International Journal of Networking and Computing – `www.ijnc.org`
ISSN 2185-2839 (print) ISSN 2185-2847 (online)
Volume 6, Number 2, pages 395–419, July 2016

An Active Access-Point Configuration Algorithm for Elastic Wireless Local-Area Network System Using Heterogeneous Devices

Md. Selim Al Mamun, Md. Ezharul Islam, Nobuo Funabiki, Minoru Kuribayashi
Department of Electrical and Communication Engineering, Okayama University
3-1-1 Tsushimanaka, Okayama 700-8530, Japan

I-Wei Lai
Department of Electronic Engineering, Chang Gung University
No. 259 Wen-Hwa 1st Road, Guishan Dist., Taoyuan, Taiwan, 333, R.O.C.

Received: February 15, 2016
Revised: June 6, 2016
Accepted: July 4 2016
Communicated by Tomoyuki Ohta

Abstract

An *Elastic Wireless Local-Area Network (WLAN)* system provides a reliable, flexible, and efficient Internet access to users through installations of heterogeneous access points (APs) including dedicated APs (DAPs), virtual APs (VAPs), and mobile APs (MAPs). The number of APs should be carefully selected to optimize the network performance. Specifically, for heavy traffic, a large number of APs are required. However, the dense deployment of APs introduces the inter-AP interferences which may eventually degrade the communication quality when the number of users are few. In this paper, we propose an *active access-point configuration algorithm* that activates or deactivates APs according to the changes of network topologies and demands of users for the elastic WLAN system. The algorithm considers the bandwidth difference among heterogeneous AP devices and the total available bandwidth in the network. The number of active APs is minimized to ensure the minimum inter-AP interference subject to the constraints. The host locations can be the candidate positions for the MAPs, because host owners may use them for the Internet access. The effectiveness of the proposed algorithm is demonstrated using the WIMNET simulator.

Keywords: Elastic wireless local-area network system, Access point configuration algorithm, Dedicated access point, Virtual access point, Mobile access point

1 Introduction

The IEEE 802.11 wireless local area networks (WLANs) have been deployed everywhere for the Internet access [1]. 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 access points 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 deteriorate the communication performance [2]. Besides, the configuration of these APs should be arranged according to the traffic demands and the network topologies.

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 users or students increases in the afternoon during weekdays, while it decreases in the morning or evening, and 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 [3].

Developing countries like Bangladesh and Myanmar suffer from unreliable and slow Internet access. The major reason for this is *load shedding*, i.e. the fluctuation and the discontinuity of electricity supply for the time being. In Bangldesh and Myanmar, only 60% and 30% of people use electricity respectively [4, 5] and suffer *load shedding* frequently, because the production of power is comparatively less than the growing demand. The power authority stops the supply to a certain area to give the flow of electricity to other areas. This problem is dominant when the extra energy is required, e.g. air conditions in summer. The irregular flow of electricity often causes damages to network devices. In such uncertain cases, only some APs are given 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. Internet cables are often cut off at construction sites. Network devices and Internet cables can be stolen to hamper the continuity of Internet access. Sometimes, Internet service providers restrict the bandwidth of clients for their own selfish ends. All the reasons mentioned above decrease the available bandwidth from its expected value.

As the usage fee is continuously falling down with time, the Internet access through 3G/4G mobile networks is increasing in these countries at recent time. Currently, so many people are using the Internet through mobile network devices such as mobile routers, pocket routers, and smartphones. Besides, it has become common for them to share the use of the Internet 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 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 such circumstances, we have studied the elastic WLAN system for burning demands in developing countries. The elastic WLAN system allows dynamic number of active APs according to the network environment. The more traffic, the more active APs are required. Figure 1 shows the topology of a simple elastic WLAN system with heterogeneous AP devices.

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

A host represents a device capable of connecting to the Internet through WLAN interface, such as a personal computer (PC) or a smart phone. However, in our current implementation of the elastic WLAN system, only a PC using Linux can be used for a host to allow the use of necessary Linux commands there. We note that a PC using Linux can be used for the management server. Through numerical experiments in different network environment and network instances, we demonstrate the effectiveness of proposed algorithm. The throughput of the resulting WLAN system is evaluated by using the *WIMNET* simulator. The experimental results show the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows: Section 2 describes the related works and technologies. Section 3 describes the elastic WLAN system with heterogeneous AP devices. Section 4 formulates the AP configuration problem for the elastic WLAN system. Section 5 describes the active AP configuration algorithm with limited MAP placement. Section 6 evaluates the proposal through simulations. Finally, Section 7 concludes this paper with some future works.

