throughput

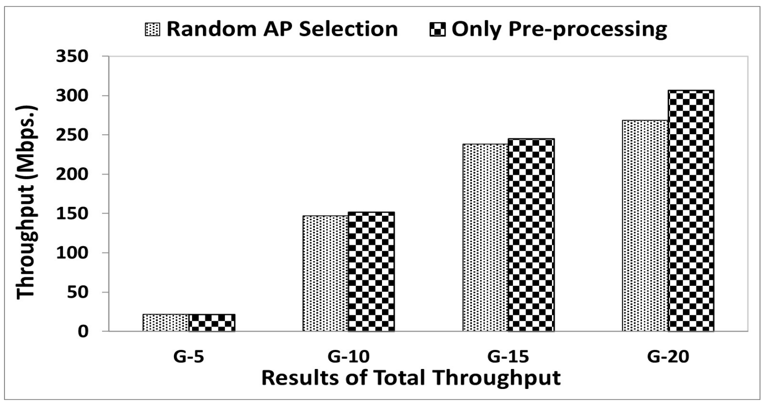


(b) Total throughput

Figure 6.14: Throughput comparison between random AP selection and pre-processing methods for random topology.



(a) Min. host, ave. min. host, and ave. host throughput



(b) Total throughput

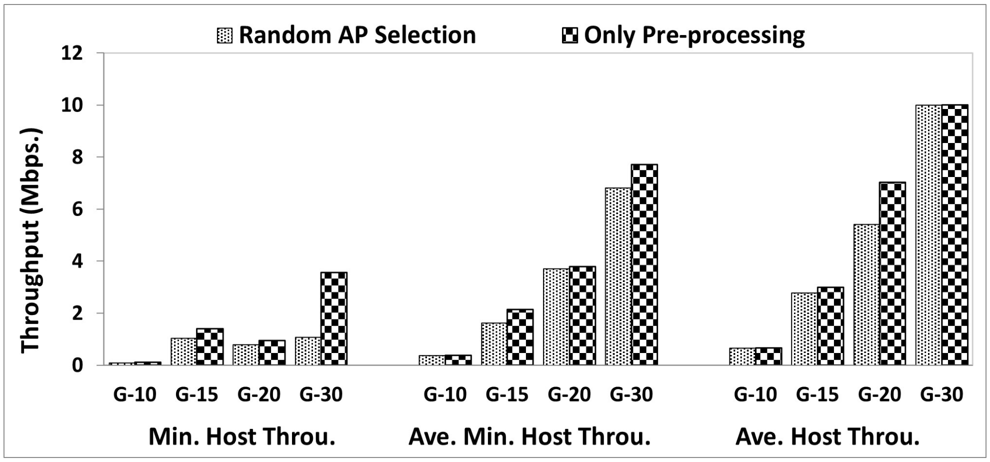
Figure 6.15: Throughput comparison between random AP selection and pre-processing methods for regular topology 1.

and the total throughput respectively. The simulations demonstrate that the random *AP* selection method yields the lower throughput performance than the pre-processing method across all the G values. For $G = 30$, although the random *AP* selection method results in the slightly higher total throughput, the minimum host throughput and the average minimum host throughput are lower than those of the pre-processing method. The average host throughput is same for both methods.

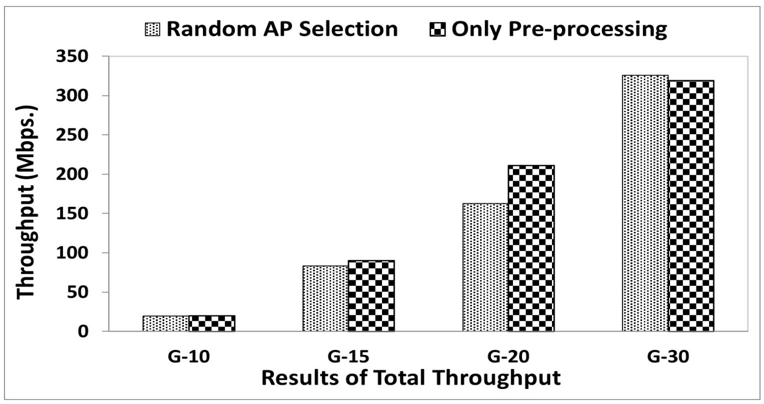
From the above simulation results, illustrate that for all the random and regular topologies, the *random AP selection method* yields the lower throughput performance compared to the *pre-processing method*. These results validate the effectiveness of the proposal of first extension.

6.4.4 Results for Greedy Approach

Tables 6.17, 6.18, and 6.19 show the simulation results for the greedy approach applied to random topology, regular topology 1, and regular topology 2, respectively. In the greedy approach, the *CB channel* is assign with maximum power to all active *APs* instead of joint optimization. However, the results indicate that this approach leads to worse throughput performance compared to the original method, as well as the first and second extensions.



(a) Min. host, ave. min. host, and ave. host throughput



(b) Total throughput

Figure 6.16: Throughput comparison between random AP selection and pre-processing methods for regular topology 2.

Table 6.17: Greedy approach simulation results for random topology.

G mbps	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	2.27	2.59	3.39	4.09	50.87
20	2.85	3.33	4.21	4.93	63.23
30	3.11	4.97	5.7	6.28	85.52
35	5.28	6.22	7	7.75	105.01

Table 6.18: Greedy approach simulation results for regular topology 1.

G mbps	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
5	0.69	0.82	1.07	1.25	21.52
10	3.24	4.16	4.69	5.17	93.9
15	3.96	5.02	6.01	6.81	120.17
20	5.88	7.67	7.69	8.16	153.98

Table 6.19: Greedy approach simulation results for regular topology 2.

