

A Study of TCP Throughput Fairness Control Method for Multiple-Host Concurrent Communications in Wireless Local-Area Network

2020, March

Rahardhita Widyatra Sudibyo

Graduate School of
Natural Science and Technology

(Doctor's Course)
OKAYAMA UNIVERSITY

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

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by

Professor Satoshi Denno

and

Professor Yasuyuki Nogami

OKAYAMA UNIVERSITY, March 2020.

TO WHOM IT MAY CONCERN

We hereby certify that this is a typical copy of the original doctor thesis of
Rahardhita Widyatra Sudibyo

Signature of
the Supervisor

Seal of

Prof. Nobuo Funabiki

Graduate School of
Natural Science and Technology

Abstract

Recently, the IEEE802.11n *Wireless Local Area Networks (WLANs)* have been widely adopted around the world due to the characteristics of the flexible coverage, the simple installation, and the low cost. WLAN provides the Internet access by wireless medium and offers a lot of benefits such as mobility, reliability, and portability. Hence, the popularity of WLAN is increasing in government organizations, private companies, and schools for accessing the Internet.

In WLAN, the positions of hosts are mostly non-uniformly, and the number of user or the traffics tend to fluctuate unpredictably. Therefore, we have studied the *elastic WLAN system* that dynamically adapts the network configuration depending on traffic demands and network conditions, to reduce the energy consumption and improve the performance. We have implemented the *elastic WLAN system testbed* adopting *Raspberry Pi* as the software *access point (AP)* using *hostapd*.

The *fairness* of the throughput quality among the hosts in WLAN is important to ensure the fair *quality of services (QoS)* among the users. Since a large number of network applications in the Internet adopt the *transmission control protocol (TCP)*, the TCP fairness is exceedingly critical. However, our preliminary experiments using the elastic WLAN system testbed have revealed that the *TCP throughput unfairness* will appear among hosts, when they are located at different distances from the AP and are concurrently communicating with the same AP. It is assumed that this unfairness is caused by differences in the *modulation and coding scheme (MCS)* and the *TCP window size*.

In this thesis, firstly, we propose the *TCP throughput fairness control method* for two concurrently communicating hosts in the *elastic WLAN system*, to overcome the TCP throughput unfairness problem among two hosts. By introducing a delay at the packet transmission to a faster host at the AP, the slower host will obtain more transmission opportunities. This delay is initially estimated from the measured *received signal strength (RSS)* from every host at the AP. Then, the delay is controlled by the *PI controller* to make the *fairness index* be 1 while measuring the throughput to each host.

Secondly, we generalize the *TCP throughput fairness control method* to deal with any number of concurrently communicating hosts with the same AP. The *target throughput* is introduced as the equal target throughput among the hosts. This target throughput is dynamically updated by the measured throughputs of the hosts, since the throughput is varied in each network field and topology.

Finally, for evaluations, we implement the proposal in the elastic WLAN system testbed using *Raspberry Pi* AP and carry out extensive throughput measurements in indoor fields up to four hosts. The experiment results confirm the effectiveness of the proposals.

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

Acknowledgements

It is my great pleasure to express my heartiest thankfulness to those who gave me the valuable time and supported me in making this dissertation possible. I believe that you are the greatest blessing in my life. Thanks all of you to make my dream successful.

I owe my deepest gratitude to my supervisor, Professor Nobuo Funabiki, who has supported me throughout my thesis with his patience and knowledge. I am greatly indebted to him, whose encouragements, advices, and supports from the beginning enabled me to develop the understanding of this subject, not only in scientific but also in life. He gave me wonderful advices, comments, and guidance when writing papers and presenting them. Thanks for making me what I am today.

I am deeply grateful to my co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for their continuous supports, guidance, and proofreading of this work.

I wish to express my sincere gratitude to Associate Professor Minoru Kuribayashi for his valuable suggestions during my research. I also want to express my gratitude to the course teachers during my Ph.D. study for enlightening me with wonderful knowledge.

I would like to acknowledge Sanyo Electronic Industries Co.,Ltd. in Okayama, Japan and the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT) for financially supporting my Ph.D. study, and the Politeknik Elektronika Negeri Surabaya (PENS), Surabaya, Indonesia for giving me the study leave permission for this study.

I would like to thank for the fruitful discussions and cooperations with many people including Prof. Wen-Chung Kao, Dr. Zainal Arief, Dr. Dadet Pramadihanto, Dr. Sritrusta Sukaridhoto, Dr. Amang Sudarsono, Dr. Nobuya Ishihara, Dr. Md. Manowarul Islam, Dr. Md. Selim Al Mamun, Dr. Khin Khin Zaw, Dr. Kyaw Soe Lwin, Dr. Sumon Kumar Debnath, Ms. Mousumi Saha, Mr. Kwenga Ismael Munene, Mr. Samsul Huda and Mr. Hendy Briantoro. I would like to convey my respect to all the members of FUNABIKI Lab for their support during the period of this study.

Finally, I am eternally grateful to my beloved parents, brother, and all of my friends who always encourage and support me throughout my life. Your supports and understanding gave me the strength and inspiration to overcome any difficulty in my life.

Rahardhita Widyatra Sudibyo
Okayama University, Japan
March 2020

List of Publications

Journal Papers

1. **R. W. Sudibyo**, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A Proposal of TCP Fairness Control Method for Two-Host Concurrent Communications in Elastic WLAN System Using Raspberry Pi Access-Point," *Int. J. Adv. in Sci., Tech, and Eng. Sys.*, vol. 4, No. 6, pp. 10-18, Nov. 2019.
2. **R. W. Sudibyo**, N. Funabiki, M. Kuribayashi, K. I. Munene, 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," Accepted in *Int. J. of Mobile Comp. and Multimedia Comm. (IJMCMC)*, February 2020.

International Conference Papers

3. **R. W. Sudibyo**, N. Funabiki, and W.-C. Kao, "A proposal of hardware channel bonding for IEEE802.11n wireless network using Raspberry Pi," *Proc. IEEE Int. Conf. Consum. Elect. Taiwan (ICCE-TW)*, pp. 277-278, May 19 - 21, 2018.
4. **R. W. Sudibyo**, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A TCP fairness control method for two-host concurrent communications in elastic WLAN system using Raspberry Pi access-point," *Proc. 2nd Int. Conf. Commun. Eng. Tech. (ICCET 2019)*, pp. 76-80, April 2019.

Other Papers

5. **R. W. Sudibyo**, K. S. Lwin, N. Funabiki, M. Saha, and M. Kuribayashi, "A Study of Channel Bonding Configuration and Performance for Raspberry Pi Access-Point in Wireless Local-Area Network," *IEICE Tech. Rep.*, SRW2018-10, pp. 7-12, Aug. 2018.
6. **R. W. Sudibyo**, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A TCP fairness control method for concurrent communications in elastic WLAN system using Raspberry Pi access-point," *IEICE Tech. Rep.*, NS2018-175, pp. 96-100, Dec. 2018.
7. **R. W. Sudibyo**, N. Funabiki, K. I. Munene, and M. M. Islam, "A Proposal of Dynamic Delay Optimization Function in TCP Fairness Control Method for Two-Host Concurrent

Communications in Elastic WLAN System,” IEICE General Conf., BS-4-32, pp. S70-S71, March. 2019.

8. **R. W. Sudibyo**, N. Funabiki, K. I. Munene, and M. M. Islam, “A Generalization of TCP Fairness Control Method Using Transmission Delay for Multiple-Host Concurrent Communication in Elastic WLAN System,” IEICE Society Conf., BS-4-9, pp. S-27-28, Sept. 2019.

List of Figures

1.1	Illustration of elastic WLAN.	2
2.1	Components of IEEE802.11 WLANs.	6
2.2	Operating modes of IEEE802.11 networks.	7
2.3	Extended service set (ESS).	8
2.4	Current and future WiFi Standards.	9
2.5	WiFi channels in 2.4 GHz band.	12
2.6	WiFi channels in 5 GHz band.	12
2.7	IEEE802.11n channel bonding concept.	13
2.8	Comparison between SISO and 4×4 MIMO technology.	13
2.9	OFDM sub-carriers with different channel width.	14
3.1	Design of Elastic WLAN system.	21
4.1	Asahi Riverbed.	27
4.2	TCP link configuration.	28
4.3	Outdoor measurement field.	28
4.4	Throughput at different locations for $H2$	29
4.5	RSS at different locations for $H2$	29
5.1	Background of TCP throughput fairness control method.	32
5.2	TCP link configuration.	35
5.3	Outdoor field.	36
5.4	Results of throughput in outdoor field for 1) <i>without proposal</i>	37
5.5	Results of throughput in outdoor field for 2) <i>initial delay only</i>	37
5.6	Results of throughput in outdoor field for 3) <i>proposal</i>	37
5.7	Results of overall throughput in outdoor field.	38
5.8	Observed TCP fairness index in outdoor field.	38
5.9	Experimet field for two-room case.	39
5.10	TCP fairness index results in indoor field for two-room case.	39
5.11	Indoor field.	39
5.12	Results of throughput in indoor field for 1) <i>without proposal</i>	40
5.13	Results of throughput in indoor field for 2) <i>initial delay only</i>	40
5.14	Results of throughput in indoor field for 3) <i>proposal</i>	40
5.15	Results of overall throughput in indoor field.	40
5.16	TCP fairness index results in indoor field for four-room case.	41
5.17	Time scenario for dynamically connecting/disconnecting host.	41
5.18	Individual throughput of $H1$ and $H2$ with dynamic hosts.	42

6.1	TCP link configuration.	46
6.2	Experiment field for two-host, three-host, and four-host case.	47
6.3	Two-host case individual throughput.	47
6.4	Two-host case fairness index comparison.	48
6.5	Three-host case individual throughput.	48
6.6	Three-host case fairness index comparison.	49
6.7	Four-host case individual throughput.	49
6.8	Four-host case fairness index comparison.	49
6.9	Experiment field for dynamically connecting/disconnecting host.	50
6.10	First scenario for dynamic host.	50
6.11	Second scenario for dynamic host.	51
6.12	First scenario individual throughput.	51
6.13	Second scenario individual throughput.	52

List of Tables

2.1	IEEE802.11 Standards.	8
2.1	IEEE802.11 Standards.	9
2.2	Characteristics of common IEEE802.11 standards.	10
2.3	IEEE802.11n specification.	11
2.4	Effects of channel bandwidth and spatial stream's selection towards IEEE802.11n's throughput.	12
2.5	Example of <i>Jain's fairness index</i>	17
3.1	Device environment and software in testbed.	23
4.1	Device specifications.	28
5.1	Hardware and software specifications.	35
6.1	Device specifications.	46

Contents

Abstract	i
Acknowledgements	iii
List of Publications	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Background	1
1.2 Contributions	2
1.3 Contents of Thesis	3
2 IEEE 802.11 Wireless Network Technologies	5
2.1 Overview of IEEE802.11 WLAN	5
2.1.1 Advantages of WLAN	5
2.1.2 Components of IEEE802.11 WLAN	6
2.1.3 Operating Modes for IEEE802.11 WLAN	7
2.1.4 IEEE802.11 Standards for WLAN	7
2.2 IEEE802.11n Protocol	11
2.3 Features of IEEE802.11n Protocol	11
2.4 Linux Tools for Wireless Networking	14
2.4.1 'arp-scan' - to Explore Currently Active Devices	14
2.4.2 'bc' - to Calculate Arithmetic	15
2.4.3 'hostapd' - to Make AP-mode Linux-PC	15
2.4.4 'ssh' - to Remotely Execute Command	15
2.4.5 'iw' - to Collect Information of Active Network Interface	15
2.4.6 'iperf' - to Measure Link Speed	16
2.4.7 'tc' - to Manipulate Traffic Control Setting	16
2.5 Jain's Fairness Index	16
2.6 Summary	17
3 Review of Previous Studies	19
3.1 Elastic WLAN System	19
3.1.1 Overview	19
3.1.2 Related Works in Literature	20

3.1.3	Design and Operational Flow	20
3.2	Throughput Estimation Model for IEEE802.11n Protocol	21
3.2.1	Overview of Model	21
3.2.2	Receiving Signal Strength Estimation by Log-Distance Path Model	21
3.2.3	Throughput Estimation by Sigmoid Function	22
3.2.3.1	Multi-Path Effect Consideration	22
3.3	Testbed Implementation Using Raspberry Pi	23
3.3.1	Implementation Environment/Platform	23
3.3.2	System Topology	24
3.3.3	AP Configuration of Raspberry Pi	24
3.4	Summary	25
4	Experimental Observations of TCP Throughput Unfairness Problem of Multiple-Host Concurrent Communications	27
4.1	Introduction	27
4.2	Experiment Setup	28
4.3	Experiment Results	29
4.4	Summary	30
5	Proposal of TCP Throughput Fairness Control Method for Two-Host Concurrent Communications	31
5.1	Introduction	31
5.2	Idea	31
5.3	Initial Delay Calculation	32
5.4	Dynamic Delay Optimization by PI Controller	32
5.4.1	Testbed Implementation	33
5.4.1.1	RSS Measurement	33
5.4.1.2	Transmission Delay Application	33
5.4.1.3	Dynamic Delay Optimization	34
5.5	Evaluations with Static Hosts	34
5.5.1	Experiment Setup	35
5.5.2	Experiments in Outdoor Field	35
5.5.2.1	Experiment Field	36
5.5.2.2	Throughput Results	36
5.5.2.3	Fairness Index Results	38
5.5.3	Experiments in Indoor Field	38
5.5.3.1	Experiments in Two-room Case	38
5.5.3.2	Experiments in Four-room Case	39
5.6	Evaluations with Dynamic Hosts	41
5.6.1	Experiment Setup	41
5.6.2	Results	42
5.7	Summary	42
6	Proposal of TCP Throughput Fairness Control Method for Multiple-Host Concurrent Communications	43
6.1	Introduction	43
6.2	Target Throughput	43

6.3	Delay Optimization by PI Controller	44
6.4	Testbed Implementation	44
6.4.1	RSS Measurement	44
6.4.2	Transmission Delay Application	45
6.4.3	Dynamic Delay Optimization	45
6.5	Evaluation with Static Hosts	46
6.5.1	Experiment Setup	46
6.5.2	Experiment Field	47
6.5.3	Results for Two-host Case	47
6.5.4	Results for Three-host Case	48
6.5.5	Results for Four-host Case	48
6.6	Evaluation with Dynamic Hosts	50
6.6.1	Experiment Setups	50
6.6.2	Results for First Scenario	51
6.6.3	Results for Second Scenario	51
6.7	Summary	51
7	Related Works in Literature	53
8	Conclusion	57
	References	59

Chapter 1

Introduction

1.1 Background

The Internet has been growing rapidly everywhere by providing effective and efficient communication environments. People can enjoy opportunities to communicate with various people and to access to various information through the Internet. The Internet is affecting any society because it is versatile and convenient. The Internet has attracted inordinate amounts of public attentions [1].

The IEEE 802.11 *wireless local-area network (WLAN)* is most commonly used in many locations as an access network to the Internet. WLAN has been extensively applied around the world due to its characteristics of the easy installation, the flexible coverage, and the low cost [2]. WLAN offers communication abilities through the wireless medium. WLAN has several benefits such as mobility, flexibility, and portability. Hence the popularity of WLAN is increasing in offices, schools, hotels, and various public spaces [3].

The *IEEE 802.11 standard* was first released from the IEEE LAN/MAN committee in June 1997 [4, 5]. Then, upgrades of the standard have been released to catch up with advancements in communication technologies. Among several protocols on the *IEEE802.11 standard*, the *IEEE 802.11n* has been commonly used in many fields, although the maximum throughput is lower than a newer protocol *IEEE 802.11ac*. The IEEE 802.11n offers a variety of enhanced features over previous IEEE 802.11 protocols namely, 11a, 11b, 11g.

In WLAN, hosts are mostly non-uniformly located [6], and the number of users or the traffics tends to fluctuate unpredictably depending on the time and day of the week [7, 8]. A WLAN user may exist at any location if it is within the wires range of an *access point (AP)*. Users can join and leave WLAN frequently. For example, in a university, the number of network users increases in the afternoon during weekdays, while it decreases in the morning or evening, and the whole days at the weekend. Besides, the conditions of network devices and communication links may be influenced by different factors, such as weather changes, bandwidth controlled by their authorities, power shortages, or device failures [9].

We have studied the *elastic WLAN system* that dynamically adapts the network configuration depending on traffic demands and network conditions, to reduce the energy consumption and improve the performance [10–15]. Three types of APs, namely, a *dedicated AP (DAP)*, a *virtual AP (VAP)*, and a *mobile AP (MAP)* are considered in them. A *DAP* represents a commercial AP, a *VAP* does a host PC of a user in the network that installs the software of AP functions [16, 17], and a *MAP* does a *mobile router*. In addition, we have implemented the testbed for experiments using *Raspberry Pi* as the software *access point (AP)* using *hostapd*. *Raspberry Pi* is a card-size, single-board computer that has the built-in wireless network interface (NIC) supporting

IEEE802.11n [18].

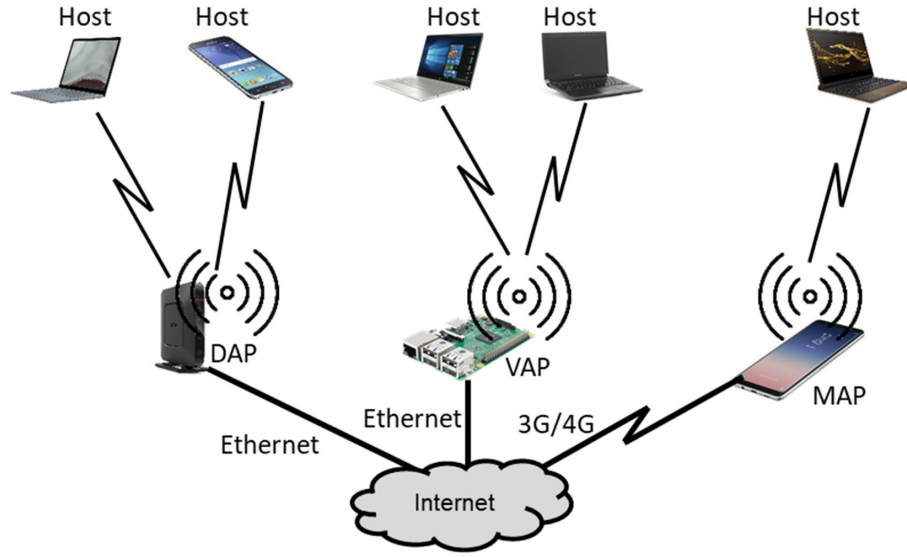


Figure 1.1: Illustration of elastic WLAN.

Throughput fairness is one of interdisciplinary research issues, which is typically related to resource allocations of the network. In WLAN, all the hosts should gain the fair amount of bandwidth or throughput for the fair *quality of service (QoS)*. It is important for service providers to ensure the fair quality of the services to the network users [20]. Fairness issues in WLAN have been explored extensively [19, 36–41]. Since most of the network applications use TCP, the *TCP fairness* is most important.

