Technological University Dublin
ARROW@TU Dublin

Doctoral

Engineering

2013-11

An Autonomous Channel Selection Algorithm for WLANs

Fuhu Deng
Technological University Dublin, uhu.deng@student.dit.ie

Follow this and additional works at: https://arrow.tudublin.ie/engdoc

 Part of the Digital Communications and Networking Commons

Recommended Citation

Fuhu, D. (2013) *An Autonomous Channel Selection Algorithm for WLANs.* Doctoral Thesis, Technological University Dublin. doi:10.21427/D73S4K

This Theses, Ph.D is brought to you for free and open access by the Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Doctoral by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie, vera.kilshaw@tudublin.ie.

Funder: China Scholarship Council



Dublin Institute of Technology
ARROW@DIT

Articles | Communications Network Research Institute

2014-02-12

An autonomous channel selection algorithm for WLANs

Fuhu Deng

Follow this and additional works at: http://arrow.dit.ie/commart

 Part of the Digital Communications and Networking Commons

This Theses, Ph.D is brought to you for free and open access by the Communications Network Research Institute at ARROW@DIT. It has been accepted for inclusion in Articles by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.



An Autonomous Channel Selection Algorithm for Wireless LANs

By

Fuhu Deng

B. Eng., M. Eng.

A thesis submitted to the Dublin Institute of Technology

for the degree of

Doctor of Philosophy



Supervisor: Dr. Mark Davis

School of Electronic and Communications Engineering

November 2013

To my mother and father
Xiu Wu and Danian Deng

Abstract

IEEE 802.11 wireless devices need to select a channel in order to transmit their packets. However, as a result of the contention-based nature of the IEEE 802.11 CSMA/CA MAC mechanism, the capacity experienced by a station is not fixed. When a station cannot win a sufficient number of transmission opportunities to satisfy its traffic load, it will become saturated. If the saturation condition persists, more and more packets are stored in the transmit queue and congestion occurs. Congestion leads to high packet delay and may ultimately result in catastrophic packet loss when the transmit queue's capacity is exceeded. In this thesis, we propose an autonomous channel selection algorithm with neighbour forcing (*NF*) to minimize the incidence of congestion on all stations using the channels. All stations reassign the channels based on the local monitoring information. This station will change the channel once it finds a channel that has sufficient available bandwidth to satisfy its traffic load requirement or it will force its neighbour stations into saturation by reducing its PHY transmission rate if there exists at least one successful channel assignment according to a *predicting* module which checks all the possible channel assignments. The results from a simple C++ simulator show that the *NF* algorithm has a higher probability than the dynamic channel assignment without neighbour forcing (*NONF*) to successfully reassign the channel once stations have become congested. In an experimental testbed, the Madwifi open source wireless driver has been modified to incorporate the channel selection mechanism. The results demonstrate that the *NF* algorithm also has a better performance than the *NONF* algorithm in reducing the congestion time of the network where at least one station has become congested.

Declaration

I certify that this thesis which I now submit for examination for the award of _____, is entirely my own work and has not been taken from the work of others, to save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for another award in any other third level institution.

The work reported on in this thesis conforms to the principles and requirements of the DIT's guidelines for ethics in research.

DIT has permission to keep, lend or copy this thesis in whole or in part, on condition that any such use of the material of the thesis is duly acknowledged.

Signature_____ Date_____

Acknowledgements

Foremost, I would like to express my deepest gratitude to my supervisor, Dr. Mark Davis, for his excellent guidance during my graduate study, endless support and patience throughout the thesis process the Dublin Institute of Technology. I also wish to thank Prof. Gerald Farrell, for his support of my application and providing me with the opportunity to learn in an innovative, responsive and caring learning environment. I also wish to give my thanks to Prof. Zhiguang Qing and Proc. Bing Wu for believing in me and encouraging me to pursue my Ph.D. in DIT.

The Communications Network Research Institute (CNRI) is a wonderful place to work and my colleagues gave me great support I needed. I wish to thank my colleges, Chengzhe Zhang, Jianhua Deng, Mirek Narbutt, Mustafa Ramadhan, Stuart Wallace, Tanmoy Debnath, Yi Ding, and Ying Chen. It has been a pleasure to have they around with their contagious positive attitude. I am particularly grateful to Ying Chen for the indispensable advice he provided.

I wish to thank my all my dear friends as well as a sincere apology that I am not listing you all by name. I will mention a few people who have been extremely supportive during my study in Ireland: Erqiang Zhou, Jianfeng Wu, Jianhua Deng, Rong Hu, Yi (一) Ding and Yi (熠) Deng. They gave me invaluable support and for providing me with a roof for the first week when I have arrived in Ireland. I will never forget the time we spent together to make dinner to celebrate the Spring Festival every year.