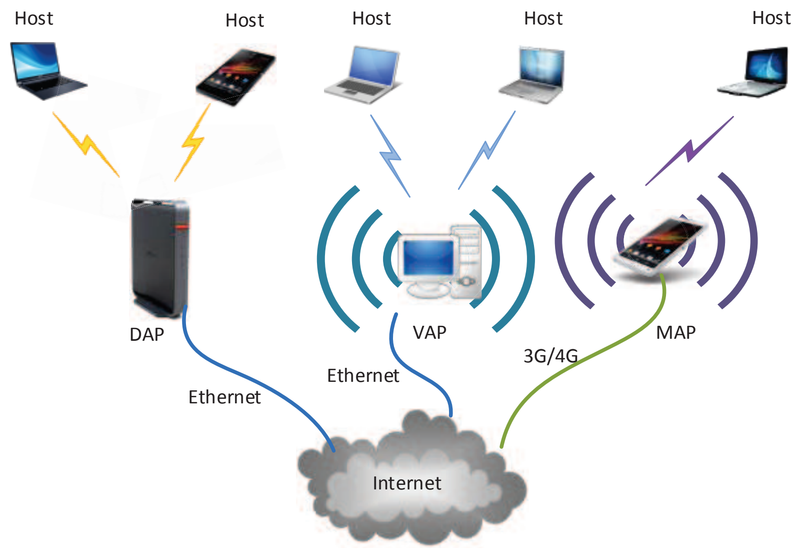

Figure 1: Illustration of elastic WLAN with heterogeneous devices.

2 Related Works and Technologies

In this section, we briefly describe the related works and technologies for the elastic WLAN system and the active AP configuration algorithm.

2.1 Heterogeneous AP Devices

First, we introduce three different types of APs that we use in our AP configuration algorithm. We also discuss the characteristics and the speed difference of these APs.

A DAP is a wireless AP that adopts the IEEE 802.11n wireless protocol and connects PCs to the Internet. A commercial DAP using IEEE 802.11n 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. Several Internet connection mediums including wired, wireless, or cellular communications are available for the VAP. A network device can connect to a VAP the same way as it does to a conventional DAP. Most VAPs support the IEEE 802.11n protocol [6] with a maximum of $54Mbps$ transmissions.

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

2.2 Features of IEEE 802.11n

The IEEE 802.11 protocol is a standard created by the IEEE 802 LAN/MAN (Local Area Network/Metropolitan Area Network) Standards Committee. It specifies an over-the-air interface between a wireless client and a base station or between two wireless clients within a local area in either

---

[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.

fixed, portable, or moving stations mode [7]. IEEE 802.11n is an amendment to the IEEE 802.11 2007 wireless networking standard. This standard was introduced with 40 MHz bandwidth channel (channel bonding), MIMO, frame aggregation, and security improvements over its predecessors. Table 1 briefly summarizes the IEEE 802.11n protocol.

Table 1: IEEE 802.11n specification.

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

For the *channel bonding* technique, each channel can double the physical data rate 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 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. In orthogonal frequency division multiplexing (OFDM) encoding, data are carried on several parallel data streams or channels using a large number of closely spaced orthogonal signals. In the *short guard interval*, the guard interval is reduced to 400 ns, which can increase the data rate. In the frame aggregation, multiple frames can be transmitted as one aggregated frame using MAC protocol data unit (MPDU), which reduces the overhead of medium access.

2.3 Throughput Estimation For IEEE 802.11n Protocol

The link speed or throughput is affected by many factors like modulation and coding schemes, transmission powers, transmission distances, and even designs of network adapters [8]. Therefore, the theoretical computation of the link speed is difficult. In this work, we take an alternative approach which conducts real-world measurements to model the actual link speed. Through measurements [9], we derived the link speed ($Mbps$) function $f(x)$ with the independent variable transmission distance x (m).

Table 2: Devices and softwares 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	Windows7
PC2	model	Ultra book Lesance NB S3532-SP
	CPU	Intel(R) Core i3 2350M (2.3 GHz)
	OS	Windows7
NIC		Buffalo WZR-G1750DHP
software		iperf2.0.5
protocol		TCP

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

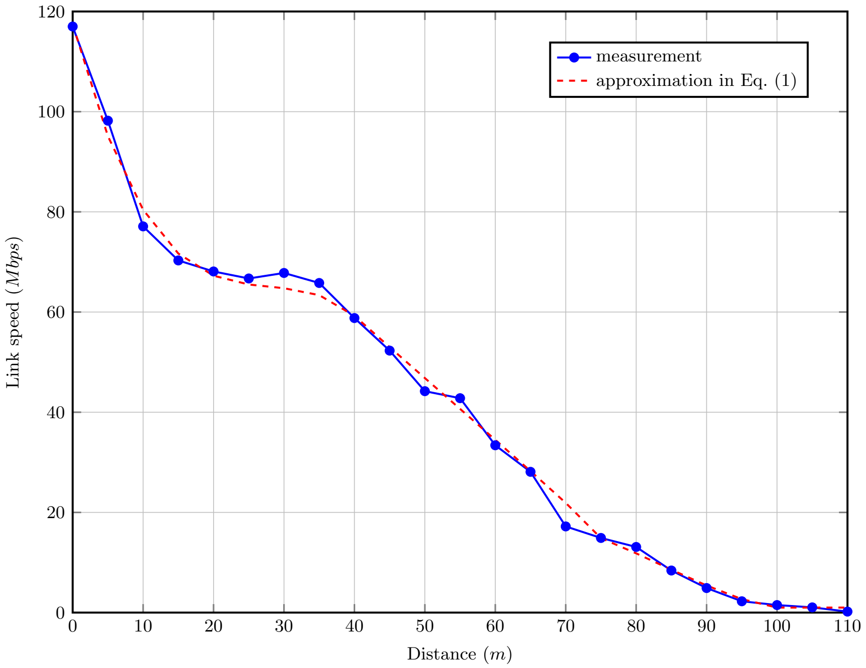

Figure 2: IEEE 802.11n link speed measurement results.

