A Study of Throughput Control Method for Concurrently Communicating Multiple Hosts in Wireless Local-Area Network

September, 2022

Md. Mahbubur Rahman

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 2022.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Md. Mahbubur Rahman

Signature of

the Supervisor

Prof. Nobuo Funabiki

Seal of

Graduate School of

Natural Science and Technology

Abstract

Nowadays, the *IEEE 802.11 wireless local-area network (WLAN)* system is the most popular medium for the Internet access around the world due to the characteristics of high speed data rates, the low installation/management costs, and the high flexibility. A WLAN user can access to the Internet by connecting with a nearby *access point (AP)* through wireless signals. As a result, WLAN has become the default media for accessing Internet. Hence, WLAN has been deployed at offices, schools, and public transportations including buses, trains, and airplanes. In the network field, multiple APs are often deployed to provide the flawless Internet access by extend the network coverage range, and to support a large number of hosts.

In WLAN, the locations of the users are generally non-uniform and their traffic patterns fluctuate frequently by the time and day of the week. Hence, the *elastic WLAN system* has been studied that dynamically controls the network configuration according to traffic demands while reducing power consumption and improving the network performance. The *elastic WLAN system testbed* has been developed to conduct experiments for this study. By running *hostapd*, *Raspberry Pi* is used as the software AP.

In WLAN, the fair throughput service is important to offer the equal *quality of service (QoS)* among the hosts in the network. Particularly, as the demand for real time multimedia applications increases, such as online meeting tools, the fairness becomes the critical issue. However, in WLAN, the throughput may not be fairly shared among the hosts in the network when they are concurrently communicating. The different *received signal strengths (RSS)* at the hosts from the AP/APs may result in differing throughputs among them. The slower *RSS* may lead to the use of a slower *modulation and coding scheme (MCS)* at the far hosts compared to the near hosts, resulting in lower throughputs. The interference caused by co-located APs and hosts in the same network field may further enhance the throughput disparities. Besides, a host may suffer from *insufficient throughput*, although it may need the high throughput to download large files, for example. In this case, the necessary throughput should be allocated to the host by sacrificing the other hosts.

On the other hand, a host may be connected with a server on the Internet that needs the small throughput for the running application. Consequently, the throughput achieved by the host can be smaller than the fair throughput in the WLAN. In this case, this host is referred to as the *saturated host*, and the maximum achieved throughput is the *saturated throughput*. A WLAN should assign the *saturated throughput* to the *saturated host*, and the remaining bandwidth should be shared among the other hosts to avoid wasting bandwidth.

Previously, to realize the fair throughput service among the multiple hosts associated with a single AP, I studied the *TCP fairness control method* in WLAN that controls the *packet transmission delays* at the AP using the *proportional integral (PI) controller*. Unfortunately, this method was limited to one AP in the network, although multiple APs are common and are often interfered with each other in WLAN. Our previous method may have the slow convergence to achieve the fair throughput since the *delay* is changed continuously using the *PI controller*. Furthermore, it is

difficult to assign different throughputs to the hosts if necessary, and it cannot be applied when a *saturated host* appears in the network.

To address the above drawbacks, in this thesis, I propose a *throughput control method* to solve the throughput unfairness/insufficiency problem for concurrently communicating hosts in WLAN. It provides the fair or necessary throughput to the hosts in WLAN, when they are concurrently communicating with same or different APs. This method 1) measures the *single throughput* and the *concurrent throughput* for each host, 2) calculates the *channel occupying time* from the measurement results, 3) derives the *target throughput* to achieve the request, and 4) controls the traffics to satisfy the *target throughput* of every host by applying the existing *traffic shaping* technique at the AP using the Linux command *tc*. It employs the *hierarchical token bucket (HTB)* queuing discipline.

I implemented the proposed *throughput control method* in the *elastic WLAN system testbed* that uses *Raspberry Pi* devices for the APs. Then, firstly, I evaluated the proposal when multiple hosts concurrently communicating with the same AP by considering different throughput requests. The experiment results in various scenarios confirm that this approach can achieve the fair or necessary throughputs to the hosts.

Secondly, I evaluated the proposal by extensive experiments when multiple hosts concurrently communicating with different multiple APs under the equal throughput scenario. The experiment results show the throughput fairness becomes close to one in any topology, which confirms the effectiveness of our proposal.

In future studies, I will consider further enhancements of the *throughput control method* and *the WLAN system testbed* implementation, and their evaluations in various network fields.

Acknowledgements

I am delighted to express my heartiest thankfulness to those who gave me valuable time and supported me in making this dissertation possible. Thanks all of you to make my dream successful.

I wish to express my sincere gratitude to my supervisor, Professor Nobuo Funabiki for his excellent guidance, valuable suggestions, persistent encouragement, and other helpful assistance throughout my Ph.D. study. He provided me with insightful comments and guidance that helped me to complete my research papers and present them in a productive way. In addition to that, he was always patient and helpful whenever I needed his guidance and assistance in my academic and daily life in Japan. I have been fortunate to work with someone like him. In fact, it would not have been possible to complete this thesis without his guidance and support.

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

I would like to express my profound gratitude to Associate Professor Minoru Kuribayashi in Okayama University for his valuable discussions and support during my research. I also want to express my gratitude to the course teachers during my Ph.D. study for sharing their great ideas and knowledge with me.

I would like to acknowledge the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT) for financially supporting my Ph.D. study at Okayama University.

I would like to convey my appreciation to all the members of FUNABIKI Lab. Especially, I would like to give special thanks to Ms. Keiko Kawabata, Dr. Kwenga Ismael Munene, Dr. Md. Manowarul Islam, Dr. Samsul Huda, Dr. Rahardhita Widyatra Sudibyo, Dr. Hendy Briantoro, Dr. Yan Watequlis Syaifudin, Dr. Htoo Htoo Sandi Kyaw, Dr. Hein Htet, Dr. Pradini Puspitaningayu, Mr. Yuanzhi Huo, Mr. Sujan Chandra Roy, Mr. Yohanes Yohanie Fridelin Panduman and all members in general.

Last but not least, I am eternally grateful for the sincere friendship, inspiration, motivation and support of my dear parents, brothers, sisters and all my friends.

Md. Mahbubur Rahman
Okayama University, Japan
September, 2022

List of Publications

Journal Paper

1. **M. M. Rahman**, N. Funabiki, K.I. Munene, S. C. Roy, M. Kuribayashi, and W. -C. Kao, "A Throughput fairness control method for concurrent communications in wireless local-area network with multiple access-points," J. Commun., vol. 17, no. 8, pp. 592-599, Aug. 2022.

International Conference Paper

2. **M. M. Rahman**, N. Funabiki, K. I. Munene, H. Briantoro, F. Akhter, M. Kuribayashi, and M. M. Islam, "A proposal of throughput allocation method for concurrently communicating two or three hosts in wireless local-area network," Int. Conf. on Emerg. Techn. for Commun. (ICETC), C1-4, Dec. 2020. (Online, Japan)

Other Paper

3. **M. M. Rahman**, N. Funabiki, R. W. Sudibyo, K. I. Munene, H. Briantoro, and M. M. Islam, "An Improvement of TCP Throughput Fairness Control Method for Concurrent Communications in Elastic WLAN System," IEICE Tech. Report, NS2020-25, pp. 19-24, Jun. 2020. (Online, Japan)

List of Figures

List of Tables

List of Abbreviations

AP access point

BSS basic service set

CB channel bonding

DAP dedicated access point

ESS extended service set

HTB hierarchical token bucket

IBSS independent basic service set

ISM industrial scientific and medical

MAC media access control

MAP mobile access point

MCS modulation and coding scheme

MIMO multiple input multiple output

NIC network interface card

OFDM orthogonal frequency division multiplexing

OFDMA orthogonal frequency division multiple access

PC personal computer

RF radio frequency

RSS received signal strengths

SSID service set identifier

tc traffic control

TCP transmission control protocol

VAP virtual access point

WLAN wireless local area network

List of Notations

H_i i[th] host, where $i = 1, 2, ..., n$

S_i single throughput for the host H_i

C_i concurrent throughput for the host H_i

t_i target throughput for the host H_i

t_{min} minimum target throughput

d_i rate and ceil value for the host H_i

R_i measured throughput for the host H_i at each time step

Contents

Chapter 1

Introduction

1.1 Background

Recently, the Internet has become one of the most powerful tools for communicating effectively and efficiently in our regular lives. It creates opportunities for people to communicate with one another, create digital contents, access various information, and solve different problems regardless of where they are located. The influence of the Internet on our society is growing rapidly due to its flexible and convenient functionality [1].

The IEEE 802.11 *wireless local-area network (WLAN)* or WiFi is the most widely used wireless network technology for accessing the Internet services around the world [2–4]. WLAN allows the user to access the Internet through the wireless medium which makes it the default mode of Internet access. It offers several advantages such as flexibility, fast data transfer, and easy installation. Hence, it has been deployed at offices, universities, schools, and various public transport as buses, trains, and even in airplanes [5–7]. WLAN networks are often installed with multiple *access points (APs)* to extend coverage area, and support a large number of WLAN users to ensure seamless Internet access. Besides, channel bonding (CB) plays an important role in IEEE 802.11n to increase transmission capacity by combining two adjacent 20 MHz channels to form a single 40 MHz channel [8].

The *IEEE 802.11 standard* was initially introduced in June 1997 [9, 10]. Since then, the *IEEE 802.11 standards* have been updated with the newer standards by adding new features to improve the performance of WLAN. Currently, the *IEEE 802.11n standard* is most widely used among them, despite the fact that its maximum throughput is lower than that of newer standards like IEEE 802.11ax. Several improved features have been included in the *IEEE 802.11n standard*, including channel binding (CB), multiple input multiple output (MIMO), and frame aggregation, compared to previous standards, namely 11a, 11b and 11g.

In WLAN, the number of hosts and their traffics are frequently changing by day of the week and by time of day [11, 12]. Besides, the WLAN performance can be affected by a variety of factors, such as device failures, power shortages and bandwidth controls by the network administrators. Therefore, we have studied the *elastic WLAN system* that dynamically controls the network configuration according to throughput/traffic demands and devices conditions to minimize the power consumption while improving the WLAN performance [13–15].

Figure 1.1 illustrates the example topology of the *elastic WLAN system*. It considers three types of APs, specifically, a *dedicated AP (DAP)*, a *virtual AP (VAP)*, and a *mobile AP (MAP)*. A *DAP* express a commercial AP, a *VAP* denotes a host PC of a user in the network that installs the software for AP functions, and a *MAP* does a mobile router. Additionally, the *elastic WLAN* *system* has been implemented by running *hostapd* software onto *Raspberry Pi*, which act as an AP. The *Raspberry Pi* is a small card-sized computer. It contains a built-in *wireless network interface card (NIC)* for IEEE 802.11n and runs on a Linux-based operating system called Raspbian.

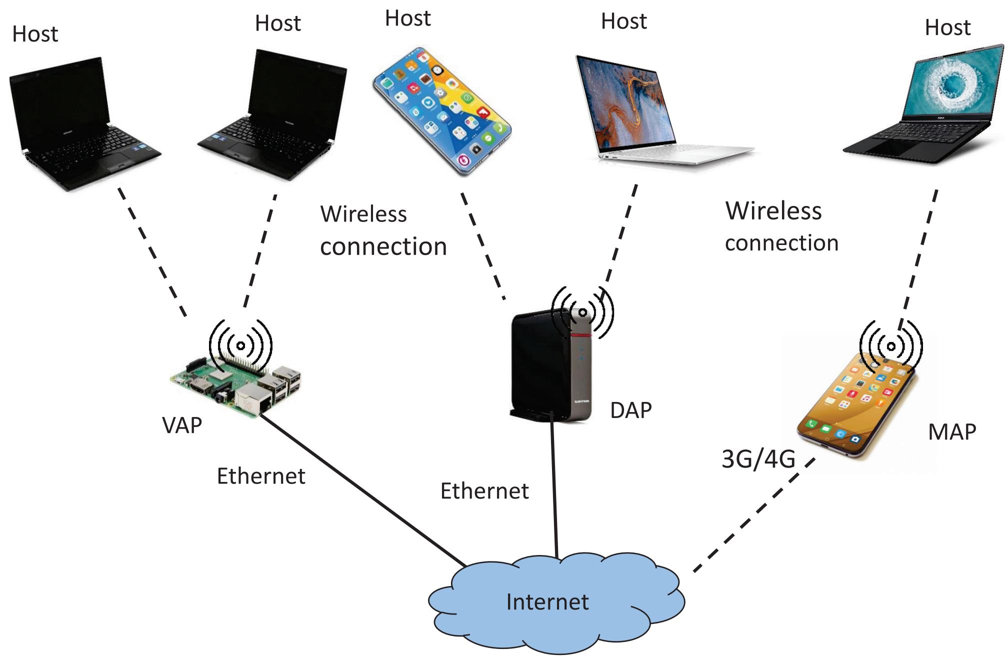


Figure 1.1: Overview of elastic WLAN system topology.

The throughput fairness has become a critical issue in WLAN as demands for the real-time multimedia applications such as streaming video and the online meeting have grown. For service providers, it is imperative to provide *fair quality of service (QoS)* to the WLAN users. However, WLAN cannot ensure the throughput fairness among the concurrently communicating hosts in the network. The throughput depends on the distance of the host from the AP/APs and the interference from the co-located APs in the same network field. Furthermore, a host may suffer from insufficient throughput, even though it may need the high throughput to download large files. In this case, the required throughput should be allocated to the host by reducing the other hosts.

In fact, our preliminary testbed experiments have expressed the throughput unfairness appears among the hosts when they are concurrently communicating with the same AP from different relative distances. Also, they found that the throughputs are much different among the concurrently communicating hosts when the multiple APs are installed with the CB channels in WLAN. It can happen due to the unequal *received signal strengths (RSS)* at the hosts and the interferences from nearby APs. Basically, the hosts far from the AP/APs receive lower *RSS* than a hosts near to it. The lower *RSS* can lead to the use of a slower *modulation and coding scheme (MCS)* at distant hosts compared to nearby hosts, which can cause the lower throughput at the distant hosts. As a result, the throughput unfairness problem becomes severe among the hosts in the WLANs. Hence, the throughput fairness issue in WLAN has been studied for the *transmission control protocol (TCP)*, since the TCP is used as a key Internet services such as the world-wide web, electronic mail, and video streaming [16, 17].

In WLAN, a host may be connected to an Internet server that needs the small throughput for the application, or has the small processing capability, or has the small bandwidth section. Then, the achieved throughput of the host can be saturated and be smaller than the fair throughput in the WLAN. For example, the popular video meeting service *zoom* requires $2Mbps$ for the single screen [18], which is much smaller than the available bandwidth of *IEEE 802.11n* WLAN. In this thesis, this host is called the *saturated host*, and the maximum achieved throughput is the *saturated throughput* for convenience. To avoid wasting the limited bandwidth in WLAN, the saturated throughput should be assigned to the *saturated host*, and the remaining bandwidth be shared among the other hosts.

Previously, to achieve the fair throughputs among the hosts, we have studied the *TCP fairness control method* that controls the *packet transmission delays* at the AP using the *PI control* [19,20]. However, this method cannot be applied to a common WLAN with multiple APs where they are often interfered with each other. This method initially calculates the *packet transmission delay* based on measured RSS of the hosts. Then, the *proportional integral controller* continuously changes the delay to achieve the fair throughputs among the hosts, which can cause a slow convergence. Furthermore, it is hard to allocate different throughputs to the hosts even if necessary, and it cannot assign the proper throughput to the hosts when the *saturated host* presents in WLAN.

1.2 Contributions

The following research contributions are included in this thesis. To address the drawbacks of the *TCP fairness control method*, I propose a *throughput control method* to overcome the throughput unfairness/insufficiency problem for concurrently communicating hosts in WLAN [21, 22]. This method 1) measures the single and concurrent throughput for each host, 2) calculates the *channel occupying time* from them, 3) derives the target throughput to achieve the fair or request throughput, and 4) controls the traffics to satisfy the target throughput of every host by applying *traffic shaping* at the AP. In this case, we adopted *hierarchical token bucket (HTB)* queuing discipline to implement the *traffic shaping* [23].

It is crucial to provide the proper bit rate for each host in order to achieve fair or demanding throughput. For this reason, the *target throughput* is introduced, which determines how many bits should be transmitted per second by each host. The *target throughput* for each host is derived from the measured *single* and *concurrent throughput* for every host. The *single throughput* gives the maximum average bit rate of the wireless link between the host and the AP. The *concurrent throughput* gives the channel occupying time by this link per one second, when it divides the *single throughput*. The remaining time is occupied by the other links. Then, if the concurrent throughput is replaced by the target throughput, this relationship is still true. Based on these observations, the procedure of calculating the target throughput for each host is derived.

For performance evaluations, I implement the proposed method in the *elastic WLAN system testbed* by using *Raspberry Pi APs*. Firstly, I evaluated the proposal by conducting extensive with different throughput requests scenarios when multiple hosts concurrently communicate with the same AP. The experiment results confirm that this approach can achieve the fair or necessary throughputs to the hosts.

Secondly, I evaluated the proposal by extensive experiments when multiple hosts concurrently communicating with different multiple APs under the equal throughput scenario. The experiment results show the throughput fairness becomes close to one in any topology, which confirms the effectiveness of our proposal.

1.3 Contents of Thesis

The remaining part of this thesis is organized as follows.

Chapter 2 reviews the IEEE 802.11 wireless network technologies related to this thesis, including features of the IEEE 802.11n protocol, software tools in the Linux operating system, the fairness index, and the traffic shaping technology.

Chapter 3 reviews our previous studies to this thesis.

Chapter 4 describes the experimental observations of the throughput unfairness problem at concurrent communications with multiple-hosts.

Chapter 5 presents the fair throughput control method.

Chapter 6 presents the demanding throughput control method.

Chapter 7 describes the implementation and evaluations of the proposals.

Chapter 8 reviews relevant works in literature.

Finally, Chapter 9 concludes this thesis with some future works.

Chapter 2

Background Technologies

This chapter introduces wireless network technologies for backgrounds of this dissertation. At first, we discuss the advantages, components, types, and standards of *IEEE 802.11 protocols*. Next, the *IEEE 802.11n protocol* and its main features are described. We then outline *Linux* tools and commands for wireless networking that are used for measuring and implementing the elastic WLAN system testbed. Lastly, we review the *traffic shaping* for controlling the network traffic and the *Jain's fairness index*.

2.1 IEEE 802.11 WLAN Overview

The *IEEE 802.11 Wireless local-area network (WLAN)* technology defines the specifications of *physical (PHY)* and *media access control (MAC)* layers to provide high-speed data communication. WLAN extends a wired LAN to enable user mobility by providing wireless connectivity and supporting the flexibility of data communications [24]. It uses *radio frequency (RF)* technology to send and receive data over the air, thus reducing the cost of wiring at home or workplace. Therefore, WLANs are widely adopted in many places, including at home, school, and in the office.

2.1.1 Advantages of WLAN

There are several advantages of WLAN over traditional wired LANs. Some of them are outlined below.

- *Mobility:*
  User mobility is offered by wireless networking over wired networking. The users with a wired network connection must use a wired line to stay connected. In WLAN, users stay connected to the network while moving around a local coverage area.

- *Simple and quick deployment:*
  The network cables between hosts and access points can be eliminated by using WLAN. This makes WLAN installations much quicker and simpler than wired LAN installations.

- *Cost:*
  The cost of installing and maintaining wired LANs is typically higher than that of installing and maintaining WLANs. WLAN reduces the costs associated with cabling and the associated installation and repair work. Since, the WLAN simplifies moving, adding, and

re-configuring, resulting in a decrease in indirect costs such as user downtime and administration.

- *Flexibility:*
  The use of a WLAN reduces the need to run cables through walls and ceilings. The network coverage area of WLAN can be easily expanded as the network media is everywhere.

- *Scalability:*
  WLANs can be designed according to the topologies required. It can accommodate a high number of users and cover a vast region by adding access points.