G mbps	Min. h. throu.	Ave. Min. h throu.	Ave. h. throu.	Ave. Max. h. throu.	Total throu.
10	0.11	0.22	0.44	0.55	13.13
15	0.59	0.92	1.71	2.17	51.23
20	0.69	1.54	2.46	2.91	73.83
25	1.27	3.07	4.69	5.63	140.84
30	3.36	4.8	6.09	7.39	182.8

Table 6.20: Throughput performance of random topology.

Results	Method	Min. h.	A. min. h.	A. h.	Max. h.	A. max. h.	Total
Throu. (Mbps.)	Original	13.8	23.72	30.56	54.88	36.05	450.46
	Greedy	13.51	17.11	20.3	29.71	23.05	304.63
	Only pre.	18.3	27.31	33.48	63.3	39.14	502.11
	With pre. & post.	18.37	28.3	34.91	63.19	39.16	509.48
Improvement with original (%)	Only pre.	32.61	15.13	9.55	15.34	8.578	11.46
	With pre. & post.	33.11	19.31	14.23	15.14	8.62	13.10
Improvement with greedy (%)	Only pre.	35.45	59.61	64.92	113.03	69.81	64.82
	With pre. & post.	35.97	65.40	71.97	112.66	69.89	67.24

Table 6.21: Throughput performance of regular topology 1.

Results	Method	Min. h.	A. min. h.	A. h.	Max. h.	A. max. h.	Total
Throu. (Mbps.)	Original	12.50	25.42	31.41	55.05	36.68	628.52
	Greedy	13.77	17.67	19.46	28.81	21.39	389.57
	Only pre.	16.98	32.32	36.24	80.41	42.68	725.17
	With pre. & post.	18.35	30.80	37.00	81.78	43.88	740.25
Improvement with original (%)	Only pre.	35.84	27.14	15.37	46.06	16.35	15.37
	With pre. & post.	46.8	21.16	17.79	48.55	19.62	17.77
Improvement with greedy (%)	Only pre.	23.31	82.91	86.22	179.11	99.53	86.14
	With pre. & post.	33.26	74.31	90.13	183.85	105.14	90.01

6.4.5 Discussions of Simulation Results

Tables 6.20, 6.21, and 6.22 show the throughput performance for random topology, regular topology 1, and regular topology 2, respectively. The results demonstrate that the second extension results are better than the original, greedy, and first extension, which confirms the effectiveness of the proposal.

The results of the first extension improvement show that the pre-processing stage enhances the solution quality by optimizing the promising active *APs* compared to the original and greedy method. Additionally, the results of the second extension demonstrate that the post-processing stage has an impact on the host association and improves the throughput performance, which is better than the original, greedy method and the first extension. These results highlight the superior throughput performance of our proposal.

Table 6.22: Throughput performance of regular topology 2.

Results	Method	Min. h.	A. min. h.	A. h.	Max. h.	A. max. h.	Total
Throu. (Mbps.)	Original	4.68	13.8	26.4	57.71	37.13	793.05
	Greedy	6.02	10.55	15.39	25.24	18.65	461.83
	Only pre.	7.54	18.7	29.75	57.81	38.44	892.14
	With pre. & post.	11.78	21.31	31.01	63.77	39.18	930.21
Improvement with original (%)	Only pre.	61.11	35.51	12.68	0.17	3.52	12.49
	With pre. & post.	151.71	54.42	17.46	10.51	5.52	17.29
Improvement with greedy (%)	Only pre.	25.24	77.25	93.31	129.02	106.11	93.17
	With pre. & post.	95.68	101.99	101.49	152.63	110.08	101.42

6.5 Summary

In this chapter, I presented the *preprocessing stage* for the *active AP configuration algorithm*. Firstly, I presented the *preprocessing stage* including both the exhaustive and heuristic approaches. Secondly, I evaluated the *preprocessing stage* through simulations. Thirdly, I presented the application of the *preprocessing stage* to the *AP joint optimization algorithm* in WLAN with two extensions. In the first extension, the *pre-processing stage* of the *active AP configuration algorithm* is considered to select the promising candidate *APs* from a large number of *APs*. In the second extension, the *post-processing* stage is presented to refine the AP-host associations, and the use of both stages are considered. Lastly, the effectiveness of the proposals is evaluated through simulations using the *WIMNET* simulator. In the next chapter, I will present related works to this thesis in literature.

Chapter 7

Related Works in Literature

In this chapter, I survey works in literature related to this study. A significant amount of research works has addressed the problem related to interferences in WLAN, to enhance the throughput performance through the AP allocation, and the power optimization with the channel assignment. Within this survey, no works have reported together the simultaneous optimization of the transmission power, the frequency channel, the channel bonding, the AP allocation, and the AP-host associations in WLAN.

In [96], Lee et al. proposed a heuristic AP placement scheme for reliable wireless communications in industrial environments. They aimed to minimize the number of APs while providing each mobile user with at least two simultaneous links with different APs. Their scheme relied on the direct measurement of signal strength, and used passive repeaters to improve the communication reliability. The authors demonstrated the effectiveness and feasibility of their approach through experiments and simulations.

In [97], Khattak et al. discussed the challenges of using WLAN fingerprinting for indoor localization systems in smart and sustainable cities due to limited non-overlapping channels in WLAN that cause interferences. To address this, they proposed a channel assignment strategy based on the hearability and mutual distance of the APs. They present a simulation model, which demonstrates that the proposed strategy reduces interferences among neighboring APs.