To observe the TCP unfairness problem in WLAN, we are conducted TCP throughput measurements using testbed when two hosts are concurrently communicating with the same AP [42] in the outdoor environment of *Asahi Riverbed*. By using *Homedale* [43], this environment is cleared of interfering signals.

The measurement results revealed that the TCP throughput is not fair among the two hosts when they are located at different relative positions from the AP. It is assumed that this unfairness was caused by differences in the *TCP window size* and the *modulation and coding scheme (MCS)* among them. Because of them, the packet transmission interval of the farthest host hereafter becomes longer, and the interval for the nearest host becomes shorter.

1.2 Contributions

In this thesis, we propose the *TCP throughput fairness control method* for multiple-host concurrent communications to enhance the throughput performance of each host in WLAN.

Firstly, we propose the *TCP throughput fairness control method* for two concurrently communicating hosts in the elastic WLAN system, to overcome the TCP throughput unfairness problem among two hosts [42, 44–46]. The *delay* is implemented at the AP in the packet transmission to the nearer host. After that, it can be predicted that the nearer host will decrease the throughput, and the farther host will enhance it by obtaining more transmission opportunities. This delay is firstly calculated by the *received signal strength (RSS)* from hosts. Then, the throughput fairness

index between the two hosts is continuously observed during communications, and the delay is dynamically optimized by using the *PI controller* [92] to achieve the desired throughput fairness.

Secondly, we generalize the *TCP throughput fairness control method* to deal with any number of concurrently communicating hosts with the same AP [48]. The *target throughput* is introduced as the equal target throughput among the hosts. This target throughput is dynamically updated by the measured throughputs of the hosts, since the throughput is varied in each network field and topology.

Finally, for performance evaluations, we implement the proposal in the elastic WLAN system testbed using *Raspberry Pi* AP, and carry out extensive throughput measurements in indoor fields up to four hosts. The experiment results confirm the effectiveness of the proposals.

1.3 Contents of Thesis

The remaining part of this thesis is organized as follows.

In Chapter 2, we introduce 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, and the fairness index.

In Chapter 3, we review our previous related studies.

In Chapter 4, we describe the TCP unfairness problem of multiple-host concurrent communications.

In Chapter 5, we present the *TCP throughput fairness control method* for two-host concurrent communications, the implementation in the elastic WLAN system testbed, and the evaluations.

In Chapter 6, we present the *TCP throughput fairness control method* for multiple-host concurrent communications and the evaluations.

In Chapter 7, we review relevant works in literature.

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

Chapter 2

IEEE 802.11 Wireless Network Technologies

In this chapter, we briefly introduce wireless network technologies for backgrounds of this dissertation. First, we review *IEEE802.11 protocols*. Then, we discuss the *IEEE802.11n protocol*, especially, the key features of the protocol. After that, we outline Linux tools and commands for WLANs that are used for our measurements and the implementation of elastic WLAN system. Finally, we review the Jain's fairness index as an indicator in our thesis.

2.1 Overview of IEEE802.11 WLAN

The IEEE802.11 *Wireless local-area network (WLAN)* [49] is an alternative or extension to a wired LAN, and supports flexibility for data communications. IEEE802.11 standards determine *physical (PHY)* and *media access control (MAC)* layer specifications for implementing high-speed *wireless local area network (WLAN)* technologies. WLAN is an extension to a wired LAN that enables the user mobility by the wireless connectivity and supports the flexibility in data communications. It can reduce the cabling costs in the home or office environments by transmitting and receiving data over the air using *radio frequency (RF)* technology.

2.1.1 Advantages of WLAN

WLAN offers several advantages over the traditional wired networks. Specific benefits are included in the following [49]:

- *User mobility:*
In a wired network, users need to use wired lines to stay connected to the network. On the other hand, WLAN allows user mobility within the coverage area of the network.
- *Easy and rapid deployment:*
WLAN can exclude the requirement of network cables between hosts and connection hubs or APs. Thus, the installation of WLAN can be much easier and quicker than wired LAN.
- *Cost:*
The initial installation cost can be higher than the wired LAN, but the life-cycle cost can be significantly lower. In the environment requiring frequent movements or reconfigurations of the network, WLAN can provide the long-term cost profit.

- *Flexibility:*
The network coverage area by WLAN can be easily expanded because the network medium is already everywhere.
- *Scalability:*
WLAN can be configured in a variety of topologies suitable to applications. WLAN can support both peer-to-peer networks suitable for a small number of users and full infrastructure networks of thousands of users.

2.1.2 Components of IEEE802.11 WLAN

IEEE802.11 WLAN consists of four primary components as shown in Figure 2.1 [49]:

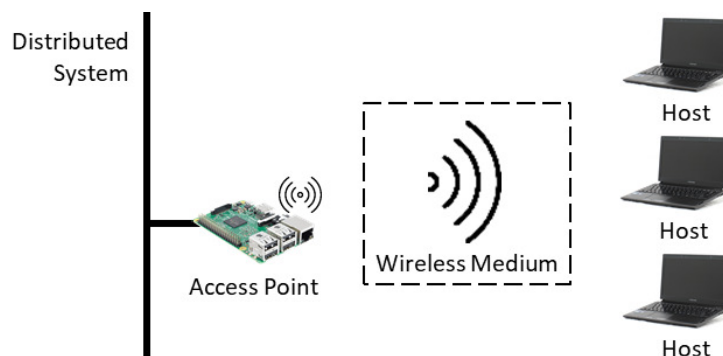


Figure 2.1: Components of IEEE802.11 WLANs.

- *Stations or hosts:*
WLAN transfers data between *stations or hosts*. A station in WLAN indicates an electronic device such as a desktop/laptop PC, a smartphone, or a tablet that has the capability of accessing the network over the wireless network interface card (NIC).
- *Access points (APs):*
An AP acts as the main radio transceiver or a generic base station for WLAN that plays the similar role as a hub/switch in a wired Ethernet LAN. It also performs the bridging function between the wireless and the wired networks with some other tasks.
- *Wireless medium:*
The IEEE802.11 standard uses the wireless medium to flow the information from one host to another host in a network.
- *Distribution system:*
When several APs are connected together to form a large coverage area, they must communicate with each other to trace the movements of the hosts. The distribution system is the logical component of WLAN which serves as the backbone connections among APs. It is often called as the *backbone network* used to relay frames between APs. In most cases, *Ethernet* is commonly used as the backbone network technology.

2.1.3 Operating Modes for IEEE802.11 WLAN

The elementary unit of IEEE802.11 WLAN is simply a set of hosts that can communicate with each other known as the *basic service set (BSS)*. Based on the types of BSS, the IEEE802.11 supports two operating modes as illustrated in Figure 2.2.

- *Independent or ad hoc mode:*

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

- *Infrastructure mode:*

In this mode, the stations exchange their information through an AP as shown in Figure 2.2(b). A single AP acts as the main controller to all the hosts within its BSS, known as *infrastructure BSS*. In this mode, a host must be associated with an AP to obtain network services [50].

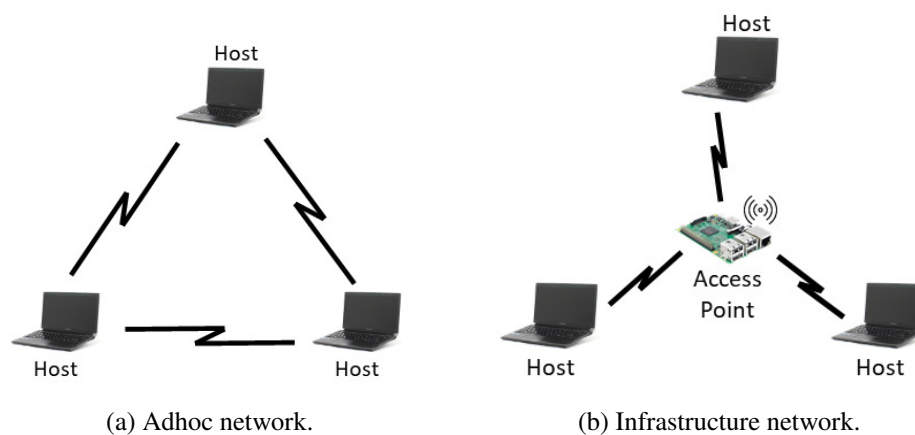


Figure 2.2: Operating modes of IEEE802.11 networks.

In addition, multiple BSSes can be connected together with a backbone network to form *extended service set (ESS)* as shown in Figure 2.3. ESS can form a large size WLAN. Each AP in ESS is given an ID called the *service set identifier (SSID)*, which serves as a “network name” for the users. Hosts within the same ESS can communicate with each other, even if they are in different basic service areas.

2.1.4 IEEE802.11 Standards for WLAN

The IEEE802.11 working group has improved the existing PHY and MAC layer specifications to realize WLAN at the 2.4-2.5 GHz, 3.6 GHz and 5.725-5.825 GHz unlicensed ISM (*Industrial, Scientific and Medical*) frequency bands defined by the ITU-R. In this working group, several types of IEEE Standard Association Standards are available, where each of them comes with a letter suffix, covers from wireless standards, to standards for security aspects, Quality of Service (QoS) and others, shown in Table 2.1 [50–55].

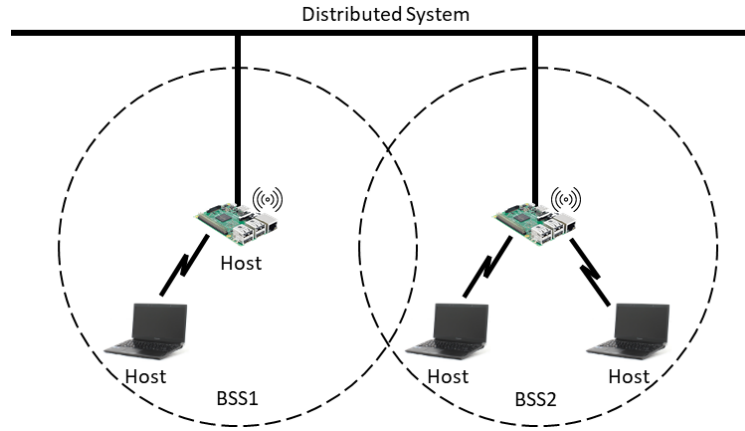


Figure 2.3: Extended service set (ESS).

Table 2.1: IEEE802.11 Standards.

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

Table 2.1: IEEE802.11 Standards.

Standard	Purpose
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 <i>1Gbps</i> for multi-station operation and <i>500Mbps</i> 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 shows the current and future WiFi standards. Among these standards, the common and popular ones are IEEE802.11a, 11b, 11g, 11n, and 11ac. For the physical layer, the 11a/n/ac use *Orthogonal Frequency Division Multiplexing (OFDM)* modulation scheme while the 11b uses the *Direct Sequence Spread Spectrum (DSSS)* technology. 11g supports both technologies. Table 2.2 summarizes the features of these common WiFi standards [49, 56, 57].

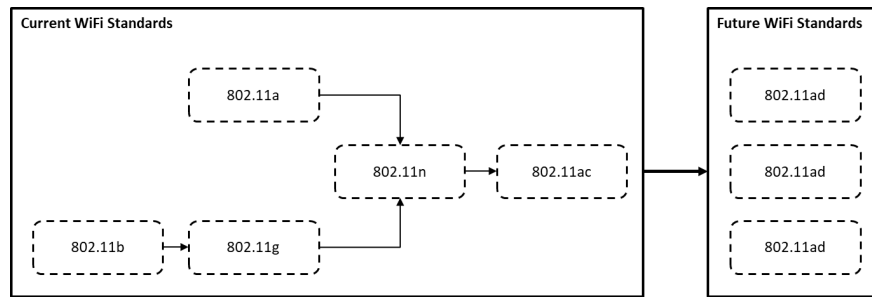


Figure 2.4: Current and future WiFi Standards.

- *IEEE802.11b*: IEEE802.11b operates at 2.4 GHz band with the maximum data rate up to 11 Mbps. 11b is considered to be a robust system and has a capacity to compensate the

Table 2.2: Characteristics of common IEEE802.11 standards.

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

¹ CCK: Complementary Code Keying

² DSSS: Direct Sequence Spread Spectrum

same IEEE802.11 protocols. Because of the interoperability feature between products from different vendors, this standard has not only boosted the manufacturing of the products but also motivated the competitions between WLAN vendors. The limitation of this standard is the interference among the products using *industrial, scientific and medical* (ISM) band that uses the same 2.4 GHz band of frequency [58, 59].

- *IEEE802.11a*: IEEE802.11a operates at 5 GHz ISM band. It adopts on orthogonal frequency division multiplexing (OFDM) coding scheme that offers a high data rates up to 6, 12, 24, 54 Mbps, and sometimes beyond this speed in comparison to 11b. Two main limitations of 11a are the compatibility issue of the 11a products with 11b products and the unavailability of 5 GHz band with free of cost for all the countries in the world [58, 59].
- *IEEE802.11g*: IEEE suggested 11g standard over 11a to improve the 2.4 GHz 11b technology. 11g introduces two different modulation techniques including the *packet binary convolution code (PBCC)* that supports the data rate up to 33Mbps and the *orthogonal frequency division multiplexing (OFDM)* that supports up to 54Mbps data rate. Compatibility issues are also resolved in 11g products with 11b products [58, 59].
- *IEEE802.11n*: The primary purpose of initiating the 11n standard to improve the usable range and data rate up to 600Mbps. 11n supports both of 2.4 GHz and 5 GHz ISM band *unlicensed national information infrastructure* (UNII) band, and is backward compatible with earlier standards. It introduces new technology features including the use of *channel bonding* and *multiple antennas* to get the better reception of the RF signals to enhance the throughput and coverage range [58, 60].
- *IEEE802.11ac*: The aim of the 11ac standard to improve individual link performance and total network throughput of more than 1Gbps. Many of the specifications like static and dynamic channel bonding and simultaneous data streams of 11n have been kept and further en-

hanced for 11ac to reach the gigabit transmission rate. It supports static and dynamic channel bonding up to 160MHz and *Multi-User Multiple-Input-Multiple-Output (MU-MIMO)*. 11ac operates only on the 5 GHz band [61–63].

2.2 IEEE802.11n Protocol

In this section, we overview the IEEE802.11n protocol that has been used for our throughput measurements and implementations. IEEE802.11n is an amendment to the IEEE802.11 2007 wireless networking standard. This standard was introduced with 40 MHz bandwidth channels, *Multiple-Input-Multiple-Output (MIMO)*, frame aggregation, and security improvements over the previous 11a, 11b, and 11g standards. Table 2.3 shows a brief summary of this standard.

Table 2.3: IEEE802.11n specification.

Specification	IEEE802.11n	
Frequency Band	2.4 GHz	5 GHz
Simultaneous Uninterrupted Channel	2 ch	9 ch
Available Channel	13 ch	19 ch
Max. Speed	600Mbps	
Max. Bandwidth	40 MHz	
Max. Spatial Streams	4	
Subcarrier Modulation Scheme	64 QAM	
Release Date	Sept 2009	

The IEEE802.11n is available both on 2.4 GHz and 5 GHz bands. Nowadays, 2.4 GHz is very popular as it was inherited from the IEEE802.11g. This frequency band has become crowded with lots of WiFi signals using the same channel. As a result, these WiFi signals with adjacent channels will suffer from interferences between them, and end up with throughput performance degrades [54, 56].

For 2.4 GHz band, there is a limited number of non-interfered channels, which are Channel 3 and Channel 11 in 40 MHz bandwidth. While for 20 MHz bandwidth, Channel 1, Channel 6, and Channel 11 are free from interference. In overall, the wider bandwidth will reduce the number of free channels. Figure 2.5 [55] shows the WiFi channels for IEEE802.11n 2.4 GHz band.

In the 5 GHz band of IEEE802.11n protocol, it has 19 uninterrupted channels available with the 20MHz bandwidth. In the 40MHz bandwidth, which doubles the channel width from the 20MHz, there are nine channels. For the 80MHz bandwidth, there are four of them. Figure 2.6 shows these WiFi channels for the IEEE802.11n 5 GHz band [64].

2.3 Features of IEEE802.11n Protocol

IEEE802.11n protocol incorporates several new technologies to boost up its performance. The standard uses the multiple antennas technology, channel bonding, frame aggregation, and security improvements mechanism to improve the throughput. In this section, we describe these features of IEEE802.11n Protocol.

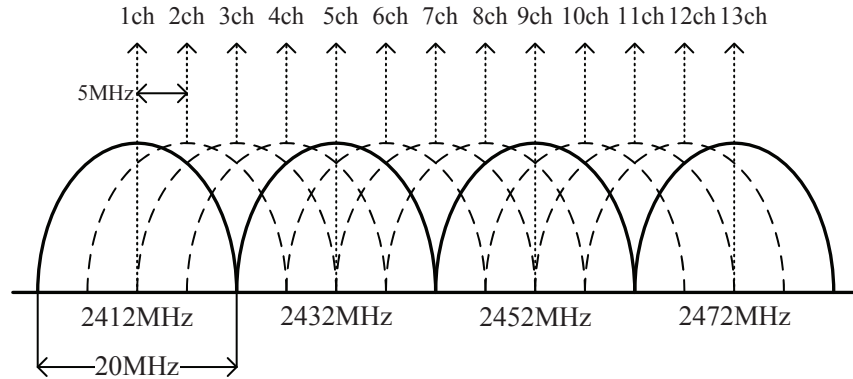


Figure 2.5: WiFi channels in 2.4 GHz band.

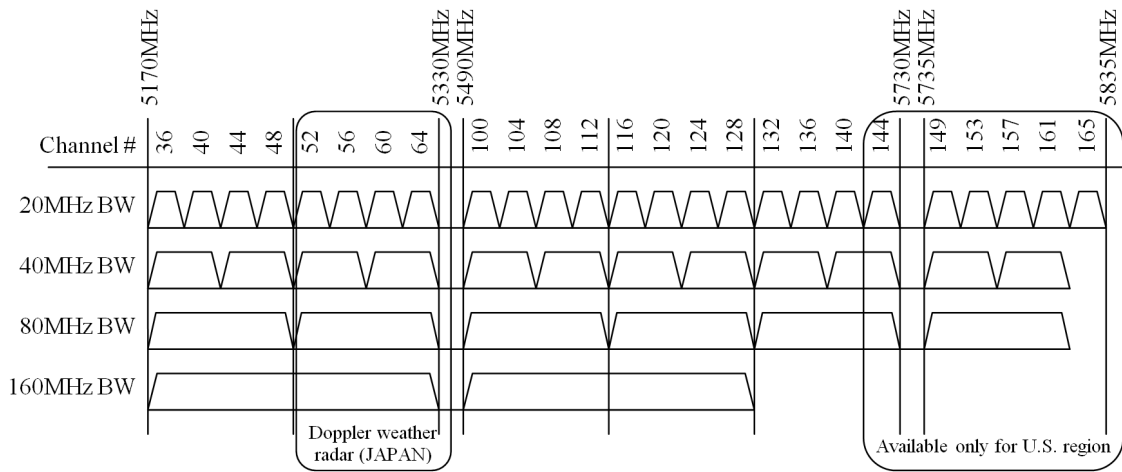


Figure 2.6: WiFi channels in 5 GHz band.