2.1.2 IEEE 802.11 WLAN Components

IEEE 802.11 WLAN consists of four primary components as shown in Figure 2.1 [24]:

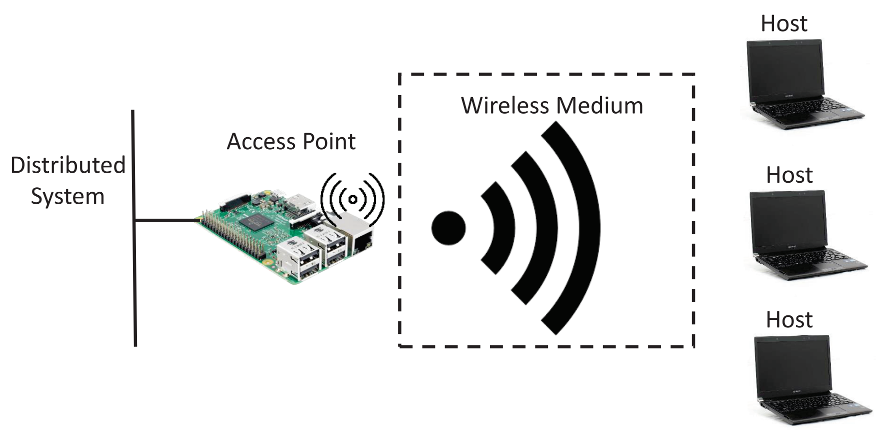


Figure 2.1: Components of *IEEE 802.11* WLANs.

- *Stations or hosts:*
  The *station* or *host* is an electronic device with a wireless *network interface card (NIC)* that can access to the network through WLAN. The device can be a smartphone, a desktop/laptop PC, or a tablet.

- *Access points (APs):*
  In WLAN, the AP is the main radio transceiver that performs the same function as the hub or switch in a wired Ethernet LAN. Additionally, it functions as the bridge between wireless and wired networks.

- *Wireless medium:*
  The *IEEE 802.11* utilizes wireless medium to transmit information/data from one host to another within a network.

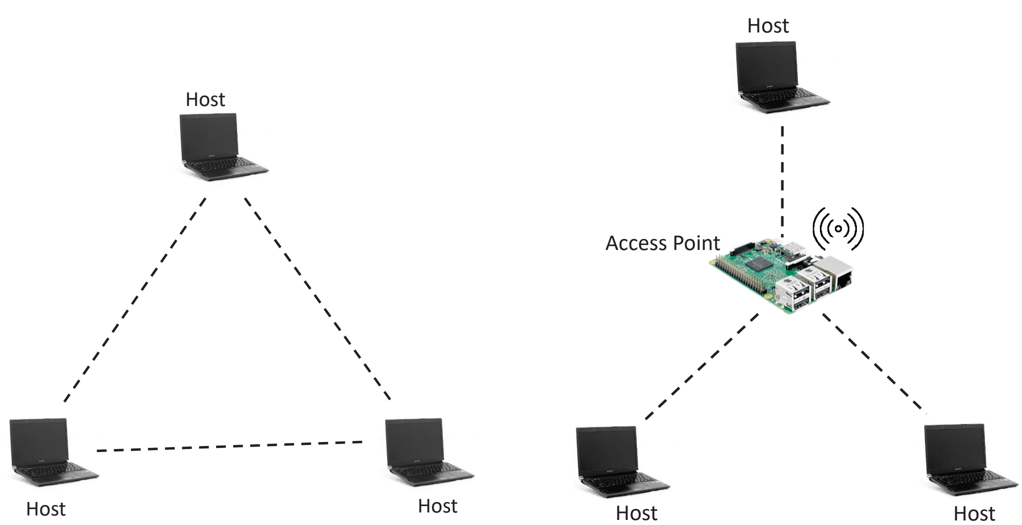
- *Distribution system:*
  In *IEEE 802.11* standards, the distribution system refers to the infrastructure that connects several APs in order to trace the movements of the hosts. The distribution system is a logical component of WLAN that acts as the backbone connection among the APs. It is also commonly known as the *backbone network* for relaying data between APs. Typically, *Ethernet* is used as the backbone.

2.1.3 Types of WLANs

The *IEEE 802.11* WLAN is functioning on a *basic service set (BSS)*, which comprises a set of hosts that can communicate with each other. Two different types of WLAN are supported by the IEEE 802.11 standard depending on the type of BSS as shown in Figure 2.2.

- *Independent or ad hoc type:*
  This type allows stations to communicate directly with each other without an AP. It is also known as an *independent BSS (IBSS) mode* as shown in Figure 2.2(a). In permanent networks, this type of *ad hoc network* is rarely employed due to the lack of required performance and security issues.

- *Infrastructure type:*
  This type exchanges information of stations through an AP as shown in Figure 2.2(b). A single AP serves as the main controller for all the hosts in its BSS, which is called the *infrastructure BSS*. In this case, a host must be associated with an AP to get access the network services [25].



(a) Adhoc network. (b) Infrastructure network.

Figure 2.2: Types of *IEEE 802.11* networks.

In addition, WLAN can be extended further by connecting multiple BSSs with a backbone network to form *extended service set (ESS)* as shown in Figure 2.3. Each AP in ESS is assigned an ID called a *service set identifier (SSID)*, which acts as the ”network name” for the users. The hosts within the same ESS can exchange information with each other, even if they are located in different BSS.

2.1.4 IEEE 802.11 Standards for WLAN

The IEEE 802.11 working group has been enhanced the specifications of existing PHY and MAC layers to support WLAN at the 2.4-2.5 GHz, 3.6 GHz and 5.725-5.825 GHz unlicensed ISM *(Industrial, Scientific and Medical)* spectrum bands defined by the ITU-R. This working group offers several types of IEEE Standard Association Standards, each prefixed with a letter. These include

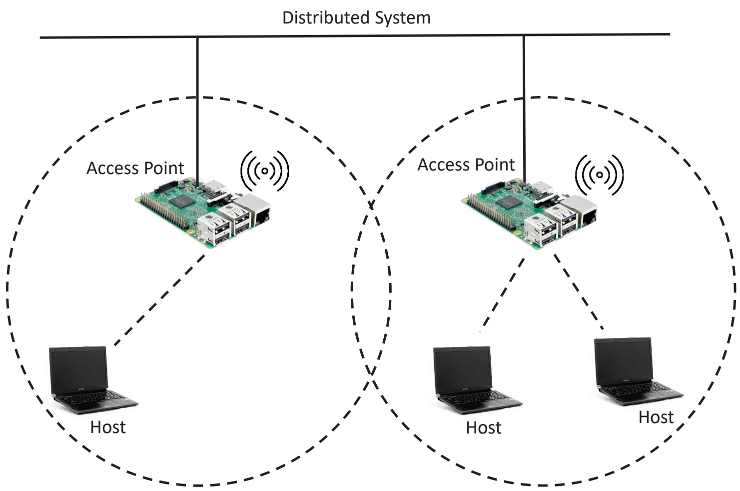


Figure 2.3: Extended service set (ESS).

wireless standards, security standards, quality of service (QoS), and so on as shown in Table 2.1 [25–30].

Table 2.1: *IEEE 802.11* standards.

Standard	Purpose
802.11a	Wireless network bearer operating in the 5GHz ISM band with up to $54Mbps$ of data transmission rate
802.11b	Operates on the 2.4 GHz ISM band, with data rates of up to $11Mbps$
802.11c	Includes bridge operation that links LANs with similar or identical MAC protocol
802.11d	Support for additional regulations in different countries
802.11e	An enhancement to the WLAN 802.11a and 802.11b specifications in terms of QoS and prioritization
802.11f	Inter-Access Point Protocol for handover, the standard was withdrawn
802.11g	2.4GHz ISM band, maximum data rate of $54Mbps$
802.11h	Transmit power control (TPC) and dynamic frequency selection (DFS)
802.11i	Authentication and encryption
802.11j	Standard of WLAN operation in the 4.9 to 5 GHz band to conform to the Japan's rules
802.11k	Measurement reporting and management of the air interface between several APs
802.11l	Reserved standard, to avoid confusion
802.11m	Provides an integrated view of the 802.11 base standard through continous monitoring, management, and maintenance
802.11n	Operate in the 2.4 and 5 GHz ISM bands, data rates up to $600Mbps$
802.11o	Reserved standard, to avoid confusion
802.11p	Provide wireless access for vehicular environments (WAVE)
802.11r	Fast BSS Transition, supports VoWiFi handoff between access points to enable VoIP roaming on a WiFi network with 802.1X authentication

Table 2.1: *IEEE 802.11* standards.

Standard	Purpose
802.11s	Wireless mesh networking
802.11t	Wireless Performance Prediction (WPP), this standard was cancelled
802.11u	Improvements related to ”hotspots” and 3rd party authorization of clients
802.11v	To enable configuring clients while they are connected to the network
802.11w	Protected Management Frames
802.11x	Reserved standard, to avoid confusion
802.11y	Introduction of the new frequency band, 3.65-3.7GHz in US besides 2.4 and 5 GHz
802.11z	Extensions for direct link setup (DLS)
802.11aa	Specifies enhancements to the IEEE802.11 MAC for robust audio video (AV) streaming
802.11ac	Wireless network bearer operating below 6 GHz to provide data rates of at least $1Gbps$ for multi-station operation and $500Mbps$ on a single link
802.11ad	Wireless Gigabit Alliance (WiGig), providing very high throughput at frequencies up to 60GHz
802.11ae	Prioritization of management frames
802.11af	WiFi in TV spectrum white spaces (often called White-Fi)
802.11ah	WiFi uses unlicensed spectrum below 1GHz, smart metering
802.11ai	Fast initial link setup (FILS)
802.11aj	Operation in the Chinese Milli-Meter Wave (CMMW) frequency bands
802.11ak	General links
802.11aq	Pre-association discovery
802.11ax	High efficiency WLAN, providing 4x the throughput of 802.11ac
802.11ay	Enhancements for Ultra High Throughput in and around the 60GHz Band
802.11az	Next generation positioning
802.11mc	Maintenance of the IEEE802.11m standard

Figure 2.4 illustrates the current and future WLAN standards. Among them, the IEEE 802.11a, 11b, 11g, 11n, 11ac are the most commonly used, and the latest is 11ax. . For the physical layer, the 11a/n/ac adopts *orthogonal frequency division multiplexing (OFDM)* modulation scheme while the 11b applies the *Direct Sequence Spread Spectrum (DSSS)* scheme. The 11ax uses *orthogonal frequency division multiple access (OFDMA)*, which is a multi-user version of OFDM. Table 2.2 summarizes the features of these common WiFi standards [24, 31–34].

The IEEE 802.11a, b, and g standards are assumed to have medium security since they use the *wired equivalent privacy (WEP)* security technology. The WEP encryption uses symmetric RC4 stream ciphers with 40-bit and 104-bit encryption keys [35]. The IEEE 802.11n, ac and ax are assumed to have higher security because they adopt the advanced *Wi-Fi protected access (WPA)* encryption system called *temporal key integrity protocol (TKIP)* with *message integrity check (MIC)*.

- *IEEE 802.11b:* The IEEE 802.11b standard was published in September 1999 and operates

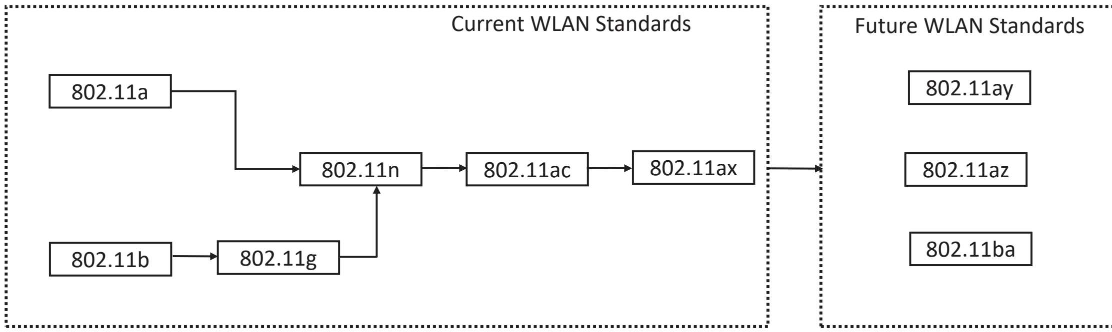


Figure 2.4: Current and future WLAN standards.

at 2.4GHz band with a maximum data rate up to 11$Mbps$. It is considered to be a robust system and has the ability to compensate *IEEE 802.11* protocol. This standard has not only boosted the production of WLAN products, but also motivated the competition between WLAN vendors due to its interoperability. The limitation of this standard is the interference among the products using *industrial, scientific and medical* (ISM) band that uses the same 2.4 GHz band of frequency [36, 37].

- *IEEE 802.11a:* The IEEE 802.11a also released in September 1999. It operates at 5GHz ISM band and data rates offer up to 54$Mbps$ by adopting orthogonal frequency division multiplexing (OFDM) coding scheme. The main limitations of 11a are the incompatibility of 11a products with 11b products and the lack of availability of free 5 GHz bands for all countries in the world [36, 37].

- *IEEE 802.11g:* The IEEE 802.11g was published in 2003. This new 11g standard was proposed over 11a to improve the 2.4GHz 11b standard. The 11g standard introduces two different modulation techniques, including *packet binary convolution code (PBCC)* that supports data rates up to 54$Mbps$ and *orthogonal frequency division multiplexing (OFDM)* which supports data rates up to 54$Mbps$. Compatibility problems are also resolved in 11g products with 11b products [36, 37].

- *IEEE 802.11n:* The IEEE 802.11n standard was issued in 2009. The main purpose of introducing the 11n standard is to improve the usable range and the data rate up to 600$Mbps$. It supports both of 2.4GHz and 5GHz ISM band as *unlicensed national information infrastructure* (UNII) bands, and is backward compatible with earlier standards. It introduces new technology features including the use of *channel bonding* and *multiple-input-multiple-output (MIMO)* to get the better reception of the RF signals to improve the throughput and coverage range [36, 38].

- *IEEE 802.11ac:* The IEEE 802.11ac standard was introduced in December 2013, which operates only in the 5 GHz band with a data throughput of more than 1$Gbps$. It supports static and dynamic channel bonding up to 160MHz and *multi-user multiple-input-multiple-output (MU-MIMO)* [39–41].

- *IEEE 802.11ax:* The IEEE 802.11ax standard was approved on February 2021, which operates in the frequency bands between 1GHz and 7.125GHz and the data rate up to 9.6$Gbps$. 11x focuses on improving throughput per area or the ratio between the overall network

Table 2.2: Features of common IEEE 802.11 standards.

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

[1] CCK: Complementary Code Keying

[2] DSSS: Direct Sequence Spread Spectrum

throughput and the network area size. It supports channel bonding up 160MHz. *IEEE 802.11ax* adopts *orthogonal frequency division multiple access (OFDMA)* technique that is commonly used in cellular networks [42–46]. However, only few devices are compatible with this standard now.

2.2 IEEE 802.11n Protocol

This section overviews the *IEEE 802.11n* protocol that has been adopted in our throughput measurements, models, and implementations in this thesis. This protocol applies several enhanced features, such as CB, MIMO, and frame aggregation, over previous *IEEE 802.11* protocols, 11a, 11b, 11g. Table 2.3 summarizes the main features of *IEEE 802.11n* standard.

Table 2.3: IEEE 802.11n specifications.

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

The IEEE 802.11n supports both 2.4GHz and 5GHz ISM bands. Nowadays, the 2.4GHz is the most widely used frequency band. It becomes congested with many WiFi signals using the same channel or partially overlapping channels. Hence, these WiFi signals with adjacent channels will suffer from interferences among them, and degrade the overall network throughput. [29, 31, 34].

In the 2.4GHz band, the number of non-interfering channels is limited, which are Channel 3 and Channel 11 in 40 MHz bandwidth. While for the 20MHz bandwidth, channel 1, channel 6 and channel 11 are free of interference. However, the wider bandwidth will reduce the number of free channels. Figure 2.5 [30] describes the WiFi channels for *IEEE 802.11n* 2.4 GHz band.

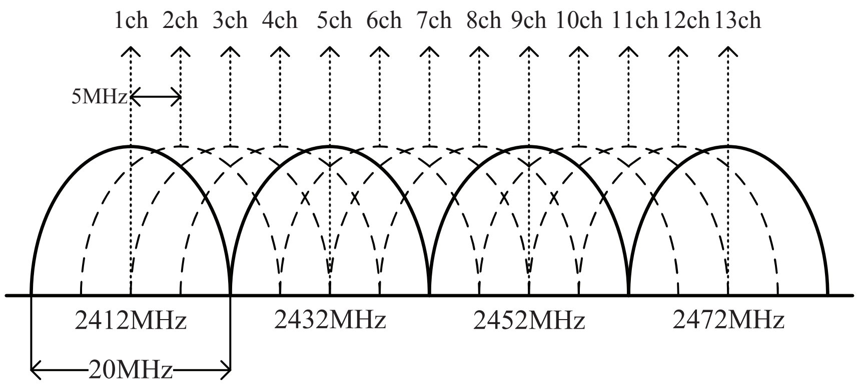


Figure 2.5: WiFi channels in 2.4 GHz band.

In the 5GHz band of the IEEE 802.11n protocol, it is available with 19 uninterrupted channels with a bandwidth of 20MHz and 9 channels with 40MHz bandwidth. For the 80MHz bandwidth, there are four of them. Figure 2.6 shows these WiFi channels for the *IEEE 802.11n* 5GHz band [47].

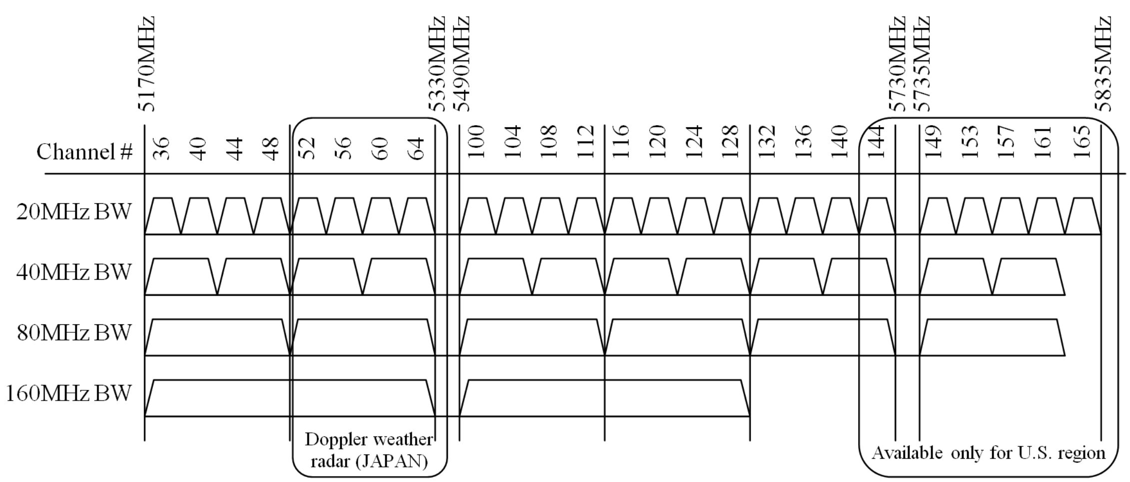


Figure 2.6: WiFi channels in 5 GHz band.

2.3 Features of IEEE 802.11n Protocol

The *IEEE 802.11n* protocol adopts several new technologies to improve the performance. The standard uses the MIMO technology, CB, frame aggregation, and security improvements mechanism to enhance the throughput. In this section, we describe these features of the *IEEE 802.11n* Protocol.

2.3.1 Channel Bonding

IEEE 802.11n introduces *channel bonding (CB)* as a new enhanced feature to increase network capacity. In the *CB*, two adjacent 20MHz channels are combined into a single channel to operate each channel at a 40MHz bandwidth [48] as shown in Figure 2.7. However, the use of channel bonding may reduce the number of non-interferred channels for other devices, since there are only two non-interfered bonded channels for *IEEE 802.11n* protocol in the 2.4GHz band. Table 2.4 summarizes the channel bonding for the 13 20MHz channels at 2.4GHz band [34, 49]. In Table 2.5, different channel bandwidths and spatial streams are described in regards to *IEEE 802.11n* throughput.

Table 2.4: Channel bonding in IEEE 802.11n.

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

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

Stream number	Bandwidth	
	20 MHz	40 MHz
one Streams	72.2$Mbps$	150$Mbps$
two Streams	144.4$Mbps$	300$Mbps$
three Streams	216.7$Mbps$	450$Mbps$
four Streams	288.9$Mbps$	600$Mbps$

2.3.2 Partially Overlapping Channels

In *IEEE 802.11n*, at the 2.4GHz band, each channel has a bandwidth of 20MHz and the adjacent channels are 5 MHz apart. Therefore, all the adjacent channels are partially overlapped with each other. Thus, each channel partially overlaps with at least three of its neighbors.