In [98], Garroppo et al. presented energy conservation in enterprise WLANs during off-peak hours without compromising coverage and quality of service. The method involves three strategies: AP Management, Device Association, and Power Adjustment. The approach includes creating a comprehensive WLAN model and solving a mathematical problem using a developed algorithm. The aim is to save energy while maintaining WLAN performance.

In [99], Verma et al. conducted a detailed review of the latest features of *Multi-AP Cooperative (MAP-Co)* technology. They also highlighted various drawbacks, such as the need for multi-AP channel sounding, coordinated *Orthogonal Frequency Division Multiple Access (OFDMA)*, coordinated spatial reuse, coordinated beamforming, and joint transmission. Additionally, this study discussed potential future directions and challenges that may arise with use of *MAP-Co* over the emerging *Wireless Local Area Networks (WLANs)*.

In [100], Ali et al. proposed a model to identify the optimal locations of outdoor access points in a large university campus with limited resources. They collected data on the current locations of 17 outdoor access points, 25 demand points, and 14 new potential locations. The *Set-Covering* model suggests that 21 APs are required to cover all the demand points, while the *maximal-coverage location problem (MCLP)* model shows that 17 access points can cover 92.34% of the demand.

In [101], Naif et al. proposed a multi-level optimization algorithm for *access point (AP)* placement in a multi-floor building using *Binary Particle Swarm Optimization (BPSO)*. The algorithm considers five pairs of weights, signal thresholds, and *received signal strength (RSS)* measurements. They compared the algorithm's results with the current AP deployment in the target building and showed that it outperforms the current AP deployment in terms of *RSS*, path loss, interference, and the number of APs required. However, their study did not consider the effects of different sources of interference and co-channel interference.

In [102], Liu et al. proposed a method to optimize APs in WLANs and used *Fruit Fly Optimization Algorithm (FOA)* to jointly optimize the location and transmit power of each AP, reducing power consumption and improving coverage rate. Redundant APs were removed below a threshold. However, interference with concurrent communication was not considered.

In [103], Roy et al. proposed an algorithm for dual interfaces with channel bonding that optimizes network performance and reduces the number of active APs in dense WLAN environments. The algorithm uses a throughput estimation model, a greedy algorithm, a local search method, and a simulated annealing technique. The proposal was evaluated through simulations and experiments using *Raspberry Pi* APs and *Linux* PCs with positive results. However, the AP location and power optimization were not considered.

In [104][105], Bejerano et al. proposed a load balancing approach among APs to mitigate congestion and fair distributions of users. Each host monitors the wireless channel qualities that it experiences from nearby APs and reports them to the network control center that determines the *AP-host associations*. Since the objective function of the proportional fairness is non-linear, its implementation is much more challenging, where detecting the bottleneck users and finding their normalized bandwidth is *NP-hard*. However, their proposal does not consider the joint optimization for concurrent communication.

In [106], Tewari et al. presented a combined *transmission power* and *POC assignment* optimization algorithm to maximize the network performance in dense WLAN. Efficient use of *POC* can increase the frequency reuse by reducing the interference range. However, the excessive use of *POCs* may cause adjacent channel interferences. To improve the quality of service they proposed an effective power control method, where the effectiveness is confirmed by simulations and does not consider the AP locations.

In [107], Kachroo et al. have proposed an algorithm that incorporates both the channel assignment and the transmission power control for a multi-rate WLAN that uses conventional $20MHz$ *non-CB POC channels*. The algorithm starts by assigning channels to the APs while keeping the transmission power constant. It then optimizes the transmission power by gradually increasing or decreasing it to maximize the *signal to interference plus noise ratio (SINR)*. However, their proposal does not consider the host association.

In [20], Mittal et al. adopted a game theoretic approach to balance the loads among the APs. Users associated with highly congested APs are moved to less loaded APs for better throughputs. However, concurrent communication interference was not considered.

In [108], Shindo et al. presented a *virtual access point (VAP)* aggregation method based on the *received signal strength indicator (RSSI)* and the *current bandwidth (BW)* usage. This method avoids the throughput degradation of mobile nodes due to the *VAP* aggregation and reduces the number of running APs. Their proposal does not consider joint optimization for concurrent communication.

In [109], Broustis et al. proposed the "AP association approach" to improve network throughput and balance loads among *access points (APs)* in a dense WLAN. They used a measurement-driven framework with three objective functions: frequency/channel selection, user association based on AP load and signal strength, and power control for each AP. However, applying all of them can lead to suboptimal performance. A host is initially associated with the AP that provides the highest receiving signal strength. Overloaded hosts are migrated to lightly loaded APs. However, concurrent communication interference was not considered.

In [110], Prommak et al. proposed a novel network design algorithm for WLANs, focusing on the optimal access point placement and the frequency channel assignment to enhance *Quality of Service (QoS)*. They use a cross-layer approach, considering both the physical layer and data link layer functionalities, and formulate a multi-objective optimization problem. Numerical results show that the proposed model improves signal coverage and system throughput by balancing the interference level and the MAC protocol.

In [111], Uemura et al. discussed *Wireless Internet-access Mesh NETwork (WIMNET)*, which used two types of access points (APs): *smart APs (SAPs)* and *conventional APs (CAPs)* to provide scalable and cost-effective internet access. To improve network performance, the authors proposed an algorithm to optimize *SAP* allocation. The proposed algorithm aims to minimize transmission time and balance *SAP* loads. The effectiveness is verified through simulations of various network topologies.