- **Channel Bonding:**

In the channel bonding, each channel can operate with the 40MHz bandwidth by using two adjacent 20MHz channels together to double its physical data rate [65] as shown in Figure 2.7. However, the usage of the channel bonding will reduce the available non-interfered channels for other devices as there are only two non-interfered bonded channels available for IEEE802.11n protocol at 2.4 GHz band. Table 2.4 shows the usage of different channel bandwidths and spatial streams towards the throughput of IEEE802.11n.

Table 2.4: Effects of channel bandwidth and spatial stream's selection towards IEEE802.11n's throughput.

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

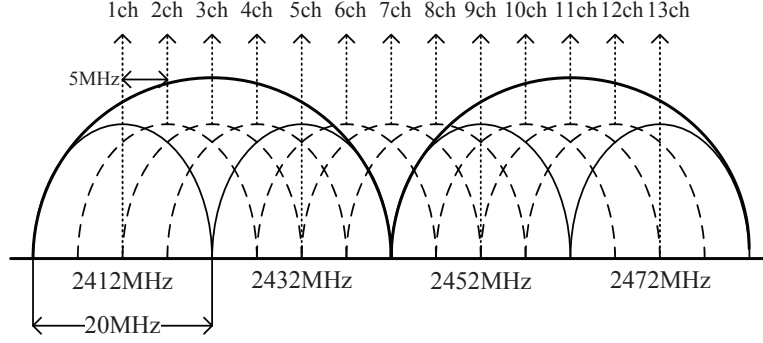


Figure 2.7: IEEE802.11n channel bonding concept.

- **MIMO (Multiple-Input-Multiple-Output):**

In MIMO, the throughput can be linearly increased to the number of transmitting (T_X) and receiving (R_X) antennas up to four times, without the additional bandwidth or transmission power. The multiple antenna configurations in *MIMO* can overcome the detrimental effects of multi-path and fading, trying to achieve high data throughput in limited bandwidth channels. The performance can be enhanced over the single antenna technology in *Single-Input Single-Output* (SISO). For example, in the 4×4 MIMO, four independent data streams can be multiplexed and transmitted simultaneously with the *spatial division multiplexing* (SDM), to speed up the transmission capacity by quadruple as shown in Figure 2.8. When the *space-*

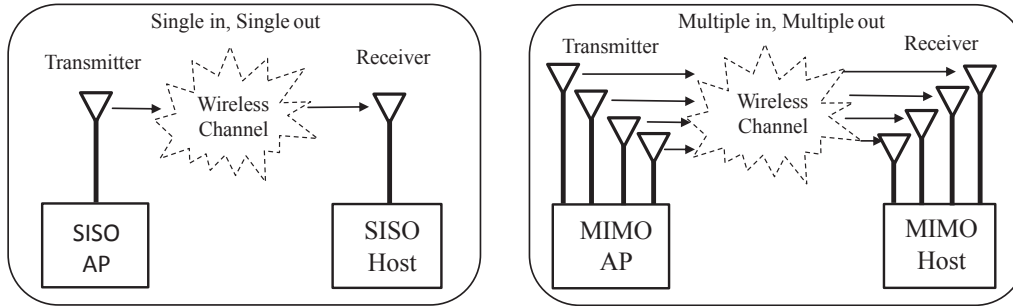


Figure 2.8: Comparison between SISO and 4×4 MIMO technology.

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

- **Frame Aggregation:**

IEEE802.11n also provides the performance improvement through the frame aggregation in the MAC layer, besides MIMO. The frame aggregation can transmit multiple frames by one big frame with a single pre-amble and header information to reduce the overhead by them. IEEE802.11n introduces the *Aggregation of MAC Service Data Units* (A-MSDUs) and *Aggregation of MAC Protocol Data Units* (A-MPDUs). Frame aggregation is a process of packing multiple A-MSDUs and A-MPDUs together to reduce the overheads and average them over multiple frames, thereby increasing the user level data rate [66].

- **Modulation and Coding Scheme:**

Various modulation, error-correcting codes are used in IEEE802.11n, which are represented

by a Modulation and Coding Scheme (MCS) index value or *mode*. IEEE802.11n defines 31 different modes and provides the greater immunity against selective fading by using the Orthogonal Frequency Division Multiplexing (OFDM). This standard increases the number of OFDM sub-carriers of 56 (52 usable) in *High Throughput (HT)* with 20MHz channel width and 114 (108 usable) in HT with 40MHz. Each of these sub-carriers is modulated with *BPSK*, *QPSK*, *16-QAM*, or *64-QAM*, and *Low-Density Parity-check Code (LDPC) Forward Error Correction (FEC)* coding rate of 1/2, 2/3, 3/4 or 5/6 [66].

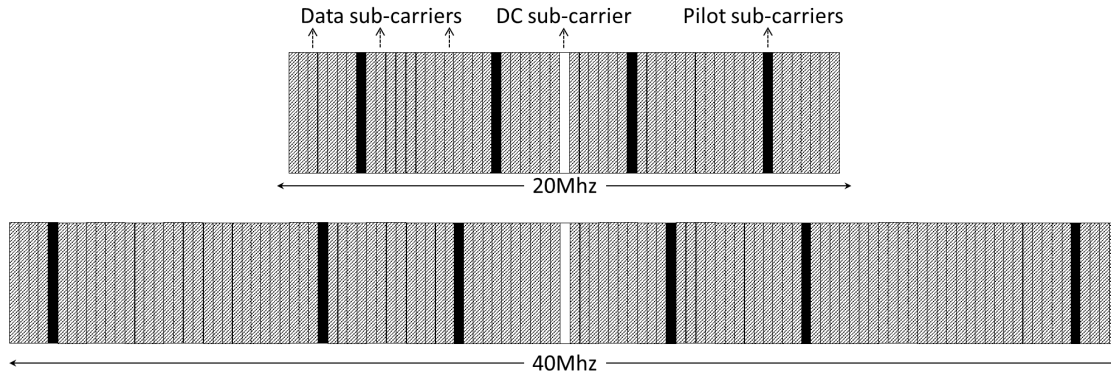


Figure 2.9: OFDM sub-carriers with different channel width.

2.4 Linux Tools for Wireless Networking

As an open-source operating system, Linux has been used as a platform to implement new algorithms, protocols, methods, and devices for advancements of wireless networks [68]. In this section, we give the overview of the Linux tools and software used for the measurements performed throughout the thesis and the implementation of the *elastic WLAN system*.

2.4.1 ‘arp-scan’ - to Explore Currently Active Devices

arp-scan [69] is a command-line tool using the ARP protocol to discover and fingerprint the IP hosts in the local area network. This tool can discover all the devices using IP addresses in the network. *arp-scan* works on an IEEE802.11 wireless (WiFi) network as well as a wired Ethernet, where the wireless network uses the same data-link protocol. In the Linux operating system, *arp-scan* can be installed by downloading the source code from [70] 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=eth0 --localnet
```

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

2.4.2 'bc' - to Calculate Arithmetic

bc command is used for doing arithmetic calculation in bash or shell script with Linux or Unix operating system [71]. This command is an *arbitrary-precision calculator language* with syntax similar to the C programming language. *bc* command is usually installed by default on the Ubuntu distribution. It can be installed using the following command manually:

```
$ sudo apt-get install bc
```

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

hostapd is a Linux tool for the AP and the authentication server. It implements IEEE802.11 AP managements, along with other IEEE802.1X protocols and security applications. In the Linux operating system, *hostapd* can be installed by downloading the source code from [72] or using the following command:

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

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

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

2.4.4 'ssh' - to Remotely Execute Command

ssh is an abbreviation of *Secure Shell* that is a cryptographic network protocol to securely initiate a shell session on a remote machine [73, 74]. It is operated in two parts: *SSH client* and *SSH server*, and establishes a secure channel between them over an insecure network. The open source version of *ssh* is *OpenSSH* [75] that can be installed using the following command [76]:

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

The following list shows an example to remotely access the AP through the network using *ssh* [73, 74, 77]:

```
$ ssh username@172.24.4.2  
username@172.24.4.2's password:
```

Here, 172.24.4.2 represents the IP address of the AP.

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

iw [79] is a command-line Linux tool to display and change the parameters of the active network interface for wireless operations. This is nl80211 based CLI configuration utility for wireless devices. It supports all new drivers that have been added to the kernel recently. It can also be used to display the wireless network parameters and statistics. *iw* is usually installed by default in the Ubuntu distribution. It can also be installed manually using the following command:

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

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

```
$ iw dev wlan0 scan
```

2.4.6 'iperf' - to Measure Link Speed

iperf [80] is a software to measure the available throughput or bandwidth on IP networks. It supports both TCP and UDP protocols along with tuning various parameters related to timing and buffers, and reports the bandwidth, the loss, and other parameters. *iperf* is usually installed by default in the Ubuntu distribution. It can also be installed manually using the following command:

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

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

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

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

2.4.7 'tc' - to Manipulate Traffic Control Setting

tc [81] is a command-line Linux tool to configure the kernel packet scheduler. It helpful tools to simulate packet delay and loss for UDP or TCP applications, or limit the bandwidth usage of a particular service to simulate Internet connections (DSL, Cable, T1, etc). *tc* is usually installed by default in the Ubuntu distribution. It can also be installed manually using the following command:

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

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