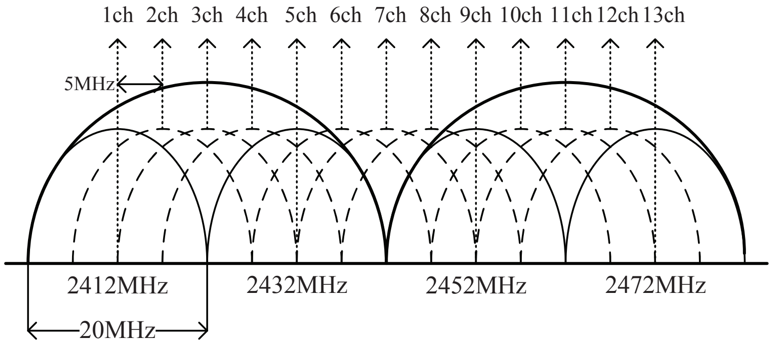


Figure 2.7: IEEE802.11n channel bonding concept.

2.3.3 Multiple Input Multiple Output (MIMO)

The MIMO is the another feature of *IEEE 802.11n* standard which improve the throughput drastically by increasing the number of transmitting (T_X) and receiving (R_X) antennas up to four times, without the additional bandwidth or transmission power. It can overcome the effects of multi-path and fading, and achieve the high data throughput in limited bandwidth channels. The performance can be improved over the single antenna technology in *single-input single-output (SISO)*. The comparison between SISO and 4×4 MIMO as shown in figure 2.8. When the *space-time block coding* (*STBC*) is applied in the 4×4 MIMO link, the sender can transmit four copies of the data streams over four antennas to enhance the reliability and the effective range of data transmissions.

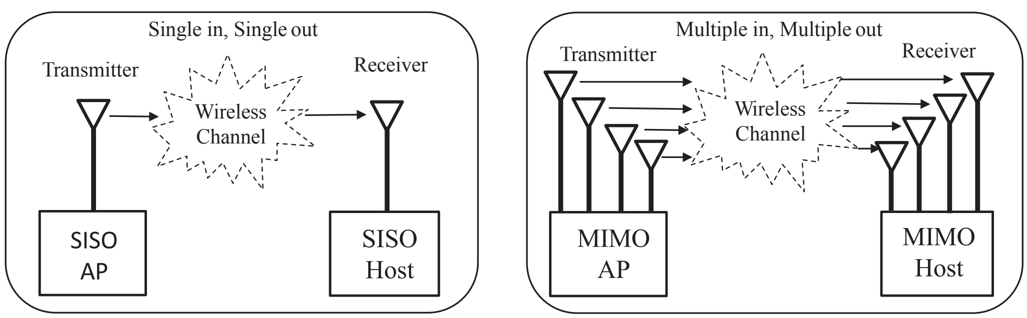


Figure 2.8: Comparison between SISO and 4×4 MIMO technology.

2.3.4 MAC Layer Enhancements

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

- *Frame Aggregation:*
  The IEEE 802.11n standard offers the performance improvement through frame aggregation in the MAC layer, in addition to MIMO. In frames aggregation, multiple frames are transmitted in a single big frame with a single pre-ample and header information to reduce their overhead. The IEEE 802.11n adopts the *Aggregation of MAC service data units (A-MSDUs)* and *aggregation of MAC protocol data units (A-MPDUs)*. Frame aggregation is a process

of packing multiple A-MSDUs and A-MPDUs together to reduce the overheads and average them over multiple frames, thereby increasing the user level data rate [50].

- *Modulation and Coding Scheme:*
  The *IEEE 802.11n* standard uses a number of modulation and error-correction codes, represented by a modulation and coding scheme (MCS) index value or *mode*. In *IEEE 802.11n*, 31 different modes are defined, which provides a higher level of protection against selective fading through the use of *orthogonal frequency division multiplexing (OFDM)*. This standard increases the number of OFDM sub-carriers of 56 (52 usable) in *high throughput (HT)* with $20MHz$ channel width and 114 (108 usable) in HT with $40MHz$. Each of these sub-carriers is modulated with *BPSK*, *QPSK*, *16-QAM*, or *64-QAM*, and *low-density parity-check code (LDPC) forward error correction (FEC)* coding rate of 1/2, 2/3, 3/4 or 5/6 [51]. The coding rates depends on link quality of the hosts. The good link quality can use higher coding rates and transmit more data. On the other hand, the worse the radio condition, the lower the coding rate and less data can be sent.

2.4 Linux Tools for Wireless Networking

Linux has been used as an open-source operating system for implementing algorithms, protocols, methods, and devices that have enabled the advancement of wireless networks [52]. This section describes the tools and software used in the thesis to perform the measurements throughout and to implement the *elastic WLAN system*.

2.4.1 'arp-scan' - to Explore Currently Active Devices

arp-scan [53] is a command line tool that uses the *address resolution protocol (ARP)* to detect and fingerprint the IP addresses of the hosts in the local area network. It works on both IEEE802.11 wireless network and wired Ethernet netowrk, where the wireless network uses the same data-link protocol. In *Linux*, *arp-scan* can be installed by downloading the source code from [54] or using the following command:

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

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

```
$ arp-scan --interface=wlan0 --localnet
```

–interface=eth0 defines the interface to be used for scanning the devices. The *arp-scan* uses *–localnet* to scan all the possible IP addresses in the network that connect to the interface, which is defined by the interface IP address and net mask. Here, the interface *wlan0* is used as an example.

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

The *hostapd* software enables a network interface card to serve as an AP and authentication server. It implements IEEE802.11 AP managements as well as other IEEE802.1X protocols and security applications. In *Linux*, *hostapd* can be installed by downloading the source code from [55] or using the following command:

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

The Linux PC can be setup as an AP using a command-line interface after installing this tool on a Linux PC with WLAN driver support for AP mode. The *hostapd* can be started or stopped by the following commands:

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

2.4.3 'ssh' - to Remotely Execute Command

Secure Shell (ssh) is a cryptographic network protocol that allows you to securely start a shell session on a distant machine [56, 57]. It operates in two parts: *SSH client* and *SSH server*, and it builds a secure connection between them over an unsafe network. The open source version of *ssh* is *OpenSSH* [58], it can be installed using the following command [59]:

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

The following command line example demonstrates how to use *ssh* to remotely access the AP across the network [56, 57, 60]:

```
$ ssh username@192.168.10.11
username@192.168.10.11's password:
```

Here, 192.168.10.11 represents the IP address of the AP.

2.4.4 'iw' - to Collect Information of Active Network Interface

iw [61] is a command-line Linux tool which allows to view and modify the parameters of the active network interface for wireless operations. It is usually installed by default in the *Ubuntu* distribution. The following command can be used to install it manually:

```
$ sudo apt-get install iw
```

The following list shows the use of *iw* to view the information of the currently associated AP using the network interface *wlan0*:

```
$ iw dev wlan0 scan
```

2.4.5 'iperf' - to Generate Network Traffic

iperf [62] is a software tool to measure the network throughput by generating network traffic. It supports both TCP and UDP protocols. *iperf* is normally installed by default in the *Ubuntu* distribution. It can also be installed manually using the following command:

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

To generate the traffics between two devices using *iperf*, one of them uses the server-mode and the other one uses the client-mode, where the data packets are transmitted from the client to the server. The following list shows the typical use of *iperf* on the server and client side:

```
$ iperf -s  //server side
$ iperf -c 192.168.10.11 //client side
```

Here, 192.168.10.11 defines the IP address of the server. In this thesis, we use *iperf* to generate the TCP traffic between an AP and a host through the *IEEE 802.11n* protocol.

2.4.6 'iftop' - to Measure Link Speed

iftop [63] is a real time network monitoring tool, which is installed at the AP only to estimate the throughput by capturing the transmitted packets between the server and the host [64]. It displays the bandwidth usage per host. Before installing it, two additional packages, *libpcap* and *libncurses*, need to be installed as the root user. The following commands show the procedure:

```
$ sudo apt-get install libpcap0.8 libpcap0.8-dev libncurses5
    libncurses5-dev
$ sudo apt-get install iftop
```

In this thesis, we use *iftop* to measure the throughput between an AP and a hosts. Figure 2.9 shows the output display of throughput monitoring by iftop.

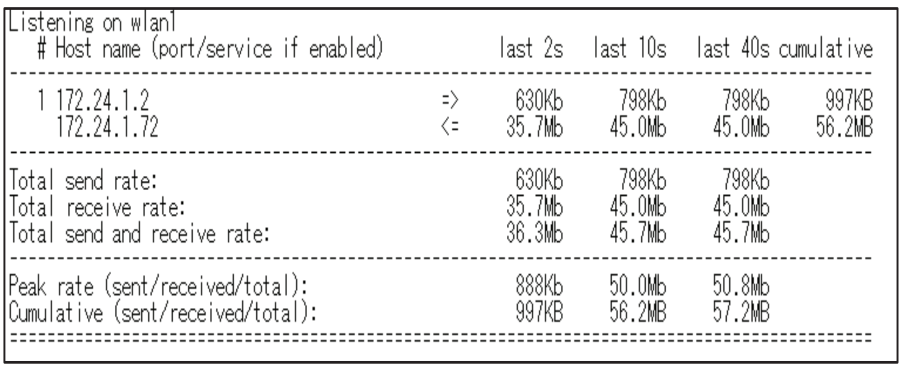


Figure 2.9: Throughput monitoring by *iftop*.

2.4.7 'traffic control (tc)' - to Manipulate Traffic Control Setting

The *tc* [65] tool is used to control network traffic by configuring the *Linux kernel*. It allows shaping, scheduling, policing, and dropping of the network traffic. These elements are used for controlling the network bandwidth to provide guaranteed service to the users. Traffic shaping permits to control the transmission rate, the scheduling technique arranges or rearranges the network traffic before they enter or leave various queues, a policer can limit traffic to a specific queue, dropping discards a packet with a certain criteria or overloaded period. The *tc* usually installed by default in the *Ubuntu* distribution. In *Ubuntu*, *tc* is bundled with the *iproute2* package and it can be installed manually using the following command:

```
$ apt-get install iproute2
```

2.5 Traffic Shaping

This section describes the *traffic shaping* that has been adopted for controlling the network traffic. *Traffic shaping* can control the packet flow rate at certain bit rate, to provide the dedicated bandwidth service for the specific user. In *Linux*, *traffic control (tc)* command can be used for *traffic shaping*. There are three components for the *tc* command, namely, *queueing discipline (qdisc)*, *classes*, and *filters*. The *qdisc* scheduler is categorized into two groups of *classless qdisc* and *classful qdisc*. The *classful qdisc* permits to categorize the traffic that have different treatments. In contrast, the *classless qdisc* does not allow to classify the traffic.

In this thesis, we adopt the *classful hierarchical token bucket (HTB) queuing discipline (qdisc)* [66] to control the traffics at the specific rate. The *HTB* uses *token buckets* for the link sharing classes. Figure 2.10 shows the hierarchical shape of *HTB qdisc*. In *HTB*, traffic shaping occurs only in the child class, not in the parent or root class. The child classes can borrow tokens from their parent class until the token is available in the parent class. Each child class contains two parameters, *ceil* and *rate*, to specify the amount of traffics allocated to each class. The *rate* refers to the guaranteed bandwidth of the whole class and the *ceil* refers to the maximum bandwidth of each traffic. In this thesis, we give the same value to them.

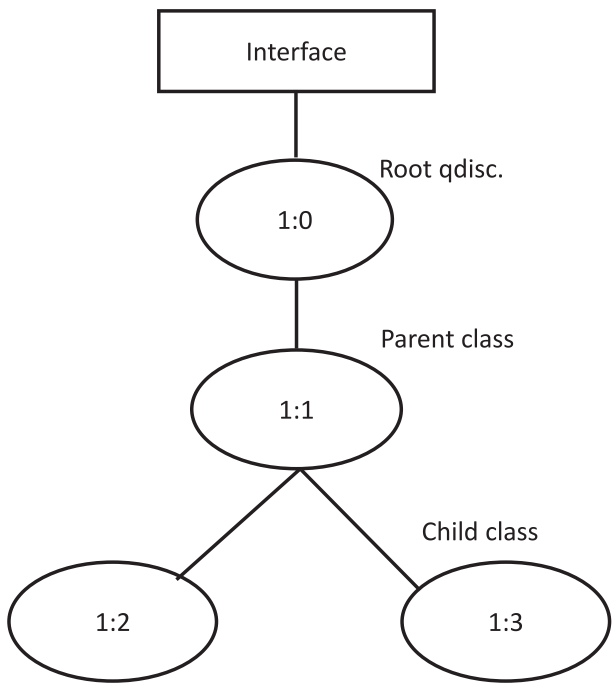


Figure 2.10: Hierarchical view of *HTB qdisc*.

2.6 Jain's Fairness Index

This section overviews the *Jain's fairness index* [67] that has been used as the throughput fairness indicator in experiments. The *Jain's fairness index* represents the fairness among the values with a real number between 1 and 0. 1 indicates that they are totally fair, and 0 does not at all. It is given by:

$$\textit{fairness index} = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \times \sum_{i=1}^{n} x_i^2} \tag{2.1}$$

where n represents the total number of throughputs of the hosts in this thesis and x_i does the i-th throughput.

2.7 Summary

In this chapter, we introduced the wireless network technologies, the main features of *IEEE 802.11n* protocol, *Linux* tools, *traffic shaping*, and *Jain's fairness index* which are adopted in this thesis for experiments, implementations, and simulations purposes. In the next chapter, we will review our previous studies related to this thesis.

Chapter 3

Review of Previous Studies

This chapter reviews briefly our previous studies relevant to this thesis. First, we review the study of the *elastic WLAN system* [13–15]. Then, we review the implementation details of the *elastic WLAN system* testbed using *Raspberry Pi* for APs [14,33]. Lastly, we review the study of the *TCP fairness control method* in WLAN [19,20].

3.1 Elastic WLAN System

This section reviews the *elastic WLAN system*. This system dynamically activates or deactivates the APs based on traffic demands and network conditions to increase the throughput performance and reduce the energy consumption.

3.1.1 Overview

WLANs have been widely used in many places including government offices, educational institutions, and public places such as trains, buses, or airplanes. In these instances, unplanned or autonomous APs can degrade performance or waste energy. WLANs can have over-allocation issues due to redundant APs having overlapping coverage areas. At the same time, WLANs can be overloaded with hosts and can be suffered from poor performances. Therefore, WLANs should be adaptive according to traffic demands and network conditions by changing the number of active APs and the hosts connected to them. Along this context, we have studied *elastic WLAN systems*.

The motivation for studying the *elastic WLAN systems* is summarized as follows:

1. Reduction of operating costs and energy consumption:

    - In general, organizations such as companies, educational institutions, and government offices allocate the high number of APs to offer the sufficient performance at the peak time, and to keep the WLANs running entire days. Limited APs may be used during the off-peak time and holidays. With the elastic WLAN system, this problem can be solved by minimizing the number of APs based on traffic demands.
    - The lack of reliable electricity supplies causes unreliable Internet connections in developing countries for the time being. In such situations, the *elastic WLAN system* can efficiently utilize the available power source to improve the network performance.

2. Enhancement of WLAN performance:

- When the current active APs are not sufficient to cover the user's demands, additional APs must be added to satisfy the required traffic demands.
- When the WLAN performance suffers due to shortages of *internet service provider (ISP)* connections or power supplies, mobile APs using cellular networks are activated to maintain the WLAN performance.
- In the dense WLAN using a high number of active APs, users may experience interferences due to overlapping signals. In such cases, the *elastic WLAN system* can dynamically change the channels that are assigned to APs, to reduce interferences and improve the WLAN performance.

3.1.2 Design and Operational Flow

Figure 3.1 shows an example topology of the *elastic WLAN system*. The management server has the administrative access to all the devices including the APs and hosts to control the network configuration through the following steps:

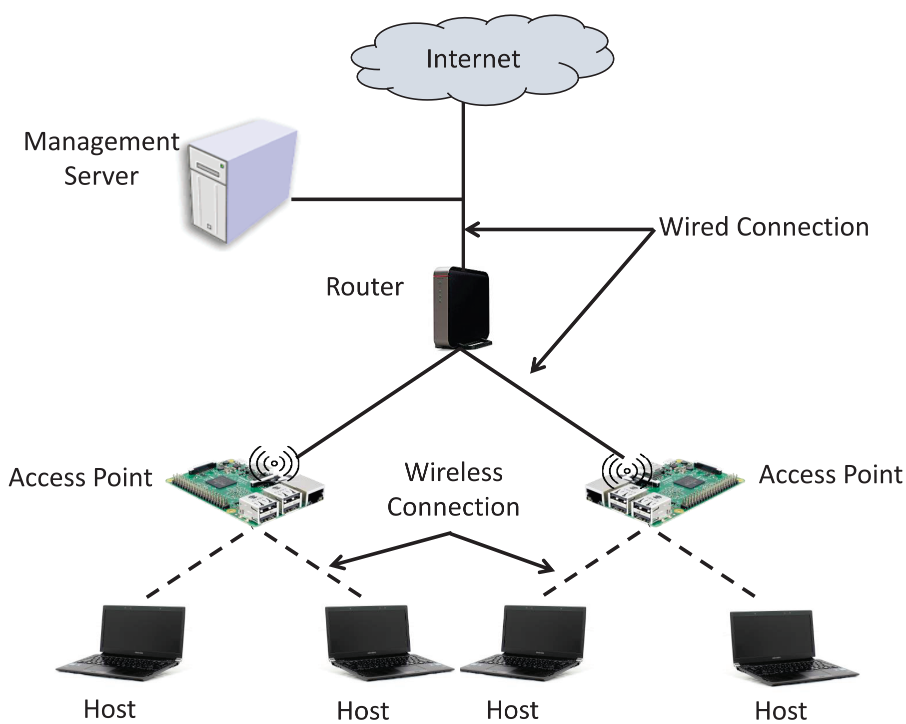


Figure 3.1: Design of Elastic WLAN system.

1. Initially, the server inspects the network devices and collects all the information required for the AP configuration algorithm.
2. Then, the server executes the AP configuration algorithm using the collected information. Therefore, the output of the algorithm contains a list of active APs, host associations, and allocated channels.
3. In the end, the server implements this output in the network by activating or deactivating the APs, changing the host associations, and allocating the channels.

3.2 Testbed Implementation using Raspberry Pi

This section discusses the *elastic WLAN system* testbed implementation using the *Raspberry Pi* and Linux PCs. *Raspberry Pi* is a single-board computer with a wireless network interface (NIC) that supports *IEEE 802.11n*. Hence, the use of *Raspberry Pi* in the elastic WLAN system is significant for the use at any country.

3.2.1 Implementation Environment/Platform

The *elastic WLAN system* testbed was initially implemented on the Linux operating system for the server, the APs, and the hosts. Linux has been adopted as the platform to implement new algorithms, protocols, methods, and devices for advancements of wireless networks. It offers many tools which are easily configurable and have flexibility to use and integrate with other tools. [52]. In this testbed, the system is implemented on Ubuntu, which is the most popular platform. Implementations of the *elastic WLAN* system testbed on various platforms will be in future studies.

Table 3.1 illustrates the devices and software are used for the testbed implementation of the system. The *IEEE 802.11n* protocol is used for any communication link with the channel bonding. *Raspberry Pi 3* with *TP-Link TL-WN722N* wireless NIC [68] adapter was used for the channel bonding.

Table 3.1: Device environment and software in testbed.

AP	
model	Raspberry Pi 3
CPU	Broadcom BCM2837 @1.2GHz
RAM	1GB LPDDR2 900MHz
NIC chipset	Broadcom BCM43438
Operating System	Linux Raspbian
software	hostapd
server and hosts	
model	1. Toshiba Dynabook R731/B 2. Toshiba Dynabook R734/K
CPU	1. Intel Core i5-2520M @2.5GHz 2. Intel Core i5-4300M @2.6Ghz
RAM	4GB DDR3 1333MHz
Operating System	Linux Ubuntu 14.04 LTS
software	iperf 2.0.5, iftop
main router	
model	Buffalo WZR-1750DHP

3.2.2 System Topology

Figure 3.1 illustrates the simple topology of the *elastic WLAN system*. *Raspberry Pi* is adopted for the AP and the *Linux laptop PC* is for the server and the hosts. With the administrative access to all the APs and hosts, the server can manage and control them. There is a wired connection between the APs and the server and the hosts and the APs are connected through wireless.

3.2.3 AP Configuration of Raspberry Pi

This section explains how to configure *Raspberry Pi* to use *hostapd* [69, 70] for AP.