In [112], Yang et al. proposed a method to optimize the selection of measurement points for the access point (AP) localization to reduce manual efforts and improve accuracy. In their proposal, the next measurement point is determined based on real-time data and is located at the intersection of the coverage areas of APs, whose locations are roughly estimated from previous measurements.

In [113], Arivoli et al. proposed a load-power control algorithm that used *PWmin* and *PWmax* power control techniques to optimize the transmission power based on the network load and distance. The proposed algorithm was simulated using *NS-2*, showing improvements in network throughput, delay, and energy savings.

In [114], Adewalet et al. compared various channel assignment schemes for mobile *WiMAX* networks, highlighting that the dynamic guard channel assignment minimizes the call blocking probability better than other schemes. *MATLAB* simulations showed that the dynamic guard channel assignment outperforms the non-prioritized, prioritized guard channel and the prioritized guard channel with buffer schemes.

In [115], Kobayashi et al. addressed the issue of optimizing Wi-Fi APs by considering the type and volume of traffic generated by users, rather than assuming a fixed AP position. The authors proposed *CHASA*, a system that dynamically updates the optimal location of a movable AP based on the type of traffic. This system used a decision tree for traffic classifications and a hill climbing method to find the optimal AP position.

In [116], Alakhras et al. proposed a two-step algorithm for resource allocations in wireless networks, focusing on subcarrier allocations based on channel quality and multicast service priority, followed by power reallocation to enhance system capacity while ensuring QoS. The interval-based algorithm allocates subcarriers and power to multicast groups, aiming to maximize throughput and minimize complexity.

In [117], Mestre et al. proposed an algorithm that selects the best AP based on the network delay rather than RSSI. The proposed method improved the overall network performance by choosing APs with the lower delay.

In [118], Deng et al. optimized the layout of WLAN APs to improve location reliability and accuracy by adding minimally redundant APs. They proposed a method to ensure that even if one AP fails, the remaining APs can still cover all the areas with minimal location error.

Chapter 8

Conclusion

In this thesis, I presented studies of enhancements of the *active access-point configuration* for 2.4GHz IEEE 802.11n wireless local-area networks (WLANs).

Firstly, I proposed the measurement results and the modifications of the *throughput estimation model* for concurrent communication of multiple links under various conditions with 11 and 13 *POCs* respectively for upto three APs. Through experiments, I observed that the total throughput gradually improves as the number of wall obstacles or the physical distance between *APs* increases when three *APs* are placed in different rooms in the network field. Then, I extend the model modification with 13 *POCs* considering the interference effects among APs under different numbers of wall obstacles in the presence of concurrent communications. I verified the accuracy of the modified throughput estimation model through simulations and experiments where estimated throughputs and measured ones matched well.

Secondly, I proposed the *network configuration optimization algorithm* for *IEEE 802.11n WLAN* with three *Raspberry Pi* APs, by utilizing the modified throughput estimation model. This algorithm optimize the configuration of the network by selecting the CB or non-CB, channel assignment, and associated hosts for each AP, to maximize the throughput performance of the network. The effectiveness of the proposal is confirmed through simulations and testbed experiments in different network topologies.

Thirdly, I proposed the *preprocessing stage* for the active AP configuration algorithm to reduce the CPU time by confining the search space by selecting promising candidates for active APs. Both the exhaustive and heuristic approaches are presented for this stage. The effectiveness of the proposal is evaluated through simulations using the *WIMNET simulator*.

Lastly, I proposed the *application of the preprocessing stage* of the active AP configuration algorithm. Here, I proposed *two extensions* to the *AP joint optimization algorithm* to improve the performance of *WLAN*. In the first extension, I adopted the *pre-processing stage* of the *active AP configuration algorithm* in the *AP joint optimization algorithm* to select the promising candidate *APs*. And, in the second extension, I proposed the *post-processing stage* of the algorithm, which refines *AP-host* associations to further improve the overall network performance. The effectiveness of the proposal is evaluated through simulations using the *WIMNET simulator*.

In future studies, I will study further enhancements of the *active access-point configuration* for 2.4GHz IEEE 802.11n wireless local-area networks (WLANs) by improving the related models and algorithms. Then, I will evaluate them in various network fields and topologies.

Bibliography

[1] M. S. Gast, “802.11 wireless networks - the definitive guide,” 2nd ed., O’Reilly, Sebastopol, Apr. 2005.

[2] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, “IEEE 802.11 wireless local area networks,” IEEE Commun. Mag., vol. 35, no. 9, pp. 116-126, Sept. 1997.

[3] M. Balazinska and P. Castro, “Characterizing mobility and network usage in a corporate wireless local-area network,” in Proc. Int. Conf. Mobile Syst. Appl. Serv. pp. 303-316, 2003.

[4] S. Zeadally, O. Bello, “Harnessing the power of Internet of Things based connectivity to improve healthcare,” Int. Things (2019) 100074.

[5] H.A. El Zouka, M.M. Hosni, “Secure IoT communications for smart healthcare monitoring system,” Int. Things (2019) 100036.

[6] D. Mocrii, Y. Chen, P. Musilek, “IoT-based smart homes: A review of system architecture, Software, Communications, Privacy and Security,” Int. Things 1 (2018) 81–98.

[7] M. M. Islam, N. Funabiki, R. W. Sudibyo, K. I. Munene, W. C. Kao, “A dynamic access-point transmission power minimization method using PI feedback control in elastic WLAN system for IoT applications,” Int. Things 8 (2019) 100089.

[8] J. O. Agyemang, J. J. Kponyo, G. S. Klogo, J. O. Boateng, “Lightweight rogue access point detection algorithm for WiFi-enabled internet of things (IoT) devices,” Int. Things 11 (2020) 100200.

