A Study of Access Point Configuration Algorithm for Elastic WLAN System and Its Implementation Using Raspberry Pi

September 2017

Md. Selim Al Mamun

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, September 2017.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Mr. Md. Selim Al Mamun

Signature of

the Supervisor

Professor Nobuo Funabiki

Seal of

Graduate School of

Natural Science and Technology

Abstract

Wireless Local-Area Network (WLAN) has become popular today. The WLAN provides the flexible Internet access to users with low costs through installations of *access points (APs)* in network fields. Many organizations including universities and companies are using WLANs. To satisfy the increasing number of users, a WLAN can be expanded easily by deploying additional APs in the network field. However, the random deployment of APs can degrade the communication performance due to radio interferences among them. On the other hand, when a small number of users use the WLAN, the number of APs should be reduced to minimize the operational costs. Thus, it is expected to realize the WLAN system that is adaptive and elastic to the traffic load in the network.

In this thesis, we firstly present the concept of the *elastic WLAN system* as a solution to the above-mentioned problems in WLANs. The elastic WLAN system controls the number of active APs, the operating channels of these active APs, and the host associations to the active APs in the network, so that it can save energy, reduce interferences, and improve the performance of networks.

Secondly, we present the *active AP configuration algorithm* for the elastic WLAN system using heterogeneous access points (APs). These APs can include dedicated commercial APs, mobile routers, and software APs using personal computers (PCs). For this purpose, we formulate the *AP configuration problem* as a combinatorial optimization problem and prove the *NP-completeness* of its decision problem. Then, we propose the heuristic algorithm composed of seven phases to dynamically control the network topology by activating or deactivating APs to be matched with the traffic load in the network. We verify the effectiveness of our algorithm through extensive simulations using the WIMNET simulator.

Unfortunately, this algorithm assumes that every AP uses a different channel from the other ones to avoid interferences among them, although the number of non-interfered channels in IEEE 802.11 protocols is limited. Therefore, thirdly, we propose the extension of the AP configuration algorithm to consider the channel assignment to the active APs under this limitation. For the proper channel assignment, AP associations of some hosts are modified here to improve the network performance by averaging loads among channels. The effectiveness of this extension is also evaluated using the WIMNET simulator.

Finally, we present the implementation of the *elastic WLAN system* using the proposed algorithm. The number of active APs and the associations of hosts are dynamically optimized based on the traffic load in the network. For implementation, Linux PCs are adopted for the hosts and *Raspberry Pi* are for the APs. Using several Linux commands, this system collects the necessary information for the *AP configuration algorithm* to determine the activations/deactivations of APs, the host associations, and the channels for active APs. Based on the algorithm outputs, it controls the active APs, their channels, and the associations of hosts automatically. The feasibility and performance of this elastic WLAN system implementation is verified through experiments using testbeds.

In future, we will study further enhancements of the AP configuration algorithm and the elastic WLAN system implementation, and their evaluations in various practical scenarios.

To My Parents, Brother and Sister

Acknowledgements

This is my great pleasure to thank those who made this dissertation possible. I want to say so much, but I can hardly find the words. Therefore, I will just say that you are the greatest blessings in my life.

I would like to express my deepest and sincere gratitude to my supervisor, Professor Nobuo Funabiki, who has supported me throughout my thesis with his wide knowledge and superhuman patience. His encouragement and personal guidance have provided a good basis for this thesis. He gave me wonderful advices, thoughtful comments, and guidance when writing papers and presenting them. During my Ph. D. study, the most important thing he taught me is the principle of being scientific in life. Thanks for making me what I am today.

I would like to express my sincere gratitude to my co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for their continuous supports, guidance, and proofreading of this work.

I would like to express my sincere gratitude to Prof. Minoru Kuribayashi in Okayama University and Dr. Shigeto Tajima in Osaka University for their valuable suggestions during my research. I also want to express my gratitude to the course teachers during my Ph. D. study for enlightening me with wonderful knowledge.

I would like to acknowledge Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) for financially supporting my doctoral course study in Okayama University.

I would like to thank for the helpful discussions from many people including Dr. Md. Ezharul Islam, Prof. I-Wei Lai, Mr. Nobuya Ishihara, Dr. Chew Chang Choon, Ms. Khin Khin Zaw, Dr. Kyaw Soe Lwin, Mr. Sumon Kumar Debnath, Mr. Md. Manowarul Islam, Mr. Ismael Munene, and Mr. Rahardhita Widyatra Sudibyo. I would like to convey my thanks to all the members of FUNABIKI Lab for their supports during the period of this study.

Finally, I am forever indebted to my parents for all the years of love and support, endless patience and encouragement when it was most required. Your support and understanding gave me the strength to continue fighting. Without you I would give up.

Thank you a billion times!

Md. Selim Al Mamun
Okayama University
September 2017

List of Publications

Journal Paper

1. **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, Internet: `http://www.ijnc.org/index.php/ijnc/article/view/134`.

2. **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, Internet: `http://www.ijnc.org/index.php/ijnc/article/view/153`.

3. 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, Internet: `http://www.ijnc.org/index.php/ijnc/article/view/150`.

4. I. W. Lai, N. Funabiki, S. Tajima, **M. S. A. Mamun**, and S. Fujita, "Cross-layer selective routing for cost and delay minimization in IEEE 802.11 ac wireless mesh network," Wireless Networks, pp.1-13, Feb. 2017.

International Conference Paper

5. **M. S. A. Mamun**, S. K. Debnath, K.S. Lwin, and N. Funabiki, "A channel assignment extension of active access-point configuration algorithm for elastic WLAN system under limited channels," Int. Symp. Computing and Networking, pp. 318-324, Nov. 2016.

6. **M. S. A. Mamun**, M. E. Islam, N. Funabiki, and I.-W. Lai "Active access-point configuration algorithm with dynamic mobile router placement for elastic WLAN system," Int. Symp. Computing and Networking, pp. 246-252, Dec. 2015.

7. K. S. Lwin, N. Funabiki, K. K. Zaw, **M. S. A. Mamun**, and M. Kuribayashi, "A minimax approach for access-point setup optimization using throughput measurements in IEEE802.11n wireless networks," Int. Symp. Computing and Networking, pp. 311-317, Nov. 2016.

Other Papers

8. **M. S. A. Mamun**, M. E. Islam, and N. Funabiki, "An active access-point configuration algorithm for wireless local area networks with heterogeneous devices," IEICE Tech. Rep., vol. IEICE-115, no. 159, pp. 147-152, July 2015.

9. **M. S. A. Mamun**, M. E. Islam, S. K. Debnath, and N. Funabiki, "Evaluations of active access-point configuration algorithm under limited mobile router allocations," IEICE Society Conference, Mar. 2016.

10. **M. S. A. Mamun**, M. E. Islam, N. Funabiki, and I. -W. Lai, "Evaluations of active access-point configuration algorithm with dynamic mobile router placement under various mobile router speeds and wall effects," The 17th IEEE Hiroshima Section Student Symposium (HISS 2015), pp. 415-418, Nov. 2015.

11. K. S. Lwin, N. Funabiki, K. K. Zaw, **M. S. A. Mamun**, and M. Kuribayashi, "Throughput measurements for access-point setup optimization in IEEE802.11n wireless networks," IEICE Tech. Rep., vol. IEICE-116, pp.45-50, July 2016.

12. S. K. Debnath, N. Funabiki, K. S. Lwin, **M. S. A. Mamun**, R. W. Sudiby, and S. Huda, "Raspberry Pi configuration for access-point and its throughput measurements in IEEE802.11n wireless networks," IEICE Tech. Rep., vol. IEICE-116, no.382, pp.101-106 , Dec. 2016.

13. M. E. Islam, K. S. Lwin, **M. S. A. Mamun**, N. Funabiki, and I. Lai, "Measurement results of three indices for IEEE802.11n wireless networks in outdoor environments," The 17th IEEE Hiroshima Section Student Symposium (HISS 2015), pp. 410-414, Nov. 2015.

14. M. E. Islam, N. Funabiki, K. S. Lwin, **M. S. A. Mamun**, S. K. Debnath, and M. Kuribayashi, "Performance evaluations of elastic WLAN system testbed," IEICE Society Conference, Mar. 2016.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

Nowadays, the Internet plays a very important role today in daily lives. With rapid developments of high-speed communication technologies and small-sized inexpensive communication devices, it has become easy to access to a variety of information, data, and services through the Internet. It is very common for people to use social networking sites in order to regularly contact with their friends and relatives over the Internet. This increasing usage of the Internet leads to strong demands of a secure, reliable, and high-speed Internet services for us. A common solution to meet these demands is the use of a *wireless local area network (WLAN)* with low cost WLAN enabled devices [1].

WLANs have been widely deployed for the Internet access, since they use wireless medium over the air. This feature makes WLANs inexpensive, flexible, and scalable [2]. The wireless aspect of the WLAN leads the users to move freely in the network field without losing connections. It has provided a solution to one of the key challenges characterizing wired networks that restrict the user mobility due to the wired medium.

With the rapid advancements of wireless communication technology, WLAN users can set up their WLAN systems using a variety of low cost access points (APs) such as commercial APs and mobile routers according to their budgets and demands. Unfortunately, these APs are often installed and used in a service field randomly, which can cause poor network performances due to overlapping of the same frequency signals [3]. Besides, the configuration of these APs should be arranged properly based on communication demands in the network field, where redundant APs should be turned off for energy saving and interference prevention.

In the real world scenarios, the number of users in a network often fluctuates depending on time and day of week. For example, in a university, the number of network users or students increases in the afternoon during weekdays, while it decreases in the morning or evening, and the whole days in the weekend. Besides, conditions of network devices or communication links are changed due to power shortages, device failures, regulations of bandwidth by authorities, and even by the weather [4].

Developing countries like Bangladesh and Myanmar suffer from the unreliable and slow Internet access. The major reason is the *load shedding*, i.e. the fluctuation and the discontinuity of the electricity supply for the time being. In these countries, only 60% and 30% of the people can use electricity respectively [5][6]. They suffer from the load shedding frequently, because the power production is comparatively less than the growing demand. The power authority stops the supply to certain areas so as to supply electricity to other areas. This problem is dominant when the extra energy, such as air conditioning in summer, is required. The irregular flow of electricity often causes damages to network devices. In such uncertain cases, a very limited number of APs can be supplied power using back up power sources by generators and batteries.

Another reason for slow and interrupt Internet connections in developing countries is the fluctuation of the allocated bandwidth to the organization. Network cables are often cut off at construction sites. Network devices and cables are stolen in many places. They hamper the continuity of the Internet access. In addition, Internet service providers restrict the bandwidth of clients for their own selfish ends. These reasons can decrease the available bandwidth, which is far from its expected value.

The Internet access through 3G/4G mobile networks is increasing in these countries at recent time, as their usage fees are continuously falling down with time. Currently, many people are using the Internet through mobile network devices including mobile routers, pocket routers, and smart-phones. Besides, it has become common for them to share the Internet use among friends, families, and groups by them these days.

The guaranteed throughput, which allows the access to a Web site or the use of an e-mail without strong stress, may not be essential for the personal use of the Internet, because the Internet access fee by a mobile network may still matter for them. However, for the commercial use in a school or a company, the consistent Internet access with the guaranteed throughput is more desirable and significant than the cost.

Under these circumstances, we have studied the *elastic WLAN system* for burning demands in developing countries. Figure 1.1 shows the topology of a simple elastic WLAN system with heterogeneous AP devices. The elastic WLAN system allows the dynamic control of active APs according to the network environment. As the more traffic demands appear, the more active APs are required. The implementation of the elastic WLAN system, which is adaptive to the network changes, is important for energy efficient, reliable, and high performance wireless communication services.

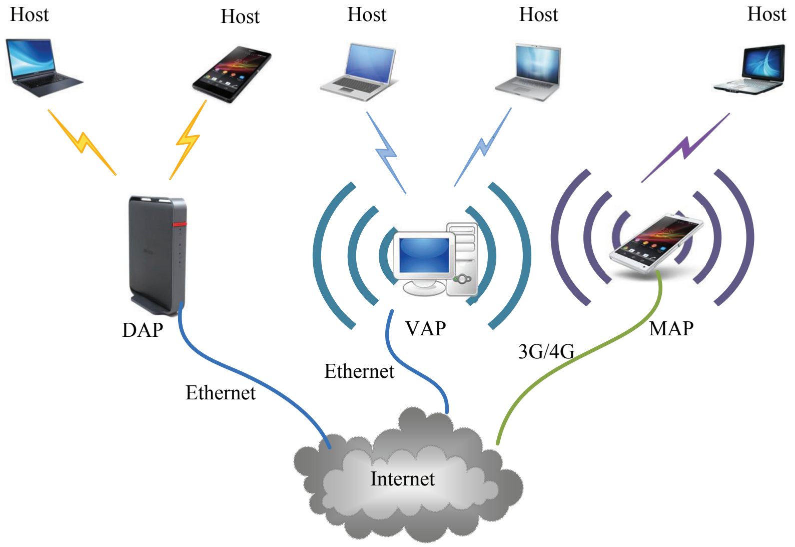


Figure 1.1: Illustration of elastic WLAN with heterogeneous devices.

1.2 Contributions

In this thesis, we have carried out research works in the following three stages.

In the first stage [7–9], we propose the *active AP configuration algorithm* to activate the minimum number of APs in the network while satisfying the constraints of throughputs for the hosts. Three types of APs, namely *dedicated APs (DAPs)*, *virtual APs (VAPs)*, and *mobile APs (MAPs)*, are considered for this algorithm. Virtual APs use personal computers (PCs) of users installing the software for AP functions. Mobile APs are often called *mobile routers*. To connect with the backbone network in the Internet, DAPs and VAPs use wired connections, whereas MAPs use cellular networks. For the positions of MAPs, the same locations as the hosts should be considered, because the host owners may use MAPs for the Internet access.

In this stage, first, we formulate the *AP configuration problem* as a combinatorial optimization problem, and prove the *NP-completeness* of its decision version through the reduction from the *minimum set cover problem* that has been known as a *NP-complete* problem [10]. Then, we propose its heuristic algorithm composed of seven phases. The effectiveness of the algorithm is verified through numerical experiments in different network instances using the WIMNET simulator.

In the proposed algorithm, we assume that each AP can use a different channel from the other APs to avoid interferences among them at all. In reality, the number of non-interfered channels available in *IEEE 802.11* protocols is limited [11]. Some APs have to share the same channel if the number of APs is larger than the number of available non-interfered channels. Therefore, the channels should be assigned properly to the active APs to avoid the interference as much as possible to improve the network performance.

Therefore, in the second stage [12, 13], we propose the *channel assignment extension* of the active AP configuration algorithm for the elastic WLAN system to solve the above mentioned problem. One channel is assigned to every active AP from the limited number of the orthogonal channels in a way of minimizing the overall interference in the network. Besides, AP associations of hosts are modified to improve the network performance by averaging loads among channels. After the channel assignment, the communication loads among the channels may become imbalanced, because the previous algorithm finds the AP-host associations assuming the infinite number of the orthogonal channels. The effectiveness of the proposed extensions is verified through simulation in two network topologies using the WIMNET simulator.

In the third and final stage [13], we implement the model for the *elastic WLAN system*. In this system, the number of active APs and the associations of hosts is optimized by the proposed algorithm. In the implementation, *Raspberry Pi* [14] is adopted for the AP and Linux PC is for the host. Raspberry Pi is a small-size low-cost computer, and has become popular in academics and industries around the world. The use of Raspberry Pi in the elastic WLAN system is important for its dissemination in developing countries. 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 [15]. By using several Linux tools and commands, this system collects necessary information of the network in real time for the algorithm input, and optimizes the topology of the network by turning APs on/off, and changing the AP-host associations according to the algorithm output. Finally, the elastic WLAN system implementation is evaluated through experiments using the testbed.

In short, the main contributions of this thesis are:

- proposal of the active AP configuration algorithm that can dynamically adjust the number of active APs in the network.

- proposal of the channel assignment extension to the active APs from the limited number of non-interfered channels in a way to minimize the overall interference in the network.

- implementation of the elastic WLAN system using Raspberry Pi and Linux PCs.

1.3 Contents of Thesis

The remaining part of this thesis is organized as follows.

Chapter 2 introduces related wireless network technologies, such as IEEE 802.11n protocol, heterogeneous AP devices, channels in IEEE 802.11 protocols and important WLAN tools in the Linux operating system.

Chapter 3 describes the elastic WLAN system briefly.

Chapter 4 proposes the active AP configuration algorithm for the elastic WLAN system.

Chapter 5 describes the channel assignment extension of the AP configuration algorithm.

Chapter 6 evaluates the AP configuration algorithm through simulations in several network instances using the WIMNET simulator.

Chapter 7 presents the implementation of the elastic WLAN system using Raspberry Pi and Linux PC.

Finally, Chapter 8 concludes this dissertation with some future works.

Chapter 2

Background Technologies

In this chapter, we briefly introduce some wireless network technologies as backgrounds for this dissertation. First, we give an overview of *IEEE 802.11 protocols*, specially *IEEE 802.11n protocol*, and our link speed estimation model for this protocol. Then, we introduce three different types of APs assumed in the AP configuration algorithm and discuss the characteristics and the speed difference of them. Then, we discuss the available channels in the IEEE 802.11n protocol. Finally, we briefly describe Linux tools and commands used in the implementation of elastic WLAN system.

2.1 Overview of 802.11 WLAN

A *wireless local area network (WLAN)* is an alternative or an extension to a wired LAN that supports flexibility in data communication. A WLAN can minimize the need for wired connections by transmitting and receiving data over the air using the radio frequency (RF) technology. Thus, a WLAN can combine the data connectivity with the user mobility. IEEE 802.11 protocols provide standards for WLANs.

2.1.1 Advantages of WLAN

A WLAN offers the following convenience, productivity, and cost advantages over traditional wired networks [1]:

- *Mobility:* Users move, but data is usually stored centrally. Thus, users can access data while they are in motion. In wired networks, users need to use wired lines to stay connected to the network.

- *Easy and faster deployment:* A WLAN system can eliminate the need to put cables between hosts and network connection hubs. Thus, the installation can be fast and easy.

- *Flexibility:* The coverage area by a WLAN can be easily expanded because the network medium can cover it without additional costs.

- *Cost:* The initial installation cost can be higher than the cost of wired LANs, but the life-cycle cost can be significantly lower. In environments requiring frequent moves and changes, a WLAN can provide the long-term cost benefit.

- *Scalability:* According to specific applications, a WLAN system can be configured in a variety of topologies. They can support a peer-to-peer network suitable for a small number of users and a full infrastructure network for thousands of users.

2.1.2 Components of 802.11 WLANs

A WLAN system with IEEE 802.11 protocols consists of four major physical components shown in Figure 2.1:

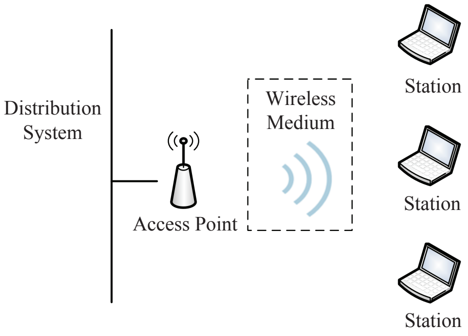


Figure 2.1: Components of WLAN system.

- *Stations:* Data are transferred between *stations* in a network. Stations are computing devices such as desktop or notebook computers, or hand-held devices with wireless network interfaces. *Stations* are often called *hosts*.

- *Access points (APs):* To deliver a data frame from an 802.11 network to other networks, it needs the conversion. An AP performs the wireless-to-wired bridging function along with some other tasks.