1) Install the *hostapd* software on *Raspberry Pi* to enable the AP functionality by running the following command:

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

2) The following steps can be used to configure *hostapd*:

   i. Create the configuration file, *hostapd.conf*, inside */etc/hostapd/*, and set the necessary configuration options in this file to create the wireless network.
   ii. To start *hostapd* during booting, edit the *etc/default/hostapd/* file to set the absolute path to the configuration file as:
   DAEMON_CONF="/etc/hostapd/hostapd.conf",
   run

   ```
   $ sudo nano etc/init.d/hostapd/
   ```

   and change
   DAEMON_CONF=/etc/hostapd/hostapd.conf.

3) Set up the *Raspberry Pi's* wireless connection as static by editing the *"/etc/network/interfaces"* file and adding the static IP address information.
4) Install the DHCP server that allows wireless connections to automatically receive the dynamic IP addresses by using the command:

```
$ sudo apt-get install isc-dhcp-server
```

5) Configure the DHCP server using the following steps:

   i. Edit the *dhcpd.conf* file inside the folder */etc/dhcp/* and put necessary configuration options.
   ii. Edit the */etc/default/isc-dhcp-server* file as follows to make the wireless adapter the default for DHCP requests:
   INTERFACE="wlan0".

6) Configure the *network address translator (NAT)* by editing the */etc/sysctl.conf* file, so that multiple clients can be connected with the AP and have all data 'tunneled' through the single Ethernet IP address.

3.3 TCP Fairness Control Method

This section reviews the study of the *TCP Fairness Control Method*. This method can achieve the relatively fair throughput among the hosts when they are concurrently communicating with the single AP.

3.3.1 Overview of TCP Fairness Method

In WLAN, the throughput unfairness occurs when multiple hosts communicate with the same AP simultaneously. WLAN should provide fair throughput among the communicating hosts. To achieve this goal, we studied the method of controlling the packet transmission delay at one AP using *PI controller*.

3.3.2 Initial Delay Calculation

This method initially calculates the packet transmission delay based on measured RSS of the hosts by using the Eq. (3.1). Then, it dynamically changes the delay using the *PI controller* [71] to achieve the fair target throughput among the hosts. The *PI controller* calculates an error value as the difference between a measured throughput and a desired target throughput and minimizes the error by adjusting the process control inputs.

$$D_i(0) = \frac{RSS_i}{-x}\left(\frac{RSS_i}{RSS_{min}}\right)^2 e^{y(RSS_i - RSS_{slow})} \tag{3.1}$$

where RSS_i expresses the measured *RSS* at the AP from host i, RSS_{slow} does *RSS* from the slowest host among the hosts, RSS_{min} does the minimum *RSS* that a host can successfully receive packets from the AP, and x and y define the constant parameters.

3.3.3 Target Throughput

In this method, the target throughput is calculated with the following equation in Eq. (3.2).

$$TH^{tar}(0) = \frac{1}{\sum_{i=1}^{N} \frac{1}{z_i}}(1 - \delta N) \tag{3.2}$$

where z_i expresses the estimated throughput for host i by using the *throughput estimation model* [72], N does the total number of hosts associated with the AP, and δ defines the constant interference-loss parameter.

Then, the target throughput is updated based on the measured target throughput by using the following equation in Eq. (3.3).

$$TH^{tar}(m) = \frac{\sum_{i=1}^{N} TH_i(m)}{N} \tag{3.3}$$

Here, $TH_i(m)$ expresses the measured throughput for host i at time-step m.

3.4 Summary

In this chapter, firstly, we reviewed the study of the elastic WLAN system. Then, we reviewed the implementation details of the elastic WLAN system. Lastly, we reviewed the TCP fairness control method. However, the TCP fairness method was limited to one AP in the network, despite the fact that multiple APs are common in WLANs and can be interfered with each other frequently, and may have the slow convergence to achieve the fair throughput. In addition, it is hard to allocate demand throughput to the host even if necessary, and cannot be applied when saturated host appears in WLAN. In the next chapter, we will introduce the throughput unfairness problem of concurrent communications with multiple hosts.

Chapter 4

Experimental Observations of Throughput Unfairness Problem of Concurrent Communication with Multiple-Hosts

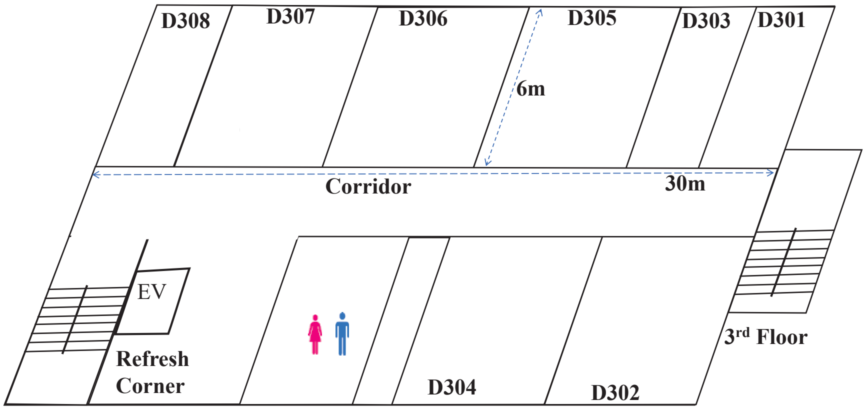
This chapter presents experimental observations of the throughput unfairness problem of multiple hosts communicating concurrently. First, we show the throughput results when multiple hosts concurrently communicate with a single AP. Then, we present the throughput results when the multiple hosts concurrently communicate with multiple APs.

4.1 Throughput Unfairness When Hosts Concurrently Communicate with Single AP

In WLAN, the throughput unfairness may appear among the hosts when they concurrently communicate with the same AP at the different relative distances from it.

4.1.1 Experiment Setup and Field

In our experiments, we performed throughput measurements of concurrently communicating two hosts with the same AP in the corridor of Engineering Building #2 in Okayama University (indoor environment) and Asahi riverbed (outdoor environment). Figure 4.1(a) illustrates the experiment field for indoor environment where interferences from other WLANs exist, while Figure 4.1(b) does the outdoor environment without any interference. In both fields, the hosts communicate with the *Raspberry Pi* AP using the *IEEE802.11n* 20MHz channel at 2.4GHz. In this case, we used single spatial stream. The *iftop* is adopted to measure the throughput, and and the same TCP traffic was generated using *iperf 2.0.5 software* with $477KB$ TCP window size and $8KB$ buffer size. Figure 4.2 illustrate the network topology for single AP with two hosts. The *Toshiba Dynabook R731/B* laptop PCs are used for the server and the hosts. Table 4.1 shows the detail of the hardware and software specifications. Figure 4.3 describes the locations of the AP and two hosts in the measurements. The host H_1 is fixed at 0m distance from the AP, and the host H_2 is moved from 0m to 20m with the 5m interval from the AP.



(a) Indoor environment.

(b) Outdoor environment.

Figure 4.1: Experiment fields.

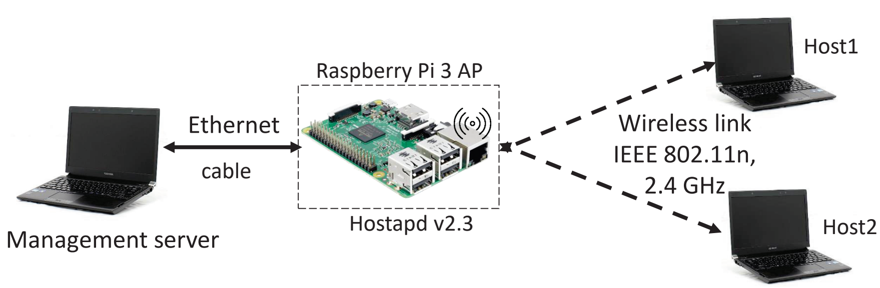


Figure 4.2: Testbed topology for single AP with two hosts.

4.1.2 Throughput Results

Figure 4.4 shows throughput measurement results for both indoor and outdoor environments. The results show that the throughputs are similar at both hosts when they are located with the same distance ($0m$) from the AP. However, the throughput difference between the two hosts increases as the distance between H_2 and the AP increases. It is noticed that the throughput of H_1 increases,

Table 4.1: Hardware and software specifications.

access point	
model	Raspberry Pi 3
CPU	BCM2837 1.2GHz, Broadcom
RAM	LPDDR2 900MHz 1GB
NIC chipset	1. BCM43438, Broadcom 2. Atheros AR9002U (USB)
Operating System	Linux Raspbian
software	hostapd, iftop
server and hosts	
model	1. Toshiba Dynabook R731/B 2. Toshiba Dynabook R734/K 3. Fujitsu Lifebook S761/C
CPU	1. Intel Core i5-2520M @2.5GHz 2. Intel Core i5-4300M @2.6GHz 3. Intel Core i5-2520M @2.5GHz
RAM	4GB DDR3 1333MHz
Operating System	Linux Ubuntu 14.04 LTS
software	iperf 2.0.5
main router	
model	Buffalo WZR-1750DHP

although the location is fixed in both network fields, which leads to the throughput unfairness between the hosts.

4.2 Throughput Unfairness When Hosts Concurrently Communicate with Multiple APs

In WLAN using multiple APs and channel bonding (CB) channels, the throughputs are much different among concurrently communicating hosts due to their unequal *received signal strength*

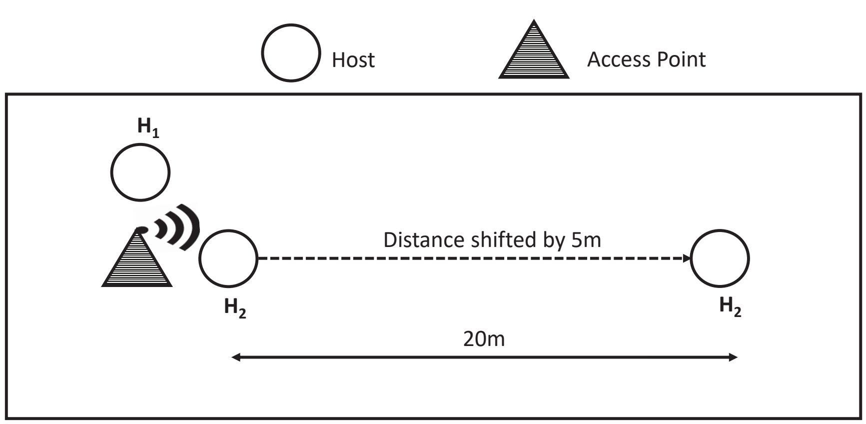
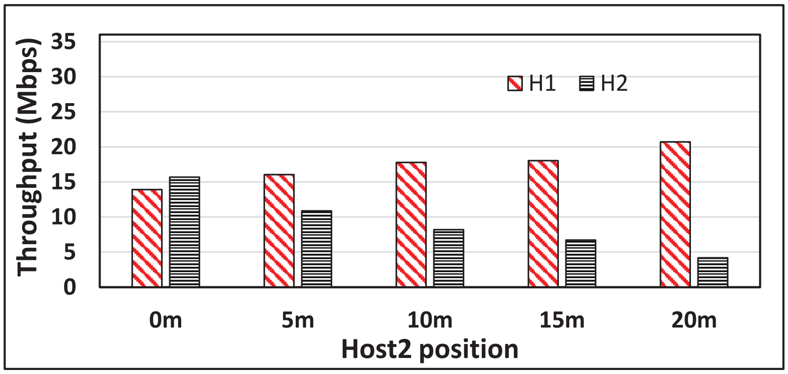
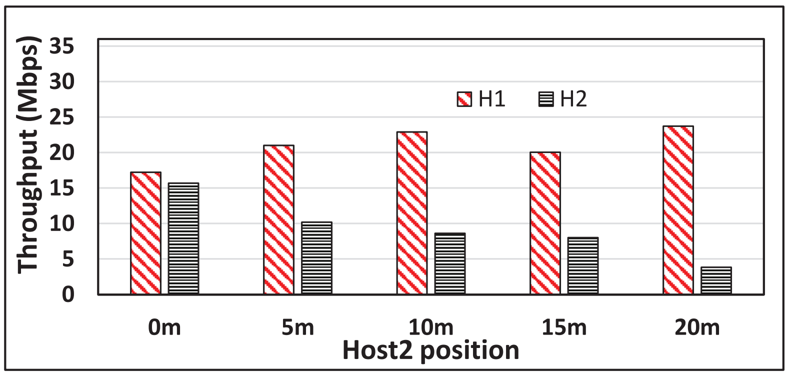


Figure 4.3: Location of the AP and hosts in the experiment field for single AP.



(a) Indoor result.



(b) Outdoor result

Figure 4.4: Throughput unfairness observations between two hosts for single AP.

(RSS) and interferences if the distances between the hosts and the APs are different.

4.2.1 Experiment Setup and Field

We conducted experiments of measuring the throughputs of three concurrently communicating hosts with three different APs. Figure 4.5 shows the experiment field in the third floor of Engineering Building #2 at Okayama University. The host locations are depicted by the circles and the AP locations are by the triangles. The distance between the host and the AP was different from each other. The host1 (H_1) was connected to AP_1 from a distance of 9.5m, host2 (H_2) was to AP_2 from a distance of 0.5m, and host3 (H_3) was to AP_3 from a distance of 5m. Figure 4.6 shows the testbed topology for three APs with three hosts. The *Toshiba Dynabook R731/B* and *Fujitsu Lifebook S761/C* laptop PCs are used for the hosts and the server, respectively. Table 4.1 shows the detail of the hardware and software specifications. The *TCP traffic* is generated by *iperf* software where the *iftop* is used for throughput measurement.

Three *Raspberry Pi 3* were used for experiments. However, the *Raspberry Pi 3* does not support *channel bonding (CB)*. Thus, we used *USB NIC* [68] adapter for CB channels. The *IEEE802.11n* protocol was adopted with *40MHz* bonded channels at 2.4GHz. To reduce the interferences as much as possible, the three bonded channels, 1 + 5, 9 + 13, and 5 + 9, are assigned to AP_1, AP_2, and AP_3, respectively, by following the assignments in [73]. The following Linux commands are adopted for running CB channel at the AP.

```
ieee80211n=1
channel=1
```

```
ht_capab=[HT40+][SHORT-GI-20][SHORT-GI-40][DSSS_CCK-40][MAX-AMSDU-3839]
```

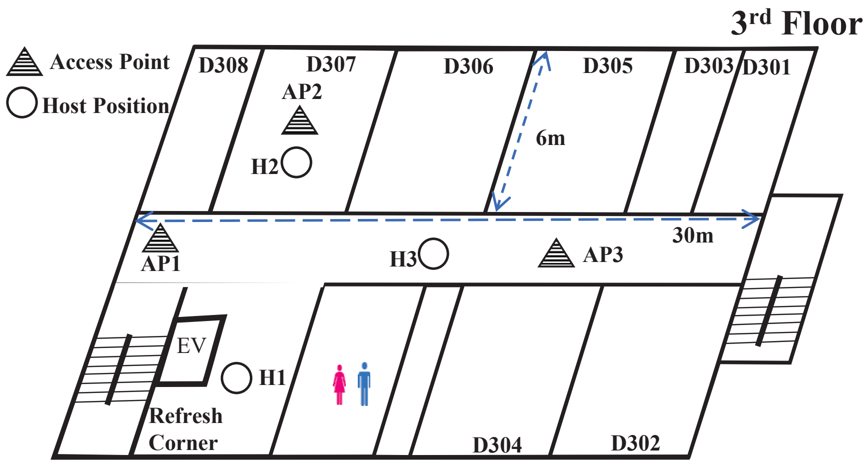


Figure 4.5: Experiment field for multiple APs.

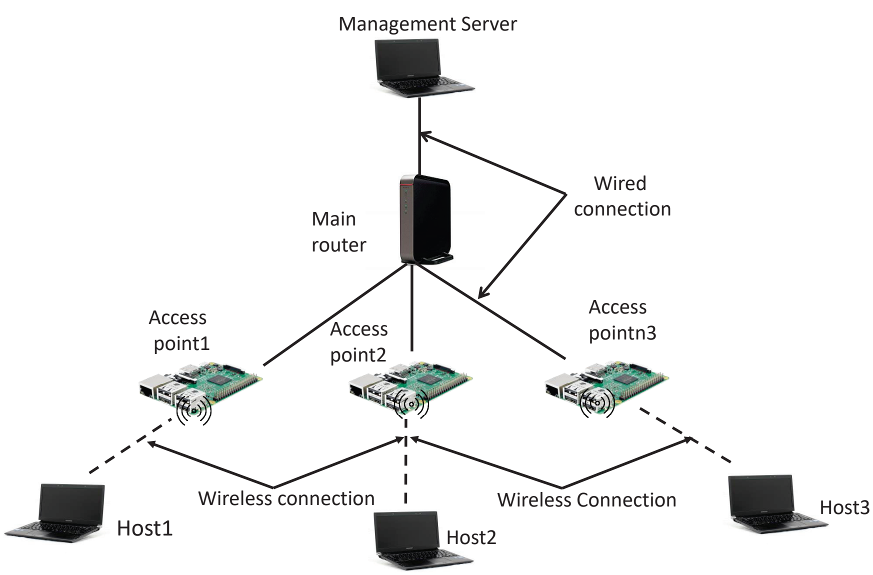


Figure 4.6: Testbed topology for three APs with three hosts.

4.2.2 Throughput Results

From Figure 4.7, the *throughput unfairness* among the hosts was observed. The distance between AP_2 and H_2 was smallest. Thus, the throughput of the link $AP_2 - H_2$ was largest among the three links. On the other hand, the throughput of the link $AP_1 - H_1$ was smallest because the distance was longest and the elevator attenuated the signal as the obstacle of the transmission path. Additionally, interference caused by co-located APs also causes throughput disparity. However, it is desired that the *throughput fairness* among them should be available even in this network with the objective of universal services among the users.

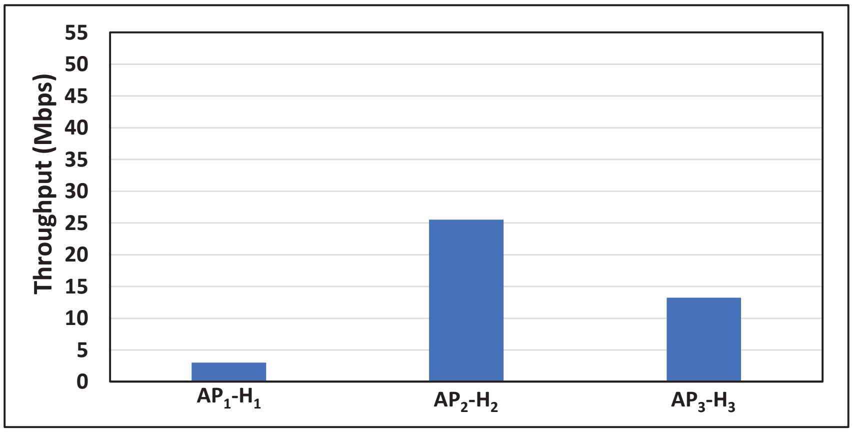


Figure 4.7: Throughput unfairness observations in dense WLANs at concurrent communication.

4.3 Summary

In this chapter, we presented experimental observations of the throughput unfairness problem of multiple-host concurrent communications. First, we presented the throughput results when multiple-hosts concurrently communicate with single AP. Then, we showed the throughput results when multiple-hosts concurrently communicate with different multiple APs. In the next chapter, we will propose the *fair throughput control method* for multiple-hosts concurrent communications in WLAN.

Chapter 5

Proposal of Fair Throughput Control Method

This chapter presents the *fair throughput control method* for multiple-hosts concurrently communicating in WLAN. First, we describe the observations for proposal and *channel occupying time* of hosts. Then, we present the calculation of *equal target throughput provision* as the common goal among the hosts.

5.1 Introduction

In Chapter 4, we observed that the throughput unfairness issue occurs among multiple hosts when they communicate with a single AP or closely-located multiple APs at the same time. The testbed experiment results showed that the throughput is not fairly distributed between the two hosts when they are concurrently communicating from different relative distances with a single AP. In addition, it is found that when multiple APs and CB channels are adopted in WLAN, the throughputs are different greatly among the concurrently communicating hosts due to the unequal received signal strengths at the hosts and to the interferences from the APs. The hosts far from the AP/APs receive lower *RSS* than the hosts near to them, which leads to the use of a slower *modulation and coding scheme (MCS)* and to the lower throughputs. The near hosts get higher chances to transmit/receive the packets than the hosts distant from the AP. Hence, it occupies a large amount of bandwidth during communication compared to far hosts.