[9] H. D. Ghael, Dr. L Solanki, and G. Sahu, “A review paper on Raspberry Pi and its applications”, Int. J. Adv. Eng. Manage. (IJAEM), vol. 2, no. 12, pp. 225-227, 2021.

[10] Raspberry Pi, Teach, Learn, and Make with Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/.

[11] S. K. Pritish Sachdeva, “A review paper on Raspberry Pi”, vol. 4, no. 6, Dec. 2014.

[12] H. Chaudhari, “Raspberry Pi technology: a review,” vol. 2, no. 3, 2015.

[13] S. K. Debnath, N. Funabiki, K. S. Lwin, Md. S. A. Mamun, R. W. Sudibyo and S. Huda, “Raspberry Pi configuration for access-point and its throughput measurements in IEEE802.11n wireless networks,” IEICE Tech. Rep., NS2016-137, pp. 101-106, Dec. 2016.

[14] R. W. Sudibyo, K. S. Lwin, N. Funabiki, M. Saha, and M. Kuribayashi, “A study of channel bonding configuration and performance for Raspberry Pi access-point in wireless local-area network,” IEICE Tech. Rep., SRW2018-10, pp. 7-12, Aug. 2018.

[15] "Supplement to 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: high-speed physical layer in the 5 GHz band," IEEE STD 802.11a-1999, 1999.

[16] "IEEE Std 802.11-2012, IEEE standard for information technology- telecommunications and information exchange between systems local and metropolitan area network - specific requirements - Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE STD 802.11-2007, pp. 1184, 2007.

[17] J. Talvitie, M. Renfors, and E. S. Lohan, "A comparison of received signal strength statistics between 2.4 GHz and 5 GHz bands for WLAN-based indoor positioning," in Proc. IEEE Globecom Work., pp. 1-6, 2015.

[18] K. Zhou, X. Jia, Y. Chang, and X. Tang, "Partially overlapping channel assignment for WLANs using SINR interference model," Int. J. Comm. Syst, vol. 27, no. 11, pp. 3082-3095, March 2014.

[19] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: turning a peril into an advantage," in Proc. ACM Conf. Inter. Meas., pp. 311-316, 2005.

[20] K. Mittal, E. M. Belding, and S. Suri, "A game-theoretic analysis of wireless access point selection by mobile users," Computer Commun., vol. 31, no. 10, pp. 2049-2062, Jan. 2008.

[21] National Instrument, Introduction to wireless LAN measurements from 802.11a to 802.11ac, 2018.

[22] M. S. A. Mamun, M. E. Islam, N. Funabiki, M. Kuribayashi, and I.-W. Lai, "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, July 2016.

[23] 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, July 2017.

[24] 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 802.11n wireless networks," Int. J. Netw. Comput., vol. 7, no. 2, pp. 187-207, July 2017.

[25] K. I. Munene, N. Funabiki, M. M. Rahman, H. Briantoro, S. C. Roy, and M. Kuribayashi, "A throughput drop estimation model and its application to joint optimization of transmission power, frequency channel, and channel bonding in IEEE 802.11n WLAN for large-scale IoT Environments," Int. Things 20 (2022) 100583.

[26] M. Saha, R. W. Sudibyo, N. Funabiki, and M. Kuribayashi, "Modifications of throughput estimation model for concurrent communications of multiple Raspberry Pi access-points in wireless local-area network," IEICE Tech. Report, SRW2018-9, pp. 1-6, Aug. 2018.

[27] M. Saha, R. W. Sudiobyo, Funabiki, N., Kuribayashi, M., S. K. Debnath, S. K., and Kao, W-C. (2019) "Throughput Measurement and Estimation Model for Concurrent Communications of Multiple Raspberry Pi Access-Points in Wireless Local-Area Network", Wireless Networks, J. Comm., vol. 14, no. 11, pp. 1009-1016.

[28] M. Saha, N. Funabiki, R. W. Sudibyo, M. Kuribayashi, and W.-C. Kao, "A Network Configuration Optimization Algorithm for Wireless Local-Area Network with Three Raspberry Pi Access-Points under Concurrent Communications," International Conference on Future Computer and Communication, 2019.

[29] M. Saha, N. Funabiki, R. W. Sudiobyo, M. Kuribayashi, S. K. Debnath, and W. C. Kao, "A network configuration optimization algorithm for wireless local-area network with three Raspberry Pi access-points under concurrent communications," Int. J. Future Comp. Com., vol. 8, no. 2, pp. 55-62, 2019.

[30] N. Funabiki ed., "Wireless mesh networks," InTech-Open Access Pub., Jan. 2011, Internet `http://www.intechopen.com/books/wireless-mesh-networks`.

[31] M. Saha, N. Funabiki, S. K. Debnath, W.-C. Kao, S. Tajima, and T. Higashino, "A proposal of preprocessing stage for active access point configuration algorithm in elastic WLAN system," ICIET, pp. 284-289, March 28–30, 2020.

[32] S. Banerji and R. S. Chowdhury, "On IEEE 802.11: wireless LAN technology," Int. J. Mobi. Net. Commun. & Telematics, vol. 3, no. 4, Aug. 2013.

[33] K. Shi, Y. shu, and J. Feng, "A MAC layer congestion control mechanism in IEEE 802.11 WLANs,"" 4th Int. Conf. on Commun. and Netw., (ChainaCom-2009), pp.1-5, Aug. 2009.

[34] I. Syed, and B. Roh, "Adaptive backoff algorithm for contention window for dense IEEE 802.11 WLANs," Mobile Inf. Sys., pp. 1-11, May 2016.