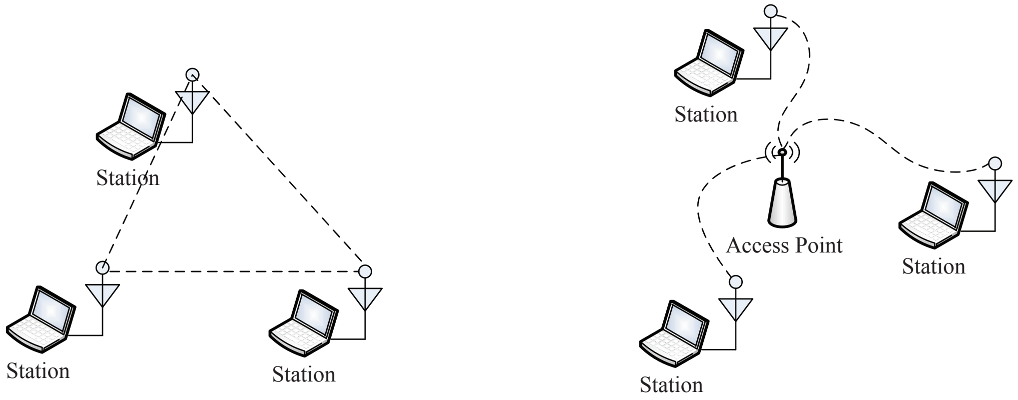
- *Wireless medium:* The 802.11 standard uses a wireless medium to transmit frames from a station to a station.

- *Distribution system:* If several APs are connected together to form a large coverage area, they must communicate with each other to track the movements of the mobile stations. The distribution system is the logical component of the 802.11 standard which is used to forward frames to their destinations. It is often called as the *backbone network*. Mostly, *Ethernet* is used as the backbone network technology.

2.1.3 Types of WLANs

The basic building block of an 802.11 network is simply a group of stations that communicate with each other that is called the *basic service set (BSS)*. BSS can be of two types as illustrated in Figure 2.2.

- *Independent or ad hoc network:* This is a local area network (LAN) that can be formed spontaneously as devices connect each other. Instead of relying on a base station or an AP to coordinate the flow of messages, the stations forward packets to and from each other directly.



(a) Independent network

(b) Infrastructure network

Figure 2.2: Types of 802.11 networks.

There is no AP in an independent BSS as shown in Figure 2.2(a). The ad hoc network is rarely used due to reasons related to performance and security.

- *Infrastructure network:* Infrastructure networks or *infrastructure BSS* use APs as shown in Figure 2.2(b). APs are used in all communications in the infrastructure network. If one station wants to communicate with another station, it always needs to communicate through the AP. Thus, in an infrastructure network, any station must be associated with an AP to obtain network services.

- *Extended service areas:* Multiple BSS can be chained together with a backbone network to form the *extended service set* (ESS) as shown in Figure 2.3. It can form large size wireless networks. All the APs in an ESS are given the same *service set identifier* (SSID), which serves as a network "name" for the users. Stations within the same ESS can communicate with each other, even if they are in different basic service areas.

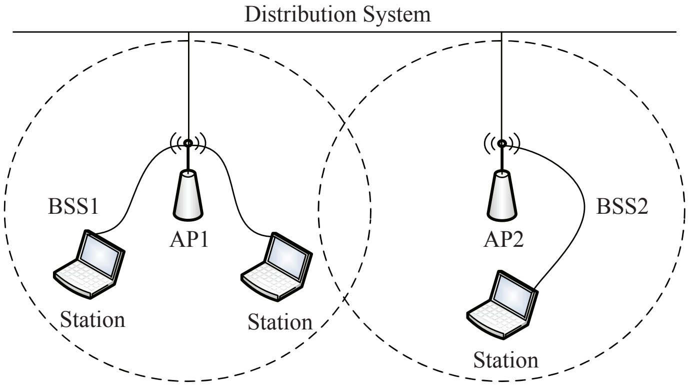


Figure 2.3: Extended service set.

2.2 IEEE 802.11n Protocol

In this section, we briefly give the overview the IEEE 802.11n protocol that we have used for throughput measurements and implementations of the elastic WLAN system. IEEE 802.11 protocols is a group of the standards created by the IEEE 802 LAN/MAN (Local Area Network/Metropolitan Area Network) Standards Committee. They specify over-the-air interfaces between a wireless client and a base station or between two wireless clients within a local area in either fixed, portable, or moving stations mode [16].

IEEE 802.11n is an amendment to the IEEE 802.11-2007 wireless networking standard. This standard was introduced with the 40 MHz bandwidth channel called *channel bonding*, the Multiple-Input-Multiple-Output (MIMO), the frame aggregation, and security enhancements over its predecessors. Table 2.1 briefly summarizes the IEEE 802.11n protocol.

Table 2.1: IEEE 802.11n specification.

specification	**IEEE 802.11n**	
frequency band	2.4 GHz	5 GHz
number of available channels	13	19
number of uninterfered channels	2	9
maximum speed	600$Mbps$	
maximum bandwidth	40 MHz	
number of maximum streams	4	
maximum modulation	64 QAM	

In the channel bonding technique, two adjacent channels within a given frequency band are combined together to increase the transmission speed. In the IEEE 802.11n protocol, the physical data rate can be doubled by using two adjacent 20 MHz channels simultaneously.

The spatial multiplexing and the space-time block coding are two MIMO-specific innovations of the IEEE 802.11n. For the spatial multiplexing, the transmitter transmits multiple independent data streams simultaneously from multiple antennas to increase the data rate. For the space-time block coding, the transmitter transmits dependent data stream which is spatially and time encoded to increase the signal reliability. Table 2.2 shows the throughput enhancement of IEEE 802.11n protocol using channel bonding and MIMO.

Table 2.2: Effects of channel bonding and MIMO on IEEE 802.11n 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$Mbps$

IEEE 802.11n also provides the performance improvement through the frame aggregation in the MAC layer. The frame aggregation can transmit multiple frames as one big frame with a single pre-ample and header information to reduce the overhead by them. IEEE 802.11n introduces Aggregation of MAC Service Data Units (A-MSDUs) and Aggregation of MAC Protocol Data Units

(A-MPDUs). The 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 [17].

2.2.1 Link Speed Change Feature of IEEE802.11n

The link speed or throughput is affected by many factors such as the modulation and the coding scheme, the transmission power, the transmission distance, and even the design of the network adapters [18]. Therefore, the theoretical computation of the link speed is very difficult. In this thesis, we take an alternative approach of conducting real-world measurements to model the actual link speed.

- **Link Speed Model**

  Through measurements [19], we derived the following link speed ($Mbps$) function $f(x)$ with the transmission distance x (m).

Table 2.3: Devices and software for measurements of IEEE 802.11n link speed.

PC1	model	Ultra book Lesance NB S3441/L
	CPU	Intel(R) Core i5 3317U (2.6 GHz)
	OS	Windows 7
PC2	model	Ultra book Lesance NB S3532-SP
	CPU	Intel(R) Core i3 2350M (2.3 GHz)
	OS	Windows 7
NIC		Buffalo WZR-G1750DHP
software		iperf2.0.5
protocol		TCP

$$f(x) = \begin{cases} -2.20 \times 10^{-3}x^3 + 1.85 \times 10^{-1}\,\mathrm{x}^2 - 5.33x + 117 & \text{if } 0 \leq x < 40 \\ -6.00 \times 10^{-4}x^3 + 9.50 \times 10^{-3}\,\mathrm{x}^2 - 1.73x + 117 & \text{if } 40 \leq x < 75 \\ \;\;4.38 \times 10^{-4}x^3 - 1.10 \times 10^{-1}\,\mathrm{x}^2 + 8.48x - 189 & \text{if } 75 \leq x < 100 \\ \;\;1.0 \;\; \text{if } x \geq 100 \end{cases} \tag{2.1}$$

The distance between the ith host and the jth AP is defined by the Euclidean distance:

$$d(i, j) = \sqrt{(x_i^h - x_j^{AP})^2 + (y_i^h - y_j^{AP})^2} \tag{2.2}$$

where x_i^h, y_i^h, x_j^{AP}, and y_j^{AP} are the x and y coordinate of the ith host and the jth AP.

For our measurements, two PCs with the setting listed in Table 2.3 are prepared respectively as the source and destination nodes. The link speed is measured when both end nodes adopt the IEEE 802.11n protocol. At the measurements, the link distance between two end nodes is increased from $1m$ to $110m$ with the $5m$ interval. The parameters in *iperf* are set at 50 seconds for the measurement time, and $8Kbytes$ and $477Kbytes$ for the buffer size and the window size respectively. Figure 2.4 shows the link speed measurement and estimation results.

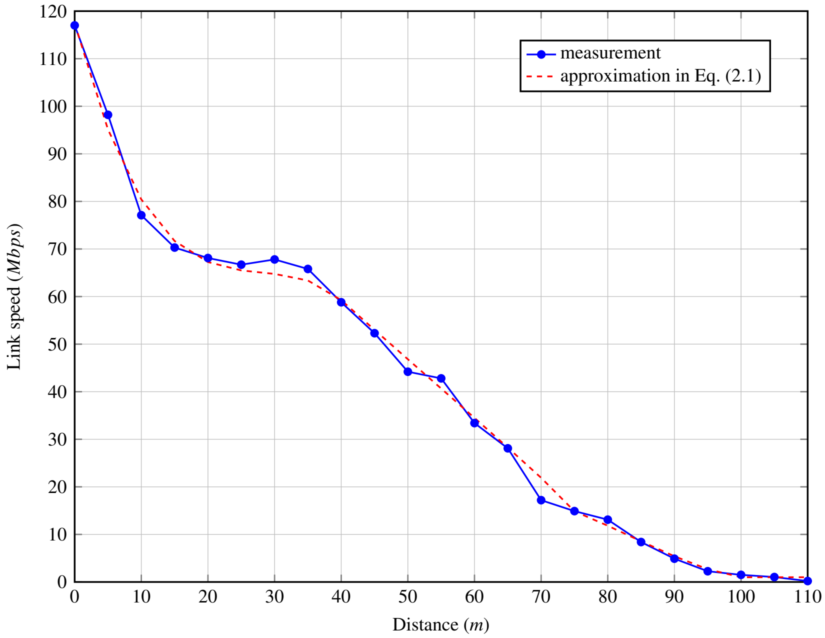


Figure 2.4: IEEE 802.11n link speed measurement results.

According to our measurements, the peak throughput becomes about $115Mbps$ at $1m$ distance. Then, the effective throughput is rapidly dropped as the link distance increases. The throughput becomes the half of the peak at $40m$ distance, and is about $10Mbps$ at $80m$ distance.

- **Wall Effect**

  Since a WLAN system is based on radio frequency (RF) signals, factors affecting RF signal strengths should be taken into considerations for planning the efficient WLAN system. As the RF signal at $2.4GHz$ or $5GHz$ becomes weak after penetrating obstacles such as concrete walls in a building, the throughput becomes low, which is called *wall effect* in this paper. The type of the materials used in the obstacle determines the drop rate of the link speed. Besides, mobile operators (carriers) offer different mobile *data plans* to meet various subscribers demands and satisfactions. Depending on a plan, several choices can be taken for the link speed to connect to the Internet, which is called the *MAP speed* in this paper. To make the WLAN adaptable to such situations, these factors must be considered.

  To examine the *wall effect*, we measured throughputs between a host and an AP that are located in different rooms separated by one concrete wall. The distance is fixed at $5m$. Then, it is confirmed that the link speed is dropped by about 15% when the signal passes through one concrete wall in a room. In simulations, the link speed is decreased by multiplying $85/100$ if there exists a wall between an AP and a host. Figure 2.5 shows the speed drop by the wall effect.

- **Device Effect**

  Furthermore, it is found that a PC provides a maximum of $54Mbps$ when it works as a VAP while a MAP supports a maximum of $30Mbps$ speed. Thus, in this thesis, the link speed estimated from the link speed function $f(x)$ is adjusted by multiplying $45/100$ for the VAP

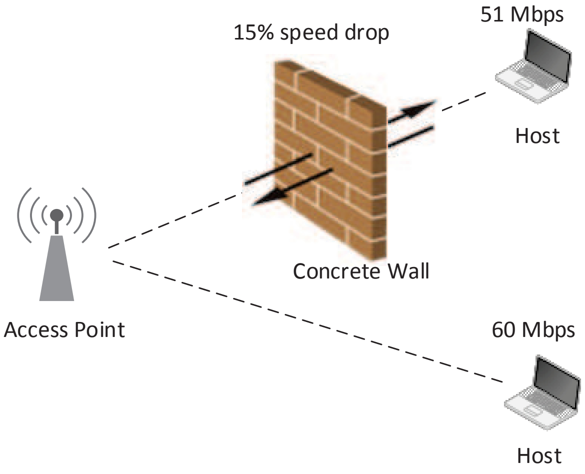


Figure 2.5: Wall effect.

(a) Dedicated AP (DAP) (b) Virtual AP (VAP) (c) Mobile AP (MAP)

Figure 2.6: Three types of APs.

and 25/100 for the MAP. The same wall effect is applied for any AP type. These parameters are used in our algorithm and the *WIMNET simulator* for simulations.

2.3 Heterogeneous Access Points

In this thesis, three types of APs are considered for the AP configuration algorithm, namely *dedicated APs (DAPs)*, *Virtual APs (VAPs)*, and *mobile APs (MAPs)*, shown in Figure 2.6.

- A *DAP* is a commercial AP that adopts the IEEE 802.11n wireless protocol and connects PCs to the Internet. A DAP usually has the coverage radius of around $110m$ and the transmission speed around $120Mbps$. However, the transmission speed varies significantly depending on the environment including obstacles, channel interferences, number of antennas, and placement heights of APs.

- A *VAP* is a software-based router using a personal computer (PC) with either Windows or Linux for the operating system. Multiple Internet connection mediums including wired, wireless, or cellular interfaces are available for the VAP. A network device can connect to a VAP the same way as it does to a conventional DAP. Most of VAPs support the IEEE 802.11n protocol with a maximum of $54Mbps$ transmissions.

- A *MAP* is a device that connects to the Internet through 3G/4G wireless technology, e.g., smart phone, while a DAP and a VAP uses the wired Ethernet to access the Internet. For this portable MAP, the power supply is unnecessary because of the built-in battery. With the rapid developments of the cellular technology, most MAPs support the IEEE 802.11n protocol, in which the transmission speed is capped to around $30 Mbps$ due to the bottleneck in the cellular network[1].

2.4 Channels in IEEE 802.11 Protocols

A WLAN using the IEEE 802.11n [16] can operate in two unlicensed frequency spectrum bands: the 2.4 GHz *Industrial, Scientific, and Medical (ISM)* band and the 5 GHz *Unlicensed National Information Infrastructure (U-NII)* band. The IEEE 802.11n standard defines the limited number of channels for use in APs and clients. One channel has 22 MHz width and is separated by 5 MHz from the adjacent channel, which is conventionally referred as the 20 MHz channel. Figure 2.7 illustrates the available channels in IEEE 802.11n for the 2.4 GHz spectrum.

In this 2.4 GHz band, 13 channels exist where only four channels can be non-interfered. If the channel bonding technique [18] is adopted, which is common in IEEE 802.11n, the 40 MHz channel is derived by bonding two adjacent 20 MHz channels. It achieves more than twice data throughputs. At the same time, it reduces the number of non-interfered channels to only two. Since the 2.4 GHz band becomes crowded, clients start using the 5 GHz band. This can carry up to nine non-interfered 40 MHz channels, but it has a shorter transmission range than the 2.4 GHz.

2.5 Linux Tools for Wireless Networking

In this section, we introduce the Linux tools and software used for the implementation of the *elastic WLAN system*.

- **'arp-scan' - to Explore Devices in Network**

  arp-scan [20] is used to scan the network devices. This tool uses the ARP protocol to discover all the devices in the network. It can be installed by downloading the source code from [21] or using the following command:

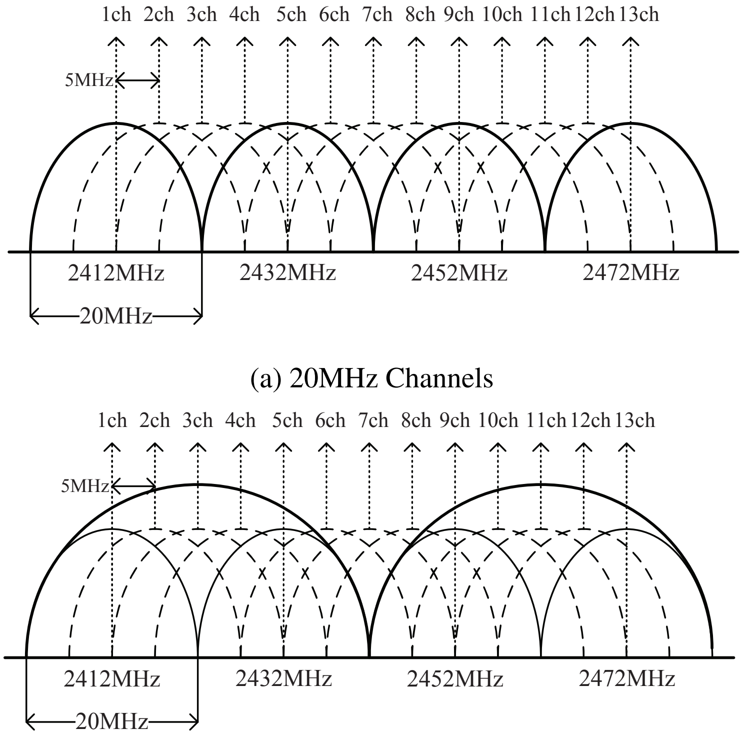
  ```
  $ sudo apt-get install arp-scan
  ```

  The corresponding Linux command to scan the network using *arp-scan* is given by:

  ```
  $ arp-scan --interface=eth0 --localnet
  ```

  --interface=eth0 represents the interface to be used for scanning devices. The use of *--localnet* makes *arp-scan* scan all the possible IP addresses in the network that are connected through this interface. The interface is defined by the interface IP address and netmask. The name of the network interface depends on the operating system, the network type (Ethernet, wireless etc.), and the interface card type. Here, the interface name *eth0* is used as an example.

[1] We should note that the DAPs and the VAPs share the gateway to the Internet. The bandwidth of this gateway becomes the total available bandwidth for the whole WLAN.



(a) 20MHz Channels

(b) 40MHz Channels

Figure 2.7: Available channels in IEEE802.11n for 2.4GHz band.