In this chapter, we propose the *fair throughput control method* of adopting *traffic shaping* [23] at the AP/APs to provide the relatively fair throughput among the concurrently communicating hosts with single/multiple APs. *Traffic shaping* can limit the bandwidth of the fastest link by controlling the packet transmissions and provide higher opportunities for the slow links to send more packets. First, this method calculates the *equal target throughput* for the throughput fairness by measuring the single and concurrent throughputs and estimating the *channel occupying time* for each host. Then, it achieves the fair throughput by applying *traffic shaping* at the AP.

In WLAN, a host may not need or cannot achieve the fair throughput with the other hosts that are associated with the same AP. When a host is taking a low-rate application service, it needs the small bandwidth constantly. Besides, if the connected server is very crowded, the host cannot consume the allocated throughput. In this thesis, such a host is called the *saturated host*, and the maximum achieved throughput is the *saturated throughput* for convenience. To avoid wasting the bandwidth, the *saturated throughput* is assigned to the *saturated host*, and the remaining bandwidth should be shared among the other hosts in WLAN.

5.2 Observations for Proposal

The proposed method is designed from the following observations:

(1) The throughput of each host can be controlled by running *tc command* at the *Linux-based* AP.

(2) The *maximum number of transmitted bits per second (bps)* for each host can be measured by running *iftop* at the AP when only one AP is active and communicating with a single host, which is called the *single throughput*.

(3) The *actual number of transmitted bits per second* for each host can be measured by running *iftop* at each AP when all the hosts are concurrently communicating, which is called the *concurrent throughput*.

(4) The *target throughput* for each host is represented by the number of bits to be transmitted per second.

(5) The *channel occupying time* per second for each host can be estimated by the ratio of the *concurrent throughput* or the *target throughput* to the *single throughput* when all the hosts are concurrently communicating.

(6) The sum of the *channel occupying time* by every host can be constant (basically, one second).

(7) The *target throughput* for each host cannot exceed the *single throughput*.

(8) The *target throughput* that is larger than the *concurrent throughput* for a host can be realized by taking the channel occupying time of the other hosts, which determines the proper *target throughput* for each host.

5.3 Single and Concurrent Throughput Measurement

In the proposed method, first, the *single throughput* and the *concurrent throughput* for every host are measured at the target AP, to calculate the proper target throughput for each host. The *single throughput* is measured for each host by limiting only the host to communicate with the AP. The *concurrent throughput* is measured by activating all the hosts to communicate with the AP/APs.

5.4 Channel Occupying Time of Hosts

Let S_i and C_i be the measured *single throughput* and the measured *concurrent throughput* for the host H_i for $i = 1, 2, ..., n$, respectively, where n is the number of hosts. When all the hosts H_1, H_2, ..., H_n are concurrently communicating with the same AP through the shared single channel, the channel occupying time by each host for one second can be estimated by $\frac{C_1}{S_1}$, $\frac{C_2}{S_2}$, ..., $\frac{C_n}{S_n}$. Then, the summation of them will become constant, basically one second, as follows:

$$\frac{C_1}{S_1} + \frac{C_2}{S_2} + ... + \frac{C_n}{S_n} = Constant \tag{5.1}$$

The *CSMA/CA protocol* in the WLAN activates the wireless links between the AP and the associated multiple hosts in turns. It basically repeats the *data transmission* of one host through the channel and the *channel idling* for the contention resolution. During the unit time of one second, the average *data transmission time* of the link with the host H_i can be estimated by $\frac{C_i}{S_i}$, because $C_i Mbit$ data is transmitted through the $S_i Mbps$ link. The *channel idling time* can be constant when the number of the contending hosts is constant, because each contention resolution time in the CSMA/CA protocol can be constant on average. To achieve the throughput request, the proposed method does not change the number of the contending hosts. It only changes the *data transmission time* of links while keeping their communications. As a result, the *channel idling time* is not changed before and after applying the proposed method. Thus, for simplicity, the *channel idling time* is neglected in this equation.

5.5 Equal Target Throughput Provision

Here, we discuss the calculation of the *target throughput* when all the hosts be assigned the same target throughput: $t_1 = t_2 = ... = t_n$. Then, to transmit t_1, t_2, ..., t_n data through S_1, S_2, ..., S_n link, the *channel occupying time* for the hosts will be $\frac{t_1}{S_1}$, $\frac{t_2}{S_2}$, ..., $\frac{t_n}{S_n}$. Therefore, their sumuation will again be constant, as follows:

$$\frac{t_1}{S_1} + \frac{t_2}{S_2} + ... + \frac{t_n}{S_n} = Constant \tag{5.2}$$

Therefore, the following result is obtained from Eq. (5.1) and Eq. (5.2), because their time must be equal.

$$\frac{C_1}{S_1} + \frac{C_2}{S_2} + ... + \frac{C_n}{S_n} = \frac{t_1}{S_1} + \frac{t_2}{S_2} + ... + \frac{t_n}{S_n} \tag{5.3}$$

5.5.1 Conventional Host Case

When there is no saturated host in the WLAN, the following result is derived from Eq. (5.3) by assigning the same target throughput to the all hosts :

$$t_1 = t_2 = ... = t_n = \frac{\sum_{i=1}^{n} \frac{C_i}{S_i}}{\sum_{i=1}^{n} \frac{1}{S_i}}. \tag{5.4}$$

5.5.2 Saturated Host Case

If the saturated host (let H_k) exists in the WLAN where the derived target throughput is larger than its *single throughput* S_k, the target throughput for each host is updated by the following procedure to avoid the bandwidth waste:

$$t_k = S_k,$$

$$t_1 = t_2 = ... = t_n = \frac{1}{\sum\limits_{\substack{i=1 \\ i \neq k}}^{n} \frac{1}{S_i}} (\sum_{i=1}^{n} \frac{C_i}{S_i} - 1). \tag{5.5}$$

5.6 Summary

In this chapter, we presented the *fair throughput control method* for multiple hosts concurrently communicating in WLAN. First, we discussed the observations for the proposal and introduced the *channel occupying time* of hosts. Then, we presented the *equal throughput control method* for the hosts. In the next chapter, we will present the proposal of the *demanding throughput control method* for multiple-hosts concurrent communications in WLAN.

Chapter 6

Proposal of Demanding Throughput Control Method

This chapter presents the *demanding throughput control method* for concurrently communicating multiple-hosts in WLAN.

6.1 Introduction

To address the issue of throughput unfairness, in Chapter 5, we proposed the *fair throughput control method* for concurrently communicating multiple-hosts in WLAN. The *traffic shaping* is adopted at the AP and control the fastest link traffic in order to give higher chance to the slowest link. However, in WLAN, the fair throughput is not always necessary since the host may need higher throughput than others depending on the type of user and the application they are using. The required throughput should be allocated to the *high priority host* by reducing the throughputs of *low priority hosts*, while they enjoy the *minimum guaranteed throughput*.

In this chapter, we propose the *demanding throughput control method* for concurrently communicating multiple-hosts with the AP. This method is designed based on observations in Section 5.2. Our method tries to satisfy the required throughput by allocating the *channel occupying time* properly to the concurrently communicating hosts. If the required throughput of a host is higher than the current one, the time allocated to the host will be increased, while the other hosts will be decreased. The proper *target throughput* for each host is estimated by measuring the single and concurrent throughput and calculating the required *channel occupying time* to accomplish the required target throughput. This method achieves the *target throughput* by applying the *traffic shaping* at the AP, it directly controls the throughput for each host.

6.2 Demand Target Throughput Provision

Here, we discuss the calculation of the target throughput for H_2, H_3, ..., H_n when the target throughput t_1 of H_1 will be controlled. t_1 must not be larger than the *single throughput* S_1 and must not be smaller than the *minimum target throughput* t_{min}. In this thesis, this *minimum target throughput* is introduced to guarantee the least throughput for any host, when some hosts request the high target throughput. Then, since another equation is necessary to give the unique values of t_2, t_3, ..., t_n for the given t_1, the equal target throughput is considered for their fairness: $t_2 = t_3 = ... = t_n$.

6.2.1 Conventional Host Case

When there is no saturated host in the WLAN, the following result is derived from Eq. (5.3):

$$t_2 = t_3 = \dots = t_n = \frac{1}{\sum_{i=2}^{n} \frac{1}{S_i}} \left(\sum_{i=1}^{n} \frac{C_i}{S_i} - \frac{t_1}{S_1} \right). \tag{6.1}$$

6.2.2 Saturated Host Case

If the saturated host (let H_k) exists in the WLAN where the derived target throughput is larger than its single throughput S_k, the target throughput for each host except t_1 (because t_1 is demanded throughput for H_1) is updated by the following procedure to avoid the bandwidth waste:

$$t_k = S_k,$$

$$t_2 = t_3 = \dots = t_n = \frac{1}{\sum_{\substack{i=2 \\ i \neq k}}^{n} \frac{1}{S_i}} \left(\sum_{i=1}^{n} \frac{C_i}{S_i} - 1 - \frac{t_1}{S_1} \right). \tag{6.2}$$

6.2.3 Minimum Target Throughput Case

If the derived target throughput for H_2, H_3, ..., H_n becomes smaller than the *minimum target throughput* t_{min}, the target throughput for every host is updated by the following procedure to ensure it.

If $S_k < t_{min}$, $t_k = S_k$, and assign the $t_2 = t_3 = \dots = t_n = t_{min}$ in Eq. (6.2) to updates the target throughput for ensuring minimum throughput and derived the equation as follows:

$$t_1 = \sum_{i=1}^{n} \frac{S_1 C_i}{S_i} - S_1 - \left(\sum_{\substack{i=2 \\ i \neq k}}^{n} \frac{t_{min} S_1}{S_i} \right). \tag{6.3}$$

Otherwise, updates the target throughput by assigning $t_2 = t_3 = \dots = t_n = t_{min}$ in Eq. (6.1) and derived the following equation:

$$t_1 = \sum_{i=1}^{n} \frac{S_1 C_i}{S_i} - \left(\sum_{i=2}^{n} \frac{t_{min} S_1}{S_i} \right). \tag{6.4}$$

6.3 Summary

In this chapter, we proposed the *demanding throughput control method* for concurrently communicating multiple-hosts in WLAN. In the next chapter, we will present the implementation and evaluation of the *throughput control method* for concurrently communicating multiple-hosts in WLAN.

Chapter 7

Implementation and Evaluation of Throughput Control Method

This chapter presents the implementation and evaluation of *throughput control method* in WLAN. First, we describe the implementation procedure of the proposed method in WLAN. Then, we evaluate the proposal through several testbed experiments using single or multiple *Raspberry Pi* APs.

7.1 Implementation Procedure of Throughput Control Method

The proposed method adopts the Linux command *tc* to apply the *traffic shaping* at the software AP using *Raspberry Pi*, which can be easily run from the application program implemented using *bash script*. Figure 7.1 illustrates the flow of the whole procedure in the proposed method. The following procedure describes how to apply the proposed method.

(1) Measure the *single throughput* for each host by activating only one AP and communicating it with the host.

(2) Measure the *concurrent throughput* for every host by activating all the APs and communicating them with their associated hosts simultaneously.

(3) Derive the *target throughput* t_i to a host.

(4) Allocate t_i to every host by assigning initial *rate and ceil* value $d_i = t_i$.

(5) Apply *traffic shaping* using the *tc command*.

(6) Measure the throughput for every host periodically while all the APs are concurrently communicating.

(7) Apply *PI feedback controller* to update d_i.

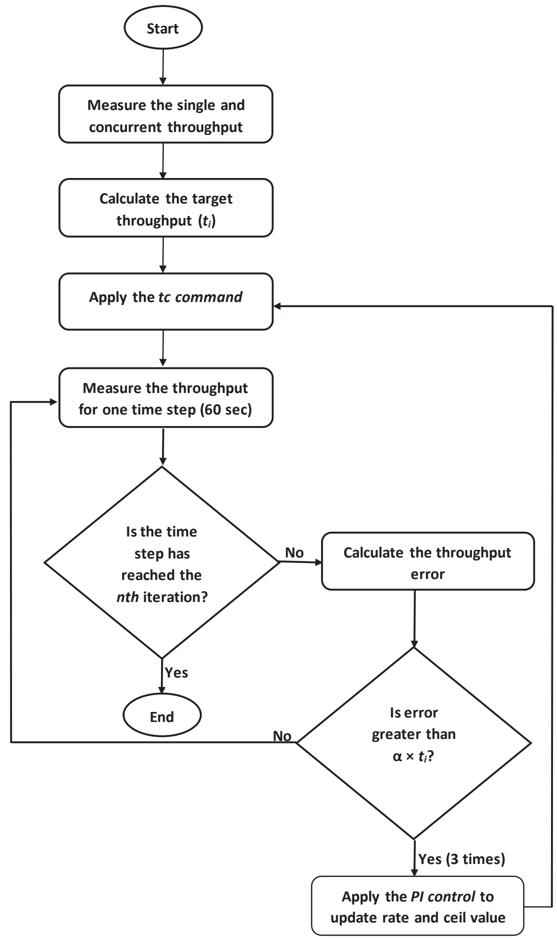


Figure 7.1: Flow of throughput control method.

7.1.1 Application of Traffic Shaping

The traffic shaping is applied using the *tc* command with the following procedure:

(1) First, create the *HTB qdisc*, generate the required number of classes for each host i, and assign the *rate* value d_i by:

- $sudo tc qdisc add dev wlan0 root handle 1: htb default.
- $sudo tc class add dev wlan0 parent 1: classid 1:1 htb rate $\sum_{i=1}^{n-1} d_i$
- $sudo tc class add dev wlan0 parent 1:1 classid 1:i htb rate d_i ceil d_i.

(2) Then, apply the d_i to the host H_i by specifying the IP address:

- $sudo tc filter add dev wlan0 protocol ip parent 1:0 prio 1 u32 match ip dst $IP of\ H_i$ flowid 1:i.

7.1.2 Optimize the Rate and Ceil Parameter value

In *traffic shaping*, the *rate* and *ceil* parameter value d_i can control the maximum bandwidth of the host at communications. Unfortunately, it does not guarantee the given specific throughput. The measured throughput will be fluctuating during communications. To address this inconvenience, the *PI feedback control* [20, 71] is adopted to dynamically update d_i in order to make the measured throughput equal to the target.

$$d_i(m) = K_P \times (t_i - R_i(m)) + K_I \times \sum_{j=0}^{m} (t_i - R_i(j)) \,. \tag{7.1}$$

where $R_i(m)$ does the measured throughput at *time step m* , and K_p and K_I express the *proportional (P)* and *integral (I)* control gain respectively ($K_P = 0.4$, $K_I = 0.5$ in this paper). The *time step* represents the time interval (60*sec* in this paper). To abstain the frequent changes of d_i, the Eq. (7.2 is used only when the throughput error $|R_i(m) - t_i|$ is greater than the given threshold $\alpha \times t_i$ for three continuous *time steps* and $\alpha = 0.2$ does the constant parameter. The updated value of d_i must be greater than or equal to the t_i . In the implementation, the following difference equation of the PI control is applied by considering the throughput difference between the current and previous steps rather than the error between target and measured throughput in *P control*, since higher value make the controller more aggressive and may adversely affect the throughput of other hosts [33].

$$d_i(m) = d_i(m-1) + K_P \times (R_i(m-1) - R_i(m)) + K_I \times (t_i - R_i(m)) \,. \tag{7.2}$$

7.2 Evaluation with iperf Traffics using Single AP

This section evaluates the proposal through testbed experiments using *iperf* traffics with one AP and up to five hosts.

7.2.1 Experiment Setup

Figure 7.2 and Table 4.1 show the network topology and the hardware and software specifications of the testbed system, respectively. *Raspberry Pi 3* is adopted as the software AP and the *Linux based* PCs are for the hosts and the management server. The *Toshiba Dynabook* laptop PCs are used for the hosts and the server. Table 7.1 describes the locations of the hosts and AP in the experiments where the indoor field of Engineering Building #2 at Okayama University in Figure 4.1 (a) was used. The measured throughput was often fluctuated. To improve measurement the accuracy, the throughput measurement for each scenario was repeated twelve times and their average result was used in evaluations. One measurement took one minute. Thus, the total measurement time for each scenario was twelve minutes.

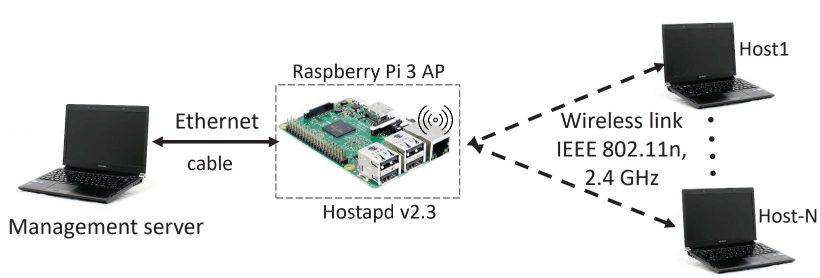


Figure 7.2: Testbed topology for iperf traffics.

Table 7.1: Device locations.

case	device location		
	D307	in front of D307	refresh corner
2 hosts	AP, H_1	–	H_2
3 hosts	AP, H_1, H_3	–	H_2
4 hosts	AP, H_1, H_3	H_4	H_2
5 hosts	AP, H_1, H_3	H_4	H_2, H_5

7.2.2 Experiment Scenarios

In our experiments, the five scenarios on target throughput conditions in Table 7.2 were assumed. In any scenario, the same TCP traffic was generated using *iperf 2.0.5 software* with $477KB$ TCP window size and $8KB$ buffer size. In this thesis, $t_{min} = 1.5Mbps$ was used for Table 7.2.

Table 7.2: Target throughput conditions in five scenarios.

scenario	condition
1) equal throughput	$t_1 = t_2 = t_3 = t_4 = t_5$
2) high priority host A	$t_1 > t_i$ and $t_i > t_{min}$
3) high priority host B	$t_1 > t_i$ and $t_i < t_{min}$
4) low priority host A	$t_1 < t_i$ and $t_i > t_{min}$
5) low priority host B	$t_1 < t_i$ and $t_1 = t_{min}$

1) Equal Throughput: All the hosts are assigned the same throughput. This scenario intends to examine the throughput fairness request among the hosts.

2) High Priority Host A: The fastest host H_1 is considered as the *high priority host* and is assigned the higher target throughput than the other hosts that are assigned the same throughput. This scenario intends to examine the simultaneous requests of the high throughput provision and the fairness among the hosts.

3) High Priority Host B: The same throughput setup is considered here except for the condition that the original target throughput by the proposal does not meet the minimum target throughput. Thus, *Minimum Target Throughput Case* in Section 6.2.3 is applied here.

4) Low Priority Host A: The fastest host H_1 is considered as the *low priority host* and is assigned the lower target throughput than the other hosts that are assigned the same throughput. This scenario intends to examine the simultaneous requests of the low throughput provision and the fairness among the hosts.

5) Low Priority Host B: The same throughput setup is considered here except for the condition that the target throughput for H_1 is considered as the *minimum target throughput*.

7.2.3 Throughput Results

Figure 7.3 illustrates the single throughput measurement results for the five hosts and the concurrent results for two, three, four, and five host cases with *iperf* traffics. Figures 7.4-7.7 show the individual host throughput results for concurrently communicating two, three, four, and five host cases, respectively. In each graph, *target thr.* represents to the derived target throughput by the proposal and *measur. thr.* does the measured throughput. The *updated target thr.* indicates that the *Minimum Throughput Case* was applied there.

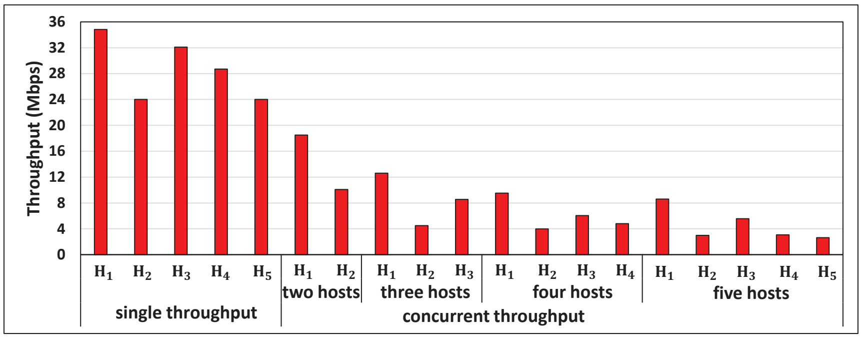


Figure 7.3: Measurement results of single and concurrent throughputs.