[35] I. Poole, "IEEE 802.11 Wi-Fi standards," Internet: `http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11-standards-tutorial.php`, Access Jan. 20, 2019.

[36] S. Banerji, "Upcoming standards in wireless local area networks," Wireless & Mobile Technol., vol. 1, no. 1, Sept. 2013.

[37] V. Beal, "What is 802.11 wireless LAN standards?," Internet: `http://www.webopedia.com/TERM/8/802_11.html`, Access Jan. 20, 2019.

[38] C. C. Choon, "A study of active access-point selection algorithm for wireless mesh network under practical conditions," Ph.D. thesis, Grad. Sch. Natural Sci. Technol., Okayama Univ., Japan, Sept. 2015.

[39] "IEEE 802.11 - Wikipedia," Internet: `http://en.wikipedia.org/wiki/IEEE_802.11`, Access Jan. 20, 2019.

[40] M. E. Islam, "A study of access-point aggregation algorithm for elastic wireless local-area network system and its implementation," Ph.D. thesis, Grad. School of Natural Science and Technology, Okayama Univ., Japan, March 2016.

[41] S. Sendra, M. Garcia, C. Turro, and J. Loret, "WLAN IEEE 802.11 a/b/g/n indoor coverage and interference performance study," Int. J. Adv. Netw. Service., vol. 4, no. 1, pp. 209-222, 2011.

[42] M. M Islam "A study of dynamic access-point configuration and power minimization in elastic wireless local-area network system," Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Sept. 2019.

[43] P. Likhar, R. S. Yadav and K. M. Rao, "Securing IEEE 802.11g WLAN using open VPN and its impact analysis," Int. J. of Netw. Security and Its Applications (IJNSA), Vol. 3, no. 6, pp. 97-113, Nov. 2011

[44] A. S. Tanenbaum, and D. J. Wetherall, "Computer networks," 5th ed., Pearson Prentice Hall, 2011.

[45] J. H. Yeh, J. C. Chen, and C. C. Lee, "WLAN standards," IEEE Potentials, vol. 22, no. 4, 2003.

[46] O. Bejarano, E. W. Knightly, and M. Park, "IEEE 802.11ac: from channelization to multi-user MIMO," IEEE Comm. Mag., vol. 51, no. 10, pp. 84-90, Oct. 2013.

[47] L. Verma, M. Fakharzadeh, and S. Choi, "Wifi on steroids: 802.11ac and 802.11ad," IEEE Trans. Wireless Comm., vol. 20, no. 6, pp. 30-35, Dec. 2013.

[48] M. S. Gast, "802.11ac: a survival guide," O'Reilly Media, Inc., 2013.

[49] B. Bellalta, "IEEE 802.11 ax: High-efficiency WLANs," IEEE Wirel. Commu. vol. 23, no. 1, pp. 38-46, 2016.

[50] E. Khorov, A. Kiryanov, A. Lyakhov and G. Bianchi, "A Tutorial on IEEE 802.11ax High Efficiency WLANs," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 197-216, 2019.

[51] IEEE, "IEEE SA Standards Board Approvals," Internet: `https://standards.ieee.org/about/sasb/sba/feb2021.html.` Accessed 25 May, 2021.

[52] IEEE, "IEEE 802.11ax-2021 - IEEE Standard for Information Technology," Internet: `https://standards.ieee.org/standard/802_11ax-2021.html.` Accessed 25 May, 2021.

[53] H. Briantoro, "A study of transmission power optimization with channel assignment consideration for access-points in wireless local-area network," Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Sept. 2021.

[54] I. M. Kwenga, N. Funabiki, M. M. Islam, M. Kuribayashi, R. W. Sudibyo, and W. -C. Kao, "A throughput estimation model under two-link concurrent communications with partially overlapping channels and its application to channel assignment in IEEE 802.11n WLAN," Int. J. Space-Base. Situ. Comp., vol. 8, no. 3, pp. 123-137, 2018.

[55] 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. Inform. Syst., vol. E104-D, no. 5, pp. 585-596, May, 2021.

[56] “WiMAX page, IEEE802.11ac works,” Internet: http://123-info.net/wimax-page/archives/918, Access Jan. 20, 2019.

[57] L. Deek, E. Garcia-Villegas, E. Belding, S. J. Lee, and K. Almeroth, “The impact of channel bonding on 802.11n network management,” in Proc. Conf. Emer. Netw. Exp. Tech. (CoNEXT), Dec. 2011.

[58] K. I. Munene, “A study of throughput estimation model and channel assignment under partially overlapping channels in IEEE 802.11n wireless local-area network,” Master thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, March. 2019.

[59] A. Saif, M. Othman, “A reliable A-MSDU frame aggregation scheme in 802.11n wireless networks,” in Proc. Comp. Scie., vol. 21, 2013.

[60] “IEEE 802.11n-2009 - Wikipedia,” Internet: http://en.wikipedia.org/wiki/IEEE_802.11n-2009, Accessed 12 Nov. 2021.

[61] NEC Inc., “WG2600HP manual,” Internet: http://www.aterm.jp/support/manual/pdf/am1-002673.pdf, Access Dec. 8, 2024.

[62] Buffalo Inc., “AirStation WZR-1750DHP user manual,” Internet: http:manual.buffalo.jp/buf-doc/35013154-01_EN.pdf, Access Aug. 13, 2016.

[63] IO data Inc., Internet: http://www.iodata.jp/product/network/wnlan/wn-ac1600dgr3/index.htm, Access Dec. 8, 2024.