- **'nm-tool' - to Obtain Signal Strength from APs**

  The Linux tool *nm-tool* [22][23] can find the receiving signal strength (RSS) at a device from all the reachable APs. In our implementation, we use this tool to get the necessary information on the host, such as the currently associated AP, the associable APs, and the receiving *signal strength* from each associable AP. *nm-tool* has been installed as a part of *NetworkManager* package [24] by default on the Ubuntu distribution. It can be installed using the following command manually:

  ```
  $ sudo apt-get install network-manager
  ```

  The simple way to run *nm-tool* to find the receiving signal strength is:

  ```
  $ nm-tool
  ```

- **'hostapd' - to Make AP-mode in Raspberry Pi**

  Raspberry Pi [14] can be set up as the AP by using *Host access point daemon (hostapd)* [25][26]. *hostapd* is capable of turning network interface cards (NICs) into access points (APs). It can be installed by downloading the source code from [27] or using the following command:

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

  The *hostapd* can be started or stopped by the following commands:

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

- **'ssh' - to Remotely Execute Command**

  ssh [28][29] is an abbreviation of *Secure Shell* that is a cryptographic network protocol to initiate a secure shell session on a remote machine. It is operated in two parts: *ssh-client* and *ssh-server*, and establishes a secure channel between them. The open source version of *ssh* is *OpenSSH* [30] that can be installed using the following command:

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

  An example to remotely execute *nm-tool* on a remote host using *ssh* is shown below:

  ```
  $ 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.

- **'nmcli' - to Change Associated AP**

  nmcli [31][32] is a command-line Linux tool to manage, configure, and control the *NetworkManager* package. This tool is pre-included in the *NetworkManager* package. It can be used to associate a host with an AP through the following command:

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

  Here *NewSSID* represents the newly associated AP for the host and *PASSWORD* does the security key of the AP.

- **'iwconfig' - to Collect Information of Active Network Interface**

  iwconfig [33] is a command-line Linux tool to display and change the parameters of the active network interface for wireless operations. It can also be used to display the wireless network parameters and statistics. This tool is usually installed by default in the Ubuntu distribution, which can also be installed manually using the following command:

  ```
  $ sudo apt-get install wireless-tools
  ```

  An example to display the information of the currently associated AP using *iwconfig* is shown below:

  ```
  $ iwconfig wlan0
  ```

  This command shows the RSS of the currently associated AP in dBm. The RSS can be converted to the estimated link speed using the sigmoid function in [34]

- **'iperf' - to Measure Link Speed**

  iperf [35] is a tool to measure the throughput between two nodes or the link speed. Both TCP and UDP protocols are supported by *iperf*. This tool 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 throughput between two devices, one should be in the server-mode while the other in the client-mode. The 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 sides for the throughput measurement:

```
$ iperf -s // server side
$ iperf -c 172.24.1.1 // client side
```

In the command, 172.24.1.1 represents the IP address of the server.

2.6 Summary

In this chapter, we presented wireless network technologies, heterogeneous WLAN devices and Linux tools that we adopted in this thesis. Since the theoretical computation of the link speed is difficult, we conducted the throughput measurement approach to estimate the link speed for IEEE 802.11n protocol. Besides, we considered the reduction of the link speed due obstacle such as walls. In the next chapter, we will introduce the elastic WLAN system.

Chapter 3

Overview of Elastic WLAN System

In this chapter, we introduce the overview of the *elastic WLAN system*. First, we discuss the motivation for the study of the elastic WLAN system. Then, we clarify the challenges of the elastic WLAN system design. Finally, we present the system design and the operational flow for this system.

3.1 Motivation

WLANs are commonly deployed to provide the Internet access in various places such as company offices, airports, shopping malls, educational institutes, hotels, and train stations. Multiple APs are deployed in each service field to provide the uninterrupted Internet connection service. These APs are often installed randomly, which may cause the poor network performance due to the interferences among them caused by using the same frequency signals. Besides, redundant APs can increase operational costs for power consumptions and their managements. Therefore, the APs should be configured properly to avoid interferences and reduce costs as much as possible.

In the network field, the distribution of users is usually non-uniform. A high concentration of users in a limited area can increase the loads on the APs allocated there. Then, the poor network performances are provided to the users. The APs in the congested area can be overloaded, whereas the APs in the less-crowded area can be under-utilized. To solve such circumstances, to change the associations of user hosts from highly congested APs to less congested APs can be useful to improve network performances.

In real worlds, APs in a network may get overloaded during office hours where the users may suffer from low network performances. In this case, it is necessary to activate additional APs so that they can ensure the expected performance. On the other hand, these APs are less loaded at the night time or weekends. In this case, such APs should be deactivated to save the energy and reduce management costs.

At the same time, due to the power shortages, the device failures, and the network backbone inefficiency, an organization receives the lower Internet bandwidth than its expected value. In such cases, the link speed between the host and the AP needs to be adjusted for a fair Internet access to all the users. Then, cellular network based devices like MAPs should be activated in the WLAN to ensure the network performance.

Therefore, WLANs is expected to be adaptive according to the traffic demands and the network conditions. This can be obtained by dynamically controlling the number of active APs in the network and by changing the associations of hosts to these active APs based on the network traffics.

To realize this goal, we have studied the elastic WLAN system. The elastic WLAN systems can play an important role for energy-saving and performance improvements for WLANs.

The motivation of the elastic WLAN system study is summarized below:

1. Reduction of operational cost and energy usage:
    - An organization generally allocates the number of APs required to ensure high performances during peak-hours, and keep the WLANs active for 24-hours a day. However, only a few of the APs are used during off-peak hours or in holidays. The elastic WLAN system activates the minimum number of APs required for the current traffic demand, and saves a huge amount of energy.
    - Developing countries like Bangladesh and Myanmar suffer unreliable Internet access due to discontinuities of electricity supplies for the time being. In such situations, the elastic WLAN system can improve the network performance through the efficient use of the available power sources.
2. Improvement of WLAN performances:
    - When APs become overloaded by increasing users and suffer from low network performances, the activations of additional APs can maintain the desirable network performances to cope with the increasing traffic demands.
    - Due to the backbone network inefficiency, power shortages, or ISP failures, an organization may receive a lower bandwidth than its expected one. In such cases, cellular network based *mobile APs* should be activated in the elastic WLAN system to ensure the desirable network performance.
    - When the number of active APs become high in a network area, users may suffer from the interferences among them due to overlapping of the frequency signals. In such situations, the elastic WLAN system can dynamically change assigned channels to APs in a way to minimize the interferences among them and improve the performance.

The dynamic configuration of the active APs and the host associations in the elastic WLAN system is provided by the *AP configuration algorithm* that is the main contribution in this thesis.

3.2 Challenges of Elastic WLAN System Design

The elastic WLAN system can be a solution of the abovementioned issues of WLANs. However, there are a substantial number of major challenges in developing the elastic WLAN system. In this section, we show the challenges to implement the elastic WLAN system in real network environments and the possible solutions to mitigate them.

1. The first major challenge is to collect the necessary network information and generate the inputs for the AP configuration algorithm in real network environments. This includes the following key points to consider:
    - How to explore the WLAN enabled devices in the network?
    - How to collect the necessary information from each host in the network? The information includes the currently associated AP, the possible associable APs and the receiving signal strength (RSS) of each host from all the reachable APs in the network.

   - How to estimate the link speed and interferences from the collected network information to use as the inputs to the AP configuration algorithm and the channel assignment algorithm?

2. The choice of operating system platform and the software is another key challenge of the elastic WLAN system. The system requires fully customizable tools and integrations of several software components developed independently.

3. The choice of the WLAN enabled devices in the elastic WLAN system is another challenge. With the rapid developments of wireless technologies, there are varieties of WLAN enabled devices in the market. The system should support the integration of heterogeneous WLAN devices.

4. The final challenge is to apply the changes in the network configuration according to the algorithm outputs. The changes should be carried out remotely and dynamically. This includes the following points to consider:

   - How to automatically activate/deactivate APs according to the network traffic if necessary?
   - How to automatically change the channel of APs to minimize the interference if necessary?
   - How to automatically change the associations of hosts to the active APs in the network if necessary?

Our solutions to address the abovementioned challenges are listed below:

- The Linux environment is adopted to implement the elastic WLAN system. The Linux environment has a lot of tools and software to manage and control the WLAN system. All of them are open source and can be easily customized. On the other hand, the network management tools in Windows operating system are less flexible, not customizable, and not open source to use.

- In our implementation, a centralized network system is assumed, where one PC serves as the control hub to all the network devices. This server has the administrative access to every device in the network for the remote command executions. This server executes the AP configuration algorithm and controls the network remotely.

- There are many tools available to explore the network condition, such as *arp-scan* [20], *nmap* [36][37], and *Wireless Network Watcher*[38]. We adopted the Linux tool *arp-scan* to discover the network devices in this work. Other tools have large response time or not flexible for customization for our study.

- To collect the network information such the receiving signal strength (RSS) of each host from the currently associated AP and the possible associable APs, the Linux tool *nm-tool* [22][23] is adopted in this study. Other tools are also examined such as *iwconfig*[33] and *wavemon*[39]. Some of them provide RSS of only the associated AP. On the other hand, *nm-tool* provides more precise and organized data to be used in the system.

- To control the network configuration according to the outputs of AP configuration algorithm, the APs must be controlled remotely. For this purpose, *Raspberry Pi* is adopted as a configurable and inexpensive AP. The AP mode of Raspberry Pi can be activated/deactivated through command lines. Raspberry Pi is small in size, cost-efficient, and consumes less power comparing to laptop PCs.

- The system sometimes needs to change the associations of hosts through command lines to control the network configuration. The Linux tools *nmcli* [31][32] and *iw* [40] are used to change the association of a host through the command line. *nmcli* is a part of the network manager tool by default.

3.3 System Design and Operational Flow

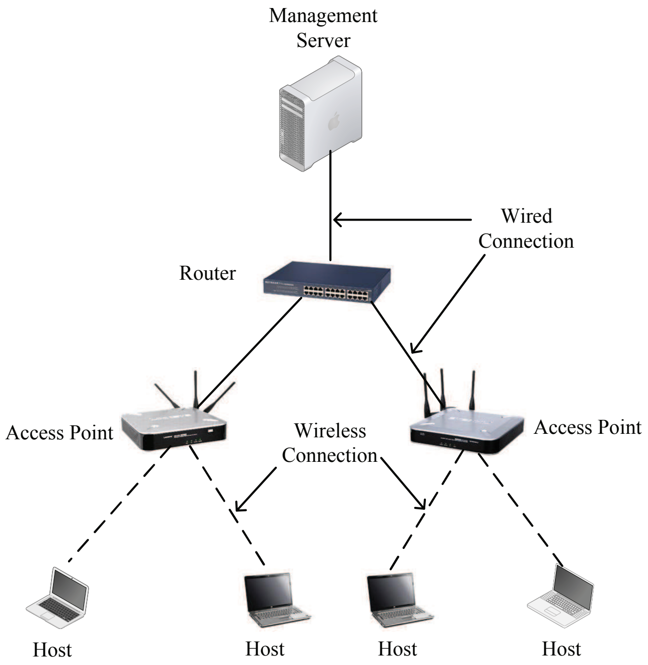


Figure 3.1: Example elastic WLAN system topology.

Figure 3.1 offers an example topology of the elastic WLAN system. The elastic WLAN system dynamically controls the number of active APs in the network by activating or deactivating APs according to the network condition. Figure 3.2 shows the operational flow of the elastic WLAN system.

Our implementation of the elastic WLAN system adopts a server to manage and control the APs and the hosts. This server has the administrative access to all the devices in the network. The server controls the system by the following five steps:

1. The server explores the devices in the network and collects the necessary information for the AP configuration algorithm.

2. The server executes the AP configuration algorithm using the inputs derived in the previous step. The outputs of the algorithm contain the list of the active APs and the host associations.

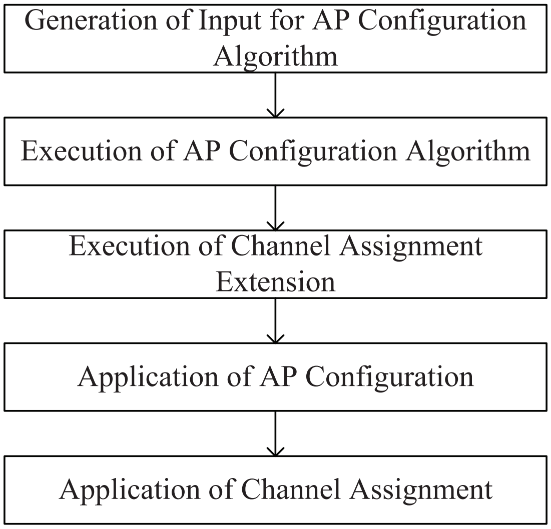

Generation of Input for AP Configuration Algorithm
Execution of AP Configuration Algorithm
Execution of Channel Assignment Extension
Application of AP Configuration
Application of Channel Assignment

Figure 3.2: Operational flow of elastic WLAN system.

3. The outputs of the AP configuration algorithm become the inputs to the channel assignment algorithm. The server executes the channel assignment algorithm, too. The output is the list of channels for the active APs.

4. The server applies the outputs of the AP configuration algorithm to the network by activating or deactivating the specified APs and changing the specified host associations.

5. The server assigns the channels to the active APs according to the output of the channel assignment algorithm.

3.4 Summary

In this chapter, we first described the motivation for the study of the elastic WLAN system. Then, we introduced the challenges in implementing the elastic WLAN system and possible solutions to overcome them. Finally, we presented the proposed design and operational flow of the elastic WLAN system. In the next chapter, we will describe the proposed active AP configuration algorithm for the elastic WLAN system.

Chapter 4

AP Configuration Algorithm

In this chapter, we present the AP configuration algorithm that dynamically controls the number of active APs in the network field by activating/deactivating APs based on traffic demands. First, we present the background of the AP configuration problem, followed by some related works. Then, we present the formulation of AP configuration problem and its proof of NP-completeness. Finally, we describe the proposed seven-phase AP configuration algorithm.

4.1 Introduction

The IEEE 802.11 wireless local area networks (WLANs) have been deployed everywhere for the Internet access [2]. Wireless connections between hosts and APs make WLANs inexpensive, flexible, and scalable. With the rapid advancements of wireless communications, WLAN users can set up their WLAN systems with different types of APs such as DAPs, VAPs and MAPs according to their budgets and demands. The channel and transmission power of each AP are generally configured by its management group.

However, if many APs are allocated and activated in the same network field, they may result in overlapping of coverage areas, and hence strong inter-AP interferences which may deteriorate the communication performance [3]. The configuration of these APs should be arranged according to the traffic demands and the network topologies.

The elastic WLAN system can adapt the network configuration with respect to network environment changes. Specifically, some APs may become crowded during peak hours and users suffer from low network performances. Then, activating new APs can ensure the expected performance.

Due to the power shortages, the device failures, and the network backbone inefficiency, an organization receives the lower Internet bandwidth than its expected value. In such cases, the link speed between the host and the AP needs to be adjusted for a fair Internet access to all the users. Then, the activations of cellular network based devices like MAPs can ensure the network performance.

In this research, we propose the *active AP configuration algorithm* to control the number of active APs for optimizing the elastic WLAN system under the minimum host throughput constraint and the use of heterogeneous APs. Three types of AP devices, namely DAPs, VAPs, and MAPs, are considered in this algorithm. MAPs are embedded with batteries and use cellular networks, so that they can be flexibly allocated at almost anywhere. In this research, we consider the locations of the hosts as the candidate positions for the MAPs, because host owners may use MAPs for the Internet access.

4.2 Related Works

Several researches have been conducted to improve the performance of the WLAN system and/or to reduce the operational cost.

In [41], the energy consumption is reduced by minimizing the transmission range of nodes while ensuring the desired coverage of the field of interest and connectivity of the network. It uses a scheduling protocol to periodically turn off communication radios i.e., deactivating nodes. Actually, they consider the dynamic deactivation of APs for wireless sensor networks. On the other hand, our approach considers a general wired-backbone WLAN.

In [42], an optimal AP placement problem for the uniform quality of services has been proposed to realize the WLAN in public.

In [43], a local search algorithm for the joint AP placement and channel assignment for IEEE 802.11 WLANs is proposed to optimize the performance of the WLAN system. In their AP placements, the loads of APs are not considered in optimizations. On the other hand, in our approach, the load of each AP is considered to optimize the throughput. The channel interference is not considered in our approach, which will be in our future works.

In [44], an optimization of the AP aggregation using virtual instances of DAPs has been proposed. The aggregation has been performed considering the receiving signal strength and the current bandwidth usage of APs.

In [45], an aggressive scheme is proposed to adapt APs according to the density of actual traffic loads. This method keeps APs inactive to the extent so that the remaining active APs can provide the coverage for the hosts. The number of active APs is changed according to the changes of traffic demands of hosts. Unfortunately, the associations of APs and hosts are optimized for given static traffic demands in this approach, where the dynamic changes of traffic demands are not considered.

In [46], an AP selection algorithm is proposed to maximize the throughput while preserving the newly arriving user throughput in a multi rate WLAN.

In [47], an active AP selection algorithm is introduced for wireless mesh networks ensuring the optimal operational cost and throughput. In [48], an AP allocation algorithm is proposed for wireless mesh networks with multiple gateways and the hop count limitation. These approaches aim to maximize the throughput by minimizing the number of hops between gateways and APs in wireless mesh networks while the number of active APs remains the same. On the other hand, in our approach, the number of active APs is minimized and the associations between APs and hosts are optimized.

4.3 Network Dynamics under Considerations

In this section, we describe the three major scenarios that we consider in this study for the elastic WLAN system.

- **Network Load Increase Scenario**: The number of users increases in the network.
- **DAP Failure Scenario**: Some DAPs do not function properly due to power shortages or device failures.
- **Bandwidth Limitation Scenario**: The total available bandwidth for the whole WLAN drops by the regulation of the authority or device failures at backbone networks.

To deal with these network dynamics, the elastic WLAN system works as follows. First, in the case of network load increase, VAPs by using user PCs are introduced. Second, in the case of DAP failures, VAPs and MAPs are gradually introduced. Last, in the case of the bandwidth limitation, MAPs are introduced in the network. The locations of hosts are considered as the candidate locations of MAPs.

4.3.1 Network Load Increase Scenario

When the network load increases, the number of hosts associated with each AP increases, and thus, throughputs for certain hosts may decrease. In this case, additional APs should be activated to meet the required throughput by increasing the internal bandwidth of the network. To satisfy the minimum host throughput constraint at every AP in the network, our algorithm introduces VAPs by utilizing PCs that connect to the Internet gateway by Ethernet and the installed VAP software there.

4.3.2 DAP Failure Scenario

This scenario considers situations where some DAPs are unable to function properly. For example, because of the power failure, only a few APs can be powered on with backup power sources like generators or batteries while others remain powered off. When some APs are out of services, the number of hosts associated with any functioning AP increases, and the overall performance may not meet the expected level. Like the previous scenario, there is enough available bandwidth but the internal network cannot fully utilize it due to the active APs shortage. If there are PCs that can be turned into VAPs, they can be used as APs to increase the bandwidth utilization in the network. To satisfy the *minimum host throughput constraint* at every AP, we introduce *VAPs* first, and then *MAPs* if necessary to ensure the minimum throughput for every host.

4.3.3 Bandwidth Limitation Scenario

The total allocated bandwidth to the whole network in an organization may be limited due to unexpected reasons like maintenances or some regulation works. This limited *total available bandwidth* should be distributed fairly to every host in the network. Otherwise, the total available bandwidth may be dried up by some greedy users, and the network becomes inaccessible to the remaining users. Thus, the adjustment of the link speed between a pair of AP and host is necessary, so that the limited bandwidth is fairly assigned to the users.

To satisfy the *minimum host throughput constraint* at every AP as best as possible, the number of active APs should be increased in the network. In this case, this limitation of bandwidth affects only the wired connection to the backbone network, and the introductions of VAPs will not increase the available bandwidth. Only MAPs can increase it as they use cellular networks. Thus, we introduce MAPs into the network to increase the available bandwidth of the network. This extra bandwidth by MAPs is not affected by the bandwidth limitation.

4.4 Formulation of AP Configuration Algorithm

In this section, we describe the active AP configuration problem to optimize the configuration of APs for the elastic WLAN system as a combinational optimization problem.

4.4.1 Basic Terminology

In this section, we briefly describe the definitions of some important terms that are used throughout this paper.

1. **Coordinate of network field(x,y):** In our paper, x and y in Eq. (2.2) represent the x-coordinate and y-coordinate in the network field of the transmission/reception device of a link.

2. **Inactive AP, active AP, and candidate AP:** In this paper, we categorize three states of an AP: *inactive AP*, *active AP*, and *candidate AP*. Initially, APs are switched off and are not connected to any host. We called them *inactive APs*. The proposed algorithm turns on some APs and connects to hosts. These APs are called *active APs*. Candidate APs are the APs that are considered that the algorithm can turn on.

3. **Available bandwidth and expected bandwidth:** *Available bandwidth* is the total speed allocated to the network field through wired connections to the backbone network. *Expected bandwidth* is the total speed required to the network field through wired connections such that every host communicates with the associated AP at the given link speed. These bandwidths are compared to find link speed reductions due to the available bandwidth limitation at the network field.

4. **Speed drop rate per wall:** In this thesis, the *speed drop rate per wall* indicates the percentage of the speed decrease from the original speed while passing through one wall, which is used for simulations. Our measurements found that the link speed decreases by about 15% when the transmitted signal passes through one concrete wall in our building. For example, in Figure 2.5, the link speed between the AP and the host is $60Mbps$ without any walls, while the link speed decreased to $51(= 60 \times 0.85)Mbps$ with a wall. The same speed drop rate per wall is adopted for any AP or wall type.

5. **Modifiable host and modifiable host list:** In our algorithm, the *modifiable host* represents the host that can be connected to two or more APs to satisfy the minimum host throughput constraint. Using Figure 4.1, the example for the modifiable host and the modifiable host list is explained. In this example, the solid line between a host and an AP represents that

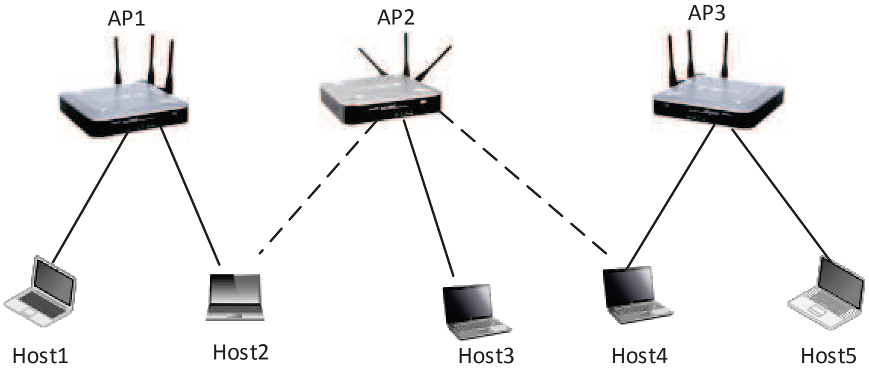


Figure 4.1: Example of modifiable hosts.

this host is connected with that AP, and the dotted line represents that the host is currently not connected with that AP but can be connected. In Figure 4.1, when $Host2$ connects with $AP1$, it can also connects with $AP2$. Hence, $Host2$ is a modifiable host for $AP1$. The modifiable host list for $AP1$ is $\{Host2\}$. Similarly, the modifiable host list for $AP2$ is $\{\}$ and the modifiable host list for $AP3$ is $\{Host4\}$.

4.4.2 Problem Formulation

The AP configuration problem in this paper is formulated as follows

1. **Hierarchical optimization :**

   (a) To minimize the number of active APs (DAPs, VAPs, and MAPs).

   (b) Holding the first objective, to maximize the minimum host throughput.

As the provisions in the active AP configuration algorithm for user increase and DAP failure scenarios, the two cost functions, E_1 and E_2 are introduced in this study. First, the cost function E_1 is used to represent the number of active APs in the network.

$$E_1 = E_1^D + E_1^V + E_1^M \tag{4.1}$$

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 as

$$T_j = \sum_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}} \tag{4.2}$$

where D_k represents the traffic of the kth host, s_{jk} represents the link speed between the jth AP to the kth host and $|\mathcal{P}_j|$ is the set of hosts that connect to the jth AP. $|\cdot|$ denotes the cardinality operation.

Then, the throughput of the ith host in the jth AP, $\mathcal{R}_{ij}$ can be calculated by

$$\mathcal{R}_{ij} = \frac{D_i}{T_j} = \frac{D_i}{\sum_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}}} \tag{4.3}$$

where D_i represents the traffic of the ith host. Then, the minimum host throughput can be defined by

$$\mathcal{R}_{min,j} = \min_{i=1,2,..,|\mathcal{P}_j|} \left[\frac{D_i}{\sum_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}}} \right] \tag{4.4}$$

Since the traffic of each host is unpredictable, we assume the identical traffic of each host, which can be represented by the unit traffic for the sake of simplicity:

$$\mathcal{R}_{min,j} = \frac{1}{\sum_k \frac{1}{s_{jk}}} \tag{4.5}$$

Now, we can formulate the objective function of the problem as follows:

$$E_2 = \min_j \left[\mathcal{R}_{min,j}\right] \tag{4.6}$$

Under the constraints, we want to maximize E_2.

2. **Inputs :**

   (a) Network topology:

   - Number of hosts: H
   - Number of APs: $N = N^D + N^V + N^M$ where N^D, N^V and N^M respectively represent the number of DAPs, VAPs, and MAPs.

   (b) Information of APs:

   - AP ID: i = 1 *to N*

   (c) Information of hosts:

   - Host ID: i = 1 *to H*

   (d) Link speed of the ith AP to the jth host, s_{ij} ($i = 1$ *to* N, $j = 1$ *to* H)
   The proposed algorithm uses link speeds between APs and hosts as the inputs. The link speeds can be estimated by measuring the signal strength of the hosts and our derived formula in Section 2.2.1.

   (e) Algorithm parameters:

   - Data plan for MAP ($Mbps$)
   - Minimum host throughput constraint: $G(Mbps)$
   - Throughput limit (bounded by the Ethernet capacity) constraint: $B^a(Mbps)$

3. **Outputs :**

   (a) The set of active DAPs, VAPs, and MAPs

   (b) The associated hosts for each active AP

4. **Constraints :**

   (a) Minimum host throughput constraint, G: every host in the network will enjoy the minimum throughput.

   (b) Throughput limit constraint, B^a: the bandwidth for the wired network must be less than the total available bandwidth of the network. The total available bandwidth of the network is determined by the external Ethernet bandwidth.

4.4.3 NP-Completeness of AP Configuration Problem

The AP configuration problem requires extensive computations to obtain the optimal solution. In this setting, we prove that the AP configuration is an NP-complete problem by reformulating this problem in the form of the *minimum set cover problem* [50], which is a well-known NP-complete problem [49].

To prove the *NP-completeness* of the AP configuration problem, first the problem is translated to its decision version. Then, the well known *NP-complete* problem can be reduced to this decision version of the AP configuration problem.

4.4.3.1 Decision Version of AP Configuration Problem

The decision version of the AP configuration algorithm is defined as follows:

- **Instance** The same inputs as the AP configuration problem and an additional constant E_0.
- **Question** Is there an AP configuration result to satisfy $E_1 \leq E_0$ such that $E_1 = E_1^D + E_1^V + E_1^M$?

4.4.3.2 Minimum Set Cover Problem

The *minimum set cover problem*, *min_set* is defined as follows

- **Instance** A collection C of subsets S_i of a finite set S for $i = 1, 2, , ..., |C|$ and a constant volume K.
- **Question** Is there an sub-collection $C' \subseteq C$ such that every element in S is included in at least one member of C' and $|C'| \leq K$?

4.4.3.3 Proof of NP-Completeness

Clearly, the AP configuration problem belongs to the class NP. Then, an arbitrary *min_set* certificate can be transformed into the following AP configuration problem instance, which proves the NP-completeness of AP configuration algorithm.

- **Input** $N = |C|$ and $H = |S|$ for any host and any AP, the set of associable hosts for the *i*th AP $= S_i$ and $sp_{ij} = 1$ for the *i*th AP to the *j*th host and vice-versa.

4.5 Active AP Configuration Algorithm

In this section, we propose the AP configuration algorithm to realize the elastic WLAN system by deploying active DAPs, VAPs and MAPs properly. The proposed algorithm first minimizes the number of active APs in the network, and then, minimizes the transmission delay to ensure the minimum host throughput for every host in the network. In addition, the proposed algorithm adjusts the host link speed if the expected bandwidth is higher than total available bandwidth. As depicted in Figure 4.2, the proposed algorithm comprises the preprocessing, the initial solution generation, the host association improvement, the AP selection optimization, the link speed normalization, the termination check, the additional VAP activation, and the additional MAP activation phases.

4.5.1 Preprocessing

In this phase, if the locations of the hosts and the APs (DAPs, VAPs, or MAPs) are given as inputs to the algorithm for simulation, the link speed between every possible pair of AP and host is estimated by using Eq. (2.1) and (2.2) and considering the speed drop rate per wall. Then, this phase initializes the variables for the next phases. This phase comprises of the following procedures:

1. For every AP, make a list of hosts that can be associated to this AP. We call this *associable host list* for the AP.

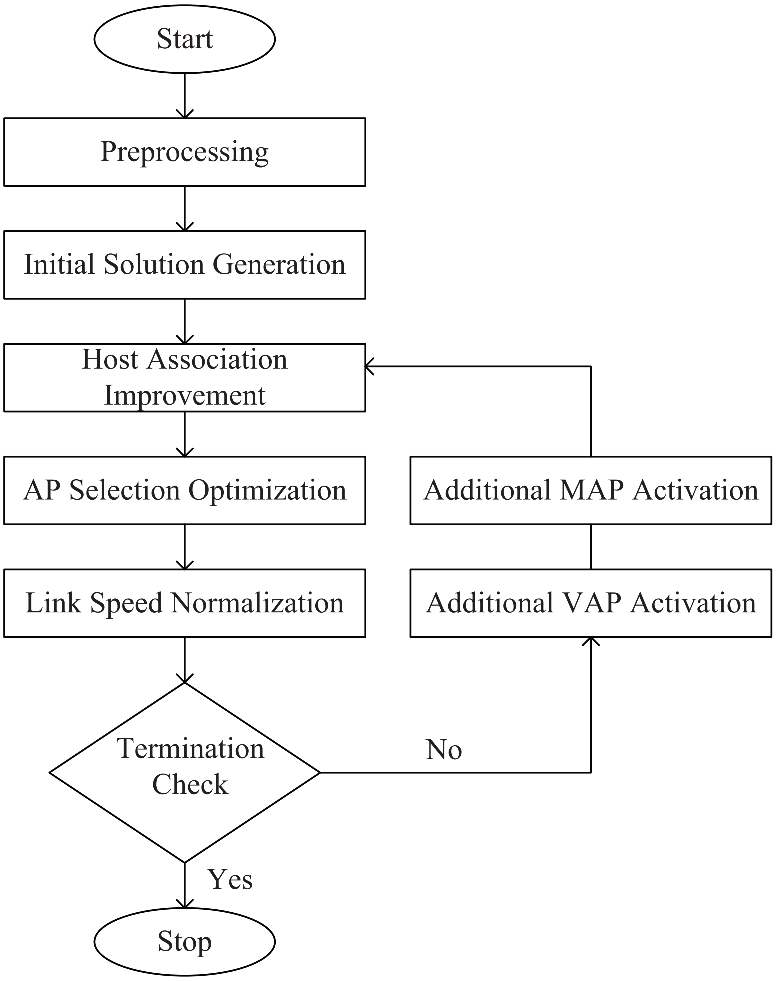

Start
Preprocessing
Initial Solution Generation
Host Association Improvement
AP Selection Optimization
Link Speed Normalization
Termination Check
No
Yes
Stop
Additional MAP Activation
Additional VAP Activation

Figure 4.2: Flow chart of the AP configuration algorithm

2. For every host, make a list of APs that can be associated to this host. They are called *associable APs* for the host.

3. Initialize each AP as a non-active AP. Initially, only the DAPs are selected as *candidate APs*.

4.5.2 Initial Solution Generation

In this phase, an initial solution is derived using a greedy algorithm [51]. This provides the initial number of active APs, E_1 and the hosts associated to these APs. This solution quality may be poor at this stage.

To find the initial solution, the number of active APs, E_1 is first initialized by $E_1 = 0$. Then, the initial AP configuration is found by repeating the following three steps until all the hosts are covered in the network:

1. From the associable host list for APs, select one AP (let AP_i) that can cover the maximum number of uncovered hosts $|\mathcal{P}_i|$ and associate all the associable hosts to AP_i.

2. Activate the AP and increment E_1 by one.

3. Update the number of remaining hosts in the associable host list that can be covered by remaining APs.

4.5.3 Host Association Improvement

In this phase, host associations of active APs found in the previous phase are improved by randomly changing the association of a host. This modification is carried out with a view to improve the overall throughput by optimizing the minimum host throughput, E_2. The host association for each AP is optimized by repeating the following steps with 10,000 times:

1. Change the association of every host to the AP that provides the maximum link speed among the associable APs. Calculate the cost function E_2 and store it as the best found cost at this stage, E_2^{best}.

2. Find the AP that gives the lowest host throughput in Eq. (4.6), and make the list of *modifiable host* that are associated with this AP and can be associated with the other APs.

3. Select one *modifiable host* randomly from the modifiable host list. Then, randomly associate this host with another associable active AP. Calculate the new cost function E_2^{new}.

4. If $E_2^{new} > E_2^{best}$, replace E_2^{best} by the newly found E_2^{new} and keep the new association. Otherwise, roll back to the previous association and the best cost function E_2^{best}.

4.5.4 AP Selection Optimization

The cost functions E_1 and E_2 are further jointly optimized in this phase by the *local search* [52] under the constraints mentioned before. This phase minimizes the number of active APs, E_1, if the current number of active APs can satisfy the minimum host throughput constraint, G. Otherwise, this phase increases the number of active APs to satisfy the minimum host throughput constraint. In both cases, the host association is optimized using Phase 4.5.3. The procedure is described as follows:

1. Initialize E_1^{best} and E_2^{best} as the current algorithm output after Phase 4.5.3. Repeat Steps 2 to 4 with $40 \times N \times H$ times.

2. While $E_2^{best} \geq G$ (minimum host throughput constraint) is satisfied, repeat the following procedures:

   (a) Select an active AP randomly such that all the hosts associated to this AP, can be associable to other active APs.

   (b) Deactivate this AP and associate all the hosts associated to this AP to other associable active APs. If there are multiple such active APs, select one in order of the largest number of unassociated hosts they can cover.

   (c) Decrease E_1 by one.

   (d) Call Phase 4.5.3 to optimize the host association for the current number of active APs.

   (e) Calculate the new value for cost function E_2, say, E_2^{new}.

   (f) If $E_2^{new} > E_2^{best}$, replace E_2^{best} by E_2^{new}, keep the current association, otherwise roll back to the previous association.

3. If $E_2^{best} \geq G$ is not satisfied, repeat the following procedures with $5 \times N$ times:
   (a) Randomly select one non-active AP and activate it.
   (b) Select an active AP randomly and deactivate it if all the hosts associated to this AP, can be associable to currently active APs.
   (c) Call Phase 4.5.3 to optimize the host association for the current active APs.
   (d) Calculate the new value for the cost function E_2 say, E_2^{new}
   (e) If $E_2^{new} > E_2^{best}$, replace E_2^{best} by E_2^{new}, keep the current association, otherwise roll back to the previous association.
4. If $E_2^{best} \geq G$ is not satisfied in the previous step, do the following:
   (a) Randomly select a non-active AP and activate it.
   (b) Increase E_1 by one.
   (c) Call Phase 4.5.3 to optimize the host associations for the current active APs.
   (d) Calculate the new value for the cost function E_2 say, E_2^{new}
   (e) If $E_2^{new} > E_2^{best}$, replace E_2^{best} by E_2^{new}.
   (f) Go back to 2.
5. Output the best found result E_1^{best} and E_2^{best}.

4.5.5 Link Speed Normalization

For the bandwidth limitation scenario, we need to adjust the link speed to reflect the total available bandwidth. First, a new constraint parameter B^a is introduced into our algorithm to represent the total available bandwidth of the network. We apply the fairness criterion when the total expected bandwidth exceeds B^a. In this case, the throughput of the ith AP is given by

$$b_i = \frac{|\mathcal{P}_i|}{\sum_{j=1}^{|\mathcal{P}_i|} \frac{1}{s_{ij}}} \tag{4.7}$$

where $|\mathcal{P}_i|$ represents the number of hosts associated with the ith AP, and s_{ij} does the link speed between the ith AP and the jth host. With the derived throughputs for APs, the total expected bandwidth B^e can be computed by

$$B^e = \sum_{i=1}^{N} b_i \tag{4.8}$$

If B^e is larger than B^a, our algorithm normalizes the link speed s_{ij} to satisfy the available bandwidth limitation:

$$\hat{s}_{ij} = s_{ij} \times \frac{B^a}{B^e} \tag{4.9}$$

where $\hat{s}_{ij}$ is the normalized link speed. It should be emphasized that only the links associated with the DAPs or VAPs are adjusted, because the MAPs using cellular networks are not influenced by the total bandwidth limitation of the wired backbone network.

To normalize the link speed, do the following:

1. Calculate the expected total bandwidth B^e using Eq. (4.7) and (4.8)
2. If $B^e > B^a$, adjust every AP-host link speed as described in Eq. (4.9).

4.5.6 Termination Check

This phase terminates the algorithm when either of the following conditions is satisfied:

1. The *minimum host throughput constraint* condition is satisfied.

2. All the APs in the network have already been activated.

4.5.7 Additional VAP Activation

If VAPs are not selected for candidate APs, they are selected as candidate APs. Go back to Phase 4.5.3. Otherwise, go to Phase 4.5.8.

4.5.8 Additional MAP Activation

In this phase, the locations of hosts are considered as the locations for the candidate MAPs. The MAPs are activated sequentially until the *minimum host throughput* constraint is satisfied. The location of a newly introduced MAP is found by the following procedure:

1. Find the APs that do not satisfy the *minimum host throughput constraint*, which are called *unhappy APs* in this thesis. This constraint for the jth AP can be checked by:

$$\frac{1}{\sum_{k=1}^{|\mathcal{P}_j|} \frac{1}{s_{jk}}} \geq G \tag{4.10}$$

   where G represents the *minimum host throughput constraint* threshold that has been specified by the network designer.

2. Find the most unhappy AP that has the lowest throughput among the unhappy APs.

3. Select the location of the associated host with this most unhappy AP that has the smallest link speed for the new MAP.

Here, we must note that when the *minimum host throughput constraint* is checked, the link speed for any DAP or VAP is adjusted by Eq. (4.9) if $B^e > B^a$. The additional MAP activation is completed by repeating the following procedure:

1. Initialize $K = 1$. K denotes the number of MAPs necessary to be activated to satisfy the constraint.

2. Apply the procedure to find the location of a newly introduced MAP described above.

3. Turn on the new MAP at the selected location, activate this MAP, and associate hosts to this MAP holding the minimum host throughput for any host.

4. Apply the *Host Association Improvement* Phase to improve the host association with a view to improve the overall throughput.

5. Apply the following *Host Association Swapping* to further improve the host association in this AP.

- Host Association Swapping
    1) Initialize E_2^{best} by the current algorithm output value of E_2.
    2) Select a new pair of hosts that satisfy the following conditions:
        a) They are not associated with the same AP.
        b) The minimum host throughput, E_2 in Eq. (4.6) is increased after the swapping.

       If no such pair is found, terminate the procedure and return the E_2^{best}.
    3) Swap their associated APs, and calculate value of E_2 i.e. E_2^{new}.
    4) If $E_2^{new} > E_2^{best}$, replace E_2^{best} by E_2^{new}. Otherwise, resume the previous host associations.
    5) Go to 2).

6. Check the *minimum host throughput constraint* in the current solution. If it is not satisfied and K is smaller than the upper limit, reset the allocated MAPs, increment K by one, and go to 2. Otherwise, terminate the algorithm and output the final solution.

4.6 Summary

In this chapter, we presented the active AP configuration algorithm for the elastic WLAN system. Three types of AP devices are considered in this algorithm, namely DAPs, VAPs, and MAPs, with their link speed differences. The AP configuration problem is formulated as a combinatorial optimization problem and is proved the NP-completeness of its decision version. The proposed heuristic algorithm is composed of seven phases. This algorithm can dynamically control the number of active APs under the constraint of minimum host throughput and heterogeneous APs in the network. In the next chapter, we will present the channel assignment algorithm for the elastic WLAN system.

Chapter 5

Channel Assignment Extension

In this chapter, we present the channel assignment extension to the AP configuration algorithm. First, we present the background of the extension, followed by some related works. Then, we formulate the channel assignment problem. Finally, we describe the procedure for the proposed channel assignment extension for the elastic WLAN system.

5.1 Introduction

In the previous chapter, we presented the AP configuration algorithm for the elastic WLAN system. We formulated the AP configuration problem, proved its NP-completeness, and proposed its seven-phase heuristic algorithm [9]. The proposed algorithm dynamically adjusts the number of active APs in the network field depending on communication demands.

This algorithm assumes that each AP can use a different non-interfered channel from the other APs to avoid interferences among them at all. In reality, the number of non-interfered channels available in *IEEE 802.11* protocols is limited [11]. Therefore, some APs have to share the same channel with each other if the number of APs is larger than the number of available non-interfered channels. Therefore, the proper channels should be assigned to the APs so that it can reduce the interference as much as possible to improve the network performance.

In this research, we propose the *channel assignment extension* of the AP configuration algorithm for the elastic WLAN system to solve the above mentioned problem. One channel is assigned to every active AP from the limited number of the non-interfered channels in a way to minimize the overall interference in the network, after the active APs and the AP-host associations are found in the previous algorithm.

Then, the *channel load averaging* is applied to balance the loads among the channels by changing some AP-host associations. After the channel assignment, the communication loads among the channels may become imbalanced, because the previous algorithm finds the AP-host associations assuming the infinite number of the non-interfered channels.

In short, the contributions of this section are summarized as below:

1. The channel assignment extension to the active APs from the limited number of non-interfered channels in a way to minimize the overall interference in the network,

2. The load averaging extension among the channels for the throughput improvement after the channel assignment.

5.2 Related Works

A variety of studies have been addressed to the channel assignment problem in WLAN. In [53], the *Least Congested Channel Search (LCCS)* approach was proposed. The network administrator conducts the on-site survey of radio frequencies to determine the number of APs and their locations that are required for the coverage. Then, these APs are configured manually with non-interfered channels to avoid interferences as much as possible. This approach is manual, whereas our proposal is automatic.

In [54], a centralized traffic and interference-aware channel assignment approach was proposed for a mesh network. Channels are assigned to APs based on the ranking that is generated from important parameters such as the aggregate traffic, the distance from the gateway node, and the number of interfaces at each node. This approach focuses on finding the optimal route in a mesh network allocating non-interfered channels, whereas our proposal assigns channels to APs in WLAN.

In [55], an interference system model using the interference graph was proposed. The degree of the interference between two channels is defined using the interference table. Then, the interference minimization problem is formulated, and the channel assignment algorithm is developed to minimize the total interference in the network. This algorithm is a simple greedy one that cannot often find optimal solutions, whereas our proposal combines a greedy algorithm and a simulated annealing to seek optimal solutions.

In [56], another interference system model was proposed to consider the channel bonding and frame aggregation mechanisms in the IEEE 802.11n protocol. First, the throughput of each host in the system is estimated. Then, a channel assignment problem is formulated and translated into a throughput optimization problem. Finally, a distributed channel assignment algorithm is developed for multi-rate 802.11n WLANs. This approach pays the higher cost to exchange the throughput information among hosts, whereas our proposal is the centralized one to avoid it.

5.3 Formulation of Channel Assignment Problem

In this section, we formulate the channel assignment problem as a combinatorial optimization problem to present the corresponding algorithm. The channel assignment problem is formulated as follows:

1. **Inputs:**
   - Output of the AP configuration algorithm: the set of active APs and their associated hosts
   - Number of non-interfered channels
2. **Outputs:**
   - Channel assignment to the active APs
3. **Objectives:**
   - The total interfered communication time E should be minimized.

$$E = \sum_{i=1}^{N}[\tau_i] = \sum_{i=1}^{N}\Big[\sum_{\substack{k\in I_i \\ c_k=c_i}} T_k\Big] \tag{5.1}$$

where $\mathcal{T}_i$ represents the interfered communication time for AP_i, T_i does the communication time for AP_i, I_i does the set of interfered APs for AP_i, and c_i does the assigned channel to AP_i.

4. **Constraint:**

    - Every AP must be assigned one channel.

5.4 Procedure of Channel Assignment Extension

In this section, we describe the procedure for the channel assignment extension to the AP configuration algorithm. Figure 5.1 shows the flow chart for the proposed channel assignment extension. The outputs of the AP configuration algorithm, namely the list of the active APs, the list of the hosts and the information of AP-host associations, become the inputs to the channel assignment extension. The procedure for the channel assignment extension is composed of the four phases.

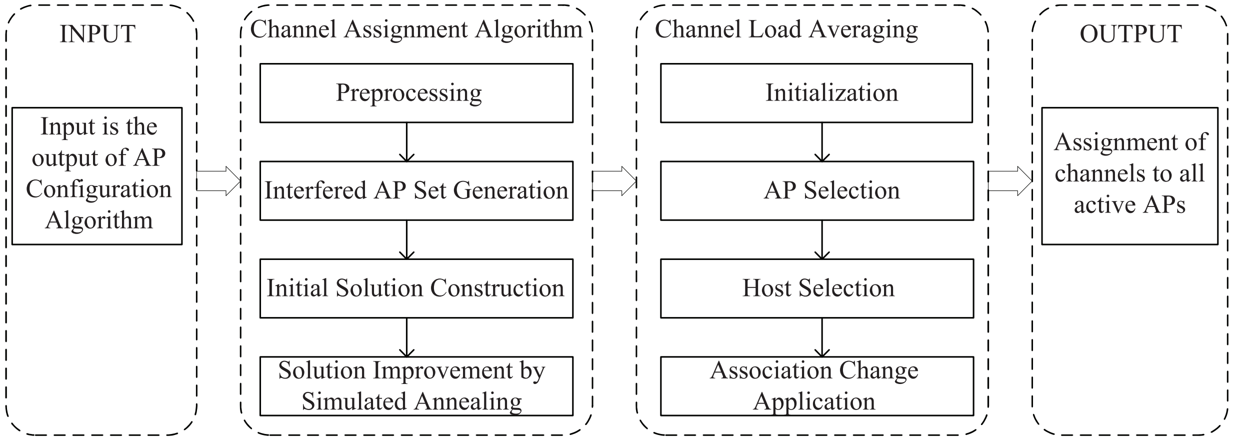

INPUT
Input is the output of AP Configuration Algorithm
Channel Assignment Algorithm
Preprocessing
Interfered AP Set Generation
Initial Solution Construction
Solution Improvement by Simulated Annealing
Channel Load Averaging
Initialization
AP Selection
Host Selection
Association Change Application
OUTPUT
Assignment of channels to all active APs

Figure 5.1: Channel assignment extension flow.

5.4.1 Preprocessing

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

1. 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 the ith AP is interfered with the jth AP in the network.

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

$$T_i = \sum_j \frac{1}{s_{ij}}. \tag{5.2}$$

   where s_{ij} represents the link speed between the ith AP to the jth host.

3. Calculate the *neighbor interfered communication time* for each AP. The neighbor interfered communication time $\widehat{T_i}$ for the ith AP is given by:

$$\widehat{T_i} = \sum_{e(i,k)=1} T_k. \tag{5.3}$$

5.4.2 Interfered AP Set Generation

The set of APs that are interfering with each other is found for each AP.

1. Sort the APs in descending order of $\widehat{T_i}$, where the tie-break is resolved by T_i.
2. Find the interfered AP set for each AP by repeating the following steps:
   (a) Initialize the interfered AP set by $I_i = \{i\}$ for AP_i.
   (b) Expand I_i by checking the APs in the sorted order in (a) whether the AP is interfered with every AP in I_i. If so, include this AP, AP_j into I_i, i.e. $I_i = I_i \cup \{j\}$.
3. Calculate the total interfered communication time $\overline{T}_i$ for the ith AP that is given by:

$$\overline{T}_i = \sum_{k \in I_i} T_k. \tag{5.4}$$

5.4.3 Initial Solution Construction

Then, an initial solution is derived using a greedy algorithm.

1. Sort the APs in descending order of the total interfered communication time $\overline{T}_i$, where the tie-break is resolved by $\widehat{T_i}$.
2. Assign a channel c to AP_i such that the interfered communication time $\mathcal{T}_i$ is minimized if assigned. $\mathcal{T}_i$ is given by:

$$\mathcal{T}_i = \sum_{\substack{k \in I_i \\ c_k = c}} T_k \tag{5.5}$$

   where c_k is the assigned channel to AP_k.
3. Repeat 2. until each AP is assigned one channel.
4. Calculate the cost function E using Eq. (5.1) and save this initial solution as the best solution E^{best}.

5.4.4 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} for the *SA repeating times* R^{SA}, where T^{SA} and R^{SA} are given algorithm parameters:

1. Randomly select one AP and one new channel for the channel change trial.

2. Calculate the interfered communication time, τ_i after assigning this new channel by:

$$\tau_i = \sum_{\substack{k \in I_i \\ c_k = c_i}} T_k. \tag{5.6}$$

3. Calculate E^{new} using Eq. (5.1) for the new channel assignment, and $\Delta E = E^{new} - E$.

4. If $\Delta E \leq 0$, accept the new channel assignment, and keep this new solution as the best one.

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

5.4.5 Channel Load Averaging

After the channel assignment to the APs, the total load can be imbalanced between different channels. That is, some channels may be crowded, and some channels may not. Because the associated hosts with each AP are fixed during the channel assignment, it is impossible to average the load by changing host associations. Thus, to further improve the performance of the network, the load averaging procedure among channels is applied to the solution from the channel assignment.

5.4.5.1 Initialization

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

5.4.5.2 AP Selection

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

1. Terminate the procedure if every AP has $1(= ON)$ AP flag.

2. Initialize the host flag by $0(= OFF)$ for every host.

3. Select one AP, say AP_i, that satisfies the two conditions:

    (a) The AP flag is OFF.

    (b) The interfered communication time τ_i is largest among the OFF APs.

4. If one AP is selected, set the corresponding AP flag ON.

5.4.5.3 Host Selection

Then, one host associated with AP_i is selected for the AP movement.

1. Select one host, say H_j, that satisfies the four conditions:

    (a) The host flag is OFF.

    (b) The host is associated with AP_i.

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

(d) The link speed of AP_i and the host is the smallest among the hosts satisfying (a)–(c).

2. If one host is selected, set the corresponding host flag ON.

3. Otherwise, go back to 5.4.5.2 for the new AP selection.

5.4.5.4 Association Change Application

Finally, the new associated AP is selected for H_j.

1. Select the AP that has the largest link speed among the APs found in 5.4.5.3−1. − (c).

2. Calculate the cost function E using Eq. (5.1) if H_j is associated with this AP.

3. If the new cost function E is equal to or smaller than the previous E, accept the new association, and go back to 5.4.5.3.

4. Otherwise, select another AP that has the next largest link speed, and go back to 2.

5. If no such AP exists, go back to 5.4.5.3 for the new host selection.

5.5 Channel Assignment Example

In this section, we will give a brief explanation of the proposed channel assignment algorithm with an example. Let us consider the following topology with 8 APs and 17 hosts as shown in Figure5.2.

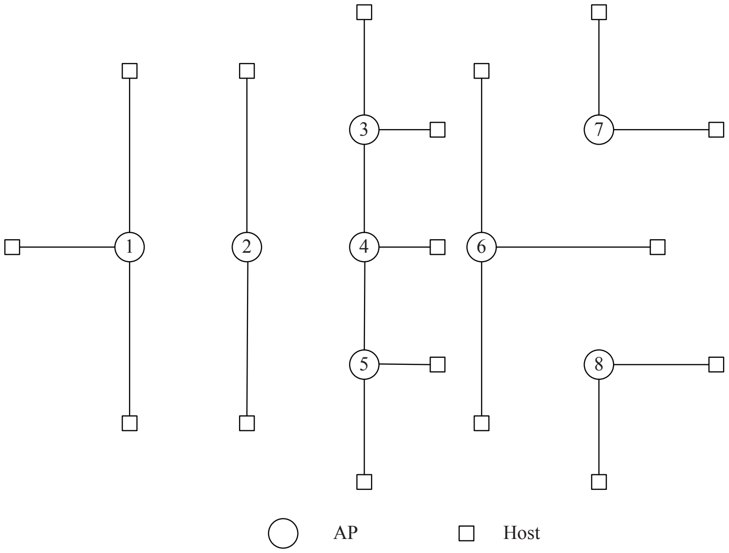


Figure 5.2: Example channel assignment topology.

The circle and rectangular shapes represent the APs and hosts. An edge between an AP and a host does the association of a host to that AP. First, the interference graph is constructed from the given topology as described in 5.4.2. Figure5.2 shows the derived interference graph. The edge between two nodes represents the interference between them. Two APs are interfered with each other if their distance is less than $110m$.

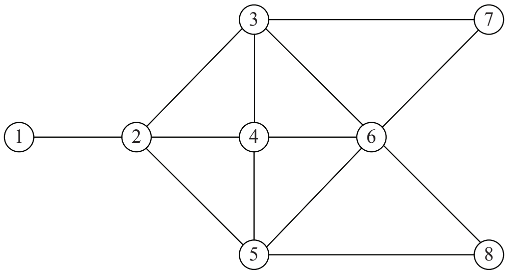


Figure 5.3: Interference graph of example channel assignment topology.

Then, the greedy channel assignment algorithm is applied as described in 5.4.3. Figure5.4. shows the channel assignment after applying the greedy algorithm. In the figure, the different non-overlapping channels are represented by different shapes.

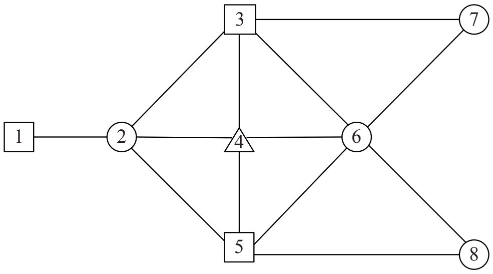


Figure 5.4: Channel assignment after greedy algorithm.

Then, the simulated annealing is applied as described in 5.4.4. Figure5.5. shows the channel assignment after applying the simulated annealing. The figure indicates that the greedy algorithm fails to assign different channels to AP#6 and AP#7, and AP#6 and AP#8, while the simulated annealing avoids them.

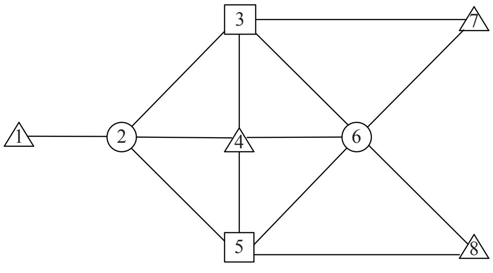


Figure 5.5: Channel assignment after simulated annealing.

5.6 Summary

In this chapter, we first presented the motivation of the channel assignment extension to the AP configuration algorithm. Then, we formulated the channel assignment problem. Finally, we described the procedure for the proposed extension channel including the load averaging for the elastic WLAN system. In the next chapter, we will evaluate the proposed AP configuration algorithm with extensions using the WIMNET simulator.

Chapter 6

Evaluation of AP Configuration Algorithm by Simulation

In this chapter, we evaluate the proposed AP configuration algorithm in various network environments using the WIMNET simulator [47]. First, we give a brief description of the WIMNET simulator and its extensions for simulating the AP configuration algorithm. Then, we evaluate the AP configuration algorithm through simulations in two different network topologies. Finally, we evaluate the channel assignment extension and the load averaging through simulations in different two network topologies.

6.1 WIMNET Simulator

The WIMNET simulator [47] has been originally developed to evaluate a large-scale *Wireless Internet-access Mesh NETwork (WIMNET)* with a reasonable CPU time on a standard PC. Some modifications have been made to the WIMNET simulator for the simulations of the elastic WLAN system with heterogeneous APs.

This simulator simulates the least functions for wireless communications between hosts and APs that are required to estimate throughputs or delays. Several network field parameters such as host locations, AP locations, link distances, communication routes, wall existences, and repeater existences are adopted in the WIMNET simulator. The sequence of the wireless communications including the host movements, the communication request arrivals, and the link activations is synchronized by a single global clock called a *time slot*. Within an integral multiple of *time slots*, a host or an AP can complete the one-frame transmission and the acknowledge reception. Different transmission rates can be set by manipulating the number of time slots required for the one link activation. Table 6.1 summarizes the hardware and software platforms used in our simulations.

Table 6.1: Simulation Environment.

simulator	WIMNET Simulator
interface	IEEE 802.11n
CPU	Intel Core i7
memory	4 GB
OS	Ubuntu LTS 14.04

In this thesis, two types of nodes, namely VAPs and MAPs, are newly introduced in the WIM-

NET simulator, in addition to existing hosts and DAPs. For each MAP, the corresponding gateway node is introduced so that every host associated with the MAP has the Internet access through this gateway.

6.2 Network Environments

In this thesis, we consider the *indoor network environment* inside a building for the target one of our proposal, because *wireless local area networks (WLANs)* are usually used there. In such environments, users often use personal computers (PCs) to access the Internet at fixed positions where chairs or tables are available. The mobility of WLAN users is much lower than that of cellular system users, because PCs are much larger and heavier than smart phones, and often require the use of both hands. Thus, WLAN users connect with WLANs while sitting on chairs and putting their PCs on tables. Besides, to consider the effects by the walls along the transmission link between two devices in simulations, it is assumed that the link speed is reduced by 15% each time the link passes through one wall. The hosts here are often regularly distributed since students have their fixed positions in the laboratories.

6.3 Evaluation of AP Configuration Algorithm

In this section, we evaluate the AP configuration algorithm in two network instances using the WIMNET simulator. First, we present simulation results under network environment changes. Then, we present the results under different values of the *MAP speed* and the *wall effect*. Finally, we compare the performance of our algorithm with other algorithms.

6.3.1 Simulations Under Network Environment Changes

The proposed AP configuration algorithm is first evaluated sequentially in three scenarios mentioned in Section 4.3. We consider the network topology which models the third floor of Engineering Building-2 in Okayama University, Japan, where it is slightly modified for simulations. There are six rooms with two different sizes, $7m \times 6m$ and $3.5m \times 6m$. We allocated two DAPs and 10 VAPs in this field. The number of MAPs is determined by the proposed algorithm. Any DAP or VAP is connected to the Internet via a wired cable, and the MAP is connected to the Internet through the cellular mobile network. Various cases are considered with the different number of hosts from 40 to 60, the different minimum host throughput constraint, G from $5Mbps$ to $10Mbps$, and the different throughput limit B^a from $50Mbps$ to $150Mbps$ for limited bandwidth scenario, otherwise ∞.

6.3.1.1 Network Load Increase Scenario Result

First, we consider the scenario where the increase of users raises the loads on the network. The number of hosts H and the minimum host throughput constraint G are both increasing. Table 6.2 summarizes the simulation results for this scenario. The table shows the required number of active APs, the analytical result estimated by our algorithm, and the numerical result generated by the WIMNET simulator for different network loads and different minimum host throughput constraints. Throughout this paper, *ana. min. host through.* represents the analytical minimum host throughput estimated by our algorithm, *num. min. host through.* represents the numerical minimum host throughput and *num. overall through.* represents the numerical overall throughput generated by the WIMNET simulator.

Table 6.2: Simulation results for network load increase scenario with DAP=2, VAP=10, and $B^a = \infty$.

G ($Mbps$)	5			10		
number of hosts (H)	40	50	60	40	50	60
active VAPs	1	2	3	6	9	10
ana. min. host through.	5.97	5.53	5.51	11.08	10.97	9.95
num. min. host through.	6.54	5.79	5.75	10.68	10.82	9.97
num. overall through.	245.47	289.63	345.39	441.59	563.92	580.11

This table indicates that, as the number of hosts increases, the algorithm increases the number of active VAPs, and that the larger minimum host throughput G requires more VAPs. Unfortunately, the algorithm cannot find a feasible solution for the extreme network load of hosts $H = 60$and the minimum host throughput $G = 10$ where more APs are presumably required.

6.3.1.2 Solution Example

To illustrate this network topology, Figure 6.1 shows one solution determined by the algorithm for the DAP failure scenario with $H = 60$, $G = 10$, one active DAP, 10 active VAPs, and five active MAPs, presented in Table 6.3. It is observed that the DAP has the largest number of associated hosts, and the MAPs have the smallest number of associated hosts. The number of associated hosts is nearly proportional to the link speed. In this figure, each DAP, VAP, and MAP is connected respectively to 10, 4 and 2 hosts and the link speed to the Internet is around $100Mbps$, $40Mbps$, and $20Mbps$.

6.3.1.3 DAP Failure Scenario Result

Then, we consider the *DAP failure scenario* where one DAP is not functioning properly. The same number of hosts H and the same value for the minimum host throughput constraint G are used as those in the previous scenario. Table 6.3 summarizes the results for this scenario.

Table 6.3: Simulation results for DAP failure scenario with DAP=1, VAP=10, data plan = $30Mbps$, and $B^a = \infty$.

G ($Mbps$)	5						10					
APType	VAP	MAP	VAP	MAP	VAP	MAP	VAP	MAP	VAP	MAP	VAP	MAP
number of hosts (H)	40		50		60		40		50		60	
active number	3	0	4	0	6	0	8	0	10	1	10	5
ana. min. host through.	5.99		5.62		6.03		10.98		10.05		10.13	
num. min. host through.	5.96		5.61		5.97		10.82		10.09		10.04	
num. overall through.	238.45		280.67		359.20		440.95		505.27		583.26	

This table indicates that our algorithm can find a feasible solution that satisfies the constraints by activating more VAPs than the previous scenario and activating new MAPs if necessary. When a DAP fails, the throughput in the network drops and additional AP activations are needed to

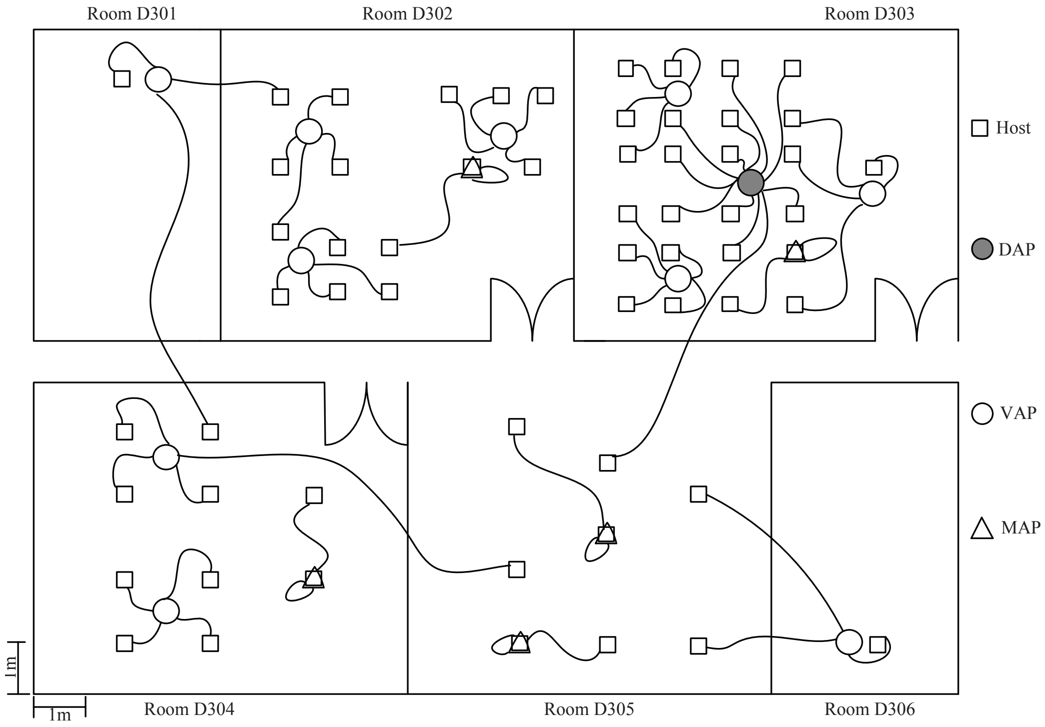


Figure 6.1: Solution of proposed algorithm for scenario with $H = 60$, $G = 10$, DAP=1, VAP=10, MAP=5, and data plan=30$Mbps$ in Okayama University instance.

maintain the minimum host throughput constraint in the network. For additional traffic demands, our algorithm first activates the VAPs, and then activates the MAPs if necessary in the network. Even if one DAP is failed, it can satisfy $G = 10$, the minimum host throughput constraint, by activating five MAPs.

6.3.1.4 Bandwidth Limitation Scenario Result

Finally, we consider the *bandwidth limitation scenario*, where the *total available bandwidth* for the network is limited. The same number of hosts H is used as those in the previous scenarios. The minimum host throughput constraint G is fixed as 5. The throughput limit constraint B^a is set 150, 100, and 50$Mbps$. If two DAPs are active, they consume all the capacity of the wired access to the Internet, where VAPs become useless in this scenario. Then, the activations of MAPs can provide the additional bandwidth if it is necessary in the network.

Table 6.4 summarizes the results for this scenario. This table indicates that our algorithm can find a feasible solution by activating more MAPs than the previous scenarios. When B^a decreases from 150$Mbps$ to 50$Mbps$, our algorithm activates more MAPs to satisfy the minimum host throughput constraint by increasing the bandwidth for the Internet connections. For 40 users, when B^a is 150$Mbps$, two DAPs along with three MAPs can satisfy the minimum host throughput constraint. When B^a is 100$Mbps$, 1 DAP and six MAPs are needed, and when B^a is 50$Mbps$, one DAP and nine MAPs are needed to satisfy the *minimum host throughput constraint*.

Table 6.4: Simulation results for bandwidth limitation scenario with DAP=2, VAP=10, $G = 5$, and data plan = $30Mbps$.

B^a	number of hosts (H)	40		50		60	
($Mbps$)	AP Type	DAP	MAP	DAP	MAP	DAP	MAP
150	active number	2	3	2	7	2	10
	ana. min. host through.	5.23		5.32		5.21	
	num. min. host through.	5.21		5.31		5.17	
	num. overall through.	213.32		267.13		316.54	
100	active number	1	6	1	9	1	12
	min. host through.	5.16		5.21		5,16	
	num. min. host through.	5.09		5.09		5.11	
	num. overall through.	209.44		259.84		307.37	
50	active number	1	9	1	12	1	14
	ana. min. host through.	5.04		5.08		5.03	
	num. min. host through.	5.01		5.03		5.01	
	num. overall through.	201.55		257.41		302.12	

6.3.2 Simulations Under Various Parameter Values

Then, we evaluate the effectiveness of the AP configuration algorithm under different values of the *MAP speed* and the *wall effect* for the algorithm. Since a MAP uses a cellular network to connect with the Internet, the access speed to the Internet can be different depending on the adopted data plan and the network environment such as the distance from the connecting base station and the number of users at the station. Thus, we evaluate the performance of the algorithm under different MAP speeds.

When there exists a wall between an AP and a host, the link speed drops significantly. Then, the overall performance of the network degrades and may not meet the throughput requirement. *MAPs* are again introduced into the network to increase the available bandwidth of the network. In [19], we found that the link speed can be dropped by about 15% when the signal passes through one 5 *inch* concrete wall in a room, and about 30% if the wall is 10 *inch*.

For this evaluation, we consider the network topology which models the second floor of Science Building in Kabi Nazrul University, Bangladesh. There are six rooms with two different sizes, $7m \times 6m$ and $3.5m \times 6m$. We allocated three DAPs in this field. Any DAP is connected to the Internet via a wired cable, and the MAP accesses the Internet by the cellular mobile network. We consider the cases with the number of hosts H = 40, 50, and 60, the minimum host throughput $G = 5$, the drop rate of the speed at a wall is 0% to 30%, and the data plan for MAPs 12, 18 and $24Mbps$ [1].

6.3.2.1 Solution Example

To illustrate this network topology, Figure 6.2 shows one solution found by the algorithm for $H = 60$, $G = 5$, and data plan for MAP = $24Mbps$ with 2 active DAPs and 8 active MAPs, which is presented in Table 6.5. Again, we observe that the DAP has the largest number of associated hosts, while the MAPs have the least number of associated hosts.

[1] To evaluate the additional MAP activation phase, any VAP is not used in this instance.

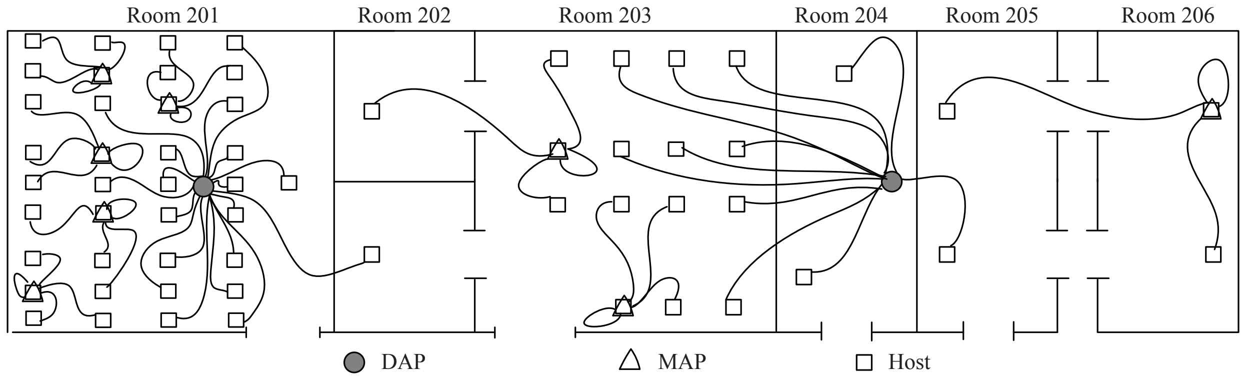


Figure 6.2: Solution for bandwidth limitation with mobile data plans scenario with $H = 60$, $G = 5$, data plan = $24Mbps$, DAP=2, and MAP=8 in Kabi Nazrul University instance.

6.3.2.2 Results for MAP Speed Changes

Here, we consider changes of the *MAP speed* for different data plans for MAPs. The number of hosts $H = 60$ and the minimum host throughput $G = 5$ are used. Table 6.5 summarizes the results for this scenario. This table indicates that when the MAP speed in the data plan increases from $12Mbps$ to $24Mbps$, our algorithm reduces the number of active MAPs.

Table 6.5: Simulation results for MAP speed changes for $G = 5$.

data plans for MAPs ($Mbps$)	12	18	24
active MAPs	11	9	8
ana. min. host through.	5.09	5.17	5.25
num. min. host through.	5.06	5.21	5.22
num. overall through.	309.43	315.99	318.23

6.3.2.3 Results for Wall Effect Changes

Then, we consider the changes of the *wall effect* for $H = 60$, the minimum host throughput $G = 5$, and the speed drop rate per wall are increasing from 0% to 30%. Table 6.6 summarizes the results for this scenario. This table indicates that as the speed drop rate becomes larger, the algorithm increases the number of active MAPs in the network to satisfy the throughput constraints.

Table 6.6: Simulation results for wall effect changes for $G = 5$

speed drop rate per wall	0%	15%	30%
active MAPs	4	8	11
ana. min. host through.	5.19	5.16	5.12
num. min. host through.	5.18	5.15	5.11
num. overall through.	323.34	318.72	312.67

6.3.3 Comparisons with Greedy Algorithm

Finally, we evaluate the performance of our algorithm by comparing it with other algorithms. To the best of our knowledge, this work is the first one that investigates this issue. Thus, we implemented a simple greedy algorithm for the AP configuration problem by ourselves for the purpose of comparisons:

1. Activate one AP (let AP_i) that satisfies the following three conditions:
   (a) It is inactive.
   (b) Its throughput is the highest among the candidates.
   (c) It can cover the maximum number of uncovered hosts among the candidates.
2. Associate the host with AP_i sequentially that satisfies the following four conditions, until no more host can be associated:
   (a) It is uncovered.
   (b) It can be associated with AP_i.
   (c) Its link speed is the maximum among the candidates.
   (d) The minimum host throughput constraint is satisfied after the association.
3. Terminate the procedure if every host is covered or every AP is activated.
4. Go to 1.

Then, we applied this greedy algorithm to Okayama University instance (1st instance) and KabiNazrul University instance (2nd instance). For 1st instance, we applied the greedy algorithm with the network load increase scenario and for the 2nd instance we applied it with the MAP speed change scenario.

Tables 6.7 and 6.8 show the comparison results respectively. These simulation results show that, except the case of $24 Mbps$ data plan in Table 6.8, the proposed algorithm activates the less number of MAPs compared with the greedy algorithm, which demonstrates the effectiveness in the first priority of our proposal, i.e., the active AP minimization. The second priority in this work is the optimization of the minimum host throughput. As demonstrated by the numerical experiments, for all the cases, the proposed algorithm delivers the higher minimum host throughput than the greedy algorithm, since the proposed algorithm yields the better host associations.

Note that for the case of $18 Mbps$ data plan, the analytical minimum host throughput of the greedy algorithm is slightly higher ($20 Kbps$). This is because that, the analytical calculation in Eq. (4.10) assumes all the hosts transmit the same amount of data, which results in several hundred $Kbps$ estimation errors as shown in Table 6.7 and Table 6.8. By far, the difference of $20 Kbps$ is due to this approximation error and hence can be ignored.

6.4 Evaluation of Channel Assignment Extension

In this section, we evaluate the channel assignment extension to the AP configuration algorithm in Chapter 5 through simulations in two network topologies using the WIMNET simulator [47]. Here, we describe the simulation platform with adopted parameters for the WIMNET simulator, the simulated network topologies, and the simulation results.

Table 6.7: Comparison results between the proposed algorithm and the greedy algorithm for Okayama University instance: network load increase scenario with DAP=2, VAP=10, $G = 5$, and $B^a = \infty$.

Number of Hosts ($Mbps$)	40		50		60	
Comparison	Greedy	Proposed	Greedy	Proposed	Greedy	Proposed
active VAPs	2	1	3	2	4	3
ana. min. host through.	5.03	5.97	5.11	5.53	5.10	5.51
num. min. host through.	5.02	6.54	5.14	5.79	5.00	5.75
num. overall through.	209.72	245.47	262.14	289.63	304.93	345.39

Table 6.8: Comparison results between the proposed algorithm and the greedy algorithm for Kabi-Nazrul University instance: MAP speed change scenario with DAP=2, $H = 60$ and $G = 5$, and $B^a = \infty$.

Data plans for MAPs ($Mbps$)	12		18		24	
Comparison	Greedy	Proposed	Greedy	Proposed	Greedy	Proposed
active MAPs	15	11	10	9	8	8
ana. min. host through.	5.03	5.09	5.19	5.17	5.20	5.25
num. min. host through.	4.81	5.06	4.95	5.21	4.75	5.22
num. overall through.	299.27	309.43	314.11	315.99	303.52	318.23

6.4.1 Network Topologies

As the first topology, the *random topology* in Figure 6.3 is considered, where 30 hosts and 8 DAPs are distributed in a 400m × 200m rectangular area. The circles and squares represent the APs and hosts respectively. The minimum host throughput constraint $G = 5$ and the bandwidth limit constraint $B^a = \infty$ are examined. For simplicity, VAPs and MAPs are not used in this topology.

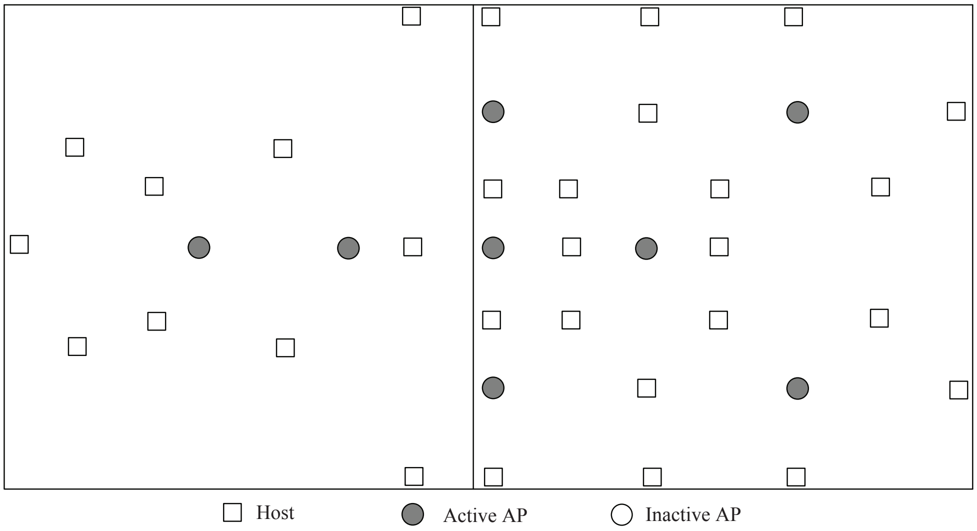


Figure 6.3: Random topolgy with 30 hosts and 8 DAPs.

Then, as the second topology, the *regular topology* in Figure 6.4 is considered, which basically models the third floor of Engineering Building-2 in Okayama University, Japan. There are six rooms with two different sizes, $7m \times 6m$ and $3.5m \times 6m$. Six DAPs and 60 hosts are regularly distributed in the field. The minimum host throughput constraint $G = 5$ and the bandwidth limit constraint $B^a = \infty$ are examined.

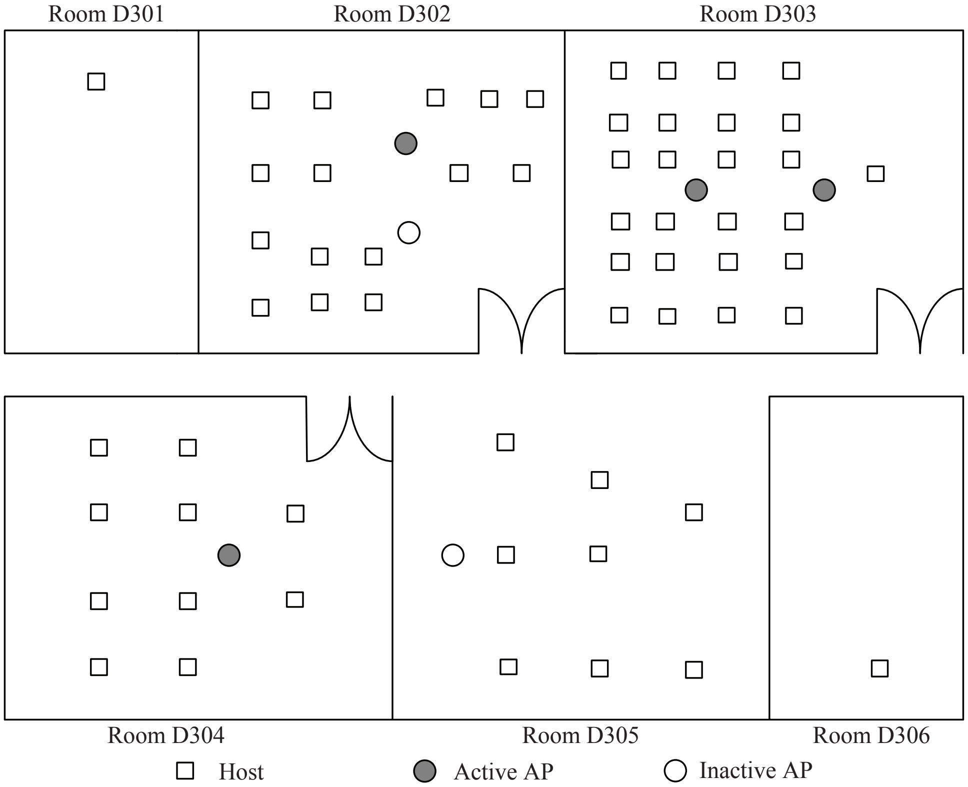


Figure 6.4: Regular topolgy with 60 hosts and 6 DAPs.

6.4.2 Evaluation of Channel Assignment Algorithm

Firstly, the channel assignment algorithm in Section 5.4 is evaluated in the two topologies with two, three, and four channels. Here, the throughput results by the random channel assignment and the algorithm assignment without using SA are also obtained through simulations, in order to compare them with the proposed channel assignment extension. To avoid the bias in the random channel assignment, 20 different channel assignments are generated by using different random numbers, and their average results are used in evaluations. Tables 6.9 and 6.10 show the minimum host and overall throughput results in the three cases for the random topology and the regular topology respectively. They indicate that the greedy initial solution in our algorithm provides better results than the random assignment, and SA can further improve them by reducing interferences. With four channels, the greedy initial solution has no interference, which cannot be improved by SA.

Figure 6.5 compares the channel assignment results before and after applying SA, where different channels are represented by different shapes for APs. They indicate that in the initial solution of the algorithm, the same channel is assigned to AP#6, AP#7, and AP#8, which causes interferences, and it is resolved by applying SA.

Table 6.9: Throughput comparisons in three cases of channel assignment for random topology.

# of channels	2			3			4		
case	random	w/o SA	with SA	random	w/o SA	with SA	random	w/o SA	with SA
# of active APs	8	8	8	8	8	8	8	8	8
min. host through.	3.63	3.87	4.73	4.68	5.10	6.61	5.04	6.61	6.61
overall through.	113.76	120.53	147.96	145.06	154.91	201.46	151.74	201.46	201.46

Table 6.10: Throughput comparisons in three cases of channel assignment for regular topology.

# of channels	2			3			4		
case	random	w/o SA	with SA	random	w/o SA	with SA	random	w/o SA	with SA
# of active APs	4	4	4	4	4	4	4	4	4
min. host through.	3.05	3.17	3.27	3.13	3.24	3.4	3.51	6.17	6.17
overall through.	188.27	200.06	201.67	191.88	201.07	205.07	222.61	402.12	402.12

6.4.3 Evaluation of Channel Load Averaging

Secondly, the channel load averaging (CLA) in Section 5.4.5 is evaluated in the two topologies with two, three, and four channels. Tables 6.11 and 6.12 compare the minimum host and overall throughput results between the cases with and without applying CLA for the random topology and the regular topology respectively. They indicate that CLA can further improve the throughput by averaging the loads among the channels, except for two channels case where the loads between two channels are already balanced before SA.

Table 6.11: Throughput comparisons between two cases of channel load averaging for random topology.

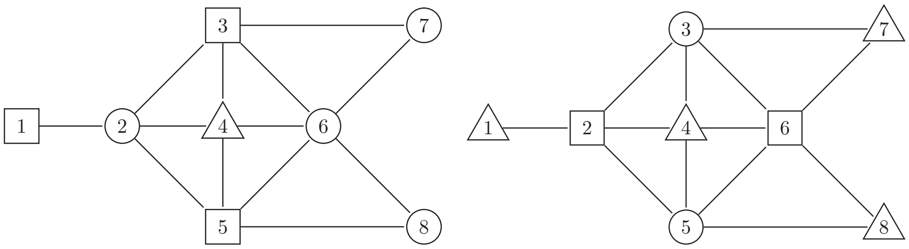
# of channels	2		3		4	
case	w/o CLA	with CLA	w/o CLA	with CLA	w/o CLA	with CLA
# of active APs	8	8	8	8	8	8
min. host through.	4.39	4.39	6.61	6.80	6.61	6.80
overall through.	147.96	147.96	201.46	219.23	201.46	219.23

Table 6.12: Throughput comparisons between two cases of channel load averaging for regular topology.

# of channels	2		3		4	
case	w/o CLA	with CLA	w/o CLA	with CLA	w/o CLA	with CLA
# of active APs	4	4	4	4	4	4
min. host through.	3.27	3.27	3.4	4.5	6.17	6.19
overall through.	201.67	201.67	205.07	283.27	402.12	404.34

6.4.4 Evaluation of Dynamic Network Load Changes

Thirdly, we evaluate the performance of the proposed algorithm when the network load changes dynamically. In simulations, the number of hosts is changed from 20 to 30 for the random topology, and from 40 to 60 for the regular topology respectively. Tables 6.13 and 6.14 summarize the



(a) Before simulated annealing

(b) After simulated annealing

Figure 6.5: Channel assignment results by algorithm for random topology with three channels.

simulation results respectively. They indicate that depending on the number of hosts, the algorithm dynamically optimizes the number of active APs in the network while satisfying the minimum host throughput constraint.

Table 6.13: Simulation results under network load changes for random topology.

# of hosts	10		20		30	
algorithm	proposed	random	proposed	random	proposed	random
# of active APs	3	3	5	5	8	8
min. host through.	5.0	4.01	5.78	4.07	6.80	5.04
overall through.	54.09	42.43	124.78	93.17	219.23.47	151.74

Table 6.14: Simulation results under network load changes for regular topology.

# of hosts	40		50		60	
algorithm	proposed	random	proposed	random	proposed	random
# of active APs	2	2	3	3	4	4
min. host through.	5.09	3.87	5.85	3.84	6.19	3.51
overall through.	209.13	163.73	303.12	209.52	404.34	222.61

6.4.5 Evaluation of Network with Uncontrolled APs

Fourthly, we evaluate the extension for the two network topologies when the number of APs that cannot be deactivated by the algorithm/system is increased from 1 to 4. Here, the APs that cannot be deactivated by the system are called*uncontrolled APs* in this paper. The uncontrolled APs are always active where their channels are assigned randomly. In simulations, the uncontrolled APs are randomly selected, and the average results are examined after 10 trials are repeated for each number of uncontrolled APs to avoid the bias in random selections.

Tables 6.15 and 6.16 summarize the simulation results for the random topology and regular topology respectively. They indicate that the network throughput drops when the number of uncontrolled APs increases in the network.

Table 6.15: Simulation results with uncontrolled APs for random topology.

# of uncontrolled APs	0	1	2	3	4
# of active APs	8	8	8	8	8
min. host through.	6.80	6.43	6.03	5.26	5.14
overall through.	219.23	197.56	189.11	161.79	156.21

Table 6.16: Simulation results with uncontrolled APs for regular topology.

# of uncontrolled APs	0	1	2	3	4
# of active APs	4	4	4	4	4
min. host through.	6.19	6.16	4.94	4.01	3.47
overall through.	404.34	402.12	307.42	282.56	220.51

6.4.6 Comparison with Existing Algorithm

Finally, the performance of the proposed channel assignment algorithm is evaluated through comparisons with the representative existing algorithm called the *ADJ-minmax approach* [55]. In the simulations of both algorithms, the same interference model is adopted for fair comparisons. Tables 6.17 and 6.18 compare the minimum host and overall throughput results between the proposed algorithm and the ADJ-minmax approach for the random topology and the regular topology respectively. They indicate that the proposed algorithm increases the overall throughput by 3% and 1% for the random topology with two and three channels, and by 1% and 38% for the regular topology with two and three channels respectively.

Table 6.17: Throughput comparisons between proposed algorithm and ADJ-minmax for random topology.

# of channels	2		3	
algorithm	proposed	ADJ-minmax	proposed	ADJ-minmax
# of active APs	8	8	8	8
min. host through.	4.7	4.7	6.8	6.8
overall through.	147.97	144.27	219.23	217.28

Table 6.18: Throughput comparisons between proposed algorithm and ADJ-minmax for regular topology.

# of channels	2		3	
algorithm	proposed	ADJ-minmax	proposed	ADJ-minmax
# of active APs	4	4	4	4
min. host through.	3.31	3.30	4.50	3.39
overall through.	201.67	200.30	283.27	204.98

6.5 Summary

In this chapter, we first introduced the WIMNET simulator and network environments for simulations. Then, we evaluated the AP configuration algorithm through simulations in two network scenarios using the WIMNET simulator. After that, we evaluated the channel assignment extension in two different network instances using the same simulator. In next chapter, we will present the implementation of the elastic WLAN system using the AP configuration algorithm and its evaluation using the testbed.

Chapter 7

Elastic WLAN System Implementation Using Raspberry Pi

In this chapter, we present the implementation of the elastic WLAN system and evaluate it using the testbed. First, we provide the background and motivation of the elastic WLAN system, followed by some related works. Then, we present the system topology for the elastic WLAN system and describe the implementation procedures. Finally, we verify the feasibility and performance of the implementation through experiments using the testbed.

7.1 Introduction

In a network, traffic demands can change by the network environments. Therefore, a WLAN system should be adaptive and elastic to the change of the network environments and traffic demands. For this purpose, in [9][13], we proposed the *active AP configuration algorithm* to activate the minimum number of APs in the network while satisfying the constraints of the throughputs offering for the hosts. Unfortunately, we have verified the effectiveness of the proposal through only simulations. To show the effectiveness in real worlds, need to we implement the elastic WLAN system using WLAN devices.

In this chapter, we present the system design of the *elastic WLAN system* and its implementation using*Raspberry Pi* for the AP and Linux PC for the host. *Raspberry Pi* is a small-size low-cost computer, and uses the open-source operating system, Linux. Thus, it has become popular in cost efficient embedded system designs including wireless networks.

In this work, we use several Linux tools and commands to collect necessary information of the system in real time for the algorithm input, and to activate/deactivate APs and the AP-host associations according to the output of the algorithm. Besides, we prepare a testbed, and evaluate the practicality and the performance of the elastic WLAN system testbed in a real network.

7.2 Related Works

In this section, we show our brief surveys to this work.

In [57], Moursy proposed a testbed implementation of an automatic network performance management framework for a mesh network.

In [58], Lei et al. proposed a campus WLAN framework based on the software defined network (SDN) technology.

In [59], Luengo et al. also proposed a design and implementation of a testbed for integrated wireless networks based on SDN. Although this framework is flexible, it requires SDN-enabled devices and network virtualizations.

In [60], Ahmed et al. describe the important design issues in preparing a large-scale WLAN testbed for evaluations of centralized control algorithms and presented experimental results. They did not analyze the power-saving and adaptive control mechanism of centralized WLANs, which is one of the main purposes of our research.

In [61], Debele et al. proposed a Resource-on-Demand (RoD) strategy for energy-saving in dense WLANs where they analyzed user behaviors in the network and formulated the stochastic characteristics. Our system also adapts with the user demands. Moreover, we present a implementation of our system in real networks.

7.3 System Topology

Figure 7.1 shows a simple network topology of the elastic WLAN system including the management server. This server has the administrative access to all the devices in the network. It collects the necessary information to the inputs of the AP configuration algorithm, executes the algorithm, and controls the activations/deactivations of the APs and the associations of the hosts according to the algorithm output. In this implementation, *Raspberry Pi* is used for the AP and a Linux laptop PC is for the host and the management server.

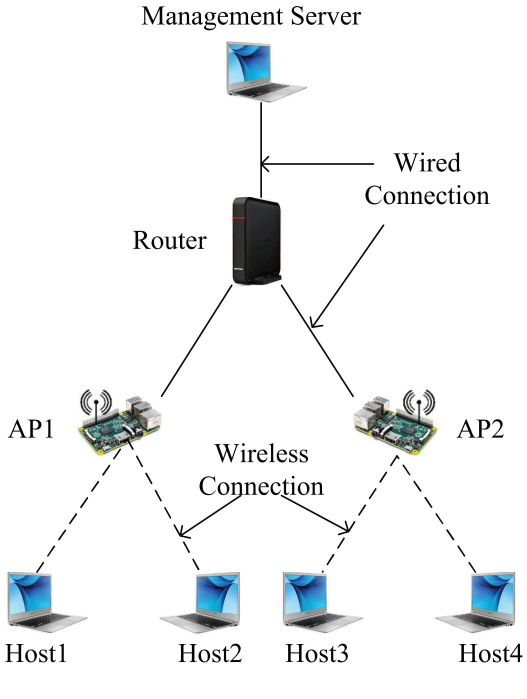


Figure 7.1: Elastic WLAN system topology.

7.4 Implementation Procedures

In this section, we describe the implementation of the elastic WLAN system using*Raspberry Pi*.The implementation for the channel assignment extension with the load averaging is also presented. Figure 7.2 shows the execution procedure for the elastic WLAN system implementation.

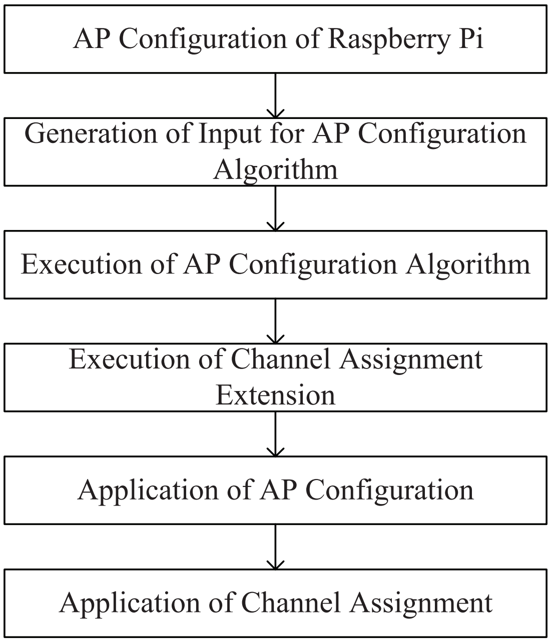

AP Configuration of Raspberry Pi
Generation of Input for AP Configuration Algorithm
Execution of AP Configuration Algorithm
Execution of Channel Assignment Extension
Application of AP Configuration
Application of Channel Assignment

Figure 7.2: Elastic WLAN system execution flow.

7.4.1 AP Configuration of *Raspberry Pi*

First, *Raspberry Pi*is set up as the AP by using *host access point daemon (hostapd)* [25][26]. *hostapd* is capable of turning network interface cards (NICs) into access points (APs) by the following steps:

1. Install the hostapd using the following command:

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

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