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

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

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 are prepared respectively as the source and destination nodes. We measured the link speed when both end nodes adopt the IEEE 802.11n protocol. At the measurements, we increased the link distance between two end nodes 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 shows the link speed measurement and estimation results. According to our measurement, 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.

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

To examine the *wall effect*, we measured throughput between two rooms separated by one concrete wall at fixed distances of $5m$ between host and AP. Then, we found that the link speed is dropped by about 15% when the signal passes through one concrete wall in a room. Besides, we 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 paper, we adjust the link speed estimated from the link speed function $f(x)$ by multiplying 45/100 for the VAP and 25/100 for the MAP. We apply the same wall effect for any AP type. We decreased the link speed by multiplying 85/100 if there exists a wall between an AP and a host. These parameters are used in our algorithm and the *WIMNET simulator* for simulations.

2.4 WIMNET Simulator

The WIMNET simulator [10] has been originally developed to evaluate a large-scale wireless Internet-access mesh network (WIMNET) with reasonable CPU time on a standard PC. Some modifications have been made to the WIMNET simulator for the simulation of elastic WLANs with heterogeneous APs. The WIMNET simulator simulates the least functions for wireless communications between hosts and APs that are required to calculate throughputs or delays. Several network field parameters such as host locations, AP locations, link distances, one or two-hop communications, wall existences, and repeater existences are adopted in the WIMNET simulator. A variety of functions such as the link speed measurement, host movement, 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 one-frame transmission and the acknowledge reception. Different transmission rate can be set by manipulating the number of time slots required for one link activation.

2.5 AP Aggregation Algorithm

The AP configuration algorithm generalizes the previously proposed AP aggregation algorithm [11, 12]. In the AP aggregation algorithm, the number of active APs are minimized through the aggregation of two or more APs into a single one in order to minimize the interference among them. We introduce the virtual function of DAP providing different SSIDs according to the policies. Every physical AP comes with the facility to create multiple SSIDs for the same DAP [13]. Utilizing this feature, sharing of devices is possible in the AP aggregation algorithm. The AP aggregation algorithm considers only DAPs for APs and the minimization of active APs. In the elastic WLAN system, we need to deal with heterogeneous AP devices such VAPs and MAPs. We also need to increase the active APs on the fly when the traffic varies.

2.6 Related Works in Literature

Several researches have been conducted to improve the performance of the WLAN system and/or to reduce the operational cost. In [14], 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 [15], an optimal AP placement problem for the uniform quality of services has been proposed to realize the WLAN in public. In [16], a local search algorithm for the joint AP placement and channel assignment for IEEE802.11 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 [17], 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 [18], 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 [19], an AP selection algorithm is proposed to maximize the throughput while preserving the newly arriving user throughput in a multi rate WLAN. In [10], an active AP selection algorithm is introduced for wireless mesh networks ensuring the optimal operational cost and throughput. In [20], 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.

3 Elastic WLAN System with Heterogeneous APs

In this section, we introduce the elastic WLAN system with heterogeneous APs.

3.1 Motivation

Unplanned installation of APs may lead to poor performance due to interferences caused by redundant APs, network overloads, and device failures. As a result, WLANs should be adaptive to the network changes by activating new APs with increasing demands or deactivating APs for decreasing demands, and changing associations of hosts to APs to optimize the network performance.

The motivation for this elastic WLAN system is to adapt the network with respect to network environments. Specifically, in the network, APs may get overloaded during office 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, cellular network based devices like MAPs can be activated in the WLAN to ensure the network performance. On the other hand, redundant APs assigned to the same channel introduces inter-AP interferences. The elastic WLAN system can reduce the interference problem by deactivating unnecessary APs.

3.2 Elastic WLAN System Design

In our current implementation of the elastic WLAN system in [21], we adopt a server to manage and control the APs and the hosts. Figure 3 shows an exemplar topology of the elastic WLAN system including the management server. This server has the administrative access to all 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 (except the MAP, i.e., the mobile routers) and the associations of the hosts according to the algorithm output.

Then, in our implementation of the elastic WLAN system, we consider to send the message to the corresponding owner of the mobile router to activate or deactivate it. Once a mobile router is turned on, the newly associating hosts with it can be associated with it remotely. To encourage the owner to follow this instruction, we assume that all the mobile routers are provided by the organization that manages the system, or that the owners of mobile routers are offered some incentives such as service fees or privileges of using resources in the organization including high performance computers, offices, or dormitories/apartments.

The transmission of this message system is not developed yet in the current implementation of elastic WLAN. This will be in our future works.

The management server explores the devices in the network and collects the necessary information for the AP configuration algorithm. This information collection procedure comprises three phases:

1. The Linux tool *arp-scan* is used to scan the network. All the commands in this paper are tested on Ubuntu 14.04 LTS operating system. The command is:

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

   Here, eth0 is the network device and 192.168.11.0/24 is the range of IP addresses to scan. The output consists of the IP and MAC addresses of the active hosts and APs in the network. We