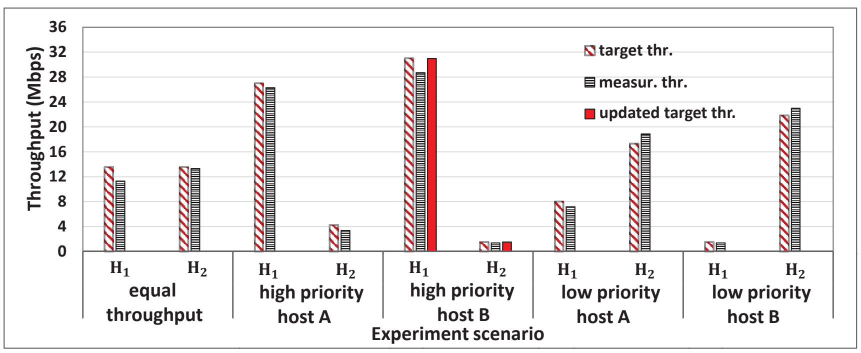


Figure 7.4: Results for two hosts case with proposal.

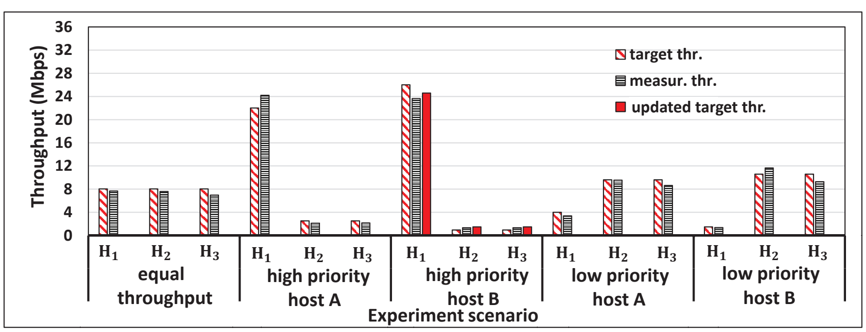


Figure 7.5: Results for three hosts case with proposal.

7.2.4 Discussions

From the measurement results, we observed the following results for these scenarios.

1) Equal Throughput: The measured throughput was similar among the hosts by assigning the equal target throughput by the proposal, although both the measured single and concurrent

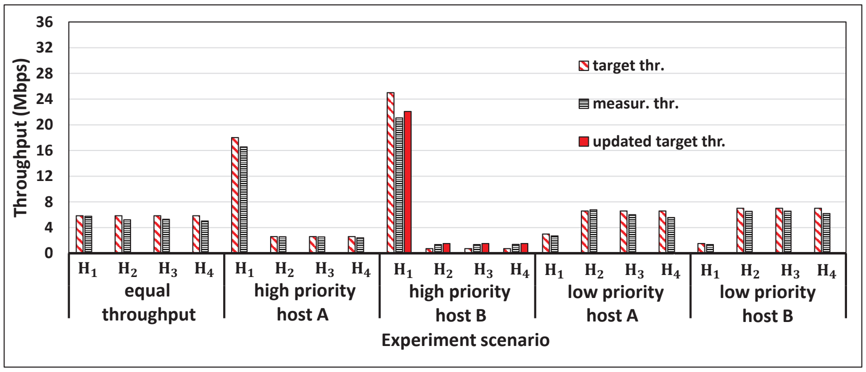


Figure 7.6: Results for four hosts case with proposal.

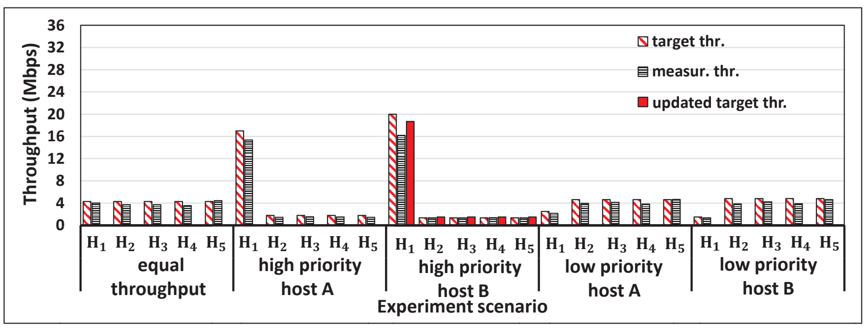


Figure 7.7: Results for five hosts case with proposal.

throughputs are different among them when the proposal was not applied. Thus, the throughput fairness request was achieved by the proposal.

2) High Priority Host A: The measured throughput of the *high priority host* H_1 was larger than its concurrent throughput and the throughputs of the other hosts that were similar to each other. The throughput of any host was larger than the *minimum target throughput*. Thus, both the high throughput provision request and the throughput fairness request were achieved.

3) High Priority Host B: As in 2, both the high throughput provision request and the throughput fairness request were achieved, while considering the *minimum target throughput*.

4) Low Priority Host A: The measured throughput of the *low priority host* H_1 was smaller than its concurrent throughput and the throughputs of the other hosts that were similar to each other. The throughput of any host was larger than the *minimum target throughput*. Thus, both the low throughput provision request and the throughput fairness request were achieved.

5) Low Priority Host B: As in 4, both the low throughput provision request and the throughput fairness request were achieved, while considering the *minimum target throughput*.

7.2.5 Proposal Accuracy

Table 7.3 shows the average of the absolute differences between the target and measured throughputs for each scenario across all host cases. The small value for any scenario ensures the effectiveness of the proposal.

Table 7.3: Average absolute differences between target and measured throughputs.

scenario	average absoluate difference (Mbps)
equal throughput	0.70
high priority host A	0.69
high priority host B	0.64
low priority host A	0.67
low priority host B	0.63

Table 7.4: Fairness index comparison using *iperf* traffic under *equal throughput scenario*.

case	fairness index[1]	
	without proposal	proposal
2 hosts	0.920	0.993
3 hosts	0.869	0.998
4 hosts	0.842	0.995
5 hosts	0.802	0.991

[1] fairness index: fairness among the values between 1 and 0, 1 indicates completely fair.

7.2.6 Fairness Index

To verify the throughput fairness for *equal throughput scenario*, Table 7.4 compares the *Jain's fairness index* [67] of the measured throughput among the hosts. It shows that the fairness index is improved singnificantly with the proposal which is very close to 1. On the other hand, without proposal the fairness index is much smaller than 1.

7.3 Evaluation with Web Traffics using Single AP

This section evaluates the proposal through testbed experiments using web application traffics as practical ones. To generate high-load traffics, the hosts are either downloading large files or accessing to video streaming from Web sites.

7.3.1 Experiment Setup

Figure 7.8 illustrates the network topology for the experiments using real web application traffics. As the web application servers in the Internet, *Ubuntu 20.04.3 OS* for file downloading [74] and *YouTube* for video streaming are adopted. In the experiments, the number of hosts is increased from two to four, where the same devices and locations in Tables 4.1 and 7.1 are used. Similarly, each experiment was conducted for *12 minutes*.

The following three scenarios of *equal throughput*, *priority host*, and *saturated host* are examined, where the measured throughput is compared with and without applying the proposal.

1) Equal Throughput Scenario: All the hosts are concurrently downloading the *Ubuntu 20.04.3 OS* files with 2.9*GB* using the web browser from the *web server*. The *equal target throughput* is

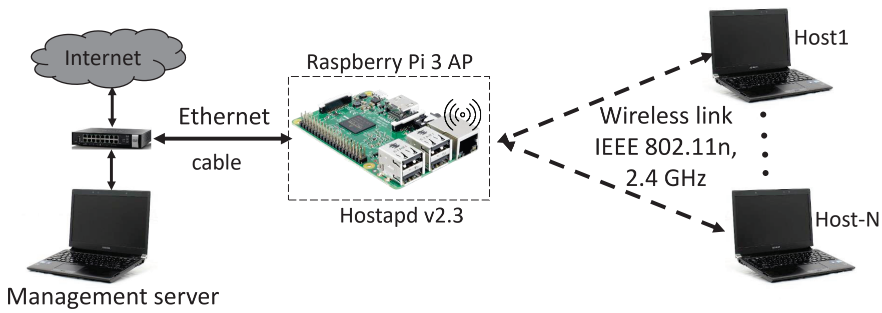


Figure 7.8: Testbed topology for web traffics.

assigned to these hosts.

2) Priority Host Scenario: All the hosts are concurrently downloading the *Ubuntu 20.04.3 OS* files. To investigate the effectiveness of the proposal, the slowest host H_2 is considered as the *priority host* and is assigned the far higher target throughput than the other hosts. This higher target throughput of the slowest host can be achieved by sacrificing the non priority hosts.

3) Saturated Host Scenario: One host H_3 is streaming video using the web browser, and the other hosts are concurrently downloading the *Ubuntu 20.04.3 OS* files. Then, H_3 is considered as the *saturated host* that cannot utilize all the available bandwidth since its application requires the much smaller one. Then, the remaining bandwidth should be allocated to the other hosts equally.

7.3.2 Results for Equal Throughput Scenario

Figure 7.9 shows the single throughput measurement results for the four hosts and the concurrent results for two, three, and four host cases with web traffics. Figure 7.10 shows the target throughput and the measured throughput for two, three, and four hosts cases. When the proposal was not applied, the throughput unfairness appeared, where the near host from the AP, H_1, achieved the higher throughput than the others. On the other hand, when the proposal was applied, the similar measured throughput was achieved to all the hosts regardless of their locations. Besides, the average absolute difference between the target and measured throughput is $0.79 Mbps$ the small value ensure the accuracy of the proposal.

Table 7.5 compares the fairness index of the measured throughputs among the hosts with and without the proposal. The proposal increases the fairness index to be close to 1. Thus, the effectiveness of the proposal in solving the throughput unfairness problem is confirmed.

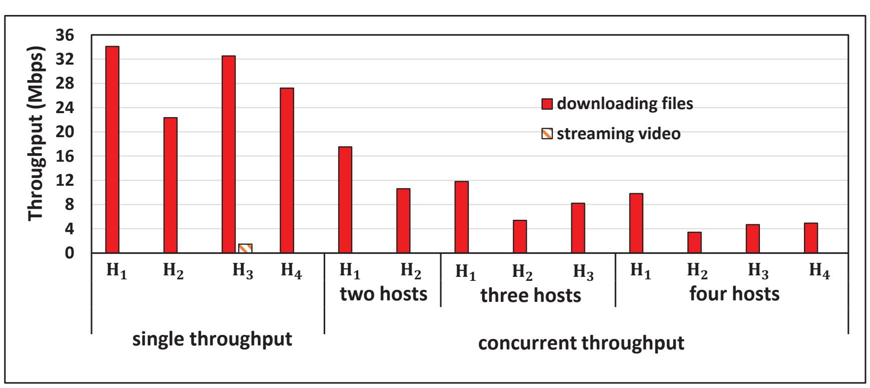


Figure 7.9: Measurement results of single and concurrent throughputs with *web* traffic.

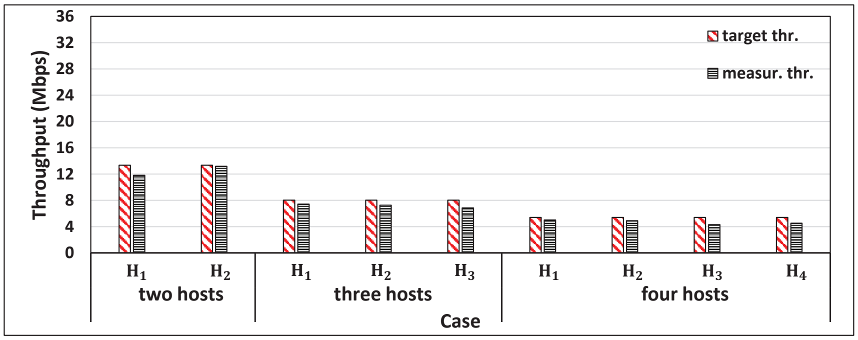


Figure 7.10: Results for *equal throughput scenario* with proposal.

Table 7.5: Fairness index comparison using *web* traffic under *equal throughput scenario*.

case	fairness index	
	without proposal	proposal
2 hosts	0.942	0.996
3 hosts	0.912	0.998
4 hosts	0.847	0.996

7.3.3 Results for Priority Host Scenario

Figure 7.11 shows the results for *priority host scenario*. Here, H_2 was selected as the priority host, because it was most distant from the AP. In three and four hosts cases, the target throughput was updated, because the original target throughput for H_2 cannot ensure the *minimum target throughput (1.5Mbps)* of the others. Then, the proposal achieved the target throughput for any host. The average absolute difference between the target and measured throughput is $0.40 Mbps$, the lower value confirm the effectiveness of the proposal.

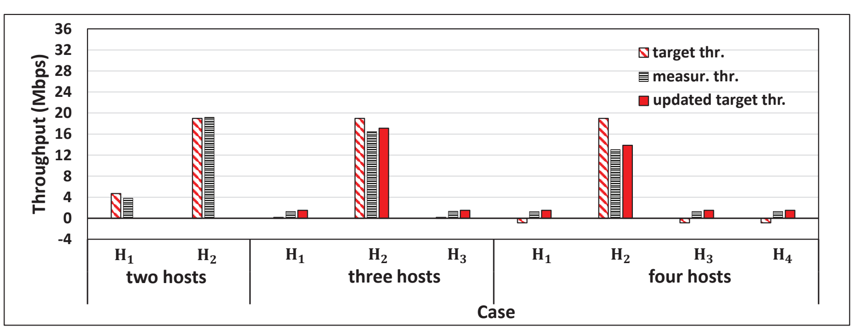


Figure 7.11: Results for *priority host scenario* with proposal.

7.3.4 Results for Saturated Host Scenario

Figure 7.12 shows the concurrent throughput measurement results for three and four host cases with the *saturated host* H_3. H_3 received the video streaming service, and was located in the same room as the AP. Figure 7.13 shows the results for *saturated host scenario*. Two hosts case was not examined because only one host remained other than the saturated host. The measured single throughput for H_3, $S_3 = 1.47Mbps$, is smaller than the obtained equal target throughput, $2.43Mbps$ for three hosts case and $2.27Mbps$ for four hosts case. Thus, S_3 was used for the target throughput of H_3, and the target throughput for the other hosts were updated. Then, the proposal achieved the target throughput for any host. The average absolute difference between the target and measured throughput is $0.48Mbps$ and $0.35Mbps$ for equal and required throughput allocation, the small values confirm the accuracy of the proposal.

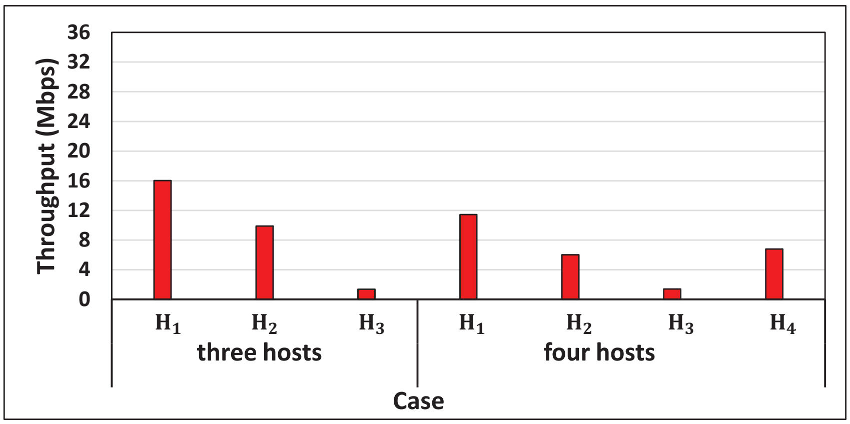


Figure 7.12: Measurement results of concurrent throughputs with *saturated host*.

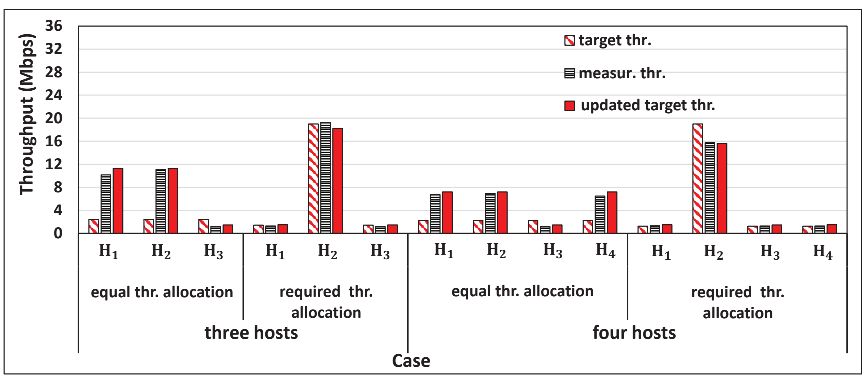


Figure 7.13: Results for *saturated host scenario* with proposal.

7.4 Throughput Comparison between the Proposal and without Proposal

Figures 7.14 and 7.15 compare the total throughput between the cases with the proposal and without the proposal. With the proposal, the total throughput is reduced by 14.36% and 14.77% on average for *iperf* and *web* traffics, which is tolerable. The packet transmissions with high bit rates to near hosts become reduced. The total throughput reduction cannot be avoided in achieving the throughput fairness by giving more packet transmissions with low bit rates to distant hosts.

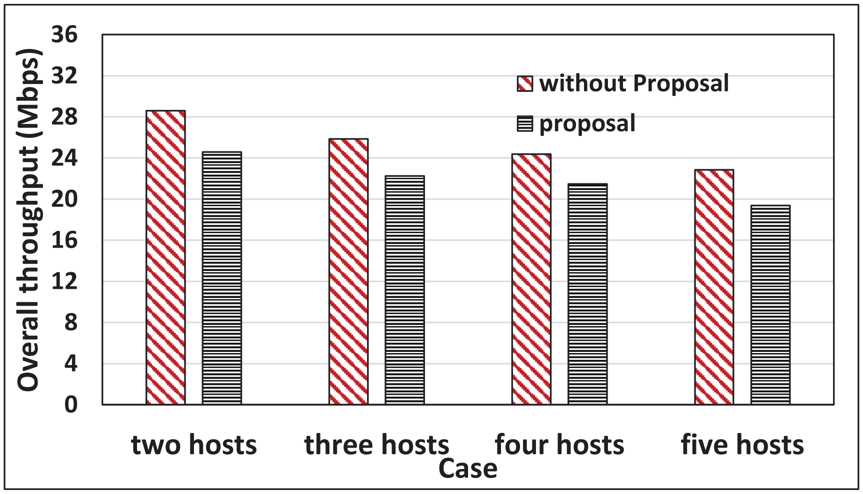


Figure 7.14: Total throughput comparisons for with and without proposal for *iperf* traffic.

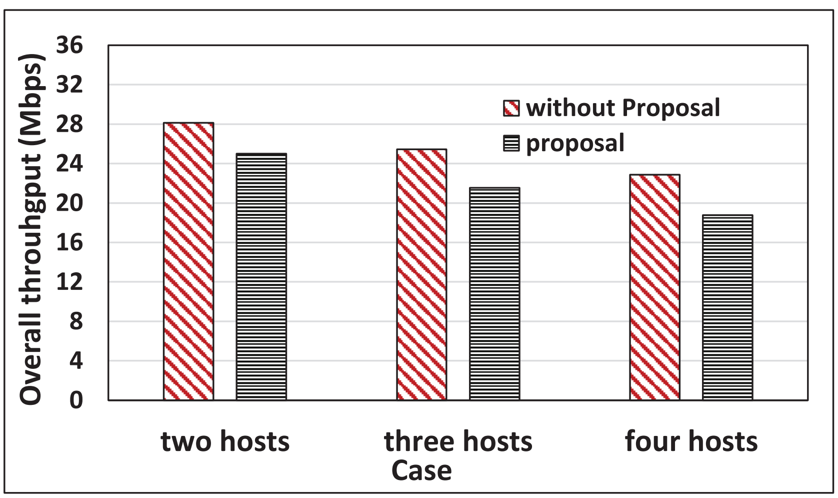


Figure 7.15: Total throughput comparisons for with and without proposal for *web* traffic.

7.5 Evaluation with Multiple APs

This section evaluates the proposal through extensive experiments using the testbed system with up to four APs and four hosts. Each AP is connected with one host.