[64] “Hostapd: the linux way to create virtual Wifi access point,” Internet: nims11.wordpress.com/2012/04/27/hostapd-the-linux-way-to-create-virtual-wifi-access-point/, Access Jan. 20, 2019.

[65] Raspberry Pi, “The official website of the Raspberry Pi project,”, Internet: http://www.raspberrypi.org/, Access Jan. 15, 2016.

[66] Wiki, “Raspberry Pi 3 Model B,” Internet: http://wiki.seeedstudio.com/Raspberry_Pi_3_Model_B/, Access Jan. 15, 2016.

[67] L. Deek, E. G. Villegas, E. Belding, S. Lee, and K. Almeroth, “Intelligent channel bonding in 802.11n WLANs,” IEEE Trans. Mobile Comput., vol.13, no.6, pp. 1242-1254, Jun. 2014.

[68] R. W. Sudibyo, K. S. Lwin, N. Funabiki, M. Saha, and M. Kuribayashi, “A study of channel bonding configuration and performance for Raspberry Pi access-point in wireless local-area network,” IEICE Tech. Rep., SRW2018-10, pp. 7-12, Aug. 2018.

[69] M. Vipin, and S. Srikanth, “Analysis of open source drivers for IEEE 802.11 WLANs,” in Proc. Int. Conf. Wireless Comm. Sens. Comput. (ICWCSC), pp. 1-5, IEEE, Jan. 2010.

[70] J. Malinen, “hostapd: IEEE 802.11 AP, IEEE 802.1 X. WPA/WPA2/EAP/RADIUS authenticator,” Internet: https://w1.fi/hostapd/, Access 12 Nov. 2021.

[71] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, “SSH, the secure shell: the definitive guide,” O’Reilly Media Inc., 2005.

[72] T. Ylonen and C. Lonvick, "The secure shell (SSH) protocol architecture," 2006.

[73] "OpenSSH," Internet: http://www.openssh.com/, Access Jan. 20, 2019.

[74] O. Terpollari, "How to install and configure OpenSSH server in Linux," Internet: http://www.tecmint.com/install-openssh-server-in-linux/, Access Jan. 20, 2019.

[75] "Running commands on a remote Linux server over SSH," Internet: http://www.shellhacks.com/en/Running-Commands-on-a-Remote-Linux-Server-over-SSH, Access Jan. 20, 2019.

[76] "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool," Internet: https://iperf.fr/, Accessed 20 May 2021.

[77] T.-D. Chiueh, P.-Y. Tsai, and I.-W. Lai, "Baseband receiver design for wireless MIMO-OFDM communications," 2nd ed. Wiley-IEEE Press., 2012.

[78] D. B. Faria, "Modeling signal attenuation in IEEE 802.11 wireless LANs," Tech. Report, TR-KP06-0118, Stanford Univ., July 2005.

[79] 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. Work. Virtual Environ. Netw.-Orient. Appli. (VENOA), pp. 701-710, July 2017.

[80] K. I. Munene, N. Funabiki, M. M. Islam, M. Kuribayashi, M. S. A. Mamun, R. W. Sudibyo, and W.-C. Kao, "An extension of throughput drop estimation model for three-link concurrent communications under partially overlapping channels and channel bonding in IEEE 802.11n WLAN," Adv. Sci. Technol. Eng. Syst. J., vol. 4, no. 4, pp. 94-105, 2019.

[81] OriginLab, Internet [online], https://www.originlab.com/, Access June 25, 2023.

[82] 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.

[83] D. P. Williamson, and D. B. Shmoys, "The design of approximation algorithms," Cambridge Univ. Press, April 2011.

[84] "Setting up a Raspberry Pi as a WiFi access point," Internet: https://cdn-learn.adafruit.com/downloads/pdf/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf, Access Jan. 20, 2019.

[85] "Arp Scan," Internet: http://www.nta-monitor.com/tools-resources/security-tools/arp-scan, Access Jan. 20, 2019.

[86] "nm-tool," Internet: http://linux.die.net/man/1/nm-tool, Access Jan. 20, 2019.

[87] "View your current network settings with nm-tool," Internet: https://nfolamp.wordpress.com/2010/05/21/view-your-current-network-settings-with-nm-tool/, Access Jan. 20, 2019.

[88] M. E. Islam, K. S. Lwin, M. S. A. Mamun, N. Funabiki, and I.-W. Lai, "Measurement results of three indices for IEEE802.11n wireless networks in outdoor environments," in Proc. IEEE Hiroshima Sect. Student Symp. (HISS), pp. 410-414, Nov. 2015.

[89] "nmcli - command-line tool for controlling NetworkManager," Internet: `http://manpages.ubuntu.com/manpages/maverick/man1/nmcli.1.html`, Access Jan. 20, 2019.

[90] "Using the NetworkManager command-line tool, nmcli," Internet: `https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/sec-Using_the_NetworkManager_Command_Line_Tool_nmcli.html`, Access Jan. 20, 2019.

[91] "Learning Linux Commands: sed," Internet: `https://linuxconfig.org/learning-linux-commands-sed`, Access Jan. 20, 2019.

[92] K. Pelechrinis, L. Broustis, T. Salonidis, S. V. Krishnamurthy, and P. Mohapatra, "Design and deployment considerations for high performance MIMO testbeds," Proc. Int. Conf. Wireless Internet (WICON), pp. 245-246, Nov. 2008.

[93] C. Newport, D. Kotz, Y. Yuan, R. S. Gray, J. Liu, and C. Elliott, "Experimental evaluation of wireless simulation assumption," J. Simu., vol. 83, no. 9, pp. 643-661, Sept. 2007

[94] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link-interference in static multi-hop wireless networks," in Proc. ACM IMC, pp. 28, Oct. 2005.