```
$ iw dev wlan0 scan
```

2.5 Jain's Fairness Index

In this section, we overview the *Jain's fairness index* that has been used for fairness indicator in our measurement and implementations. Various algorithm have been proposed to measure fairness level [19], such as *Jain's fairness index*, *entropy*, *entropy*, *max-min*, and *proportional fairness*. Between them, *Jain's fairness index* is more popularly used for quantitative fairness measures. This index was first proposed in [82] by Rajendra K. Jain. Eq. (2.1) was adopted in our thesis as the representative TCP fairness index:

$$F(n) = \frac{\left(\sum_{i=1}^k x_i\right)^2}{k \times \sum_{i=1}^k x_i^2} \quad (2.1)$$

where k represents the number of TCP connections and x_i does the throughput for the i -th connection. When all the connections receive the same throughput, this index will become 1,

Table 2.5: Example of *Jain's fairness index*.

Host-n	Throughput (Mbps)				
	Case 1	Case 2	Case 3	Case 4	Case 5
H1	30	30	30	30	30
H2	0	5	10	10	30
H3	1	3	10	20	30
H4	5	5	5	20	30
F(n)	0.35	0.48	0.67	0.89	1

which means that all the connections are 100% fair. As the disparity increases, the fairness will decrease. As an example, table 2.5 shows different *Jain's fairness index* values with different throughput performance. In this table, we can see that the allocation tends to be fairer when *Jain's fairness index* is closer to 1.

2.6 Summary

In this chapter, we presented wireless network technologies, key features of IEEE802.11n protocols, Linux tools devices, and *Jain's fairness index* that are adopted in this thesis for our experiments, implementations, and simulations. In the next chapter, we will review our previous related studies.

Chapter 3

Review of Previous Studies

In this chapter, we briefly overview our previous studies that are related to this thesis. First, we review the study of the elastic WLAN system. Then, we review the study of the throughput estimation model for the IEEE802.11n link in WLAN. Finally, we review the implementation details of the elastic WLAN system testbed using Raspberry Pi APs.

3.1 Elastic WLAN System

In this section, we review the study of the elastic WLAN system. This system can dynamically optimize the network configuration by activating/deactivating APs, according to traffic demands and network conditions.

3.1.1 Overview

Currently, WLANs have been increasingly installed in business organizations, educational institutions, and public places like buses, airplanes, or trains. In these cases, unplanned or independently controlled APs can lead to problems resulting in performance degradations and/or wastages of energy. In one hand, WLANs can suffer from over-allocation problems with redundant APs that have overlapped coverage areas. On another hand, WLANs can be overloaded with hosts suffering from low performances. Therefore, WLANs should be adaptive by changing the allocations of APs and the associations of hosts to the APs, according to the network traffic demands and conditions. To realize this goal, we have studied the elastic WLAN system. The elastic WLAN system can play an important role for energy-saving and performance improvements for WLANs.

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

1. Reduction of operational cost and energy usage:

- An organization generally allocates the number of APs required to ensure high performances during peak-hours, and keep the WLANs active for 24-hours a day. However, only a few of the APs are used during off-peak hours or in holidays. The elastic WLAN system activates the minimum number of APs required for the current traffic demand and saves a huge amount of energy.
- Developing countries suffer unreliable Internet access due to discontinuities of electricity supplies for the time being. In such situations, the elastic WLAN system can improve the network performance through the efficient use of the available power sources.

2. Improvement of WLAN performances:

- When current active APs become overloaded by increasing users and suffer from low network performances, the activations of additional APs can maintain the desirable network performance to cope with the increasing traffic demands.
- Due to the backbone network inefficiency, power shortages, or ISP failures, an organization may receive a lower bandwidth than its expected one. In such cases, cellular network base *mobile APs* should be activated in the elastic WLAN system to ensure the desirable network performance.
- When the number of active APs become high in a network area, users may suffer from the interferences among them due to overlapping of frequency signals. In such situation, the elastic WLAN system can dynamically change assigned channels to APs in a way to minimize the interferences among them and improve the performance.

3.1.2 Related Works in Literature

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

In [83], Lei et al. proposed a campus WLAN framework based on the software defined network (SDN) technology.

In [84], Luengo et al. also proposed a design and implementation of a testbed for integrated wireless networks based on SDN. Although this framework is flexible to design and manage, it requires SDN-enabled devices and network virtualizations.

In [85, 86], Sukaridhoto et al. proposed a Linux implementation design using OpenFlow of the fixed backoff-time switching (FBS) method for the wireless mesh network. Their implementation requires Linux driver kernel modifications and specific WLAN hardware interfaces.

In [87], Ahmed et al. describe the important design issues in preparing a large-scale WLAN testbed for evaluation of centralized control algorithms and presented experimental results. They did not analyzed the power-saving and adaptive control mechanism of centralized WLANs which is one of the the main purposes of our research.

In [88], Debele et al. proposed a Resource-on-Demand (RoD) strategy for energy-saving in dense WLANs where they analyzed user behavior in the network and formulated the stochastic characteristics.

3.1.3 Design and Operational Flow

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

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

1. The server will examine the network devices and then collect the necessary information for the active AP configuration algorithm.
2. Then, it will execute the active AP configuration algorithm to obtain the algorithm output including the active APs, the AP-host associations, and the allocated channels to the APs.

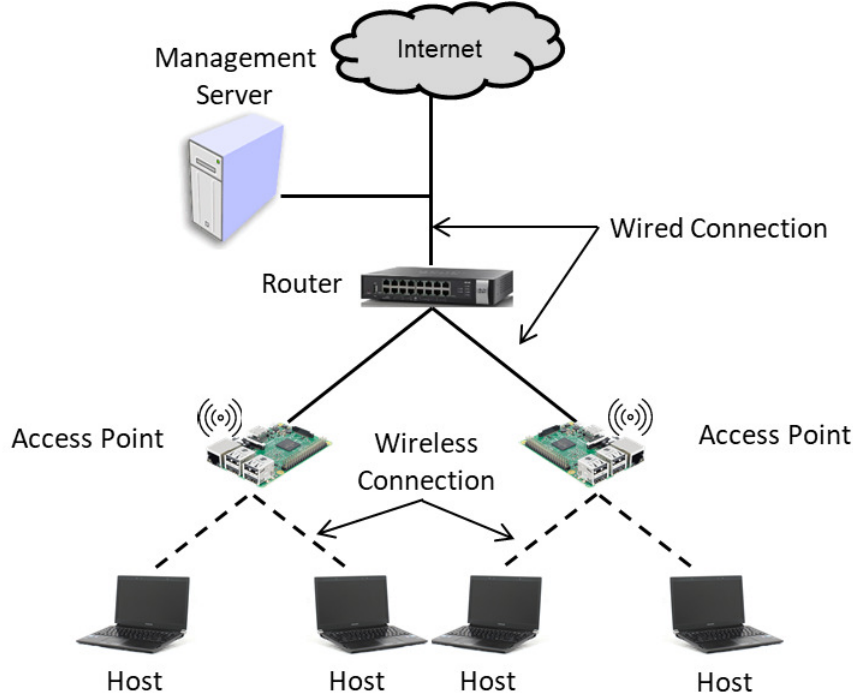


Figure 3.1: Design of Elastic WLAN system.

3. Finally, it will implement the algorithm output in the network by activating/deactivating the APs, changing the host associations, and allocating the channels.

3.2 Throughput Estimation Model for IEEE802.11n Protocol

The link speed or throughput between any AP and host depends on a variety of factors such as the modulation and coding scheme, the transmission power, the transmission distance, and obstacles. The theoretical computation of the accurate link speed is challenging [12, 89]. In this section, we review the throughput estimation model for IEEE802.11n link in WLAN.

3.2.1 Overview of Model

This model estimates the link speed or data throughput of an IEEE802.11 link in WLAN. It has two main steps to estimate the throughput between a source and a destination node. In the first step, it estimates the *receiving signal strength (RSS)* at the host by using the *log distance path loss model*. In the second step, it converts the estimated *RSS* into the corresponding throughput by the *sigmoid function* [90]. Both functions have several configuration parameters that can affect the estimation accuracy, which depends on link specifications and network field environments.

3.2.2 Receiving Signal Strength Estimation by Log-Distance Path Model

The signal strength is estimated by the log-distance path loss model and it considers the multi-path effects by considering both the direct signal along the line-of-sight (*LoS*) and the indirect signal through the non line-of-sight (*NLoS*) path in indoor environments.

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

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

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

Then, the RSS, P_d (dBm), at the host is estimated using the *log distance path loss model* by considering the distance and the obstacles loss between end nodes[89]:

$$P_d = P_1 - 10\alpha \log_{10} d - \sum_k n_k W_k \quad (3.2)$$

where P_1 represents the signal strength at 1m from the AP (source), α is the path loss exponent, d (m) does the link distance from the AP, n_k does the number of the type- k walls along the path between the AP and the host, and W_k does the signal attenuation factor (dBm) for the type- k wall in the environment. The estimation accuracy of RSS relies on the parameter values, which depend on the propagation environment [89].

3.2.3 Throughput Estimation by Sigmoid Function

From the RSS , the throughput S (Mbps) of the link is derived using following the *sigmoid function*:

$$S = \frac{a}{1 + e^{-\left(\frac{(P_d + 120) - b}{c}\right)}} \quad (3.3)$$

where a, b , and c are the constant parameters of the sigmoid function that should be optimized by parameter optimization tool [91]. The assumption of this sigmoid function for the throughput estimation is based on our real-world measurement results which clearly reflects the relationship between the RSS s and the estimated throughput.

3.2.3.1 Multi-Path Effect Consideration

Due to the *multipath effect* in indoor environments, the receiver may receive the *indirect signal* through an *NLoS* path that arrives after reflected/diffracted at some points on walls or obstacles, in addition of the *direct signal* along the *LOS* path. When the direct signal passes through multiple walls, the RSS at the receiver becomes much weaker than that of the indirect signal if it passes through much fewer walls.

The throughput estimation model considers this multi-path effect by selecting a *diffraction point* for each AP/host pair such that it is located on the wall in the same room as the host and RSS of the AP is largest. The indirect signal reaches the host through this diffracting point after the signal is attenuated there. The RSS through indirect path, P_{ind} is calculated by:

$$P_{dif} = P_1 - 10\alpha \log_{10} r - \sum_k n_k W_k \quad (3.4)$$

$$P_{ind} = P_{dif} - 10\alpha \log_{10} t - W_{dif} \quad (3.5)$$

where P_{dif} represents the RSS at the diffraction point, r (m) is the distance between the AP and the diffraction point, t (m) does the distance between the diffraction point and the host, and W_{dif} (dBm) does the attenuation factor at the diffraction point. For the estimated RSS at the receiver, the larger one between the direct or indirect signals is selected.

3.3 Testbed Implementation Using Raspberry Pi

In this section, we describe the testbed implementation of the elastic WLAN system using Raspberry Pi and Linux PCs. *Raspberry Pi* is a card-size, single-board computer that has the built-in wireless network interface (NIC) supporting IEEE802.11n. Therefore, the use of Raspberry Pi in the elastic WLAN system is significant for its disseminations in developing countries.

3.3.1 Implementation Environment/Platform

As the initial implementation platform of our elastic wireless LAN system testbed, we choose Linux operating system for the server, the APs, and the hosts. Linux has been used as a platform to implement new algorithms, protocols, methods, and devices for advancements of wireless networks. Linux is an open-source operating system, and offers a lot of tools to use. Most of them are easily configurable and have flexibility to use and integrate with other tools [68]. On the other hand, while searching for the network configuration and management tools in Windows operating system, we found most of them are less flexible and less configurable, and not open source. Currently, we are using *Ubuntu* for our implementation platform as the popular distribution of Linux environment for general-purpose users. Implementations of the elastic WLAN system on various platforms will be in our future studies.

The devices and software in Table 3.1 are used for the testbed implementation of the system. The IEEE802.11n protocol is used for any communication link with the channel bonding.

Table 3.1: Device environment and software in testbed.

devices and software		
server PC	OS	Ubuntu LTS 14.04
	model	Toshiba Dynabook R731/B
	Processor	Intel Core i5-2520M @2.5Ghz
	RAM	4GB DDR3 1333MHz
client PC 1	OS	Ubuntu LTS 14.04
	Model	Toshiba Dynabook R731/B
	Processor	Intel Core i5-2520M @2.5Ghz
	RAM	4GB DDR3 1333MHz
client PC 2	OS	Ubuntu LTS 14.04
	Model	Toshiba Dynabook R734/K
	Processor	Intel Core i5-4300M @2.6Ghz
	RAM	4GB DDR3 1333MHz
access point	OS	Raspbian
	Model	Raspberry Pi 3
	Processor	Broadcom BCM2837 @1.2Ghz
	RAM	1GB LPDDR2 900MHz
	NIC Chipset	Broadcom BCM43438
software/tools	openssh	to access remote PC and AP
	hostapd	to prepare and configure AP
	tc-tool	to manipulate traffic control setting
	iw-tool	to collect information of active network interface
	iperf	to measure link speed

3.3.2 System Topology

Figure 3.1 shows the simple network topology of the elastic WLAN system. *Raspberry Pi* is used for the AP and a *Linux laptop PC* is for the server and the host. The server can manage and control all the APs and the hosts by using the administrative access to them. The APs are connected to the server through wired connections. The hosts and the APs are connected through wireless connections.

3.3.3 AP Configuration of Raspberry Pi

This section explains how to configure *Raspberry Pi* for AP using *hostapd* daemon [16, 17].

- 1) To install the *hostapd* software onto *Raspberry Pi* that gives the AP functionality by using the command:
`sudo apt-get install hostapd`
- 2) To configure *hostapd* using the following steps:
 - i. To create a configuration file, *hostapd.conf* inside the folder */etc/hostapd/* and set the necessary configuration options in this file to produce the wireless network.
 - ii. For starting *hostapd* during the system booting, to set the absolute path of the configuration file by editing inside the folder *etc/default/hostapd/* as:
`DAEMON_CONF="/etc/hostapd/hostapd.conf",`
run
`sudo nano etc/init.d/hostapd/`, and change
`DAEMON_CONF=/etc/hostapd/hostapd.conf`
- 3) To set up the wireless connection of *Raspberry Pi* as static by editing the *"/etc/network/interfaces"* file and add the static IP address information.
- 4) To install the DHCP server that allows the wireless connections to automatically get dynamic IP addresses by using the command:
`sudo apt-get install isc-dhcp-server`
- 5) To configure the DHCP server using the following steps:
 - i. To edit *dhcpd.conf* file inside the folder */etc/dhcp/* and put necessary configuration options
 - ii. To make the wireless adapter as default for the DHCP request by editing the */etc/default/isc-dhcp-server* file as:
`INTERFACE="wlan0"`
- 6) To configure Network Address Translator (NAT) by editing the */etc/sysctl.conf* file to enable the IP forwarding, so as to allow multiple clients to connect to the AP and have all the data 'tunneled' through the single Ethernet IP.

3.4 Summary

In this chapter, firstly, we reviewed the study of the elastic WLAN system. Then, we reviewed the works of the throughput estimation model in WLAN. Finally, we reviewed the implementation details of the elastic WLAN system. In the next chapter, we will present the TCP unfairness problem of multiple-host concurrent communications.

Chapter 4

Experimental Observations of TCP Throughput Unfairness Problem of Multiple-Host Concurrent Communications

In this chapter, we present experimental observations of the TCP unfairness problem of multiple-host concurrent communications. First, we describe the experiment setup to observe the unfairness problem. Then, we show the throughput and RSS measurement results.

4.1 Introduction

In our experiments, we observed the TCP fairness among concurrently communicating multiple links with a single AP, using the elastic WLAN system testbed. The testbed measurements were carried out in the outdoor environment of *Asahi Riverbed* illustrate in Figure 4.1. By using *Home-dale* [43], we confirmed that this environment has been cleared of interfering signals.



Figure 4.1: Asahi Riverbed.

4.2 Experiment Setup

In our experiments, the TCP traffics were generated from two hosts to the server at the same time to measure the throughput under concurrent communications. Figure 4.2 shows the TCP link configuration between the server and the host through the AP. The server and the AP are connected with the 100Mbps wired Ethernet, and the AP and the host are connected to the *IEEE802.11n* wireless link. Using *iperf*, the host generates TCP traffics to the server.

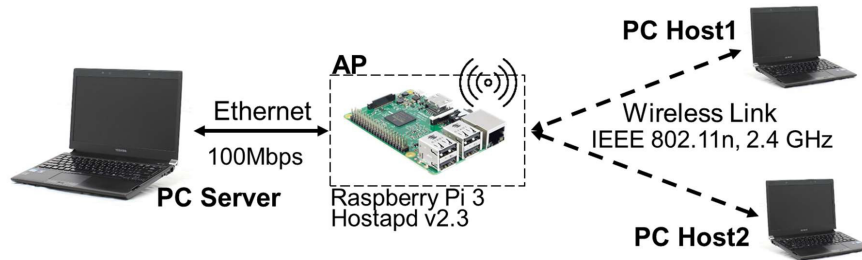


Figure 4.2: TCP link configuration.

Table 4.1 shows device and software specifications used in our experiments. One *Raspberry Pi* is employed for the AP operating at 2.4Ghz on channel 13, and three laptop PCs are for the two client hosts and the server.

Table 4.1: Device specifications.

software AP	
model	Raspberry Pi 3
CPU	Broadcom BCM2837 @1.2Ghz
memory	1GB LPDDR2 900MHz
NIC chipset	Broadcom BCM43438
OS	Raspbian
software	hostapd
link	IEEE 802.11n, 2.4Ghz (non-bonded)
server and hosts	
model	Toshiba Dynabook R731/B
CPU	Intel Core i5-2520M @2.5Ghz
memory	4GB DDR3 1333MHz
OS	Linux Ubuntu 14
software	Iperf 2.0.5



Figure 4.3: Outdoor measurement field.

Figure 4.3 illustrates the locations of the AP and two hosts in the measurements. The position of the host $H2$ is changed from 0m to 30m in the 5m interval while the position of $H1$ is fixed at 0m distance from the AP. Then, the throughput and RSS is measured at both $H1$ and $H2$. For each measurement, the traffic is generated from both hosts to the AP at the same time using *iperf* software [80].

4.3 Experiment Results

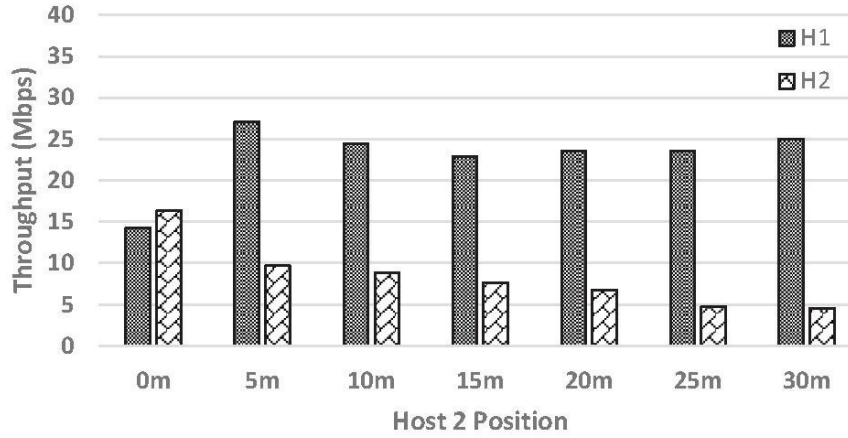


Figure 4.4: Throughput at different locations for $H2$.

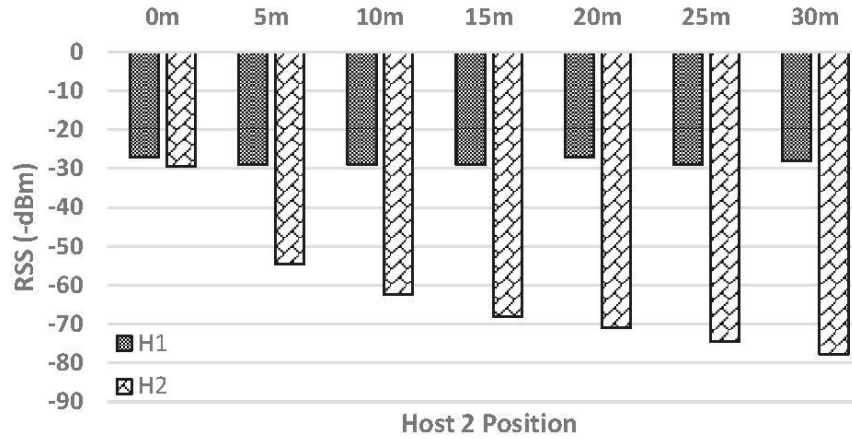


Figure 4.5: RSS at different locations for $H2$.

Figure 4.4 indicates throughput measurement results, which are similar at both hosts when the $H2$ distance from the AP is 0m. However, as the $H2$ distance increases, the throughput difference will increase accordingly. The throughput of $H2$ decrease due to the smaller RSS (receiving signal strength). Nevertheless, the throughput of $H1$ will increase, although RSS at $H1$ is constant as shown in Figure 4.5. This is considered as an inspiring result, that is to say, it seems that the nearer host to the AP will take higher transmission opportunities than the further host, which leads to the unfairness of the TCP throughput performance among them.

4.4 Summary

In this chapter, we presented experimental observations of the TCP unfairness problem of multiple-host concurrent communications. First, we described the experiment setup in experiments. Then, we showed the throughput and RSS measurement results. In the next chapter, we will present proposal of the *TCP throughput fairness control method* for two-host concurrent communications.

Chapter 5

Proposal of TCP Throughput Fairness Control Method for Two-Host Concurrent Communications

In this chapter, we present the *TCP throughput fairness control method* for two concurrently communicating hosts in the elastic WLAN system. For evaluations, we conduct experiments using the testbed in different network fields to verify the effectiveness.

5.1 Introduction

In Chapter 4, we observed the TCP fairness problem among concurrently communicating multiple links with a single AP. The measurement results using the elastic WLAN system testbed have revealed that the TCP throughput is not fair among the two concurrently communicating hosts associated with the same AP when they are located at different positions from the AP. This unfairness may be caused by the differences in the *TCP window size* and the *modulation and coding scheme (MCS)* among them. As a result, the packet transmission interval of the farther host hereafter becomes longer, and the interval for the nearer host becomes shorter.

In this chapter, we propose the *TCP throughput fairness control method* for two concurrently communicating hosts in the elastic WLAN system. The *transmission delay* is introduced at the AP in the packet transmission to the nearer host. Then, it is expected that the near host will decrease the throughput, and the far host will enhance it by obtaining more transmission opportunities. This delay is firstly calculated by the *received signal strength (RSS)* from the hosts. After that, it is dynamically controlled using the *PI controller* [92] to achieve the desired throughput fairness.

5.2 Idea

Figure 5.1(a) demonstrates the packet transmission intervals between the near and far hosts. The throughput unfairness appears by the differences in the MCS and the TCP window size. For the near host, the TCP windows size will be larger because of faster MCS compare to the far host. Then, it will occupy the larger bandwidth during the communications compared to the near host.

To overcome the issue, the *transmission delay* is intentionally introduced at the packet transmission from the AP to the near host. By reducing the bandwidth for the link with the near host,

it will give the more transmission bandwidth to the link to the far host. To achieve the throughput fairness, this delay should be controlled such that the intervals of packet transmission become equal, as shown in Figure 5.1(b).

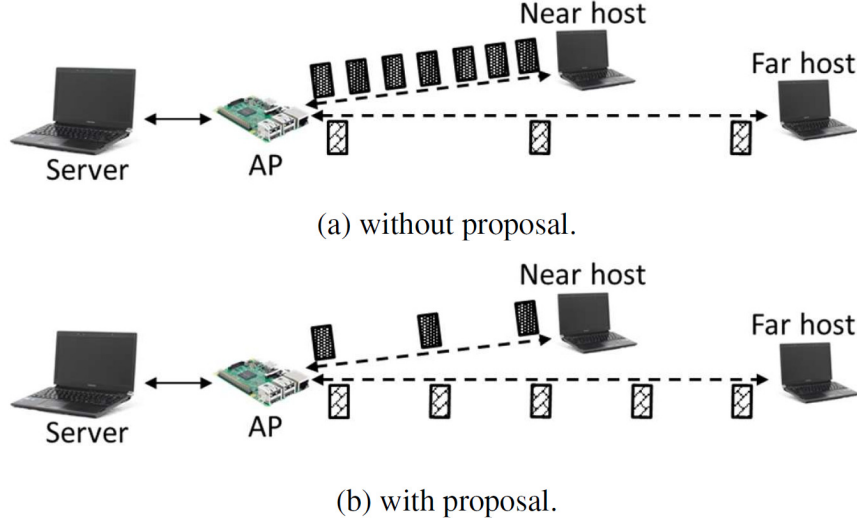


Figure 5.1: Background of TCP throughput fairness control method.

The transmission delay is initially calculated from the collected RSS at the AP from the near and far hosts, because the measured RSS influences the MCS. After that, the transmission delay is controlled dynamically by the PI control during communications to attain the fairness accurately in the network.

5.3 Initial Delay Calculation

The initial transmission delay $D(0)$ is computed from the difference of the measured RSS between the two hosts:

$$D(0) = \frac{RSS_{near}}{-a} \left(\frac{RSS_{near}}{RSS_{min}} \right)^2 e^{b(RSS_{near} - RSS_{far})} \quad (5.1)$$

The parameter values are obtained from our experimental results in [42]. $RSS_{min} = -88dBm$ from [93] is considered as the minimum RSS that a host can receive packets successfully from the *Raspberry Pi* AP. RSS_{near} represents the measured RSS at the AP from the near host, and RSS_{far} does the measured RSS from the far host. The constant parameter a and b are needed to be set depending on the environment. Here, $a = 8$, $b = 0.15$ for the outdoor field, and $a = 9$, $b = 0.17$ for the indoor field are used, respectively. If the measured RSS is smaller than RSS_{min} , the host cannot receive any packet successfully.

5.4 Dynamic Delay Optimization by PI Controller

In this stage, to achieve the throughput fairness, the delay is dynamically controlled by the *PI* (*Proportional-Integral*) controller [92]. The delay is optimized dynamically as follows:

$$D(n) = K_P \times (F_{tar} - F(n)) + K_I \times \sum_{i=0}^n (F_{tar} - F(i)) \quad (5.2)$$

Here, $D(n)$ represents the delay, $F(n)$ does the measured *fairness index* at the n -th time-step, and K_P and K_I does the gain for the P control and the gain for the I control, respectively. Where, $F_{tar} = 0.97$, $t = 20sec$, $K_P = 3$, and $K_I = 0.8$ are used. It is noted that the PI controller periodically changes the delay with the $20sec$ time-step. In the testbed implementation, the equation in (5.2) is modified as follows:

$$D(n) = D(n-1) - (K_P \times (F_{tar} - F(n))) + (K_I \times t \times (F_{tar} - F(n))) \quad (5.3)$$

The measured fairness index $F(n)$ is calculated by:

$$F(n) = \frac{(x_{far}(n) + x_{near}(n))^2}{2 \times (x_{far}(n)^2 + x_{near}(n)^2)} \quad (5.4)$$

where $x_{far}(n)$ and $x_{near}(n)$ represent the measured throughput of the far host and that of the near host at the n -th time-step, respectively.

5.4.1 Testbed Implementation

Here, we show the implementation of our proposal in the elastic WLAN system testbed.

5.4.1.1 RSS Measurement

The measured RSS from every host at the AP is collected by using the command as follows:

```
$ sudo iw dev wlan0 station dump | egrep "Station|signal:"
```

It is noted that the AP does not require to access to the host. Since the RSS is continually fluctuating, RSS is measured for $30sec$, and their average is used in Eq. (5.1).

5.4.1.2 Transmission Delay Application

Next, the transmission delay is applied to the packet transmission from the AP to the nearest host by the following procedure:

- Remove any delay previously assigned at the interface *wlan0*:

```
$ sudo tc qdisc del root dev wlan0
```
- Assign each delay at the packet transmission from the AP to every host, namely *d1* and *d2*:

```
$ sudo tc qdisc del root dev wlan0
$ sudo tc qdisc add dev wlan0 root handle 1: prio bands 3
$ sudo tc qdisc replace dev wlan0 parent 1:1 netem delay $delay1$
$ sudo tc qdisc replace dev wlan0 parent 1:2 netem delay $delay2$
```
- Enforce every delay to the packet transmission to the nearest host and the far host that is specified by the IP address:

```
$ sudo tc filter add dev wlan0 protocol ip parent 1: u32 match ip
    dst $IP address of near host$ flowid 1:1
$ sudo tc filter add dev wlan0 protocol ip parent 1: u32 match ip
    dst $IP address of far host$ flowid 1:2
```

Here, the IP address of each host is automatically set by the following program:

Linux commands for near host detection

```
#!/bin/bash
# for check total connected hosts
01: sudo hostapd_cli all_sta |
grep "dot11RSNAStatsSTAAddress="
while [ "$i" -le "$totalConnectedHost" ]
do
# for collecting the IP address
02: arp -a | grep $mac
# merge MACAddress, IP Address and RSS
03: echo $mac " " $ip " " $rss >> data.tmp
done
# sort descending data.tmp by rss value
04: sudo sort -nrk 3 data.tmp> sort.tmp
# 1st line in sort.tmp is the nearest host
```

5.4.1.3 Dynamic Delay Optimization

Firstly, the initial delay is obtained for the near host from the server by the following step:

- Explore the IP and MAC addresses of the connected hosts.
- Request the AP to measure the RSS of the hosts for 30sec and send back to the server.
- Find the average of the measured RSS.
- Compute $D(0)$ by Eq. (5.1).
- Assign $D(0)$ as the transmission delay to the nearest host.

Then, the delay is optimized dynamically by the following steps at every 20sec:

- Measure the throughput of the hosts for 20sec using *iperf*.
- Compute $F(n)$ by Eq. (5.4).
- Update $D(n)$ by Eq. (5.3).
- Set $D(n)$ as the transmission delay to the faster host at the AP.

5.5 Evaluations with Static Hosts

In this section, we evaluate the proposal through experiments when all the hosts are stationary.

5.5.1 Experiment Setup

In our experiments, we generate TCP traffics from two concurrently communicating hosts to the server to measure the throughputs. Figure 5.2 illustrates the network setup of the server, the AP, and the hosts. The server is connected with a 100Mbps wired connection to the AP, and the host is connected to the AP through the IEEE802.11n 2.4Ghz wireless link. The traffic is generated from the host to the server using *iperf*. The buffer size and the TCP window size are set to 8KB and 477KB, respectively.

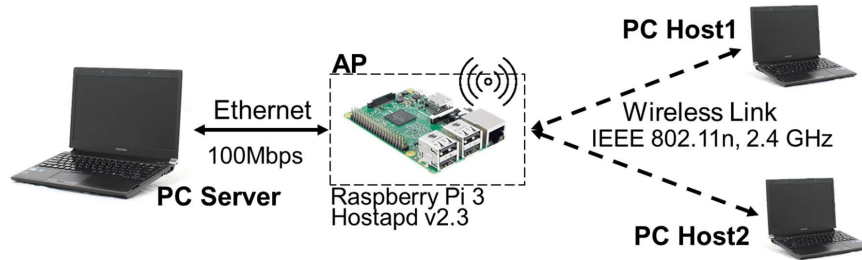


Figure 5.2: TCP link configuration.

Table 5.1 reveals software and hardware specifications for our experiments. Three laptop PCs are used for the two client hosts and the one server, and one *Raspberry Pi* is configured for the AP operating on channel 13.

Table 5.1: Hardware and software specifications.

server and hosts	
type	R731/B, Toshiba Dynabook
operating system	Linux (Ubuntu 14)
RAM	4GB DDR3-1333MHz
processor	Intel Core i5-2520M, 2.5Ghz
software	Iperf 2.0.5
access point	
type	Raspberry Pi 3
operating system	Linux (Raspbian)
RAM	LPDDR2 900MHz 1GB
processor	BCM2837 1.2Ghz, Broadcom ,
NIC	BCM43438, Broadcom
software	hostapd
link	IEEE 802.11n, 2.4Ghz (non-bonded)

5.5.2 Experiments in Outdoor Field

First, the experiments were conducted in the outdoor field.



Figure 5.3: Outdoor field.

5.5.2.1 Experiment Field

As the outdoor field, experiments were conducted in *Asahi riverbed* shown in Figure 5.3. There is no interfering signal, which is confirmed by using *Homedale*. The height of the AP and the hosts are 135cm and 70cm, respectively. During experiments, the distance of $H1$ from the AP is set 0m, 5m, 10m, and 15m. Then, for each $H1$ distance, $H2$ is located at 0m, 5m, 10m, 15m, 20m, 25m, and 30m from the AP. Finally, the throughput experiments are carried out in three methods: 1) *without proposal*, 2) *initial delay only*, and 3) *proposal*.

5.5.2.2 Throughput Results

Figure 5.4 illustrates the throughput results of the two hosts for 1) *without proposal*, where the unfairness appears between $H1$ and $H2$. Figure 5.5 shows the results for 2) *initial delay only*, where the fairness of each host is improved. Figure 5.6 shows the results for 3) *proposal*, where the fairness is achieved at any position by implementing our proposal. Figure 5.7 shows the overall throughput comparison at each $H1$ location. Nevertheless, with the proposal, the overall throughput has been reduced by 2.78% on average. The throughput drop is depending on the network environment.

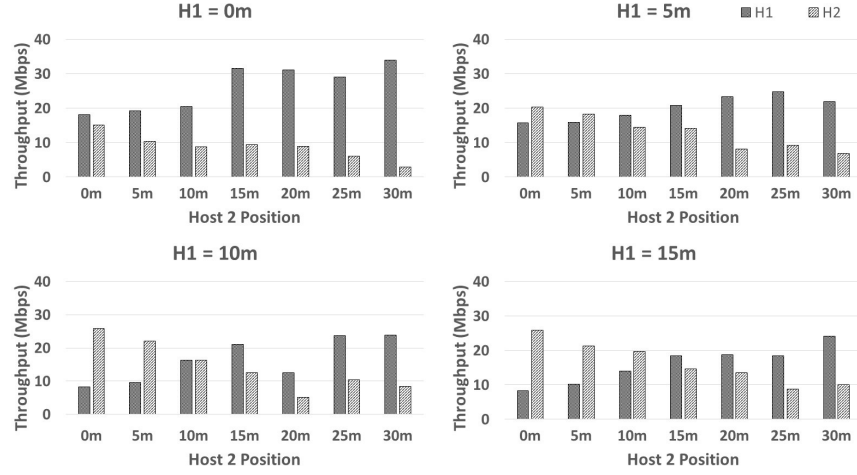


Figure 5.4: Results of throughput in outdoor field for 1) *without proposal*.

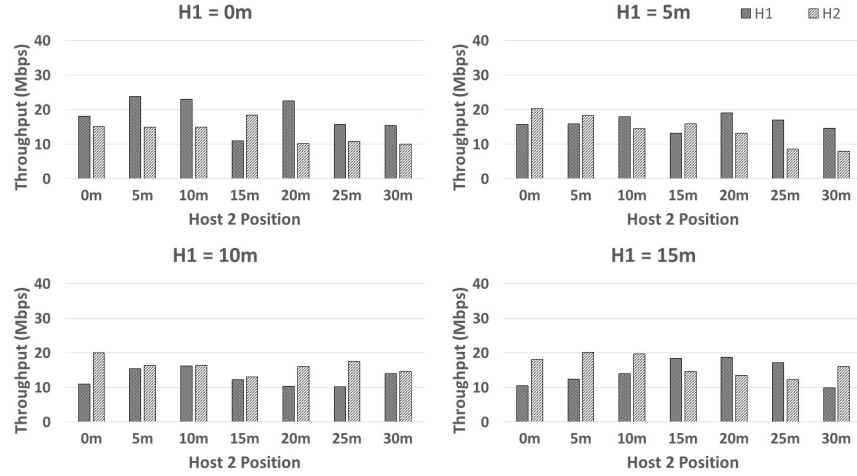


Figure 5.5: Results of throughput in outdoor field for 2) *initial delay only*.

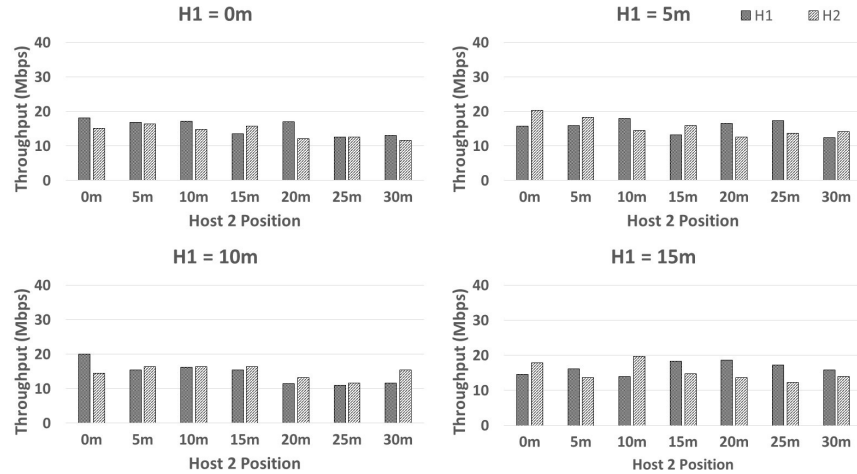


Figure 5.6: Results of throughput in outdoor field for 3) *proposal*.

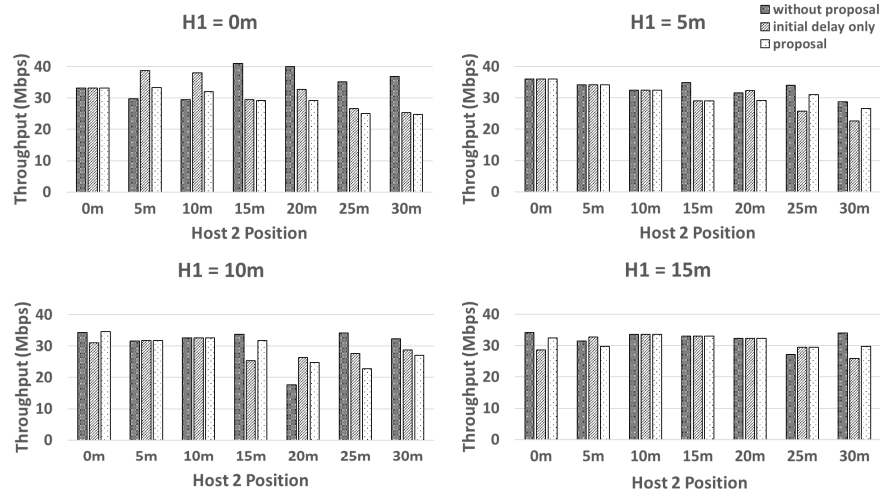


Figure 5.7: Results of overall throughput in outdoor field.

5.5.2.3 Fairness Index Results

Figure 5.8 demonstrates the *Jain's fairness index* result. From the Eq. (2.1), 1 represents the best fairness. For 1) *without proposal*, any result appears to be lower than 1. Nevertheless, for 3) *proposal*, the fairness index will generally become close to 1. It is found that the proposal achieved the throughput fairness.

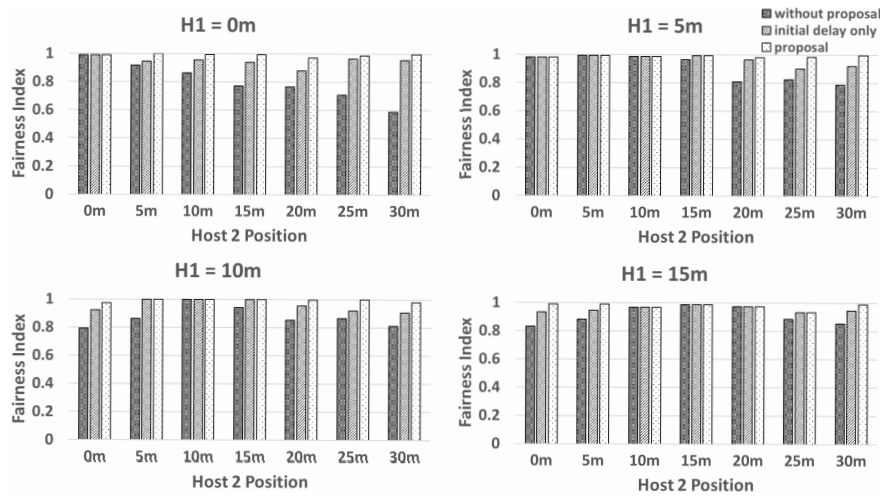


Figure 5.8: Observed TCP fairness index in outdoor field.

5.5.3 Experiments in Indoor Field

Second, the experiments were conducted in the indoor field.

5.5.3.1 Experiments in Two-room Case

The two rooms in the third floor of Engineering Building #2 at Okayama University are used in the experiments. Figure 5.9 illustrates the layout of this field.

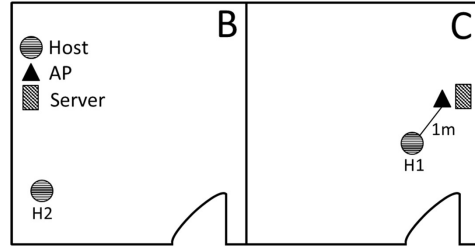


Figure 5.9: Experiment field for two-room case.

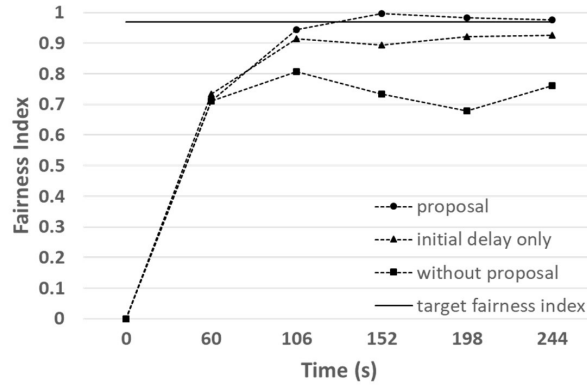


Figure 5.10: TCP fairness index results in indoor field for two-room case.

Figure 5.10 shows the fairness index results. The server collected the RSS and other necessary data during the first 60sec to measure the throughput. Then, the fairness index for each method became about 0.7. After 130sec later, the throughput fairness was achieved for 3) *proposal*. Conversely, 1) *without proposal* and 2) *initial delay only* could not achieve it.

5.5.3.2 Experiments in Four-room Case

The four rooms on the same floor were used in the experiments. Figure 5.11 illustrates the layout of this building that has a complex structure of multiple rooms and walls. This figure shows the position of the AP (triangle), the server (square), and the possible position of the hosts (circle). Here, the AP and the host are configured similarly to the outdoor field. *H1* is located at A1 and B2 in the indoor field. Then, for each *H1* position, *H2* is moved from A1 to D4, respectively.

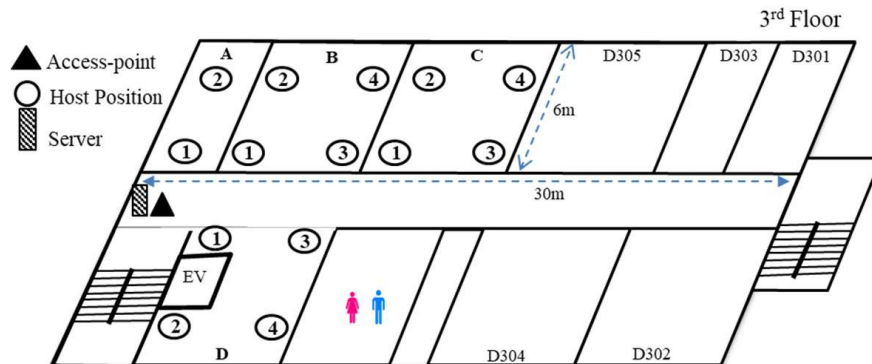


Figure 5.11: Indoor field.

Figure 5.12 illustrates the throughput results of the two hosts for 1) *without proposal*. The unfairness appears between $H1$ and $H2$. Figure 5.13 shows the results for 2) *initial delay only*, where the fairness was slightly improved. Figure 5.14 shows the results for 3) *proposal*, where the fairness was achieved by the proposal. Figure 5.15 shows the overall throughput comparison at each $H1$ location. However, by applying the proposal, the overall throughput was diminished by 14.83% on average.

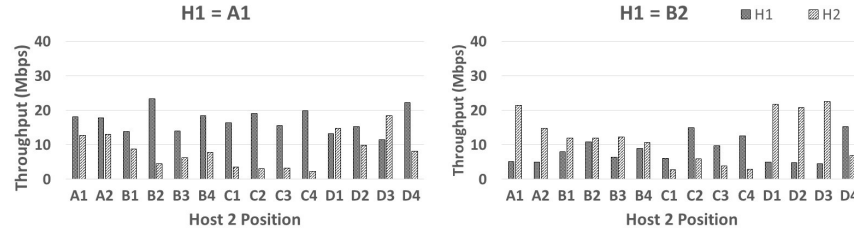


Figure 5.12: Results of throughput in indoor field for 1) *without proposal*.

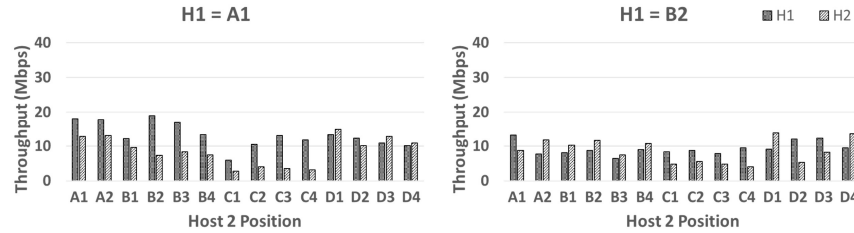


Figure 5.13: Results of throughput in indoor field for 2) *initial delay only*.

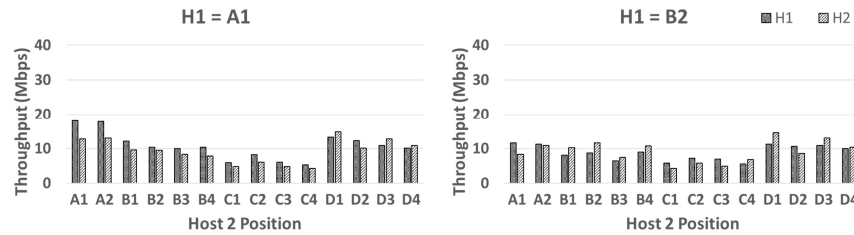