7.5.1 Experiment Setup

Figure 7.16 shows the device configuration of the testbed system. Table 4.1 shows the hardware and software specifications. *Raspberry Pi 3* with *TP-Link TL-WN722N wireless NIC* [68] adapter was adopted as the *software AP*. *Linux* laptop PCs were used for both the management server and hosts. The experiments for evaluations were conducted in the same field in Figure 4.1(a). Table 7.6 shows the locations of the hosts and the APs and the channel assignments in the experiments. Considering fluctuations of measured throughputs, the measurements were conducted 12*min* in each of six topologies, and their average results were used in evaluations.

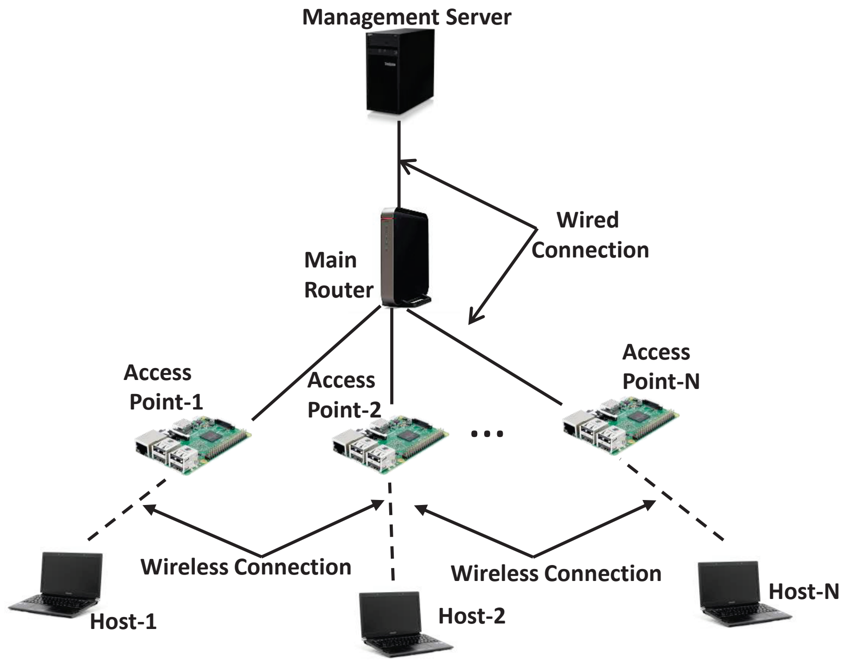


Figure 7.16: Device configuration of testbed system.

Table 7.6: Host and AP locations with channel assignments.

topology	number of APs	channel assignment				host and AP location							
		AP_1	AP_2	AP_3	AP_4	AP_1	H_1	AP_2	H_2	AP_3	H_3	AP_4	H_4
1	two	1+5	9+13	-	-	D307	in front of D307	D307	D307	-	-	-	-
2	two	1+5	9+13	-	-	in front of D307	D307	D307	D307	-	-	-	-
3	three	1+5	1+5	9+13	-	D307	D307	D307	D307	D307	D307	-	-
4	three	1+5	9+13	5+9	-	in front of D308	refresh corner	D307	D307	in front of D305	in front of D305	-	-
5	four	1+5	9+13	4+8	7+11	in front of D308	in front of D308	D307	refresh corner	D306	D306	in front of D301	in front of D301
6	four	1+5	9+13	4+8	7+11	in front of D308	in front of D308	D307	refresh corner	in front of D305	D306	refresh corner	refresh corner

7.5.2 Throughput Results

Figure 7.17 to Figure 7.22 show individual throughput results for the six topologies, respectively. Two APs, three APs, or four APs are concurrently communicating with hosts. In each figure, *single thr.* represents the measured single throughput, *concurrent thr.* does the concurrent throughput when all the hosts are communicating, *target thr.* refers to the *target throughput* that is given by the proposal, and *measur. thr.* does the measured throughput. From these graphs, we can observe the following results:

7.5.2.1 Two APs

First, we discuss the experiment results in topologies 1 and 2 where two APs are concurrently communicating with two hosts. Figure 7.17 and 7.18 show the throughput results. With the proposal, the throughput fairness was successfully achieved among the hosts. In contrast, without the proposal, the equal throughput was not achieved. However, the average measured throughput was 94.03% of the target throughput and average absolute difference between the target and measured throughput was $1.24 Mbps$. This reduction will come from the overhead of applying *traffic shaping* at the APs.

7.5.2.2 Three APs

Next, we discuss the experiment results in topologies 3 and 4 where three APs are concurrently communicating with three hosts. Figure 7.19 and 7.20 show the throughput results. The measured throughput after the proposal was similar among the hosts. The average measured throughput was 88.95% of the target throughput and average absolute difference was $1.50 Mbps$, where the overhead of applying *traffic shaping* at more APs became larger.

7.5.2.3 Four APs

Finally, we discuss the experiment results in topologies 5 and 6 where four APs are concurrently communicating with four hosts. Figure 7.21 and 7.22 show the throughput results. Again, the measured throughput after the proposal was similar among the hosts. The average measured throughput was further reduced to be 76.42% of the target throughput and also increase the average absolute difference with $2.24 Mbps$. Thus, the throughput enhancement at the increasing number of APs will be in future works.

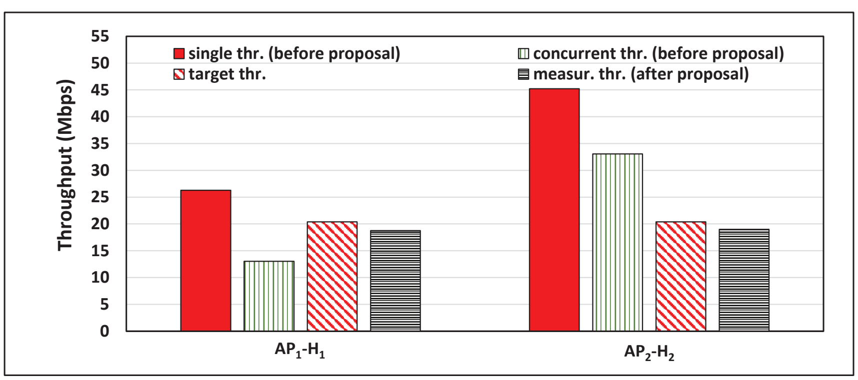


Figure 7.17: Throughput results for topology 1.

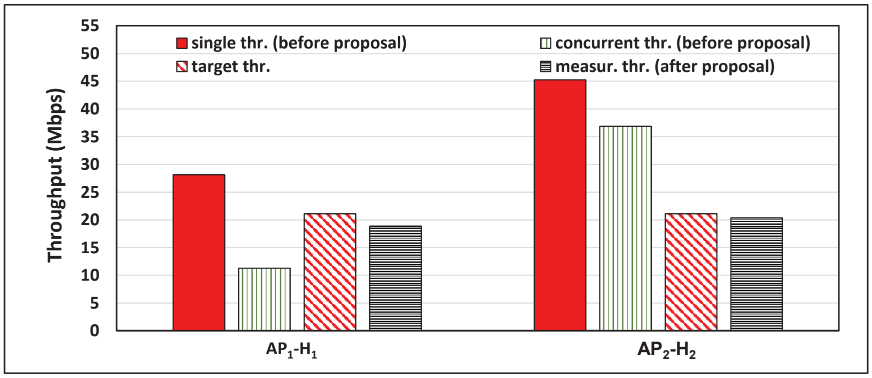


Figure 7.18: Throughput results for topology 2.

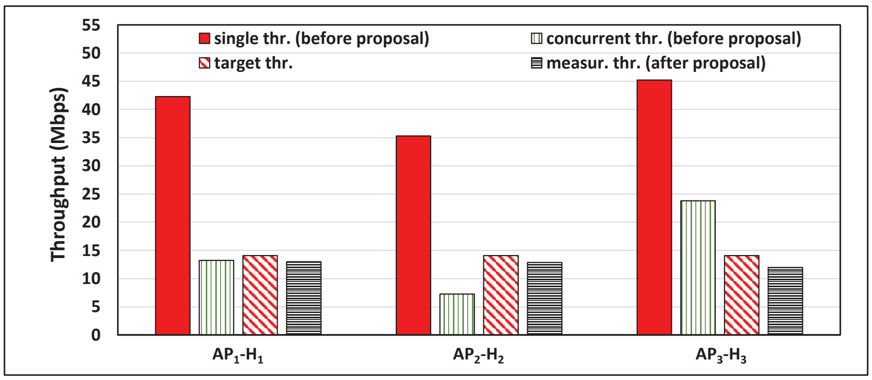


Figure 7.19: Throughput results for topology 3.

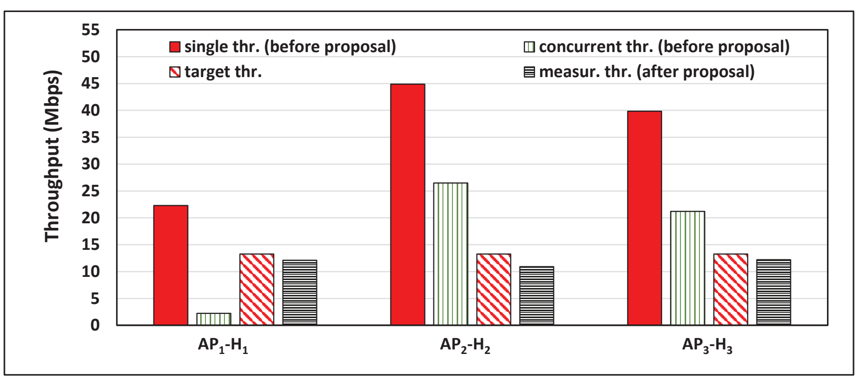


Figure 7.20: Throughput results for topology 4.

7.5.3 Fairness Index and Total Throughput Comparison

Table 7.7 compares the *Jain's fairness index* of the measured throughputs among the hosts and their totals. In any topology, this index becomes very close to 1 with the proposal, whereas it is much smaller than 1 without the proposal. However, the total throughput with the proposal is smaller than that without the proposal, because the proposal gives higher chances to slower hosts to transmit packets than to faster hosts. The solution of this tradeoff will be in future studies.

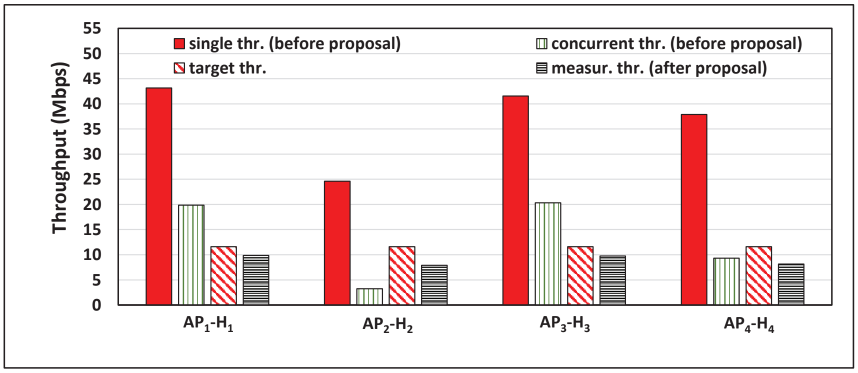


Figure 7.21: Throughput results for topology 5.

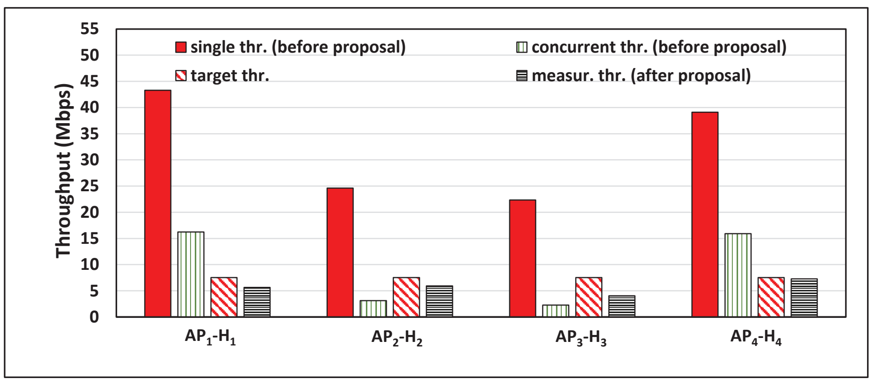


Figure 7.22: Throughput results for topology 6.

Table 7.7: Fairness index and total throughput comparisons.

topology	fairness index		total throughput	
	without proposal	proposed approach	without proposal	proposed approach
1	0.841	0.999	46.10	38.77
2	0.779	0.999	48.18	39.23
3	0.823	0.998	44.28	37.84
4	0.718	0.999	49.92	35.22
5	0.768	0.990	52.78	34.65
6	0.662	0.970	37.50	22.89

7.6 Summary

In this chapter, we presented the implementation and evaluation of *throughput control method* in WLAN. First, we introduced the implementation procedure of the proposed method in the WLAN testbed. Then, we evaluated the proposal through several testbed experiments using single or multiple APs. In the next chapter, we will present the related works of this study.

Chapter 8

Related Works in Literature

This chapter briefly discusses related works in literature. A number of research works have addressed the throughput unfairness problem in WLAN.

In [75], Hwang et al. studied the fair throughput allocation issue in a multi-rate WLAN. Hence, they proposed a network-wide association scheme with traffic control that defines the possible data traffic from an access point to clients based on throughput allocation algorithm. The proposal is verified by simulation. In contrast, our approach is implemented at the AP by using conventional Linux command.

In [76], Abuteir et al. proposed a *software defined networking (SDN)* based *wireless network assisted video streaming (WNAVS)* method to ensure the fairness among the stations. The proposal uses traffic shaping to control packets based on bandwidth allocation for users and network traffic statistics. However, their method is limited to video applications and cannot allocate the equal throughput. In contrast, our approach is generic for various applications and ensure the fair throughput among the users.

In [77], Lei et al. presented a *fair bandwidth allocation* approach, which allocates the bandwidth based on the user needs and priority. Then, they adopted the association algorithm based on client demands, which selects the optimal associating AP according to transmission time demanded by all the associating client. This method does not provide the fair throughput when multiple APs are concurrently communicating with the hosts.

In [78][79], Fang et al. and Kongsili et al. considered the air-time assignment policy for proportionally allocating the fair throughput to the hosts in WLAN while increasing the overall network throughput. On the other hand, our approach allocates the equal throughput to the concurrently communicating hosts in WLAN with single or multiple APs.

In [80], Mansy et al. presented a *quality of experience (QoE)* metric for adaptive video streams to ensure fairness at the network layer. They designed a max-min fairness problem based on this QoE metric to enforce throughput allocations in the home network. The traffic shaping is adopted to control the data traffics.

In [81], Høiland-Jørgensen et al. presented a network layer queuing management scheme to ensure the proportional fairness among the competing hosts in WLAN while increasing the overall throughput. However, their approach cannot ensure the equal throughput performance among the hosts. It was implemented at the AP with no modification at the MAC layer protocol.

In [82], Blough et al. dealt with the interference-aware proportional fairness in dense WLANs by considering the *signal to noise ratio (SNR)* level at receiving stations. In their approach, the SNR is used to estimate the optimum data transmission rate based on the channel condition in order to allocate the fair throughput among the competing hosts. While this method can increase the network throughput, there is still a problem on the throughput allocation among the hosts when considering the equal throughput performance.

In [83], Yan et al. investigated the performance anomaly problem in multi-rate WLAN. Thus, they proposed a MAC optimization technique for maintaining the proportional throughput fairness by altering the contention window based on the data rate and packet size. It was implemented at the MAC layer and was verified by simulations. In contrast, our approach is implemented in the real testbed system and ensures the fair throughput among the hosts when they are concurrently communicating with single or multiple APs.

In 0[84], Wu et al. investigated the TCP unfairness problems caused by interactions between TCP and Medium Access Control (MAC) protocol. They introduced the *Selective Packet Marking with ACK Filtering (SPM-AF)* and *Least Attained Service (LAS)* queue management scheduling techniques to mitigate the unfairness problem, and verified them through simulations. In contrast, our proposal ensures the fair (equal) or required target throughput to the hosts by controlling the packet transmission, and the effectiveness is verified by real testbed experiment.

In [85], Banchs et al. introduced an algorithm to ensure the throughput fairness in virtual WLANs by adopting the control theory. This proposal applied the proportional integrator (PI) controller, to adjust the contention window size of each virtual WLAN to achieve the optimal performance. The effectiveness of this method is verified by simulations. However, in reality, changing the contention window size is difficult because hardware modifications are required. In contrast, our proposal ensures the fair throughput or provide the necessary throughput to the hosts in WLAN. The effectiveness is verified by the real testbed where hardware modifications are not necessary.

In [86], Abeysekera et al. proposed a method to reduce the unfairness between downlink and uplink flows by amending the random backoff mechanism of the IEEE 802.11 MAC protocol at the AP. They verified it through simulations using TCP and UDP flows. On the other side, we implement a testbed for the throughput fairness in WLAN as well.

In [87], Park et al. observed the unfairness problem among the wireless stations in TCP which can be caused by interactions of the asymmetric property of the TCP congestion and the MAC contention control. Hence, they proposed a cross-layer feedback approach at the transport layer and the link layer for ensuring the per-station fairness. However, the implementation can be difficult because it works on the MAC layer. On the contrary, in our approach the throughput fairness can be achieved by assigning the equal target throughput to every host by maintain the proper communication time.

In [88], Kim et al. examined the throughput unfairness problem in WLAN that is caused by *unequal frame error rates (FERs)* among the hosts and the absence of loss differentiations in the *automatic repeat request (ARQ)* protocol, which can lead to the imbalance of the outage probability and the access probability among hosts. The authors proposed the *enhanced distributed coordination function (DCF)* by adopting the *hybrid automatic repeat request (HARQ) with Chase combining (HARQ-CC)* to solve both imbalance problems. The performance of the method was demonstrated both mathematically and through MATLAB simulations. However, for the practical implementation, the Media Access Control (MAC) layer protocol needs to be modified. On the other hand, our proposal can be implemented by calling the Linux commands from the application program. It does not need modifying the MAC protocol implementation.

In [89], Akimoto et al. observed that the locations of *mobile terminals (MTs)* in the network result in different coverages, where some terminals may cause the hidden terminal problem. This problem degrades the throughputs of the affected terminals while others have the high throughputs. To address this issue, the authors proposed the mobile terminal allocation scheme using virtual sector (VS) where terminals are classified into groups by their coverages. Terminals in one group can sense each other during data transmissions to avoid the hidden terminal problem and solve the throughput unfairness. However, our experiment results show that the throughput unfairness is observed even if stations do not suffer from the hidden terminal problem, when they communicate from different relative distances from the AP.

In [90], Priya and Murugan studied the unfairness problem for simultaneous uplink and downlink TCP flows by considering the optimum queue selection. They designed a two queue approach where the primary queue holds the TCP data packets while the secondary queue holds acknowledgement (ACK) packets. In this method, the optimal queue size is identified by the probability or priority scheduling approach. In the priority scheduling, the ACK packets are given the higher priority and are transmitted before the data packets. In the probability scheduling, the AP selects the queue based on the optimal probability p to ensure the fairness, where p is calculated considering the number of the uplink and downlink flows.

In [91], Kim et al. investigated the asymmetric behaviors between the uplink and downlink TCP flows. They designed an adaptive backoff algorithm by estimating the backlog size (number of nodes that have packets) for the uplink/downlink in order to achieve the fairness and optimize the throughput. The ideal uplink and downlink transmission probabilities are derived based on the backlog estimation as a function of the backlog size.

In [92], Lei et al. studied the airtime fairness in WLAN. They presented the *improved active queue management (IAQM)* algorithm for solving the unfairness problem of WLAN by setting the different queue length based on their data rates so that each host gets the fair channel usage time. In contrast, our proposal achieves the throughput fairness using the traffic control command for traffic shaping.

In [93], Le et al. proposed a method to solve the unfairness problem by allowing each station to choose an appropriate contention window size based on the cost function. They implemented it into the MAC layer and verified it through simulations.

In [94], Garroppo et al. observed that the performance of the 802.11 standard is severely degraded when a single station experiences the poor channel condition against the AP. This performance anomaly occurs due to the simple FIFO scheduling manner employed in the AP and the max-min fairness of the CSMA/CA protocol. In order to overcome this problem, they proposed the *Deficit Transmission Time (DTT)* scheduler to ensure the fair air-time usage to all the associated stations. The *Wireless Channel Monitor (WChMon)* tool is used to estimate the maximum attainable throughput towards the specific station. However, the major drawback of this tool is the dependence to the specific network card and driver.