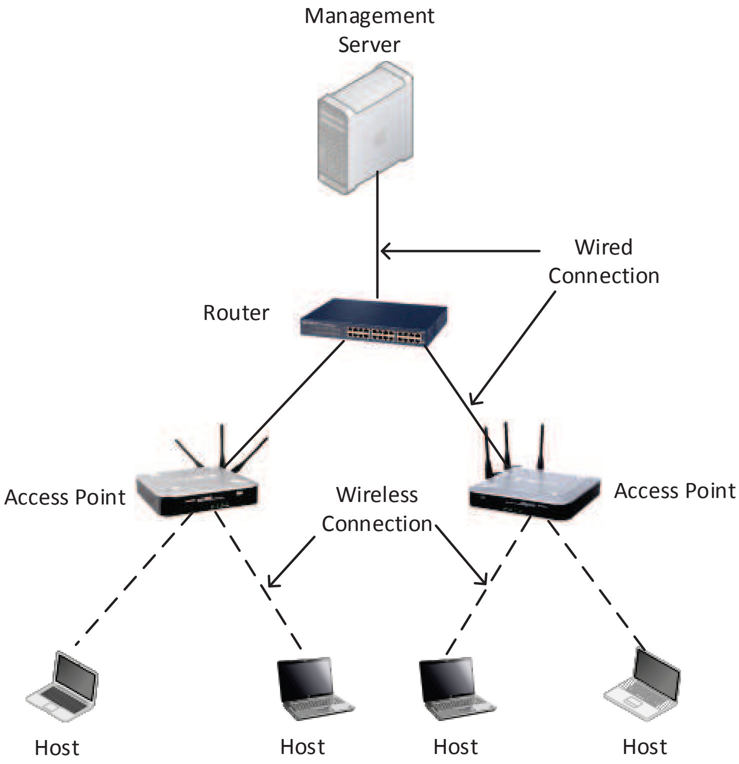

Figure 3: Elastic WLAN system topology.

assume that the management server has the list of the MAC addresses of all the permitted devices in the network beforehand. Thus, it can identify the active APs and hosts. It can also identify any prohibited or disapproved devices from the network. After this step, the server generates the following lists:

- List of active APs.
- List of active hosts.

2. Then, the management server collects the information of each active host using the list generated above. *nm-tool* is used for the purpose, which is executed using the following command:

   ```
   $ sudo nm-tool
   ```

   By executing *nm-tool* remotely from the management server on the host through the *ssh* protocol, the server can collect the following information for each host:

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

3. The management server converts each signal strength to the estimated link speed and summarizes the collected information for the input to the AP configuration algorithm.

Although in our current implementation, the proposed algorithm starts by the external control, we believe that it can be easily re-designed to start automatically. Specifically, we can consider that every time a new host joins the system, the server collects the above mentioned information for the new host to update the network information, and estimates the new load of the AP with the new host associated. Besides, whenever an existing host is disconnected from the system, the server estimates the load of associated AP. When the new load is either too high or too low, the proposed AP configuration algorithm is executed and the output is applied to the devices in the system.

3.3 Network Dynamics under Considerations

In this section, we describe the three major scenarios and the provisions that should be considered 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.

3.3.1 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.

3.3.2 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.

3.3.3 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 APs are necessary to be increased in the network. In this case, this limitation of bandwidth affects only the wired connection, and 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 Active AP Configuration Problem

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.

4.1 Basic Terminology

In this section, we briefly describe 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) 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 categorized 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 our paper, 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 simulation. 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 4, 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. We adopt the same speed drop rate per wall for any AP 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 5, we explain the example for the modifiable host and the modifiable host list. In this example, the solid line between a host and an AP represents that 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 5, 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 $\{Host4\}$.

4.2 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).

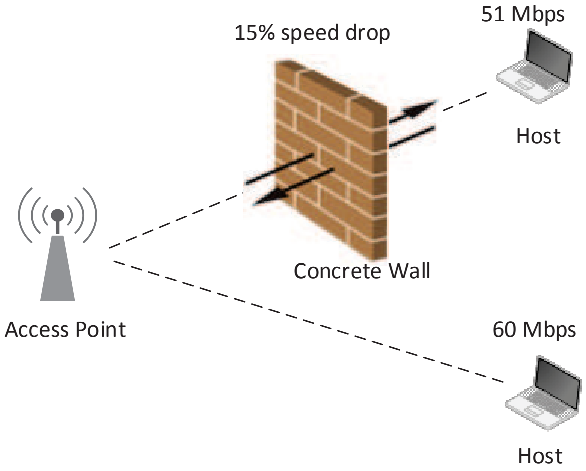

Figure 4: Wall effect.

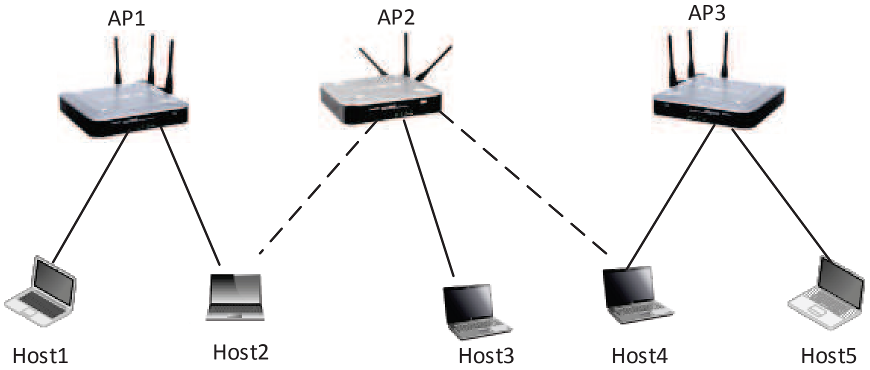

Figure 5: Example of modifiable hosts.