[95] K. Farkas, A. Huszak, and G. Godor, "Optimization of Wi-Fi access point placement for indoor localization," Network. Commun., vol. 1, no. 1, pp. 28-33, July 2013.

[96] H. U. Lee, W. S. Jeon, and D. G. Jeong, "An effective AP placement scheme for reliable WiFi connection in industrial environment," Asso. for Comp. Mach., pp. 2137-2143, April 2020.

[97] S. B. A. Khattak, M. M. Nasralla, M. Marey, M. A. Esmail, M. Jia and M. Y. Umair, "WLAN access points channel assignment strategy for indoor localization systems in smart sustainable cities," IOP Conf. Series: Ear. and Env. Sci. (ICSDI), 2022.

[98] R.G. Garroppo, G. Nencioni, L. Tavanti, B. Gendron, M.G. Scutell, "Optimal access point power management for green IEEE 802.11 networks," Sensors 2021.

[99] S. Verma, T. K. Rodrigues, Member, Y. Kawamoto, M. M. Fouda, and N. Kato, Fellow, "A survey on multi-AP coordination approaches over emerging WLANs: future directions and open challenges," arXiv:2306.04164 [cs.NI], Dec. 2023.

[100] L. M. Ali and O. Khanji, "Location-allocation optimization of Wi-Fi access points at Gaziantep University campus," in Proce. Int. Conf. Indus. Eng. Ope. Manag. Manila, pp. 2125-2136, March 2023.

[101] O. S. Naif, I. J. Mohammed, "Wireless optimization algorithm for multi-floor AP deployment using binary particle swarm optimization (BPSO)," in Proc. Int. Conf. Phys. Appl. Scie. (ICPAS), 2021.

[102] P. Liu, X. Meng, J. Wu, M. Yao, Z. Tang, "AP deployment optimization for WLAN: a fruit fly optimization approach," in Proc. Int. Conf. Comm. China (ICCC), pp. 478-483, 2019.

[103] S. C. Roy, N. Funabiki, M. M. Rahman, B. Wu, M. Kuribayashi, and W. C. Kao, "A study of the active access-point configuration algorithm under channel bonding to dual IEEE 802.11n and 11ac interfaces in an elastic WLAN system for IoT applications," Signals, vol. 4, pp. 274–296, 2023.

[104] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," IEEE/ACM Trans. Networking, vol. 15, no. 3, pp. 560-573, June 2007.

[105] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," Proc. Int. Conf. Mobile Comput. Netw., pp. 315-329, 2004.

[106] B. P. Tewari and S. C. Ghosh, "Combined power control and partially overlapping channel assignment for interference mitigation in dense WLAN," In Proceedings of AINA, pages 646-653, 2017.

[107] A. Kachroo, J. Park, H. Kim, "Channel assignment with transmission power optimization method for high throughput in multi-access point WLAN," in: Proc. Wire. Commu. Mob. Comp. Conf. 2015, pp. 314–319.

[108] H. Shindo, T. Nagai, and H. Shigeno, "Virtual access point allocation method for high density WLANs," in Proc. Int. Conf. Mob. Comput. Ubi. Network., pp. 69-76, 2012.

[109] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. P. Mhatre, "Measurement-driven guidelines for 802.11 WLAN design," IEEE/ACM Trans. Networking, vol. 18, no. 3, pp. 722-735, June 2010.

[110] C. Prommak, and B. Deeka, "Network design for quality of services in wireless local area networks: a cross-layer approach for optimal access point placement and frequency channel assignment," Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2007, Vol II, WCE 2007, 2-4 July, 2007, London, U.K., pp 1471-1475.

[111] K. Uemura, N. Funabiki, and T. Nakanishi, "A proposal of a smart access point allocation algorithm for scalable wireless mesh networks," Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, Vol II, IMECS 2010, 17-19 March, 2010, Hong Kong, pp 848-853.

[112] X. Yang, B. Chen, and F. Gao, "How to select measurement points in access point localization," Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2015, Vol II, IMECS 2015, 18-20 March, 2015, Hong Kong, pp 571-576.

[113] A. Arivoli, and P. Narayanasamy, "Cross-layer design of power assignment and power balance in wireless Ad-hoc networks," Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2015, Vol II, IMECS 2015, 18-20 March, 2015, Hong Kong, pp 583-588.

[114] A. A. Adewale, .S N. John, and E. R. Adagunodo, “Performance comparison of dynamic guard channel assignment with buffered prioritized scheme for mobile WiMAX network,” Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2016, Vol I, WCE 2016, 29 June-1 July, 2016, London, U.K., pp 566-570.

[115] Y. Kobayashi, J. Takahashi, and Y. Tobe, “A mobile Wi-Fi access point considering types and volumes of traffic,” Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2017, Vol I, WCE 2017, 5-7 July, 2017, London, U.K., pp 519-523.

[116] M. Alakhras, M. Oussalah, and M. Hussein, “Resource allocation model based on interval for wireless network with guaranteed QoS,” Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2018, Vol I, WCE 2018, 4-6 July, 2018, London, U.K., pp 399-402.

[117] P. Mestre, A. Fonseca, P. Salgado, and C. Serodio, “WiFi access point selector based on the network delay,” Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2018, Vol I, WCE 2018, 4-6 July, 2018, London, U.K., pp 409-414.

[118] C. Deng, and N. Li, “Optimization of the layout of WLAN access point based on location reliability,” Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2017, Vol II, IMECS 2017, 15-17 March, 2017, Hong Kong, pp 604-607.