Most of the works in literature focus on the throughput fairness among hosts when they communicate with one AP. They do not consider multiple APs. On the other hand, our proposal considers both single and multiple APs in WLAN, and also provides the demanded throughput to one host when multiple hosts communicate with an AP. The proposal adopts simple Linux commands for easy implementations on real devices, and is evaluated through experiments using real devices.

Chapter 9

Conclusion

This thesis presented the *throughput control method* for concurrently communicating multiple-hosts in *wireless local-area networks (WLANs)*.

Firstly, I surveyed the *IEEE 802.11* wireless network technologies related to this thesis, including the overview of IEEE 802.11 WLAN, channel access modes in IEEE 802.11 MAC, IEEE 802.11n protocols, Linux tools for wireless networking, traffic shaping, and Jain's fairness index.

Secondly, I reviewed our previous studies related to this thesis, including the *elastic WLAN system*, the testbed implementation using *Raspberry Pi* APs, and the *TCP fairness control method*.

Thirdly, I presented the measurement results for throughput unfairness observations when multiple-hosts concurrently communicate with the single/multiple APs.

Fourthly, I proposed the *fair throughput control method* to solve the throughput unfairness problem among the concurrently communicating hosts in WLAN. Also, I proposed the *demanding throughput control method* to meet the demand throughput request of the host. To meet the fair or demanded throughput request, these methods measure the single and concurrent throughput for each host, calculates the channel occupying time, derives the target throughput to satisfy the fair or demanded throughput request, and controls the traffics to achieve the target throughput of every host by applying *traffic shaping* at the AP.

Finally, I implemented the proposal in the *elastic WLAN system testbed* that uses *Raspberry Pi* for the APs. At first, I evaluated the proposal when multiple hosts communicate with the single AP at the same time by taking into account various throughput request scenarios. After that, I evaluated the proposal by extensive experiments when multiple hosts communicate with multiple APs at the same time in an equal throughput scenario. The experiment results confirmed the effectiveness of the proposal.

In future studies, I will generalize the proposal to consider multiple hosts for each AP under multiple APs and the host mobility in the network. I will also study the application of the proposal in allocating the demanded throughput to the priority host when multiple hosts communicate with multiple APs. Then, we will evaluate our proposals in various network scenarios.

Bibliography

[1] M. Graham and W. H. Dutton, Eds., "Society and the Internet: How networks of information and communication are changing our lives," 2nd ed. Oxford: Oxford Univ. Press, Jul. 2019.

[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, Sep. 1997.

[3] M. Elkhodr, S. Shahrestani, and H. Cheung, "Emerging wireless technologies in the internet of things: a comparative study," Int. J. Wireless Mob. Netw., vol. 8, no. 5, pp. 67–82, Oct. 2016.

[4] K. Pahlavan and P. Krishnamurthy, "Evolution and impact of wi-fi technology and applications: a historical perspective," Int. J. Wireless Info. Netw., vol. 28, no. 1, pp. 3-19, Mar. 2021.

[5] J. Deng, B. Liang and P. K. Varshney, "Tuning the carrier sensing range of IEEE 802.11 MAC," IEEE Glob. Telecomm. Conf. vol.5, pp. 2987-2991, 2004.

[6] S. K. Dash, S. Mishra, and J. Mishra, "Mobile data offloading using opportunistic communication and AP deployment: a case study," Int. J. Mob. Comput. Mult. Commun., vol. 8, no. 4, pp. 66-84, Oct. 2017.

[7] H. S. Oh, D. G. Jeong, and W. S. Jeon, "Joint radio resource management of channel-assignment and user-association for load balancing in dense WLAN environment," IEEE Access, vol. 8, pp. 69615-69628, Apr. 2020.

[8] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalt, "Online primary channel selection for dynamic channel bonding in high-density WLANs," IEEE Wireless Commun. Lett., vol. 9, no. 2, pp. 258-262, Feb. 2020.

[9] LAN MAN standards committee, "Part11: Wireless LAN medium access control (MAC) and physical Llayer (PHY) specifications," The Institute of Electrical and Electronics Engineering, Jun. 1997.

[10] R. B. M. A. Abdelrahman, A .B. A. Mustafa, and A. A. Osman, "A Comparison between IEEE 802.11a, b, g, n and ac Standards," IOSR J. of Comp. Eng., vol.17, issue 5, ver. III, pp. 26-29, Sep. 2015.

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

[12] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," Wireless Networks, vol. 11, no. 1-2, pp. 115-133, Jan. 2005.

[13] 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. Network. Comput., vol. 6, no. 2, pp. 395-419, Jul. 2016.

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

[15] M. M. Islam, N. Funabiki, M. Kuribayashi, M. Saha, K. I. Munene, R. W. Sudibyo, and W.-C. Kao, "A proposal of transmission power minimization extension in active access-point configuration algorithm for elastic wireless local-area network system," Int. J. Comput. Soft. Eng., vol. 4, Feb. 2019.

[16] Y. Chen, X. Wang, and L. Cai, "On achieving fair and throughput-optimal scheduling for TCP flows in wireless networks," IEEE Trans. on Wireless Commun., Vol. 15, no. 12, pp. 7996-8008, Sep. 2016.

[17] I. Moiseenko, and D. Oran, "TCP/ICN: carrying TCP over content centric and named data networks," in Proc. 3rd ACM Conf. on Information-Centric Netw., pp. 112-121, 2016.

[18] "Zoom rooms system requirements," Internet: `https://support.zoom.us/hc/en-us/articles/204003179-System-Requirements-for-Zoom-Rooms#hb48c2bfd-7da0-4290-aae8-784270d3ab3`.Accessed 20 May, 2022.

[19] R. W. Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, H. Briantoro, M. M. Islam, and W.-C. Kao, "A generalized TCP fairness control method for multiple-host concurrent communications in elastic WLAN system using Raspberry Pi access point," Int. J. Mob. Comp. and Multim. Comm. vol. 11, no. 2, pp. 18-40, Apr. 2020.

[20] R. W. Sudibyo, "A study of TCP throughput fairness control method for multiple-host concurrent communications in Wireless Local-Area Network," Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Mar. 2020.

[21] M. M. Rahman, N. Funabiki, K.I. Munene, S. C. Roy, M. Kuribayashi, and W. -C. Kao, "A throughput fairness control method for concurrent communications in wireless local-area network with multiple access-points," J. Commun., vol. 17, no. 8, pp. 592-599, Aug. 2022.

[22] M. M. Rahman, N. Funabiki, K. I. Munene, H. Briantoro, F. Akhter, M. Kuribayashi, and M. M. Islam, "A proposal of throughput allocation method for concurrently communicating two or three hosts in wireless local-area network," Int. Conf. on Emerging Tech. for Commun. (ICETC), C1-4, Dec. 2020. (Online, Japan)

[23] "Manipulate traffic control settings," Internet: `https://linux.die.net/man/8/tc`. Accessed 20 May, 2022.

[24] M. S. Gast, "802.11 wireless networks - the definitive guide," 2nd ed., O'Reilly, Sebastopol, Apr. 2005.

[25] S. Banerji, and R. S. Chowdhury, "On IEEE 802.11: wireless LAN technology," Int. J. Mobi. Net. Commun. Telematics, vol. 3, no. 4, Aug. 2013.

[26] S. Banerji, "Upcoming standards in wireless local area networks," Wirel. Mob. Technol., vol. 1, no. 1, Sep. 2013.

[27] I. Poole, "IEEE 802.11 Wi-Fi standards," Internet: https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/what-is-wifi.php. Accessed 20 May, 2022.

[28] V. Beal, "What is 802.11 wireless LAN standards?," Internet: https://www.webopedia.com/TERM/8/802_11.html. Accessed 20 May, 2022.

[29] C. C. Choon, "A study of active access-point selection algorithm for wireless mesh network under practical conditions," Ph. D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Sep. 2015.

[30] "IEEE 802.11 - Wikipedia," Internet: http://en.wikipedia.org/wiki/IEEE_802.11. Accessed 20 May, 2022.

[31] M. E. Islam, "A study of access-point aggregation algorithm for elastic wireless local-area network system and its implementation," Ph.D. thesis, Grad. School of Natural Science and Technology, Okayama Univ., Japan, Mar. 2016.

[32] S. Sendra, M. Garcia, C. Turro, and J. Loret, "WLAN IEEE 802.11 a/b/g/n indoor coverage and interference performance study," Int. J. Adv. Netw. Service., vol. 4, no. 1, pp. 209-222, 2011.

[33] M. M Islam "A study of dynamic access-point configuration and power minimization in elastic wireless local-area network system," Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Sep. 2019.

[34] K.I. Munene "A study of throughput drop estimation models for concurrently communicating links in wireless local-area networks," Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Apr. 2022.

[35] P. Likhar, R. S. Yadav and K. M. Rao, "Securing IEEE 802.11g WLAN using open VPN and its impact analysis," Int. J. of Netw. Security and Its Applications (IJNSA), Vol. 3, no. 6, pp. 97-113, Nov. 2011.

[36] A. S. Tanenbaum, and D. J. Wetherall, "Computer networks," 5th ed., Pearson Prentice Hall, 2011.

[37] J. H. Yeh, J. C. Chen, and C. C. Lee, "WLAN standards," IEEE Potentials, vol. 22, no. 4, 2003.

[38] IEEE, "IEEE Std 802.11-2012, IEEE standard for information technology," Internet: https://standards.ieee.org/findstds/standard/802.11-2012.html. Accessed 20 May, 2022.

[39] O. Bejarano, E. W. Knightly, and M. Park, "IEEE 802.11ac: from channelization to multi-user MIMO," IEEE Comm. Mag., vol. 51, no. 10, pp. 84-90, Oct. 2013.

[40] L. Verma, M. Fakharzadeh, and S. Choi, "Wifi on steroids: 802.11ac and 802.11ad," IEEE Trans. Wirel. Comm., vol. 20, no. 6, pp. 30-35, Dec. 2013.

[41] M. S. Gast, “ 802.11ac: a survival guide,” O’Reilly Media, Inc., 2013.

[42] B. Bellalta, “IEEE 802.11 ax: High-efficiency WLANs,” IEEE Wirel. Commu. vol. 23, no. 1, pp. 38-46, 2016.

[43] E. Khorov, A. Kiryanov, A. Lyakhov and G. Bianchi, “A Tutorial on IEEE 802.11ax High Efficiency WLANs,” IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 197-216, 2019.

[44] IEEE, “IEEE SA standards board approvals,” Internet: `https://standards.ieee.org/about/sasb/sba/feb2021.html.` Accessed 20 May, 2022.

[45] IEEE, “IEEE 802.11ax-2021 - IEEE standard for information technology,” Internet: `https://standards.ieee.org/standard/802_11ax-2021.html.` Accessed 20 May, 2022.

[46] H. Briantoro, “A study of transmission power optimization with channel assignment consideration for access-points in wireless local-area network,” Ph.D. thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Sep. 2021.

[47] “WiMAX page, IEEE802.11ac works,” Internet: `http://123-info.net/wimax-page/archives/918`. Accessed 20 May, 2022.

[48] L. Deek, E. Garcia-Villegas, E. Belding, S. J. Lee, and K. Almeroth, “The impact of channel bonding on 802.11n network management,” in Proc. Conf. Emer. Netw. Exp. Tech., Dec. 2011.

[49] K. I. Munene, “A study of throughput estimation model and channel assignment under partially overlapping channels in IEEE 802.11n wireless local-area network,” Master thesis, Grad. Sch. Natural Sci. Tech., Okayama Univ., Japan, Mar. 2019.

[50] A. Saif, M. Othman, “A reliable A-MSDU frame aggregation scheme in 802.11n wireless networks,” in Proc. Comp. Scie., vol. 21, 2013.

[51] “IEEE 802.11n-2009 - Wikipedia,” Internet: `http://en.wikipedia.org/wiki/IEEE_802.11n-2009`. Accessed 5 May, 2022.

[52] M. Vipin, and S. Srikanth, “Analysis of open source drivers for IEEE 802.11 WLANs,” in Proc. Int. Conf. Wireless Comm. Sensor Comput., pp. 1-5, Jan. 2010.

[53] “arp-scan user guide,” Internet: `https://linux.die.net/man/1/arp-scan`. Accessed 20 May, 2022.

[54] “Arp Scan,” Internet: `https://launchpad.net/ubuntu/trusty/+package/arp-scan`. Accessed 20 May, 2022.

[55] J. Malinen, “hostapd: IEEE 802.11 AP, IEEE 802.1 X. WPA/WPA2/EAP/RADIUS authenticator,” Internet: `https://w1.fi/hostapd/`. Accessed 20 May, 2022.

[56] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, “SSH, the secure shell: the definitive guide,” O’Reilly Media Inc., 2005.

[57] T. Ylonen and C. Lonvick, “The secure shell (SSH) protocol architecture,” 2006.

[58] "OpenSSH," Internet: http://www.openssh.com/. Accessed 20 May, 2022.

[59] O. Terpollari, "How to install and configure OpenSSH server in Linux," Internet: http://www.tecmint.com/install-openssh-server-in-linux/. Accessed 1 Sep., 2019.

[60] "Run / execute command using SSH," Internet: https://www.cyberciti.biz/faq/unix-linux-execute-command-using-ssh/. Accessed 20 May, 2022.

[61] P. Michel, "about iw," Internet: https://wireless.wiki.kernel.org/en/users/documentation/iw, Accessed 20 May, 2022.

[62] "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool," Internet: https://iperf.fr/, Accessed 20 May, 2022.

[63] "iftop: display bandwidth usage on an interface," Internet: http://www.ex-parrot.com/pdw/iftop/, Accessed 20 May, 2022.

[64] M. M. Rahman, N. Funabiki, R. W. Sudibyo, K. I. Munene, H. Briantoro, and M. M. Islam, "An improvement of TCP throughput fairness control method for concurrent communications in elastic WLAN system," IEICE Tech. Report, NS2020-25, pp. 19-24, Jun. 2020. (Online, Japan,).

[65] "TC(8)," Internet: http://man7.org/linux/man-pages/man8/tc.8.html, Accessed 20 May, 2022.

[66] MA Brown, "Traffic control how to," Internet: http://tldp.org/HOWTO/Traffic-Control-HOWTO/. Accessed 20 May, 2022.

[67] R. Jain, D. Chiu and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," East. Res. Lab., vol. 38, 1984.

[68] "TP-Link TL-WN722N," Internet: https://www.tp-link.com/us/home-networking/usb-adapter/tl-wn722n/. Accessed 20 May, 2022.

[69] "Hostapd: the linux way to create virtual Wifi access point," Internet: nims11.wordpress.com/2012/04/27/hostapd-the-linux-way-to-create-virtual-wifi-access-point/, Accessed 1 Sep., 2019.

[70] "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, Accessed 1 Sep., 2019.

[71] K. Astrom and T. Hagglund, "PID controller: theory, design, and tuning, 2nd Ed.," Inst. Socie. America, 1995.

[72] 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.

[73] H. Briantoro, N. Funabiki, M. M. Rahman, K. I. Munene, and M. Kuribayashi, “Joint optimization method of channel assignment and transmission power for concurrently communicating multiple access-points in wireless local-area network,” Int. J. Netw. Comp., vol. 11, no. 2, pp. 251-266, Jul. 2021.

[74] “Ubuntu 20.04.3 LTS Released.,” Internet: `https://www.omgubuntu.co.uk/2021/08/ubuntu-20-04-3-point-release`. Accessed 20 May, 2022.

[75] J. Hwang, H. Lim, S. Oh, and B. T. Lee, “Association scheme with traffic control for IEEE 802.11 wireless LANs,” IEICE Trans. on Commun., vol. 98, no. 8, pp. 1680-1689, Aug. 2015.

[76] R. M. Abuteir, A. Fladenmuller, and O. Fourmaux, “An SDN approach to adaptive video streaming in wireless home networks,” in Proc. IEEE Int. Wireless Commun. Mob. Comput. Conf., pp. 321–326, 2016.

[77] J. Lei, J. Tao, and S. Yang, “Joint AP association and bandwidth allocation optimization algorithm in high-dense WLANs,” Future Internet, vol. 10, no. 8, p. 73, Aug. 2018.

[78] Y. Fang, B. Doray, and O. Issa, “A practical air time control strategy for Wi-Fi in diverse environment,” in Proc. IEEE Wireless Commun. Netw. Conf. Works., pp. 1-6, 2017.

[79] L. Kongsili, A. Fujimoto, and F. Uchio, “Packet scheduling and access priority control for QoS and fairness in wireless LAN,” Procedia Comput. Sci., vol. 60, pp. 1788-1797, Jan. 2015.

[80] A. Mansy, M. Fayed, and M. Ammar, “Network-layer fairness for adaptive video streams,” in Proc. IFIP Netw., pp. 42–48, 2015.

[81] T. Høiland-Jørgensen, M. Kazior, D. Täht, P. Hurtig, and A. Brunstrom, “Ending the anomaly: achieving low latency and airtime fairness in wifi,” in Proc. USENIX Annual Tech. Conf., pp. 139-151, 2017.

[82] D. M. Blough, G. Resta, and P. Santi, “Interference-aware proportional fairness for multi-rate wireless networks,” in Proc. IEEE INFOCOM, pp. 2733-2741, 2014.

[83] M. Yan, Y. Wang, S. Wu, G. Feng, and S. Qin, “An 802.11 MAC optimization mechanism based on throughput proportional fairness in multi-rate WLAN,” in Proc. Int. Conf. Netw. Netw. Appl., pp. 196-202, 2017.

[84] Q. Wu, M. Gong, and C. Williamson, “TCP fairness issues in IEEE 802.11 wireless LANs”,” Comput. Commun., vol. 31 no. 10, pp.2150-2161, 2008.

[85] A. Banchs, P. Serrano, P. Patras, and M. Natkaniec, “Providing throughput and fairness guarantees in virtualized WLANs through control theory,” Mobile Netw. and Appl., vol. 17, no. 4, pp. 435-46, 2012.

[86] B. Abeysekera, T. Matsuda, and T. Takine, “Dynamic contention window control mechanism to achieve fairness between uplink and downlink flows in IEEE 802.11 wireless LANs,” IEEE Trans. Wireless Commun., vol. 7, no. 9, pp. 3517–3525, Sep. 2008.

[87] E. C. Park, D. Y. Kim, H. Kim, and C. H. Choi, “A cross-layer approach for per-station fairness in TCP over WLANs,” IEEE Trans. Mobile Comput., vol. 7, no. 7, pp 397-413, 2008.

[88] J. Kim, S. H. Kim, and D. K. Sung, “Hybrid ARQ- based fairness enhancement in uplink WLAN,” IEEE Trans, on Wireless Commun., vol. 17, no. 7, pp. 4362-4373, Apr. 2018.

[89] K. Akimoto, S. Kameda, and N. Suematsu, “Optimum allocation scheme for user fairness of location-based virtual sector method solving hidden terminal problem in WLAN,” IEEE Trans. on Vehicular Tech., vol. 67, no. 9, pp. 8363-8371, Jun. 2018.

[90] S. S. Priya, and K. Murugan, “Enhancing TCP fairness in wireless networks using dual queue approach with optimal queue selection,” Wireless Personal Commun., vol. 83, no. 2, pp. 1359-1372, Jul. 2015.

[91] J. Kim, H. Jin, J. B. Seo, S. H. Kim, and D. K. Sung, “Adaptive transmission control for uplink/downlink fairness in unsaturated CSMA networks,” IEEE Commun. Let., vol. 23, no. 10, pp. 1879-1882, Jul. 2019.

[92] J. Lei, Y. Wu, and X. Zhang, “An improved active queue management algorithm for time fairness in multirate 802.11 WLAN,” in Proc. 6th Computer Science On-line Conference (CSOC), Prague, Czech Republic, 2017, vol. 2, pp. 161-171.

[93] Y. Le, L. Ma, W. Cheng, X. Cheng, and B. Chen, “A time fairness-based MAC algorithm for throughput maximization in 802.11 networks,” IEEE Trans. on Comput., vol. 64, no. 1, pp. 19-31, Sep. 2013.

[94] R. G. Garroppo, S. Giordano, S. Lucetti, and L. Tavanti, “Providing air-time usage fairness in IEEE 802.11 networks with the deficit transmission time (DTT) scheduler,” Wireless Netw., vol. 13, no. 4, pp. 481-495, Aug. 2007.