(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 paper. First, the cost function E_1 is used to represent the number of active APs in the network.

$$E_1 = E_1^D + E_1^V + E_1^M \tag{3}$$

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

$$T_j = \sum_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}} \tag{4}$$

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 TH_{ij} can be defined as

$$TH_{ij} = \frac{D_i}{T_j} = \frac{D_i}{\sum\limits_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}}} \tag{5}$$

where D_i represents the traffic of the ith host. So, the minimum host throughput can be defined as

$$TH_{min,j} = \min_{i=1,2,..,|\mathcal{P}_j|} \left[\frac{D_i}{\sum\limits_{k=1}^{|\mathcal{P}_j|} \frac{D_k}{s_{jk}}} \right] \tag{6}$$

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:

$$TH_{min,j} = \frac{1}{\sum\limits_{k} \frac{1}{s_{jk}}} \tag{7}$$

So we can formulate our objective function as follows:

$$E_2 = \min_j \left[TH_{min,j} \right] \tag{8}$$

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 input. The link speeds can be estimated by measuring the signal strength of the hosts and our derived formula in Sec. 2.3.

   (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 a 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.

4.3 NP-Completeness of AP Configuration Problem

The AP configuration is a very complicated problem which 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 set cover problem, a well-known NP-complete problem [22]. To prove the *NP-completeness* of the AP configuration problem, first the problem is translated to its decision version and then a well known *NP-complete* problem is reduced to the decision version of the AP configuration problem. Here we used *minimum set cover problem* [23] as the NP-complete problem.

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 $E_1 = E_1^D + E_1^V + E_1^M$?

4.3.2 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 $|C'| \leq K$?

4.3.3 Proof of NP-Completeness

Clearly, 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 ith AP $= S_i$ and $sp_{ij} = 1$ for the ith AP to the jth host and vice-versa.

5 Active AP Configuration Algorithm

In this section, we propose the AP configuration algorithm to realize the elastic WLAN system deploying DAPs, VAPs and MAPs. The proposed algorithm first minimizes the number of active APs in the network, and then minimizes the transmission delay ensuring a minimum host throughput for every host in the network. In addition, the proposed algorithm adjusts the AP to the host link speed if the expected bandwidth is higher than total available bandwidth. As depicted in Figure 6, 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.

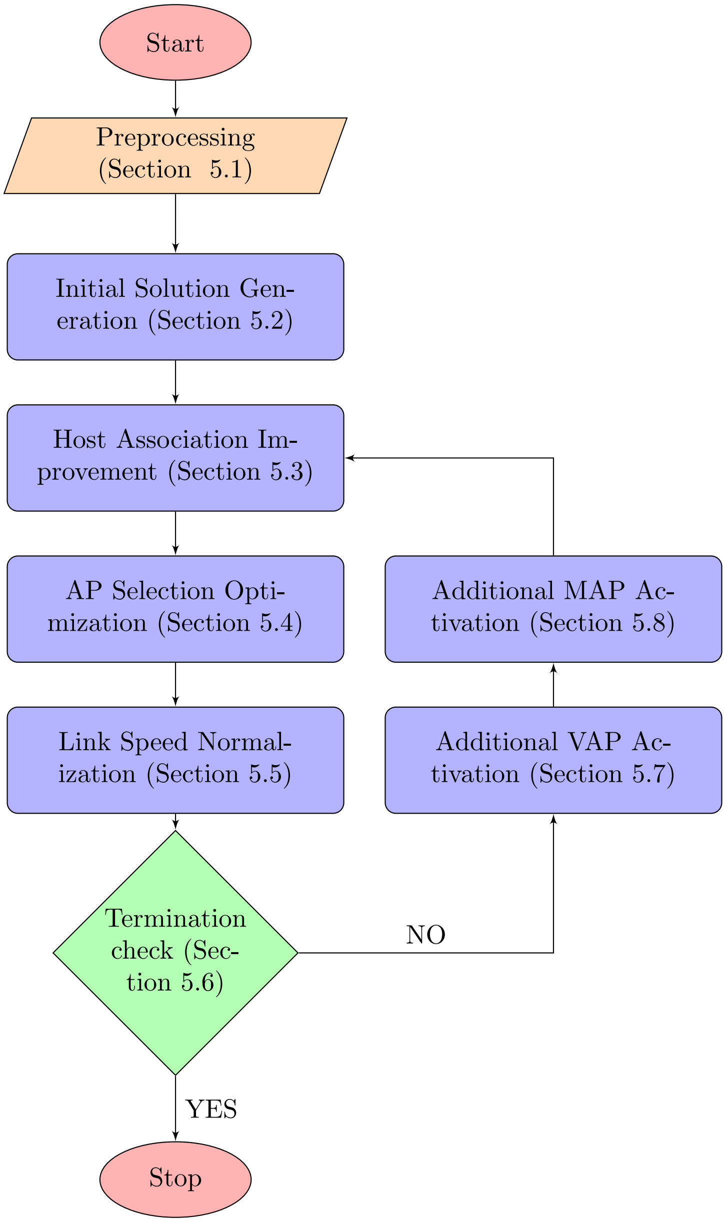

Figure 6: Flow chart of the AP configuration algorithm

5.1 Preprocessing

In this preprocessing phase, if the locations of the hosts and the APs (DAPs, VAPs, or MAPs) are given as inputs to the algorithm for simulation, it estimates the link speed between every possible pair of AP and host using Eq. (1) and (2) considering speed drop rate per wall. Then, this phase initializes 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.
2. For every host, make a list of APs that can be associated to this host. These are *associable APs* for the host.
3. Initialize each AP as non-active AP. Initially, only the DAPs are selected as *candidate APs*.