```
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 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
   ```

7.4.2 Generation of Input for AP Configuration Algorithm

The management server explores all the connected devices in the network, and generates the input data for the AP configuration algorithm by the following three steps:

1. The Linux tool *arp-scan* [20] is used to scan the network. The command is given below:

   ```
   $ sudo arp-scan --interface=eth0 192.168.11.0/24
   ```

   Here, eth0 represents the network device and 192.168.11.0/24 does the range of the IP addresses to scan. The output from the command consists of the IP and MAC addresses of the hosts and the APs available in the network. A simple C program was developed to identify the hosts and APs in this system using the MAC addresses of the devices.

   After this step, the server generates the following lists:

   - List of permitted APs.
   - List of permitted hosts.

2. The Linux command *nm-tool* [22][23] is executed remotely from the server in each host to find the receiving signal strength from each AP. The *ssh* [28][29] protocol is used for the remote procedure call. The command is given below:

   ```
   $ sudo nm-tool
   ```

   After this step, the server has the following information:

   - Currently associated AP.
   - List of the associable active APs.
   - Receiving signal strength from each associable AP.

3. The signal strength is converted to the estimated link speed using the sigmoidal function in [34]. Then, the collected information is used to generate the input file for the AP configuration algorithm.

7.4.3 Execution of AP Configuration Algorithm

After the input file is generated, the *minimum host throughput constraint* and the *bandwidth limitation constraint* are specified by the user. Then, the AP configuration algorithm is run with a given number of available channels using the following bash script:

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

In this script, *input.txt* is 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 it, the list of active APs and their association with the hosts is obtained from the algorithm.

7.4.4 Execution of Channel Assignment Extension

Then, the channel assignment extension is applied using the following bash script:

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

In this script, *HostAPassociation.txt* is the input file for the channel assignment extension that contains the list of active APs and their associations with the hosts that are derived in the previous step, and *num_of_channels* does the number of available channels.

7.4.5 Application of AP Configuration

The management server applies the algorithm output to the APs and hosts in the network by activating or deactivating APs and changing the host associations.

1. Activation/Deactivation of APs: The AP-mode of *Raspberry Pi* can be activated or deactivated by the following commands:

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

2. It is necessary to change the associations of the hosts to the active APs that are suggested by the AP configuration algorithm. The following Linux command *nmcli* [31][32] is adopted to change the association of a host and an AP:

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

   Here *NewSSID* represents the new AP for the host and *PASSWORD* does the security key of the AP.

7.4.6 Application of Channel Assignment

The management server assigns the specified channel in the algorithm output to each AP. To assign a channel in *Raspberry Pi*, the configuration file */etc/hostapd/hostapd.conf* has to be modified with this channel number. After this modification, the *hostapad* process must be restarted to make the change take effect. The Linux command *sed*[62] is used for the purpose. Then, the required commands are given below:

```
$ 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.

7.5 Evaluation by Testbed

In this section, we evaluate the elastic WLAN system testbed using the proposed AP configuration algorithm, using two network scenarios.

7.5.1 Network Scenarios

For evaluations, two network scenarios for the elastic WLAN system testbed, namely, the 3 × 3 scenario and the 5 × 5 scenario, are prepared in our building.

7.5.1.1 3 × 3 Scenario

In this scenario, three *Raspberry Pi* devices for APs and three Linux PCs for hosts are prepared in a 7m × 6m room. As shown in Figure 7.3, any AP is located 1m away from its neighbor APs, and any host is 1m away from its associated AP.

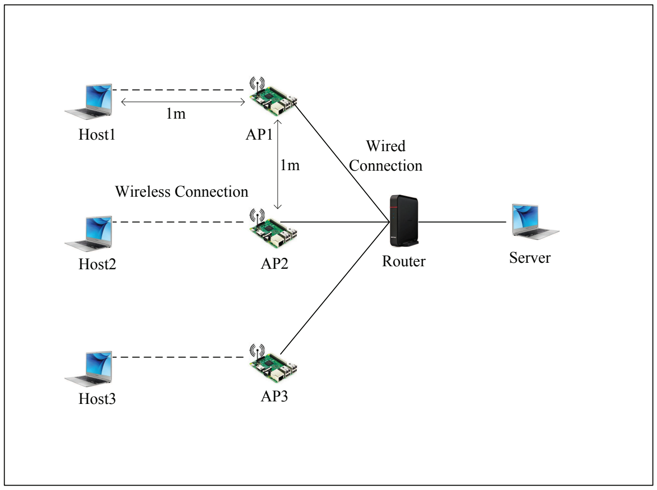


Figure 7.3: Testbed for 3 × 3 scenario.