Figure 5.14: Results of throughput in indoor field for 3) *proposal*.

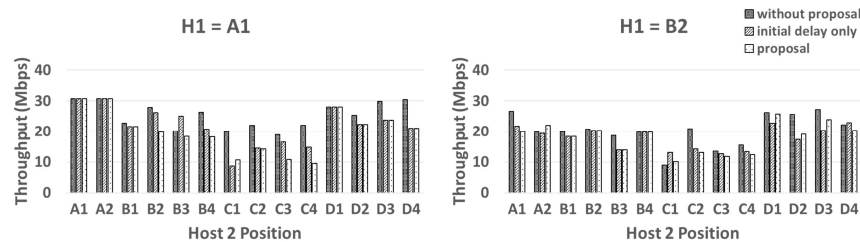


Figure 5.15: Results of overall throughput in indoor field.

Figure 5.16 shows the fairness index result of the four-room case. For 1) *without proposal*, the fairness index was smaller than 1. Nevertheless, for 2) *initial delay only*, it increased significantly. Furthermore, for 3) *proposal*, it became nearly 1. Thus, the effectiveness of the dynamic delay optimization using the PI controller is confirmed.

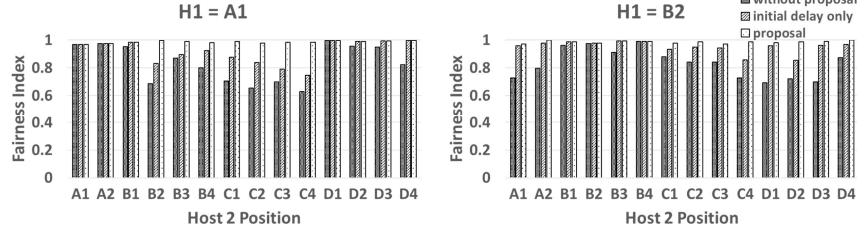


Figure 5.16: TCP fairness index results in indoor field for four-room case.

5.6 Evaluations with Dynamic Hosts

Next, we evaluated the *TCP throughput fairness control method* through experiments when one host is dynamically connected and disconnected to the AP.

5.6.1 Experiment Setup

The topology in Figure 5.9 and the devices/software in Table 5.1 were used in the experiments. Figure 5.17 indicates the 20min scenario of dynamically connecting and disconnecting one host. Initially, $H1$ is connected to the AP. At 0min, *iperf* TCP traffics were generated for 5min. At 5min, $H2$ was connected to the AP, and the proposed method started to run manually. Then, the server collected the RSS and measured the throughputs at $H1$ and $H2$. Afterward, at 10min, $H2$ was disconnected from the AP, and the server measured the throughput of $H1$. Lastly, at 15min, both $H1$ and $H2$ were connected.

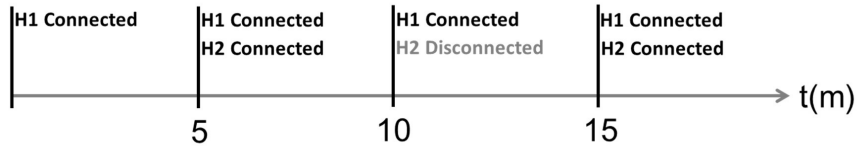
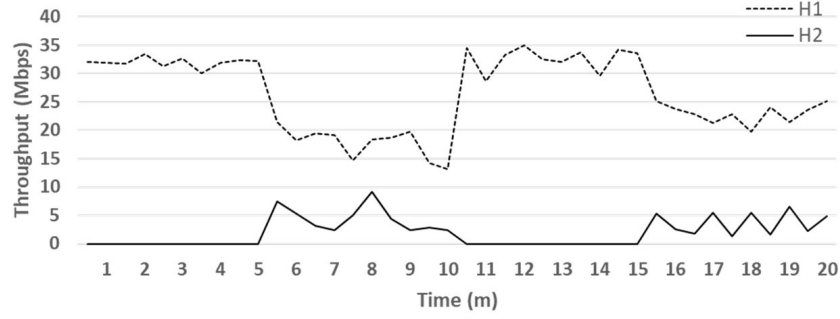
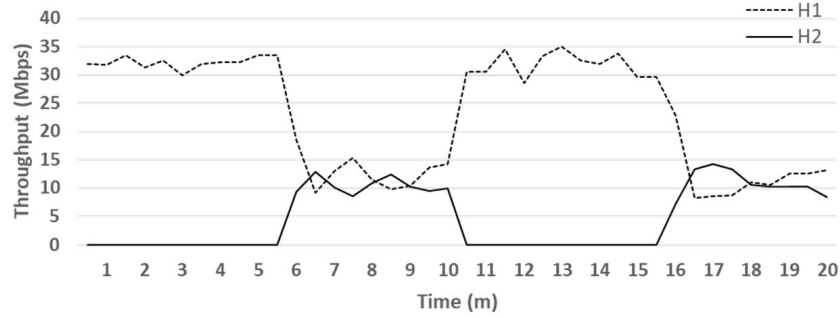


Figure 5.17: Time scenario for dynamically connecting/disconnecting host.



(a) Results of throughput without proposal.



(b) Results of throughput with proposal.

Figure 5.18: Individual throughput of $H1$ and $H2$ with dynamic hosts.

5.6.2 Results

Figure 5.18 shows the individual throughput result for $H1$ and $H2$. At $0min$, where only $H1$ was connected to the AP, the individual throughput was about $33Mbps$. Next, at $5min$, where both hosts were connected, the individual throughput of $H1$ decreased due to the concurrent communication of two hosts. Here, *with proposal* achieved the throughput fairness between two hosts, where it was around $12Mbps$ for each host. On the other hand, *without proposal* could not achieve it. Thus, our proposal has proved to reach the throughput fairness between two concurrently communicating hosts with a single AP, even if a host is continually connected and disconnected to the AP.

5.7 Summary

In this chapter, we presented the *TCP throughput fairness control method* for two concurrently communicating hosts in the elastic WLAN system. Then, we conducted experiments to verify the effectiveness in different network fields. In the next chapter, we will present proposal of the *TCP throughput fairness control method* method for multiple-host concurrent communications.

Chapter 6

Proposal of TCP Throughput Fairness Control Method for Multiple-Host Concurrent Communications

In this chapter, we present the generalized *TCP throughput fairness control method* for any number of concurrently communicating hosts with an AP. First, we describe the *target throughput* as the equal goal among the hosts. Then, we present the delay optimization using the PI controller. Finally, we evaluate the proposal through several testbed experiments.

6.1 Introduction

To overcome the throughput unfairness problem, in Chapter 5, we proposed the *TCP throughput fairness control method* for concurrently communicating two hosts in the *elastic WLAN system* using *Raspberry Pi* AP. The *transmission delay* is introduced at the packet transmission from the AP to the faster host, using the *Linux command*. This delay is dynamically controlled to achieve the acceptable throughput fairness by using the *PI feedback control*.

In this chapter, we propose the generalized *TCP throughput fairness control method* to deal with any number of concurrently communicating hosts with the same AP. The *target throughput* is newly introduced as the equal target among the hosts. The target throughput is dynamically updated using the measured throughputs of the hosts, since the throughput of a host is varied in each network field and topology.

6.2 Target Throughput

In the proposed method, the initial target throughput is derived from the estimated throughputs for the hosts using the model:

$$TH^{tar}(0) = \frac{1}{\sum_{i=1}^N \frac{1}{s_i}} (1 - \alpha N) \quad (6.1)$$

where s_i represents the estimated throughput for host i by the *throughput estimation model*, N does the number of hosts associated with the AP, and α does the constant interference-loss parameter. In this study, $\alpha = 0.06$ is used.

Then, during communications, the target throughput is updated by:

$$TH^{tar}(n) = \frac{\sum_{i=1}^N TH_i(n)}{N} \quad (6.2)$$

when either of the three conditions is satisfied:

- 1) $D_i(n) < D_{min}$,
- 2) $D_i(n) > D_{max}$, or
- 3) $|TH_i(n) - TH^{tar}(n)| < \epsilon TH^{tar}(n)$

by at least one host. $TH_i(n)$ represents the measured throughput for host i at time-step n , $D_i(n)$ does the delay given in (6.4), D_{min} and D_{max} represent the constant minimum and maximum delay, and ϵ does the constant saturation parameter, respectively. In this study, $D_{min} = 0$, $D_{max} = 200$, and $\epsilon = 0.2$ are used.

6.3 Delay Optimization by PI Controller

In the proposed method, the initial delay $D_i(0)$ for host i is calculated by:

$$D_i(0) = \frac{RSS_i}{-x} \left(\frac{RSS_i}{RSS_{min}} \right)^2 e^{y(RSS_i - RSS_{slow})} \quad (6.3)$$

where RSS_i represents measured RSS at the AP from host i , RSS_{slow} does RSS from the slowest host among the N hosts, RSS_{min} does the minimum RSS that a host can receive packets successfully from a *Raspberry Pi* AP, and x and y do the constant parameters. These parameter values of this equation are tuned from experimental results in [42]. In this thesis, $RSS_{min} = -88dBm$ [93], $x = 9$, and $y = 0.17$ are used.

Then, the delay is dynamically optimized using the *PI feedback control* [92] to let the measured throughput be equal to the target throughput:

$$D_i(n) = D_i(n-1) - (K_P \times ((TH^{tar}(n) - TH_i(n))) + (K_I \times t \times (TH^{tar}(n) - TH_i(n))) \quad (6.4)$$

where K_P and K_I represent the *P-control* and *I-control* gain, respectively. In this study, $K_P = 0.3$, $K_I = 0.07$, and $t = 20sec$ are adopted.

6.4 Testbed Implementation

Now, we present the implementation of this proposal on the elastic WLAN system testbed using *Raspberry Pi* APs.

6.4.1 RSS Measurement

The *Raspberry Pi* AP measures RSS from each host by executing the following command at the AP:

- `$sudo iw dev wlan0 station dump — egrep "Station—signal:"`.

Here, the AP does not need the root password of the host. Since the measured RSS tends to continually fluctuate, RSS is measured 30 times with the one second interval, and their average value is used in Eq. (6.1) and Eq. (6.3).

6.4.2 Transmission Delay Application

Then, the delay will be applied at the packet transmission from the AP to all the hosts except the farthest host by the following procedure:

1. Find all the associated hosts with the AP, and sort them in descending order of *RSS*:

2. Delete any previously allocated delay at the interface *wlan0*:

```
$ sudo tc qdisc del root dev wlan0
```

3. Set *delay_i* for each host *i* among the *N* hosts at the packet transmission from the AP:

```
$ sudo tc qdisc add dev wlan0 root handle 1: prio bands N  
$ sudo tc qdisc replace dev wlan0 parent 1:i netem delay delay_i
```

4. Execute the delay at the packet transmission to each host that is specified by the IP address:

```
$ sudo tc filter add dev wlan0 protocol ip parent 1: u32 match ip  
dst $IPaddressofhost_i$ flowid 1:i
```

6.4.3 Dynamic Delay Optimization

The transmission delay is optimized dynamically by the PI controller. First, the following procedure is performed on the server to assign the initial delay in the packet transmission to all the hosts at the AP:

1. Collect the MAC and IP addresses of the associated hosts with the AP.
2. Request the AP to collect *RSS* from each host with the 1sec interval for 30sec and to send them to the server.
3. Calculate the average of the received *RSS*.
4. Calculate the initial delay $D_i(0)$ for each host from the average *RSS* using (6.3).
5. Set $D_i(0)$ as the transmission delay to the corresponding host at the AP.
6. Calculate the target throughput $TH^{tar}(0)$ using Eq. (6.1).

Then, the following procedure updates the *target throughput* and dynamically optimizes the *delay*, which will operate at every 20sec:

1. Measure the throughput of each host for 20sec using *iperf*.
2. Update $D_i(n)$ for each host using Eq. (6.4).
3. Set $D_i(n)$ as the transmission delay to the faster host at the AP.
4. Update $TH^{tar}(n)$ using (6.2), if a condition is satisfied.

6.5 Evaluation with Static Hosts

In this section, we evaluate the generalized *TCP throughput fairness control method* through experiments when all the hosts are stationary.

6.5.1 Experiment Setup

In the experiments, TCP traffics are generated from the hosts to the server at the same time to measure the throughput under concurrent communications. Figure 6.1 shows the TCP link configuration between the server and the hosts through the AP. The server and the AP are connected with the 100Mbps wired Ethernet, and the AP and the host are connected through the IEEE802.11n wireless link. Using *iperf*, the host generates TCP traffics with the 477KB TCP window and the 8KB buffer to the server.

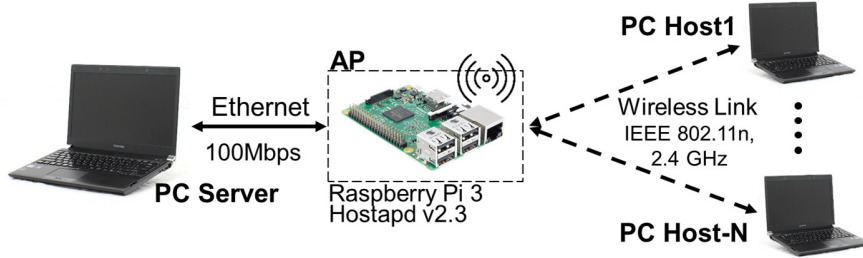


Figure 6.1: TCP link configuration.

Table 6.1 reveals device and software specifications used in the experiments. One *Raspberry Pi* is employed for the AP, which is operating at 2.4GHz on channel 13, and five laptop PCs are used for the four client hosts and the one server.

Table 6.1: Device specifications.

software AP	
model	Raspberry Pi 3
CPU	Broadcom BCM2837 @1.2Ghz
RAM	1GB LPDDR2 900MHz
NIC chipset	Broadcom BCM43438
Operating System	Linux Raspbian
software	hostapd
link	IEEE 802.11n, 2.4Ghz (non-bonded)
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
software	Iperf 2.0.5

6.5.2 Experiment Field

Three rooms in the third floor of Engineering Building #2 at Okayama University are used as the experimental field. Figure 6.2 illustrates the layout of this field for two-host case, three-host case, and four-host case, respectively, where the locations of the AP (triangle) and the hosts (circle) are shown. This field has a complex structure that consists of several rooms and walls.

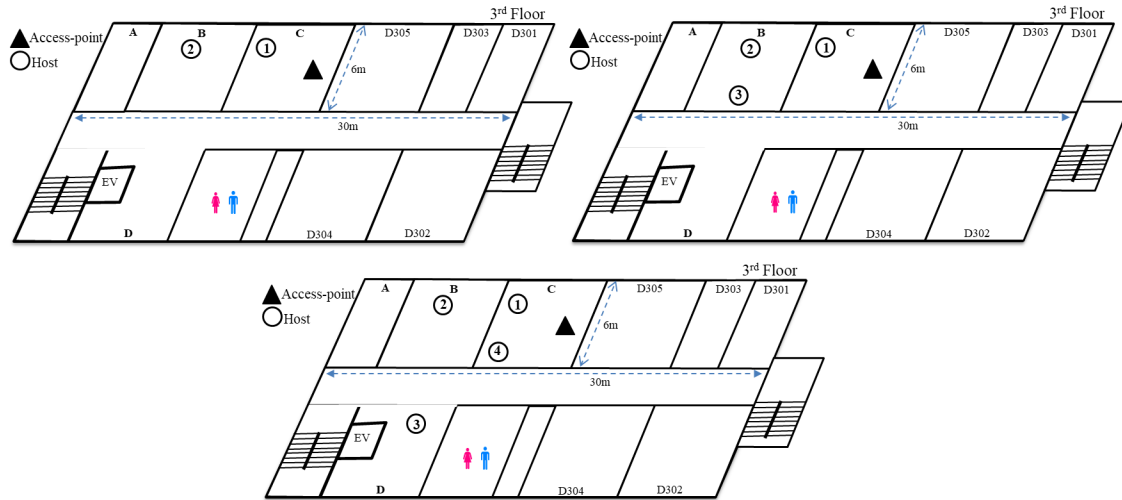


Figure 6.2: Experiment field for two-host, three-host, and four-host case.

6.5.3 Results for Two-host Case

Figure 6.3 shows the individual throughput results between $H1$ and $H2$. The proposal achieves the fairness in the third time-step without updating the target throughput. Figure 6.4 indicates the fairness index comparison. *proposal* has the similar performance with *fairnessindextarget*, where *proposal* achieves the fairness faster. On the other hand, *withoutproposal* could not achieve the fairness at all. The results have proved that the proposal works well in the two-host case as in the previous method.

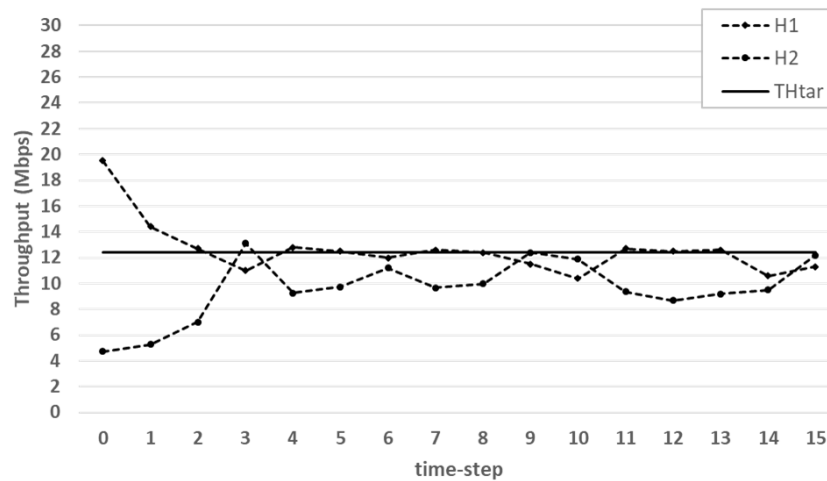


Figure 6.3: Two-host case individual throughput.

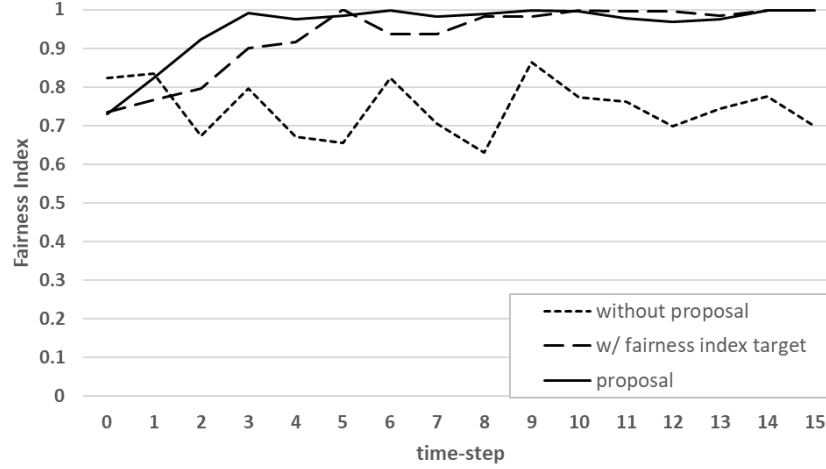


Figure 6.4: Two-host case fairness index comparison.

6.5.4 Results for Three-host Case

Figure 6.5 shows the individual throughput results for three hosts. After updating the target throughput three times, *proposal* can achieve the fairness at the 12-th time-step. Figure 6.6 compares the fairness indices between *proposal* and *withoutproposal*. *proposal* can obtain the throughput fairness, whereas *withoutproposal* fails to achieve it.

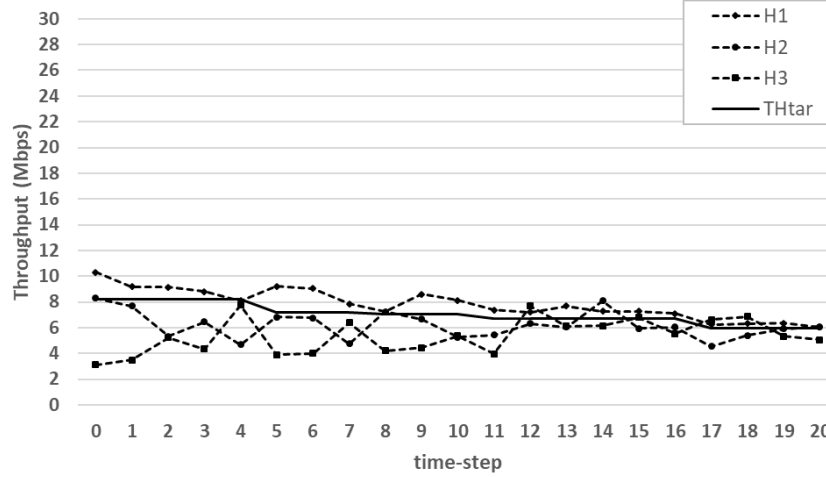


Figure 6.5: Three-host case individual throughput.

6.5.5 Results for Four-host Case

Figure 6.7 shows the individual throughput results for four hosts. After updating the target throughput three times, *proposal* can achieve the fairness at the 17th time-step. Figure 6.8 compares the fairness indices between *proposal* and *withoutproposal*. *proposal* can achieve the throughput fairness, whereas *withoutproposal* unable to fulfill it. Thus, the effectiveness of the generalized *TCP throughput fairness control method* for multiple hosts is confirmed.

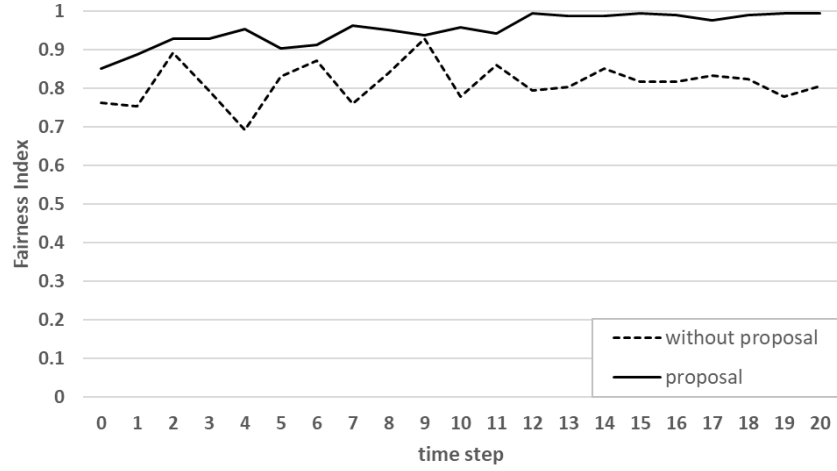


Figure 6.6: Three-host case fairness index comparison.

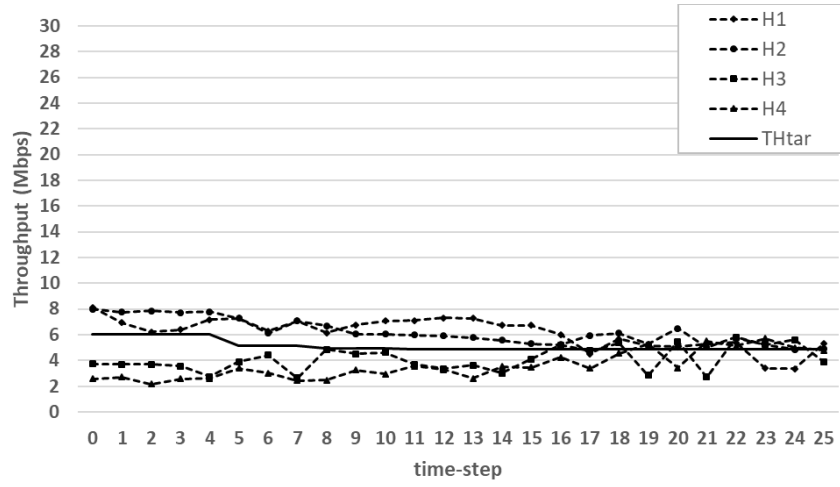


Figure 6.7: Four-host case individual throughput.

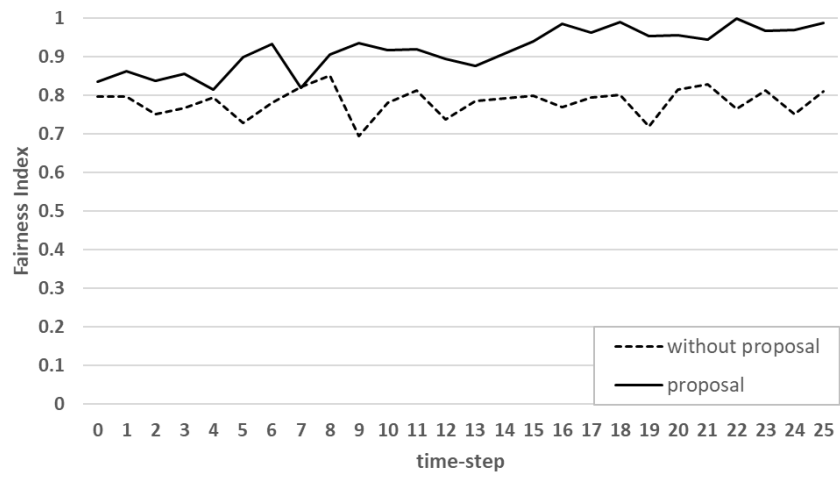


Figure 6.8: Four-host case fairness index comparison.

6.6 Evaluation with Dynamic Hosts

In this section, we evaluate the generalized *TCP throughput fairness control method* through experiments when hosts are dynamically connected and disconnected with the AP.

6.6.1 Experiment Setups

The topology in Figure 6.9 and the devices/software in Table 6.1 are used in experiments. In this topology, four hosts are prepared in three different rooms. Two scenarios are prepared to confirm the effectiveness of the proposal.

Figure 6.10 illustrates the first scenario with 140 time-steps. Hosts are connected and disconnected one by one. Initially, $H1$ is connected to the AP. At time-step 0, TCP traffics are generated during 20 time-steps. After that, $H2$ is connected to the AP, and the proposed method runs manually. Then, $H3$ is connected to the AP at the 40th time-step, and $H4$ is connected at the 60th time-step. Finally, the hosts are disconnected one by one from the 80th time-step until the 120th time-step.

Figure 6.11 illustrates the second scenario with 100 time-steps. Hosts are connected and disconnected with 20 time-step intervals by turns. First, $H1$ connects to the AP. At the initial time-step, TCP traffics are generated during 20 time-steps. Next, $H2$ and $H3$ are connected to the AP at the same time, and the proposed method is started. At the 40th time-step, only $H1$ is connected to the AP. Afterward, at the 60th time-step, all the hosts are connected to the AP. Last, $H2$, $H3$, and $H4$ are simultaneously disconnected from the AP at the 80th time-step.

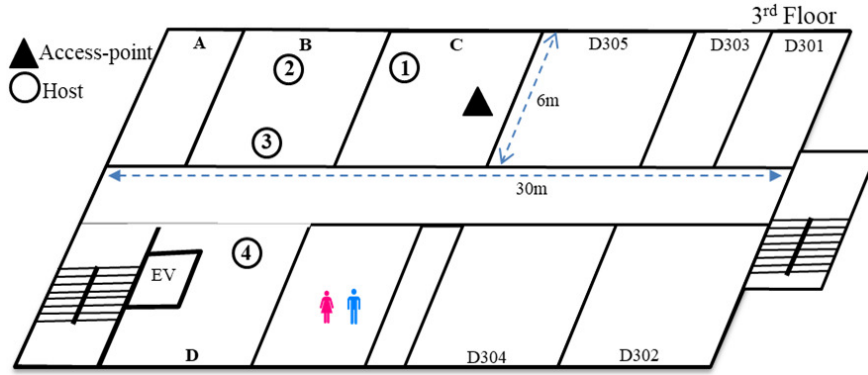


Figure 6.9: Experiment field for dynamically connecting/disconnecting host.

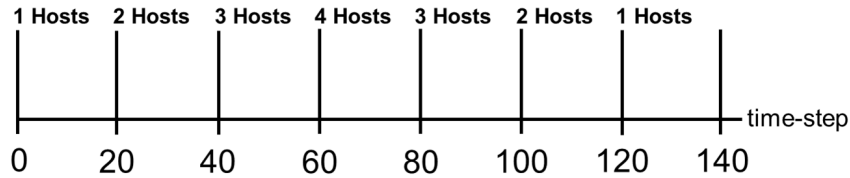


Figure 6.10: First scenario for dynamic host.

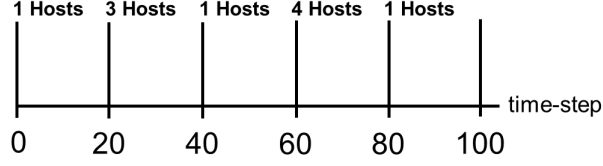


Figure 6.11: Second scenario for dynamic host.

6.6.2 Results for First Scenario

Figure 6.12 shows the individual throughput for each host in the first scenario. At the initial time-step, where only $H1$ is connected to the AP, the individual throughput is about $30Mbps$. Afterward, at the 20th time-step, $H2$ is connected, where $H1$ and $H2$ achieve the fairness after updating the target throughput once. The individual throughput of $H1$ decreases due to concurrent communications. At the 40th time-step, $H3$ is connected, and the throughput drops due to three concurrently communicating hosts. Hence, the TCP fairness achieved after updating the target throughput three times. At the 60th time-step, $H4$ is connected and the fairness is achieved after updating the target throughput three times. Next, at the 80th time-step, $H4$ is disconnected from the AP and the throughput slightly increases due to the decreasing number of the hosts in concurrent communications. At the 120th time-step, since only $H1$ is connected to the AP, the throughput increases to $30Mbps$.

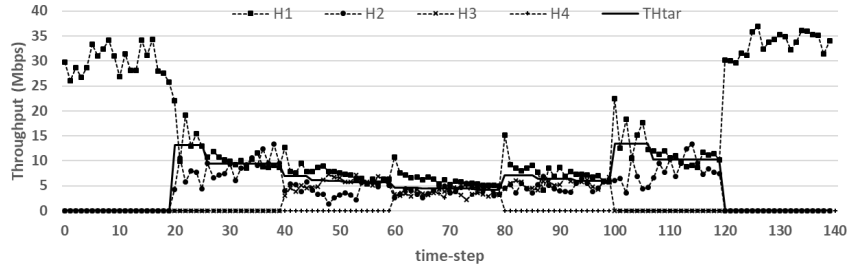


Figure 6.12: First scenario individual throughput.

6.6.3 Results for Second Scenario

Figure 6.13 shows the individual throughput for each host in the second scenario. Initially, only $H1$ is connected to the AP. Then, $H2$ and $H3$ are joined at the 20th time-step. After running the proposal, at the 33th time-step, each host has the similar throughput. After that, $H2$ and $H3$ are disconnected from the AP. At the 60th time-step, all the hosts are connected to the AP. Each host achieves the throughput fairness after updating the target throughput three times. Thus, the proposal has proved to achieve the throughput fairness up to four concurrently communicating hosts with a single AP, even if the host is dynamically connected and disconnected to the AP.

6.7 Summary

In this chapter, we proposed the generalized *TCP throughput fairness control method* for any number of concurrently communicating hosts with an AP. First, we introduced the *target throughput* as

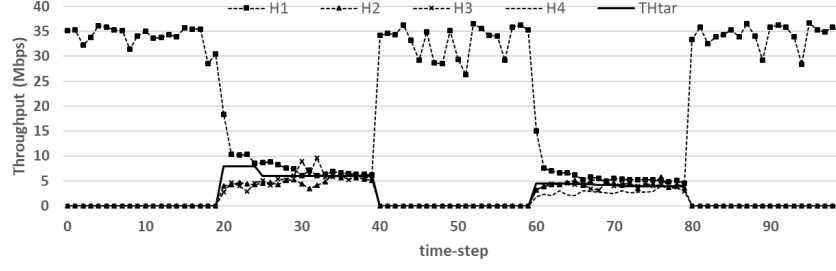


Figure 6.13: Second scenario individual throughput.

the equal goal among the hosts. Then, we presented the delay optimization using the PI controller. The experiment results confirmed the effectiveness of the proposal, where the fairness is achieved in any topology by implementing our proposal. In the next chapter, we will present the related works of this study.

Chapter 7

Related Works in Literature

In this chapter, we briefly discuss related works to this study. [19, 40] surveys several fairness issues in WLANs. [19] reveal the general view of fairness studies and existing fairness models. [40] evaluates the *elective packet marking with ACK filtering (SPM-AF)* and the *least attained service (LAS)* scheduling to minimize the unfairness caused by interactions between the TCP and the MAC protocol.

[20] observes that by adjusting the advertised *window size* in the header of the TCP ACK segment, the unfairness can occur in the network bandwidth between the uplink TCP flow and the downlink TCP flow through network measurements. They propose a mechanism of manipulating the receiver TCP window size for the upstream flow to limit the upstream traffics. Then, the downstream flow will obtain the fairness of the bandwidth.

[21] provides an analytical framework for the partitioning strategy and the rate allocation of both layered and replicated media systems over multicast IP networks in the context of *Layered Media Multicast Control (LMMC)* protocol. They assume the existence of congestion and flow control mechanisms that are capable of dynamically addressing the inter-session fairness issue, which represents the fair distribution of available bandwidths among multiple media and other sessions such as TCP sessions. Then, they formulate a two-phase optimization problem of the partitioning and the rate allocation after extrapolating the max-min fairness metric with a mathematically well-behaved function

[22] performs the fair resource allocation in the energy-constrained cooperative ad-hoc network, using a novel multiuser cooperative protocol, and the fair cooperative protocol called FAP. They present an analytical framework in which the relationship between the fairness indicator and the aggregate throughput in the energy-constrained cooperative network is characterized. It is demonstrated that an improvement in the fairness achieved by the proposed FAP can lead to substantial throughput gains over the direct transmission and the full cooperation cases.

[23] proposes a coverage algorithm for mobile sensors to achieve a coverage that will match the *over the long term and as quantified by an RMSE metric* for a given threat profile. For the well matching, they argue that the fairness measure of the performance over the shorter time scale is also important. Then show that the RMSE and the fairness are antagonistic in general, and argue the need of a combined measure of the performance.

[24] finds that the exclusive region for a specific link should be a system level concept, and should depend on system factors such as interference from/to other active links. Based on the findings, they proposed two MAC packet scheduling schemes to exploit the system capacity and, at the same time, to achieve a certain level of the fairness in UWB wireless networks.

[25] explains where the conventional fairness criterion in wired networks in terms of the flow

rate is not appropriate in wireless environments. First, they show that the conventional max-min fairness definition originally proposed for wired networks is not appropriate for wireless multi-hop networks. Second, they propose a new fairness criterion according to the contending characteristics in multi-hop wireless networks, and a new max-min fairness definition for the multi-hop wireless environment. Third, they demonstrate that their new definition is general for both wired and wireless networks.

[26] contributes to characterize the maximum throughput region attained by a distributed scheduling strategy under arbitrary topologies and interference models. The guaranteed fraction depends on the *interference degree* of the network, which is the maximum number of transmitterreceiver pairs that interfere with any given transmitterreceiver pair in the network and do not interfere with each other. Depending on the nature of communications, the transmission powers and the propagation models, the guaranteed fraction can be lower-bounded by the maximum link degrees in the underlying topology, or even by constants that are independent of the topology. They enhance the maximal scheduling that can guarantee the fairness of rate allocation among different sessions.

[27] claims that a unified framework to effectively attain different degrees of performance trade-off between throughput and fairness with QoS support and efficient resource utilization is not widely accepted. With a focus on interference-limited wireless networks, they derive a unified optimization framework to obtain the optimal relationship (i.e., trade-off curve) of the system throughput and fairness with the QoS support. By introducing a bargaining floor, the relationship curve is obtained by solving the optimization problem iteratively.

[28] develops a formal reference model that characterizes objectives of removing spatial bias (i.e., providing performance that is independent of the number of wireless hops to a wire) and maximizing spatial reuse. Then, they perform an extensive set of simulation experiments to quantify the impact of the key performance factors towards achieving these goals. Finally, they explicitly characterize the link between the fairness and the capacity with a particular focus on multi-hop wireless backhaul networks.

[29] works on *Wireless Asynchronous Transfer Mode (WATM)* medium access control. They introduce an allocation algorithm, termed as *Minmax*, which targets the fairness as defined as the growth of the queue during the bandwidth allocation for ATM. As the simulation results, *Minmax* improves cell losses by an order of magnitude, while decreasing delays in several times.

[30] considers a wireless network where all nodes need not be in the transmission range of each other. The fair allocation of the end-to-end bandwidth will require cross-layer optimizations involving the transport, network, and MAC layers. They propose a distributed method for aloha networks, which only requires two-hop information to achieve the proportional fairness.

[31] assumes the IEEE 802.11e WLAN that supports multi-rates, called multi-rate IEEE 802.11e WLAN. The fair transmission opportunity among flows is known as TXOP in the study of IEEE 802.11 MAC protocols. They propose a rate adaptive mechanism of the MAC protocol, where packet drop rates in different flows are presented and compared to show the fairness property of the proposed MAC protocol.

[32] claims that the fairness of nodes can be used as a feedback during the congestion control. They propose a rate-based *fairness-aware congestion control (FACC)* protocol, which controls congestions and achieves approximately fair bandwidth allocations for different flows. They categorize intermediate relaying sensor nodes into near-source nodes and near-sink nodes. In FACC, near-sink nodes do not need to maintain per-flow states and use a lightweight probabilistic dropping algorithm based on the queue occupancy and the hit frequency.

[33] proposes the *Event-Based Fairness (EBF)* scheme where network-wide resources are allocated to application level units (i.e., events) in video streaming applications of wireless sensor

networks. They show that the fair treatment of events, as opposed to the regular flow of frames, results in the enhanced performance in terms of the number of frames reported per event. It also improves latency.

[34,35] focuses on the fairness on the rate allocation and the congestion control. [34] examines the packet level fairness via the proportional fairness in MIMO wireless networks, and the fairness in the transmission delay, the transmission time, and the services. [35] applies the fair rate control to links, and proposes a modified proportional fairness to measure the rate of achieving the fairness.

[36] provides a general analytical framework that captures the unique character of shared wireless, and allows the modeling of a large class of system-wide fairness models via specifications of per-flow utility functions. Their work formally investigates fairness properties that can be achieved by the class of multiple access wireless MAC protocols in shared channel wireless networks in general, and ad-hoc networks in particular. They have shown results of the proportional fairness in rate controls in different protocols, such as IEEE802.11, MACAW, CB-Fair, and the PFCR protocol.