5.2 Initial Solution Generation

In this phase, an initial solution is derived using a greedy algorithm [24]. This provides the initial number of active APs, E_1 and the hosts associated to these APs. This solution 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 an 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.

5.3 Host Association Improvement

In this phase, we improve the association between active APs found in the previous phase 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. (8), 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}.

5.4 AP Selection Optimization

The cost functions E_1 and E_2 are further jointly optimized in this phase by the *local search* [25] 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, G. In both cases, the host association is optimized using Phase 5.3. The procedure is described as follows:

1. Initialize E_1^{best} and E_2^{best} as the current algorithm output after Phase 5.3. Repeat the 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 if all the hosts associated to this AP, can be associable to other active APs.
   (b) Deactivate the AP and associate all the hosts associated to this AP to other associable active APs. If there are multiple such active APs, select them in order of largest number of unassociated hosts they can cover.
   (c) Decrease E_1 by one.

   (d) Call Phase 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 a 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 5.3 to optimize the host association for the current number of active APs.
   (d) Calculate the new value for 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 5.3 to optimize the host association for the current number of active APs.
   (d) Calculate the new value for 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}.

5.5 Link Speed Normalization

For the bandwidth limitation scenario, we should adjust 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

$$b_i = \frac{|\mathcal{P}_i|}{\sum\limits_{j=1}^{|\mathcal{P}_i|} \frac{1}{s_{ij}}} \tag{9}$$

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

$$B^e = \sum_{i=1}^{N} b_i \tag{10}$$

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

$$\hat{s}_{ij} = s_{ij} \times \frac{B^a}{B^e} \tag{11}$$

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 are not influenced by the total bandwidth limitation.

To normalize the link speed, do the following:

1. Calculate the expected total bandwidth B^e using Eq. (9) and (10)
2. If $B^e > B^a$, adjust every AP-host link speed as described in Eq. (11).

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.

5.7 Additional VAP Activation

If VAPs are not selected for candidate APs, they are selected as candidate APs. Go back to Phase 5.3. Otherwise, go to Phase 5.8.

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 paper. This constraint for the jth AP can be checked by:

   $$\frac{1}{\sum_{k=1}^{|\mathcal{P}_j|} \frac{1}{s_{jk}}} \geq G \tag{12}$$

   where G represents the *minimum host throughput constraint* 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. (11) if $B^e > B^a$. The additional MAPs 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 a 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 *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.
    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. (8) 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.

6 Evaluations

In this section, we present our experimental results of the proposed AP configuration algorithm in different network environments using the WIMNET simulator. Table 3 summarizes the hardware and software platforms used for the experiments.

Table 3: Simulation Environment.

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

First, we give a brief description of the extensions of the simulator for AP configuration algorithm. Then, we present the simulation results in two different network instances.

6.1 Extensions to WIMNET Simulator

Two new types of nodes VAPs and MAPs are introduced in the WIMNET simulator in addition to hosts and DAPs. For every MAP, an additional gateway node is introduced so that every host associated with the MAP has the Internet access through this gateway.

6.2 Network Environment

In our previous studies [11, 12], we considered *indoor network environment* inside a building as the target one for our proposal, because wireless local area networks (WLANs) are usually used there. In such environments, users use their 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. To consider the effects by the walls along the transmission link between two devices for simulation, we reduce the link speed by 15% every time the link passes through one wall. The hosts here are regularly distributed since students have their fixed positions in the laboratories.

6.3 Simulations Under Network Environment Changes

The proposed active AP configuration algorithm is evaluated sequentially in three scenarios mentioned in Section 3.3. 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 2 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 accesses the Internet through the cellular mobile network. We consider the cases with the number of hosts from 40 to 60, the minimum host throughput constraint, G from $5Mbps$ to $10Mbps$, and the throughput limit B^a from $50Mbps$ to $150Mbps$ for limited bandwidth scenario otherwise ∞.

For simplicity, we assume that the link speed for a VAP or a MAP is dropped at a constant rate from the link speed for a DAP. A VAP uses a Windows-PC with larger software/hardware overheads than a DAP. A MAP uses cellular network for Internet, which is usually slower than the wired connection. The link speed assumption for VAP and MAP is described in Section 2.3.

6.3.1 Network Load Increase Scenario Result

First, we consider the scenario where increase of users raises the load on the network. The number of hosts H and the minimum host throughput constraint G are increasing. Table 4 summarizes the results for this scenario. In the table, we present the required number of active APs, analytical results estimated by our algorithm and numerical results 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 4: 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 the larger minimum host throughput G requires more VAPs. Unfortunately, the algorithm cannot find a feasible solution for the extreme network load of $H = 60$ hosts and the minimum host throughput $G = 10$ where more APs are presumably required.