7.5.1.2 5 × 5 Scenario

In this scenario, three rooms separated by walls are used, where two rooms have the size of $7m \times 6m$ and one room has the size of $3.5m \times 6m$. Five *Raspberry Pi* devices for APs and five Linux PCs for hosts are distributed in the rooms as shown in Figure 7.4. Any Linux PC is located very close to the associated AP where the distance is less than 50cm. Any AP is $4m$ away from another AP in the same room. Because the signals from the APs in different rooms become small by the walls. Thus, the interferences among the APs can be reduced.

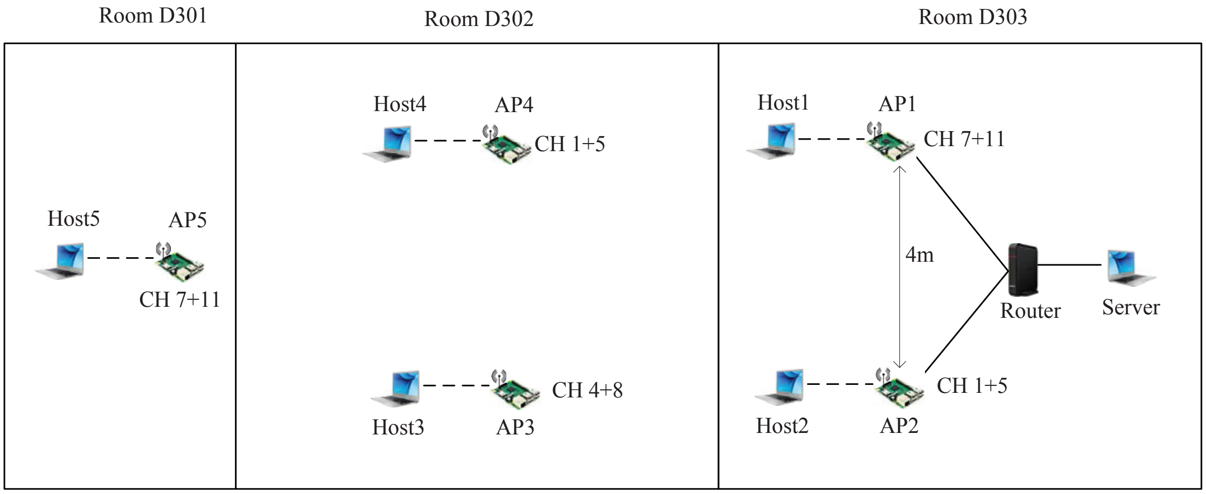


Figure 7.4: Testbed for 5 × 5 scenario.

7.5.2 Single *Raspberry Pi* AP Throughput Measurement

First, the throughput performance of the single *Raspberry Pi* AP with the associated Linux PC is measured in the same building for reference, using TCP traffics with *iperf* [35]. The average measured throughput is 42 Mbps. It is noted that all the experiments were conducted on weekends to reduce the interferences from other APs in the same building.

7.5.3 Throughput Results in Testbed

To evaluate the effect of the proposed channel assignment extension in the elastic WLAN system implementation, throughputs are measured with and without applying it using the elastic WLAN system testbed. Here, the three bonded channels of 1+5 (channel 1 and channel 5), 4+8, and 7+11 are assigned to the APs by the algorithm, where the channel bonding is adopted for 11n. On the other hands, without the algorithm, one of the three channels is randomly assigned to each AP. Any throughput is measured using *iperf* when all the hosts are communicating in parallel with the server through their connected APs. Besides, the effect of the AP configuration algorithm is evaluated with the channel assignment extension.

7.5.3.1 3 × 3 Scenario Result

Table 7.1 shows the throughput results in the testbed for the 3×3 scenario. When the number of active APs is not minimized, the overall throughput by the random and algorithm channel assignment is 47.56 Mbps and 68.10 Mbps respectively. Then, the number of active APs is minimized to one by the AP configuration algorithm, and the overall throughput becomes 32.62 Mbps, 34.63 Mbps, and 33.56 Mbps respectively, when the single bonded channel of 1+5, 4+8, or 7+11 is assigned to the single AP. Thus, the channel assignment algorithm can improve the throughput performance of the small testbed when multiple APs are active.

Table 7.1: Throughput comparisons between proposed algorithm and random assignment for 3×3 scenario.

AP minimization	no		yes		
channel assignment	random	proposed	1+5	4+8	7+11
# of active APs	3	3	1	1	1
min. host through.	13.2	18.7	9.87	10.21	10.16
overall through.	47.56	68.10	32.62	34.63	33.56

7.5.3.2 5×5 Scenario Result

Table 7.2 shows throughput results in the testbed for the 5×5 scenario. When the number of active APs is not minimized, the overall throughput for the random and proposed channel assignment is 121.11 Mbps and 151.32 Mbps respectively. Then, the number of active APs is minimized to three by the AP configuration algorithm, and the overall throughput for the random and proposed channel assignment becomes 92.05 Mbps and 118.84 Mbps respectively. Thus, the proposed channel assignment algorithm can greatly improve the throughput performance of the larger testbed, while the active AP configuration algorithm can efficiently reduce the number of active APs with keeping the throughput performance.

Table 7.2: Throughput comparisons between proposed algorithm and random assignment for 5×5 scenario.

AP minimization	no		yes	
channel assignment	random	proposed	random	proposed
# of active APs	5	5	3	3
min. host through.	19.32	27.2	14.61	19.21
overall through.	121.11	151.32	92.05	118.84

7.6 Summary

In this chapter, we presented the system design of the elastic WLAN system and its implementation using *Raspberry Pi*. The implemented elastic WLAN system executes the AP configuration algorithm to optimize the active APs and host associations based on network traffics. It also optimizes the channels of the active APs to minimize the interference in the network. In the next chapter, we will conclude this dissertation with some future recommendations.

Chapter 8

Conclusion

In this thesis, we presented the studies of the *access-point (AP) configuration algorithm* for the *elastic WLAN system*, and the design and implementation of the system using *Raspberry Pi*.

Firstly, we introduced the related wireless network technologies including the IEEE 802.11n protocol, heterogeneous WLAN devices, the link speed estimation model for IEEE 802.11n protocol, and Linux tools and commands for WLANs that are used for the implementation of elastic WLAN system.

Secondly, we introduced the motivation of the elastic WLAN system, the challenges of its implementation with possible solutions, and the system design and operational flow.

Thirdly, we proposed the AP configuration algorithm for the elastic WLAN system using heterogeneous devices. We formulated the AP configuration problem as a combinatorial optimization problem and proved the *NP-completeness* of its decision version, and proposed its heuristic algorithm composed of seven phases. The proposed algorithm dynamically optimizes the number of active APs in the network field based on communication demands. Three types of AP devices, namely DAPs, VAPs, and MAPs, with their speed differences are considered in this algorithm. The algorithm can ensure the network performance under the minimum host throughput constraint and the use of heterogeneous APs.

Fourthly, we presented the channel assignment extension of the AP configuration algorithm. One channel is assigned to every active AP from the limited number of the non-interfered channels to minimize the overall interference in the network. Besides, AP associations of hosts are modified to improve the network performance by averaging the loads among the channels.

Fifthly, we evaluated the effectiveness of the AP configuration algorithm and its extension through numerical experiments in different network instances using the WIMNET simulator. We first introduced the WIMNET simulator with the parameters for simulations, and the network environments and topologies for simulations. They showed the simulation results to verify the effectiveness of the proposal.

Finally, we introduced the design and the implementation of the elastic WLAN system using real devices. We adopted Raspberry Pi for the AP and Linux PCs for the hosts. The system uses several Linux tools and commands to collect the necessary information for the AP configuration algorithm input, and to automatically change the activation/deactivation of APs and the AP-host associations according to the algorithm output. Then, we evaluated the practicality and the performance of the elastic WLAN system in real networks using the testbed.

In future works, we will study the joint optimization of the AP configuration and the channel assignment, and improve the link speed estimation model through further extensive measurements in different conditions and environments. We will also improve the elastic WLAN system to consider the existence of uncontrollable active APs from neighboring network, to support heterogeneous operating system platforms, and investigate the performance evaluations in more practical scenarios.

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] S. Lanzisera, B. Nordman, and R. E. Brown, “Data network equipment energy use and savings potential in buildings,” Springer J. Energy Efficiency, vol. 5, no. 2, pp. 149-162, May 2012.

[4] F. Nadeem, E. Leitgeb, M. S. Awan, and S. Chessa, “Comparing the life time of terrestrial wireless sensor networks by employing hybrid FSO/RF and only RF access networks,” Proc. 5th Int. Conf. Wireless. Mobile Commun. (ICWMC-2009), pp.134-139, 2009.

[5] “Electricity sector in Bangladesh,” Internet: `https://en.wikipedia.org/wiki/Electricity_sector_in_Bangladesh`, Access Jan. 20, 2017.

[6] “Electricity to transform rural Myanmar,” Internet: `http://www.worldbank.org/en/news/feature/2015/09/16/electricity-to-transform-rural-myanmar`, Access Jan. 20, 2017.

[7] M. S. A. Mamun, M. E. Islam, and N. Funabiki, “An active access-point configuration algorithm for wireless local area networks with heterogeneous devices,” IEICE Tech. Rep., vol. IEICE-115, no. 159, pp. 147-152, July 2015.

[8] M. S. A. Mamun, M. E. Islam, N. Funabiki, and I.-W. Lai “Active access-point configuration algorithm with dynamic mobile router placement for elastic WLAN system,” Int. Symp. Computing and Networking, pp. 246-252, Dec. 2015.

[9] 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, Internet: `http://www.ijnc.org/index.php/ijnc/article/view/134`.

[10] M. R. Garey and D. S. Johnson, “Computers and intractability: A guide to the theory of NP-completeness,” Freeman, New York, 1979.

[11] M. Elwekeil, M. Alghoniemy, M. El-Khamy, H. Furukawa, and O. Muta, “Optimal channel assignment for IEEE 802.11 Multi-cell WLANs,” Signal Processing Conf. (EUSIPCO), Proc. 20th European, pp. 694-698, Aug. 2012.

[12] M. S. A. Mamun, S. K. Debnath, K. S. Lwin, and N. Funabiki, "A channel assignment extension of active access-point configuration algorithm for elastic WLAN system under limited channels," Int. Symp. Computing and Networking, pp. 318-324, Nov. 2016.

[13] 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, Internet: http://www.ijnc.org/index.php/ijnc/article/view/153.

[14] "Ultimate guide to Raspberry Pi," Comp. Shop., vol. 324, pp. 104-119, Feb. 2015, Internet: http://micklord.com/foru/Raspberry%20Pi%20Pages%20from%20Computer%20Shopper%202015-02.pdf, Access Jan. 20, 2017.

[15] M. Vipin, and S. Srikanth, "Analysis of open source drivers for IEEE 802.11 WLANs," Proc. Int. Conf. Wireless Comm. Sensor Comput. (ICWCSC 2010), pp. 1-5, IEEE, Jan. 2010.

[16] IEEE, "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," Internet: http://standards.ieee.org/getieee802/download/802.11-2012.pdf, Access Jan. 20, 2017.

[17] "IEEE 802.11n-2009 - Wikipedia," Internet: https://en.wikipedia.org/wiki/IEEE_802.11n-2009, Access Jan. 20, 2017.

[18] T.-D. Chiueh, P.-Y. Tsai, and I.-W. Lai, "Baseband receiver design for wireless MIMO-OFDM communications," 2nd ed. Wiley-IEEE Press, 2012.

[19] N. Funabiki, W. Maruyama, and T. Nakanishi, "An extension of routing tree algorithm considering link speed change in IEEE 802.11n protocol for wireless mesh network," Proc. Int. Conf. Ubiq. Intel. Comput. Auto. Trusted Comput., pp.600-605, Sept. 2012.

[20] "arp-scan user guide," Internet: http://www.nta-monitor.com/wiki/index.php/Arp-scan_User_Guide, Access Jan. 20, 2017.

[21] "arp-scan," Internet: http://www.nta-monitor.com/tools-resources/security-tools/arp-scan, Access Jan. 20, 2017.

[22] "nm-tool," Internet: http://linux.die.net/man/1/nm-tool, Access Jan. 20, 2017.

[23] "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, 2017.

[24] "Network manager," Internet: https://help.ubuntu.com/community/NetworkManager, Access Jan. 20, 2017.

[25] "Hostapd: the linux way to create virtual Wifi access point," Internet: https://nims11.wordpress.com/2012/04/27/hostapd-the-linux-way-to-create-virtual-wifi-access-point/, Access Jan. 20, 2017.

[26] "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, 2017.

[27] J. Malinen, "hostapd: IEEE 802.11 AP, IEEE 802.1 X. WPA/WPA2/EAP/RADIUS authenticator," Internet: https://w1.fi/hostapd/, Access Jan. 20, 2017.

[28] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, "SSH, the secure shell: the definitive guide," O'Reilly Media Inc., 2005.

[29] T. Ylonen and C. Lonvick, "The secure shell (SSH) protocol architecture," 2006.

[30] "OpenSSH," Internet: http://www.openssh.com/, Access Jan. 20, 2017.

[31] "nmcli - command-line tool for controlling NetworkManager," Internet: http://manpages.ubuntu.com/manpages/maverick/man1/nmcli.1.html, Access Jan. 20, 2017.

[32] "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, 2017.

[33] J. Tourrilhes, "iwconfig," Internet: http://www.linuxcommand.org/man_pages/iwconfig8.html, Access Jan. 20, 2017.

[34] M. E. Islam, K. S. Lwin, M. S. A. Mamun, N. Funabiki, and I. Lai, "Measurement results of three indices for IEEE802.11n wireless networks in outdoor environments," 17th IEEE Hiroshima Section Student Symposium, pp. 410-414, Nov. 2015.

[35] "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool," Internet: https://iperf.fr/, Access Jan. 20, 2017.

[36] G. F. Lyon, "Nmap network scanning: The official nmap project guide to network discovery and security scanning," Insecure, USA, Jan. 2009.

[37] M. Wolfgang, "Host discovery with nmap," Exploring nmap's default behavior, vol. 1, pp. 1-16, 2002.

[38] N. Sofer, "Wireless network watcher - Show who is connected to your wireless network," Internet: http://www.nirsoft.net/utils/wireless_network_watcher.html, Access Jan. 20, 2017.

[39] J. Tourrilhes, J. Morgenstern, and G. Renker, "wavemon," Internet: http://manpages.ubuntu.com/manpages/zesty/man1/wavemon.1.html, Access Jan. 20, 2017.

[40] "iw : documentation," Internet: https://wireless.wiki.kernel.org/en/users/documentation/iw, Access Jan. 20, 2017.

[41] H. P. Gupta and S. V. Rao, "Demand-based coverage and connectivity-preserving routing in wireless sensor networks," IEEE Syst. J., vol.10, no.4, pp.1380-1389, July 2014.

[42] M. Kobayashi, S. Haruyama, R. Kohno, and M. Nakagawa, “Optimal access point placement in simultaneous broadcast OFDM for public access wireless LAN,” IEICE Trans. Commun., vol.E85-B, no.10, pp.1978-1986, Oct. 2012.

[43] X. Ling and K. L. Yeung, “Joint access point placement and channel assignment for 802.11 wireless LANs,” IEEE Trans. Wireless Commun., vol.5, no.10, pp.2705-2711, Oct. 2006.

[44] H. Shindo, T. Nagai, and H. Shigeno, “Virtual access point allocation method for high density WLANs,” Proc. 6th Int. Conf. Mobile Comput. Ubiq. Netw. (ICMU-2012), pp.69-76, May 2012.

[45] F. Ganji, L. Budzisz, and A. Wolisz, “Assessment of the power saving potential in dense enterprise WLANs,” Proc. Int. Symp. Pers. Indoor. Mob. Radio Commun., pp.2835-2840, Sept. 2013.

[46] S. Miyata, T. Murase, and K. Yamaoka, “Novel access-point selection for user QoS and system optimization based on user cooperative moving,” IEICE trans. Commun. vol.E95-B, no.6, pp.1953-1964, June 2012.

[47] N. Funabiki ed., “Wireless mesh networks,” InTech-Open Access Pub., Jan. 2011, online available at: `http://www.intechopen.com/books/wireless-mesh-networks`.

[48] T. Farag, N. Funabiki, and T. Nakanishi, “An access point allocation algorithm for indoor environments in wireless mesh networks,” IEICE Trans. Commun., vol.E92-B, no.3, pp.784-793, Mar. 2009.

[49] M. R. Garey and D. S. Johnson, “Computers and intractability: a guide to the theory of NP-completeness,” W. H. Freeman & Co (Sd), New York, Jan. 1979.

[50] R. M. Karp, “Reducibility among combinatorial problems,” Springer US, pp. 85-103, 1972.

[51] 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.

[52] D. P. Williamson and D. B. Shmoys, “The design of approximation algorithms,” Cambridge Univ. Press, Apr. 2011.

[53] J. Geier, “Assigning 802.11b access point channels,” WiFi Planet, Feb. 2002.

[54] H. Skalli, S. K. Das, L. Lenzini, and M. Conti, “Traffic and interference aware channel assignment for multi-radio wireless mesh networks,” Proc. of ACM Int. Conf. Mobile Comput. Netw (MOBICOM), pp. 15-26, Dec. 2006.

[55] K. Zhou, X. Jia, Y. Chang, and X. Tang, “Partially overlapping channel assignment for WLANs using SINR interference model,” Int. J. Commun. Sys. vol. 27, no. 11, pp. 3082-3095, Nov. 2014.

[56] D. Gong, M. Zhao, and Y. Yang, “Channel assignment in multi-rate 802.11n WLANs,” Wireless Commun. Netw. Conf., pp. 392-397, Apr. 2013.

[57] A. Moursy, A. Aly, B. Xu, D. Perkins, and M. Bayoumi, “Testbed implementation for autonomic network performance management of wireless mesh networks,” Globecom Workshops, pp. 903-907, Dec., 2012.

[58] T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang, "SWAN: An SDN based campus WLAN framework," Proc. Int. Conf. Wireless Commun., Vehi. Technol., Inform. Theory., Aerospace & Electronic Systems (VITAE), pp. 1-5, IEEE, May 2014.

[59] E. Luengo, C. Weber, S. S. Heydari, and K. El-Khatib, "Design and implementation of a software-defined integrated wired-wireless network Testbed," Proc. 13th ACM Int. Symp. Mobility Manage. Wireless Access, pp 47-52, ACM, Nov. 2015.

[60] N. Ahmed and U. Ismail, "Designing a high performance wlan testbed for centralized control," Proc. The 5th Int. Conf. Testbeds and Research Infr. for the Dev. of Net. & Communities and Work., pp. 1-6, Apr. 2009.

[61] F. G. Debele, M. Meo, D. Renga, M. Ricca, and Y. Zhang, "Designing resource-on-demand strategies for dense WLANs," IEEE J. Selected Area. Commun., vol. 33, no. 12, pp. 2494-2509, Dec. 2015.

[62] "Learning Linux Commands: sed," Internet: `https://linuxconfig.org/learning-linux-commands-sed`, Access Jan. 20, 2017.