[37] presents a solution capable of enabling the TCP fairness with the minimal additional complexity. They pursue an aggregated uplink/downlink fairness by utilizing the IP rate limiter with the adaptive rate control mechanism. The feasibility and effectiveness of their mechanism are confirmed by a testbed in 802.11b WLAN. On the other hand, we implement a testbed for fairness in 802.11n as well.

[38,39] focuses on the IEEE802.11 *DCF (Distributed Coordination Function)* based WLANs. [38] proposes a simple yet highly effective scheme to ameliorate the unfairness between the uplink flow and the downlink flow by modifying the random back-off mechanism in the IEEE 802.11 MAC protocol. By considering TCP and UDP in various simulation experiments, their proposal has achieved the fairness between them. [39] observes the asymmetric behavior of the TCP congestion control and the interaction with a MAC contention control. Then, it introduces a cross-layer feedback approach in the link layer and the transport layer, and proves the effectiveness of the proposal by simulations. Since their proposal is working on the MAC layer, its implementation becomes far more challenging than our approach using Linux commands.

[41] claims that the throughput can be divided into the hosts depending on the RSS, the traffic direction, and the mechanism of RTS/CTS. The unfairness in WLAN will occur in compliance with the host positions. This phenomenon is one of the motivations of our proposal.

In this survey, most of the existing papers observe the unfairness problem in TCP flows, and try to solve the problem at the MAC layer. None of them have discussed the *transmission delay* at the AP for the TCP fairness in the proposal.

Chapter 8

Conclusion

In this thesis, we presented our studies of the *TCP throughput fairness control method* for multiple-host concurrent communications in *wireless local-area networks (WLAN)*.

Firstly, we 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/ac protocols, Linux tools for wireless networking, Jain's fairness index, and the software AP using a *Raspberry Pi* device.

Secondly, we reviewed our previous studies related to this thesis, including the *elastic WLAN system*, the *throughput estimation model*, and testbed implementation using *Raspberry Pi* APs.

Thirdly, we presented experimental observations of the *TCP throughput unfairness problem* among concurrently communicating multiple hosts. First, we described the experiment setup to observe the throughput unfairness problem. Then, we showed the throughput and RSS measurement results.

Fourthly, we presented the *TCP throughput fairness control method* for two concurrently communicating hosts in the elastic WLAN system. The *transmission delay* is introduced at the AP in the packet transmission to the nearer host. Then, it is expected that the near host will decrease the throughput, and the far host will enhance it by obtaining more transmission opportunities. For evaluations, we conducted experiments using the testbed in different network fields to verify the effectiveness of the proposal. The fairness is achieved in the experiments by applying our proposal.

Finally, we proposed a generalization of the *TCP throughput fairness control method* to deal with any number of concurrently communicating hosts. The *target throughput* is newly introduced as the equal target among the hosts, which is dynamically updated from the measured throughputs of the hosts. Then, we evaluated the proposal through testbed experiments in different network fields. The experiment results confirmed the effectiveness of the proposal, where the fairness is achieved in any topology by applying our proposal.

In future studies, we will further improve the TCP throughput fairness control method for simplifying the implementation and shortening the response time. We will also consider the application of the proposal in allocating the required bandwidth for the priority host. Then, we will continue conducting performance evaluations of our proposals in various network scenarios.

Bibliography

- [1] K. G. Coffman and A. M. Odlyzko, "Growth of the Internet," Tech. Rep. 99-08, DIMACS, 2001.
- [2] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol.35, no.9, pp. 116-126, Sept. 1997.
- [3] S. K. Dash, S. Mishra, and J. Mishra, "Mobile data offloading using opportunistic communication and AP deployment: a case study," *Int. J. Mob. Comput. Mult. Comm.*, vol. 8, no. 4, pp. 66-84, Dec. 2017.
- [4] LAN MAN Standards Committee, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," The Institute of Electrical and Electronics Engineering, June. 1997.
- [5] 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, Sept. 2015.
- [6] K. Mittal, E. M. Belding, and S. Suri, "A game-theoretic analysis of wireless access point selection by mobile users," *Comput. Commun.*, vol. 31, no. 10, pp. 2049-2062, Jan. 2008.
- [7] 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.
- [8] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wireless Networks*, vol. 11, no. 1-2, pp. 115-133, Jan. 2005.
- [9] F. Nadeem, E. Leitgeb, M. S. Awan, and S. Chessa, "Comparing the life time of terrestrial wireless sensor networks by employing hybrid FSO/RF and only RF access networks," in *Proc. Int. Conf. Wirel. Mob. Commun.*, pp. 134-139, 2009.
- [10] M. E. Islam, N. Funabiki, T. Nakanishi, and K. Watanabe, "An access-point aggregation approach for energy-saving wireless local area networks," *IEICE Trans. Commun.*, vol. E96-B, no. 12, pp. 2986-2997, Dec. 2013.
- [11] M. E. Islam, N. Funabiki, and T. Nakanishi, "Extensions of access-point aggregation algorithm for large-scale wireless local area networks," *Int. J. Netw. Comput.*, vol. 5, no. 1, pp. 200-222, Jan. 2015.
- [12] 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, July 2016.

- [13] 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. Network. Comput.*, vol. 7, no. 2, pp. 248-270, July 2017.
- [14] M. M. Islam, M. S. A. Mamun, N. Funabiki, and M. Kuribayashi, "Dynamic access-point configuration approach for elastic wireless local-area network system," in *Proc. Int. Symp. Comput. Netw.*, pp. 216-222, Nov. 2017.
- [15] M. M. Islam, N. Funabiki, M. Kuribayashi, S. K. Debnath, I. M. Kwenga, K. S. Lwin, R. W. Sudiby and M. S. A. Mamun, "Dynamic access-point configuration approach for elastic wireless local-area network system and its implementation using Raspberry Pi," *Int. J. Netw. Comput.*, vol. 8, no. 2, pp. 254-281, July 2018.
- [16] "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.
- [17] "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.
- [18] Raspberry Pi, "The official website of the Raspberry Pi project," <http://raspberrypi.org>. Accessed 1 Sep., 2019.
- [19] H. Shi, R. V. Prasad, E. Onur, and I. G. Niemegeers, "Fairness in wireless networks: issues, measures and challenges," *IEEE Comm. Survey. Tutor.*, vol. 16, no. 1, pp. 5-24, 2014.
- [20] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," in *Proc. IEEE Infocom*, April 2003.
- [21] H. Yousefizadeh, H. Jafarkhani, and A. Habibi, "Layered Media Multicast Control (LMMC): rate allocation and partitioning," *IEEE Trans. Networking*, vol. 13, no. 3, pp. 540-553, 2005.
- [22] L. Dai, W. Chen, L. J. Cimini and K. B. Letaief, "Fairness improves throughput in energy-constrained cooperative Ad-Hoc networks," *IEEE Trans. Wirel. Comm.*, vol. 8, no. 7, pp. 3679-3691, 2009.
- [23] C. Y. T. Ma, D. K. Y. Yau, J. Chin, N. S. V. Rao and M. Shankar, "Matching and fairness in threat-based mobile sensor coverage," *IEEE Trans. Mob. Comp.*, vol. 8, no. 12, pp. 1649-1662, 2009.
- [24] H. Jiang and W. Zhuang, "Effective packet scheduling with fairness adaptation in ultra-wideband wireless networks," *IEEE Trans. Wirel. Comm.*, vol. 6, no. 2, pp. 680-690, 2007.
- [25] P. Wang, H. Jiang, W. Zhuang, and H. V. Poor, "Redefinition of max-min fairness in multihop wireless networks," *IEEE Trans. Wirel. Comm.*, vol. 7, no. 12, pp. 4786-4791, 2008.
- [26] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," *IEEE Trans. Inform. Theory*, vol. 54, no. 2, pp. 572-593, 2008.

- [27] H. T. Cheng and W. Zhuang, "An optimization framework for balancing throughput and fairness in wireless networks with QoS support," *IEEE Trans. Wirel. Comm.*, vol. 7, no. 2, pp. 584-592, 2008.
- [28] V. Gambiroza, B. Sadeghi, and E. Knightly, "End-to-end performance and fairness in multi-hop wireless backhaul networks," in *Proc. 10th. Inter. Conf. Mob. Comp. Net.*, pp. 287-301, 2004.
- [29] S. Biswas and R. Izmailov, "Design of a fair bandwidth allocation policy for VBR traffic in ATM networks," *IEEE Trans. Networking.*, vol. 8, no. 2, pp. 212-223, 2000.
- [30] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in Aloha networks," *IEEE Trans. Autom. Control*, vol. 49, no. 10, pp. 1858-1863, 2004.
- [31] E. Kim and Y.-J. Suh, "A rate adaptive transmission opportunity for fairness over IEEE 802.11e wireless LANs," in *Proc. IEEE Int. Conf. Comm.*, pp. 4523-4528, 2007.
- [32] X. Yin, X. Zhou, R. Huang, Y. Fang, and S. Li, "A fairness-aware congestion control scheme in wireless sensor networks," *IEEE Trans. Veh. Tech.*, vol. 58, no. 9, pp. 5225-5234, 2009.
- [33] Y. Durmus, A. Ozgovde, and C. Ersoy, "Distributed and online fair resource management in video surveillance sensor networks," *IEEE Trans. Mob. Comp.*, vol. 11, no. 5, pp. 835-848, 2012.
- [34] M. Torabzadeh and W. Ajib, "Packet scheduling and fairness for multiuser MIMO systems," *IEEE Trans. Veh. Tech.*, vol. 59, no. 3, pp. 1330-1340, 2010.
- [35] B. Wang and D. Zhao, "Scheduling for long term proportional fairness in a cognitive wireless network with spectrum underlay," *IEEE Trans. Wirel. Comm.*, vol. 9, no. 3, pp. 1150-1158, 2010.
- [36] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. ACM Mobicom*, Aug. 2000.
- [37] N. Blefari-melazzi, A. Detti, A. Ordine, and S. Salsano, "A mechanism to enforce TCP fairness in 802.11 wireless LANs and its performance evaluation in a real test-bed," in *Proc. World. Wirel. Mob. Multi. Net.*, pp. 1-7, 2007.
- [38] B. A. H. S. 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. Wirel. Commun.*, vol. 7, no. 9, pp. 3517-3525, 2008.
- [39] E. Park, D. Kim, H. Kim, and C. Choi, "A cross-layer approach for per-station fairness in TCP over WLANs," *IEEE Trans. Mob. Comput.* vol. 7, no. 7, pp. 898-911, 2008.
- [40] Q. Wu, M. Gong, and C. Williamson, "TCP fairness issues in IEEE 802.11 wireless LANs," *Int. J. Comp. Comm.*, vol. 31, no. 10, pp. 2150-2161, 2008.
- [41] V. P. Kemerlis, E. C. Stefanis, G. Xylomenos, and G. C. Polyzos, "Throughput unfairness in TCP over WiFi," in *Proc. Wirel. On-Demand Net. Sys. Serv.*, pp. 26-31, 2006.

- [42] R. W. Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A TCP fairness control method for concurrent communications in elastic WLAN system using Raspberry Pi access-point," IEICE Tech. Rep., NS2018-175, pp. 96-100, Dec. 2018.
- [43] Software Verzeichnis development, Homedale WLAN Monitor, <http://www.the-sz.com/products/homedale>. Accessed 1 Sep., 2019.
- [44] R. W. Sudibyo, N. Funabiki, K. I. Munene, and M. M. Islam, "A Proposal of Dynamic Delay Optimization Function in TCP Fairness Control Method for Two-Host Concurrent Communications in Elastic WLAN System," in Proc. IEICE General Conf., BS-4-32, pp. S70-S71, March 2019.
- [45] R. W. Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A TCP fairness control method for two-host concurrent communications in elastic WLAN system using Raspberry Pi access-point," in Proc. 2nd Int. Conf. Commun. Eng. Tech., pp. 76-80, April 2019.
- [46] R. W. Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam, and W.-C. Kao, "A Proposal of TCP Fairness Control Method for Two-Host Concurrent Communications in Elastic WLAN System Using Raspberry Pi Access-Point," Int. J. Adv. in Sci., Tech, Eng. Sys., vol. 4, No. 6, pp. 10-18, Nov. 2019.
- [47] K. Astrom and T. Hagglund, "PID controller: theory, design, and tuning," 2nd Edition, Instrument Society of America, 1995.
- [48] R. W. Sudibyo, N. Funabiki, K. I. Munene, and M. M. Islam, "A Generalization of TCP Fairness Control Method Using Transmission Delay for Multiple-Host Concurrent Communication in Elastic WLAN System," in Proc. IEICE Society Conf., BS-4-9, pp. S-27-28, Sept. 2019.
- [49] M. S. Gast, "802.11 wireless networks - the definitive guide," 2nd ed., O'Reilly, Sebastopol, Apr. 2005.
- [50] 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.
- [51] S. Banerji, "Upcoming standards in wireless local area networks," Wirel. Mob. Technol., vol. 1, no. 1, Sep. 2013.
- [52] 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 1 Sep., 2019.
- [53] V. Beal, "What is 802.11 wireless LAN standards?," Internet: https://www.webopedia.com/TERM/8/802_11.html. Accessed 1 Sep., 2019.
- [54] 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, Sept. 2015.
- [55] "IEEE 802.11 - Wikipedia," Internet: http://en.wikipedia.org/wiki/IEEE_802.11. Accessed 1 Sep., 2019.

- [56] M. E. Islam, "A study of access-point aggregation algorithm for elastic wireless local-area network system and its implementation," Ph.D. thesis, Grad. School of Natural Science and Technology, Okayama Univ., Japan, March 2016.
- [57] 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.
- [58] A. S. Tanenbaum, and D. J. Wetherall, "Computer networks," 5th ed., Pearson Prentice Hall, 2011.
- [59] J. H. Yeh, J. C. Chen, and C. C. Lee, "WLAN standards," *IEEE Potentials*, vol. 22, no. 4, 2003.
- [60] IEEE, "IEEE Std 802.11-2012, IEEE standard for Information technology," Internet: <https://standards.ieee.org/findstds/standard/802.11-2012.html>. Accessed 1 Sep., 2019.
- [61] 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.
- [62] 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.
- [63] M. S. Gast, "802.11ac: a survival guide," O'Reilly Media, Inc., 2013.
- [64] "WiMAX page, IEEE802.11ac works," Internet: <http://123-info.net/wimax-page/archives/918>. Accessed 1 Sep., 2019.
- [65] 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.
- [66] "IEEE 802.11n-2009 - Wikipedia," Internet: http://en.wikipedia.org/wiki/IEEE_802.11n-2009. Accessed 1 Sep., 2019.
- [67] M. S. Gast, "802.11n: A survival guide," O'Reilly Media, Inc., March 2012.
- [68] 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.
- [69] "arp-scan user guide," Internet: <https://linux.die.net/man/1/arp-scan>. Accessed 1 Sep., 2019.
- [70] "Arp Scan," Internet: <https://launchpad.net/ubuntu/trusty/+package/arp-scan>. Accessed 1 Sep., 2019.
- [71] P. A. Nelson, "bc an arbitrary precision calculator language," Internet: https://www.gnu.org/software/bc/manual/html_mono/bc.html. Accessed 1 Sep., 2019.
- [72] J. Malinen, "hostapd: IEEE 802.11 AP, IEEE 802.1 X. WPA/WPA2/EAP/RADIUS authenticator," Internet: <https://w1.fi/hostapd/>. Accessed 1 Sep., 2019.

- [73] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, “SSH, the secure shell: the definitive guide,” O’Reilly Media Inc., 2005.
- [74] T. Ylonen and C. Lonvick, “The secure shell (SSH) protocol architecture,” 2006.
- [75] “OpenSSH,” Internet: <http://www.openssh.com/>. Accessed 1 Sep., 2019.
- [76] 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.
- [77] “Run / execute command using SSH,” Internet: <https://www.cyberciti.biz/faq/unix-linux-execute-command-using-ssh/>. Accessed 1 Sep., 2019.
- [78] P. Michel, “about iw,” Internet: <https://wireless.wiki.kernel.org/en/users/documentation/iw>. Accessed 1 Sep., 2019.
- [79] B. Hubert, “tc(8) - Linux man page,” Internet: <https://linux.die.net/man/8/tc>. Accessed 1 Sep., 2019.
- [80] “iPerf - The TCP, UDP and SCTP network bandwidth measurement tool,” Internet: <https://iperf.fr/>, Accessed 1 Sep., 2019.
- [81] “TC(8),” Internet: <http://man7.org/linux/man-pages/man8/tc.8.html>, Accessed 1 Sep., 2019.
- [82] 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.
- [83] T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang, “SWAN: An SDN based campus WLAN framework,” in Proc. Int. Conf. Wirel. Commun., Vehi. Technol., Inform. Theory., Aero. Electronic Syst., pp. 1-5, May 2014.
- [84] E. Luengo, C. Weber, S. S. Heydari, and K. El-Khatib, “Design and implementation of a software-defined integrated wired-wireless network Testbed,” in Proc. ACM Int. Symp. Mob. Manag. Wirel. Access, pp 47-52, Nov. 2015.
- [85] S. Sukaridhoto, N. Funabiki, T. Nakanishi, and K. Watanabe, “A design for OpenFlow implementation of fixed backoff-time switching method in wireless mesh networks,” in Proc. IEICE Gen. Conf., BS-1-26, pp. S-51-52, March 2013.
- [86] S. Sukaridhoto, N. Funabiki, D. Pramudihanto, and Z. Arief, “A fixed backoff-time switching method for wireless mesh networks: Design and Linux implementation,” in Proc. Int. Conf. Inform. Technol. Elec. Eng., pp. 248-253. Oct. 2013.
- [87] N. Ahmed and U. Ismail, “Designing a high performance wlan testbed for centralized control,” in Proc. Int. Conf. Testbed. Res. Infr. Dev. Net. Commun. Work., pp. 1-6, April 2009.
- [88] F. G. Debele, M. Meo, D. Renga, M. Ricca, and Y. Zhang, “Designing resource-on-demand strategies for dense WLANs,” IEEE J. Selected Area. Commun., vol. 33, no. 12, pp. 2494-2509, Dec. 2015.
- [89] D. B. Faria, “Modeling signal attenuation in IEEE 802.11 wireless LANs,” in Tech. Report, TR-KP06-0118, Stanford Univ., July 2005.

- [90] 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.
- [91] N. Funabiki, C. Taniguchi, K. S. Lwin, K. K. Zaw, and W.-C. Kao, "A parameter optimization tool and its application to throughput estimation model for wireless LAN," in *Proc. Int. Work. Virtual Environ. Netw.-Orient. Appl.*, pp. 701-710, July 2017.
- [92] K. Astrom and T. Hagglund, "PID controller: theory, design, and tuning, 2nd Ed.", *Inst. Socie. America*, 1995.
- [93] Cypress, "CYW43438, Single-Chip IEEE 802.11 b/g/n MAC/Baseband/Radio with Integrated Bluetooth 4.2," 2016.