6.3.2 Solution Example

To illustrate this network topology, Figure 7 shows one solution determined by the algorithm for the DAP failure scenario with $H = 60$, $G = 10$, 1 active DAP, 10 active VAPs, and 5 active MAPs, presented in Table 5. We observe 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 the figure, each DAP, VAP, and MAP is connected respectively to 10, 4 and 2 hosts and the link speed is around $100Mbps$, $40Mbps$, and $20Mbps$.

6.3.3 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 for the previous scenario. Table 5 summarizes the results for this scenario.

Table 5: 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

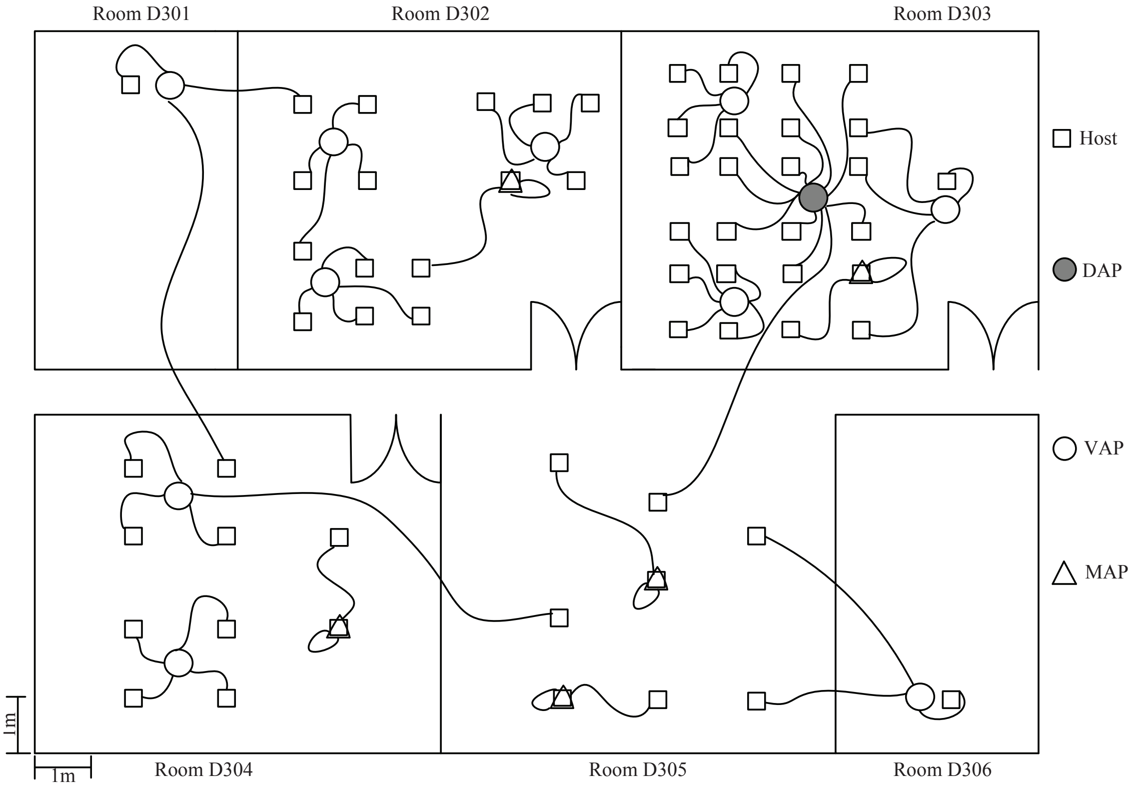

Figure 7: Solution of proposed algorithm for scenario with $H = 60$, $G = 10$, 1 DAP, 10 VAPs, 5 MAPs and data plan=$30Mbps$ in Okayama University instance.

a DAP fails, the throughput in the network drops and additional AP activations are needed to 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 $G = 10$, minimum host throughput constraint by activating 5 MAPs.

6.3.4 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 for previous scenarios, while the minimum host throughput constraint G is fixed as 5. The throughput limit constraint B^a is set 150, 100, and $50Mbps$. If two DAPs are active, they consume all the capacity of Ethernet, and VAPs become useless in this scenario. Then, activation of MAPs can provide the additional bandwidth if necessary in the network.

Table 6 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 $150Mbps$ to $50Mbps$, 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 $150Mbps$, 2 DAPs along with 3 MAPs can satisfy the minimum host throughput constraint. When B^a is $100Mbps$, 1 DAP and 6 MAPs are needed, and when B^a is $50Mbps$, 1 DAP and 9 MAPs are needed to satisfy the *minimum host throughput constraint.*

Table 6: 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	

6.4 Simulations Under Various Parameter Values

Then, we evaluate the effectiveness of the active 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 speeds to the Internet can be different depending on adopted data plans and network environments such as distances from base stations and the number of users. 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. Then, *MAPs* are again introduced into the network to increase the available bandwidth of the network. In [9], 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 3 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 $24Mbps$ [2].

6.4.1 Solution Example

To illustrate this network topology, Figure 8 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, presented in Table 7. Again, we observe that the DAP has the largest number of associated hosts, while the MAPs have the least number of associated hosts.

6.4.2 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 7 summarizes the results for this scenario. This table indicates that when MAP speed in data plans increases from $12Mbps$ to $24Mbps$, our algorithm activates the less number of MAPs for the higher speed plan than the lower speed plan.

---

[2] To evaluate the additional MAP activation phase, the VAPs are not used in this instance.

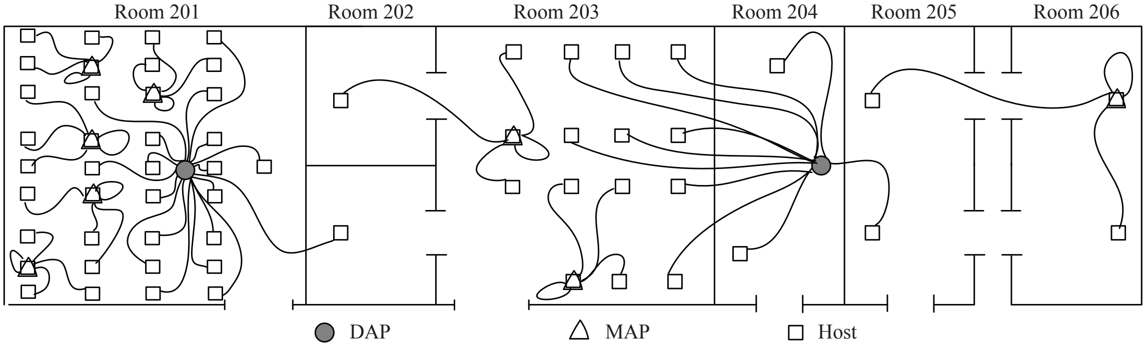

Figure 8: Solution for bandwidth limitation with mobile data plans scenario with $H = 60$, $G = 5$, data plan $= 24Mbps$, 2 DAPs and 8 MAPs in Kabi Nazrul University instance.

Table 7: 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

6.4.3 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 8 summarizes the results for this scenario. This table indicates that as speed drop rate becomes larger, the algorithm increases the number of active MAPs in the network to satisfy the throughput constraints.

Table 8: 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

6.5 Comparisons with Greedy Algorithm

Finally, we evaluated the performance of our algorithm by comparing with other algorithms. To the best of authors' knowledge, this work is the first one that investigate into this issue. Thus, we implemented a simple greedy algorithm for the AP configuration problem 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 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 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 Kabi Nazrul 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 the greedy algorithm with the MAP speed change scenario.

Tables 9 and 10 show the comparison results. These simulation results show that, except the case of 24 data plan in Table 10, the proposed algorithm activates the less number of MAPs compared with the greedy algorithm, which demonstrates 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 higher minimum host throughputs than the greedy algorithm, since the proposed algorithm yields better host associations.

Note that for the case of 18 data plan, the analytical minimum host throughput of the greedy algorithm is slightly higher ($20Kbps$). This is because that, the analytical calculation in Eq. (12) assumes all the hosts transmit the same amount of data, which results in several hundred $Kbps$ approximation errors as shown in Table 9 and Table 10. By far, the difference of $20Kbps$ is due to this approximation error and hence can be ignored.

Table 9: 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 10: 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

7 Conclusions and Future Works

This paper proposed an active AP configuration algorithm for the elastic WLAN system that activates or deactivates APs depending on traffic demands and changes of network environments. This algorithm considers three types of AP devices with difference in link speeds and network architectures. The allocated locations of MAPs are limited to positions of hosts as they can be managed by the owners. The effectiveness of the algorithm was verified through simulations in realistic scenarios using the *WIMNET simulator*. Our future works include further performance enhancements and extensions to dynamic changes of network situations.

References

[1] 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.

[2] S. Lanzisera, B. Nordman, and R. E. Brown, "Data network equipment energy use and savings potential in buildings," Energy Efficiency, vol.5, no.2, pp.149–162, May 2012.

[3] 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. Fifth Int. Conf. on Wireless and Mobile Commun. (ICWMC-2009), pp.134-139, 2009.

[4] "Electricity sector in Bangladesh," Internet: `https://en.wikipedia.org/wiki/Electricity_sector_in_Bangladesh`, Oct., 2015 [Jan. 12, 2016].

[5] "Electricity to transform rural Myanmar," Internet: `http://www.worldbank.org/en/news/feature/2015/09/16/electricity-to-transform-rural-myanmar`, Sept., 2015 [Jan. 12, 2016].

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

[7] 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," online available at: `http://standards.ieee.org/getieee802/download/802.11-2012.pdf`.

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

[9] 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. and Auto. Trusted Comput., pp.600-605, Sept. 2012.

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

[11] M. E. Islam, N. Funabiki, and T. Nakanishi, "An access-point aggregation approach for energy-saving wireless local area networks," IEICE Trans. Commun., vol.E96-B, no.12, pp.2986-2997, Dec. 2013.

[12] M. E. Islam, N. Funabiki, and T. Nakanishi, "Extensions of access-point aggregation algorithm for large-scale wireless local area networks," Int. J. Netw. Comput., vol.5, no.1, pp.200-222, Jan. 2015.

[13] B. Aboba, "Virtual access points," IEEE Contribution, Mar. 2003.

[14] H. P. Gupta and S. V. Rao, "Demand-based coverage and connectivity-preserving routing in wireless sensor networks," IEEE Syst. J., vol.PP, no.99, pp.1-10, July 2014.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] M. E. Islam, N. Funabiki, K. S. Lwin, M. S. A. Mamun, S. K. Debnath, and M. Kuribayashi, “Performance evaluations of elastic WLAN testbed,” IEICE General Conf., pp. S76-S77, Mar. 2016.

[22] 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.

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

[24] 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.

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