I wish to thank my families, my mother Xiu Wu, my father Danian Deng, my brother Xiaolong Deng, my uncle Cheng Wu and Sheng Wu, for their infinite reserves of love, suggestion, support and encouragement.

I would like to express my deepest gratitude and love to my wife, Hongmei Wu, for her support and understanding. I will spend the rest of my life showing my appreciation.

Finally, I would like to thank the Chinese Scholarship Council (CSC) for supporting the funding to make this research possible.

Abbreviations List

AARF	Adaptive Auto Rate Fallback	p. 17
AWGN	Additive White Gaussian Noise	p. 16
AMRR	Adaptive Multi Rate Retry	p. 17
AP	Access Point	p. 5
APBE	Accurate Passive Bandwidth Estimation	p. 34
ARF	Auto Rate Fallback	p. 17
ARS	Autonomous network Reconfiguration System	p. 7
BART	Bandwidth Available in Real-Time	p. 38
BPSK	Binary Phase Shift Keying	p. 16
CA	Channel Assignment	p. 51
CAR-RECA	Joint Channel Assignment and interference-aware QoS Routing algorithms	p. 55
CCM	Channel Cost Metric	p. 54
CHAT	Channel Hopping Access with Trains	p. 47
CHMA	Channel-Hopping Multiple Access	p. 47
CO-MMAC	Connection-Oriented Multi-channel MAC	p. 49
cPEAB	Cognitive Passive Estimation of Available Bandwidth	p. 34
CRAFT	Channel and Routing Assignment with Flow Traffic	p. 53
CSIE	Channel Switch Information Element	p. 12
CSMA	Carrier Sense Multiple Access	p. 9
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance	p. 9
CSMA/CD	Carrier Sense Multiple Access with Collision Detection	p. 9
CSP	Channel Switch Protocol	p. 42
CW	Contention Window	p. 10
DCF	Distributed Coordination Function	p. 9
DFS	Dynamic Frequency Selection	p. 14
DIFS	DCF Inter-Frame Space	p. 9
DR-CA	Data Rate Adaptive Channel Assignment	p. 50
DS	Direct-Sequence	p. 15
EIFS	Extended Inter-Frame Space	p. 10
FCRA	Flow-based Channel and Rate Assignment	p. 54
FCS	Frame Check Sequence	p. 5
FH	Frequency Hopping	p. 15
HAL	Hardware Abstraction Layer	p. 22
HWMP	Hybrid Wireless Mesh Protocol	p. 6
HRDS	High-Rate Direct-Sequence	p. 15
IBSS	Independent Basic Service Set	p. 6
IMM	Interface Management Module	p. 41
JCAR	Joint Channel Assignment and Routing	p. 54
JCIAR	Joint Channel allocation, Interface Assignment and Routing	p. 54
JRCA	Joint Routing and Channel Assignment protocol	p. 54
JRCAP	Joint Routing and Channel Assignment Protocol	p. 51

LCA	Local channel information assisted Channel Assignment	p. 44
LPP	Linear Programming Problem	p. 53
MAP	Mesh Access Point	p. 6
MAP	Multi-channel Access Protocol	p. 46
McMAC	Multi-Channel MAC	p. 48
MCMR	Multi-Channel Multi-Radio	p. 41
MesDRCA	Mesh-based Data Rate-Aware Channel Assignment	p. 50
MeshChop	Mesh Channel Hopping	p. 43
MesTic	a Mesh based Traffic and Interference aware Channel assignment	p. 43
MMAC	Multi-channel MAC	p. 47
MP	Mesh Point	p. 6
MPP	Mesh Portal Point	p. 6
MRC	Maximum Residual Capacity	p. 55
MRMC	Multi-Rate Multi-Channel protocol	p. 50
MR-WMN	Multi-Radio Wireless Mesh Network	p. 7
NICs	Network Interface Cards	p. 41
NF	Channel assignment algorithm with Neighbour Forcing	p. 104
NONF	Channel assignment algorithm without Neighbour Forcing	p. 104
ODC	On-Demand Switching	p. 49
OFDM	Orthogonal Frequency Division Multiplexing	p. 15
PABE	Passive Available Bandwidth Estimation	p. 35
PCB	Pause Count Back-Off	p. 11
PCF	Point Coordination Function	p. 9
PIFS	PCF Inter-Frame Space	p. 9
POS	Probability of Success	p. 54
QAM	Quadrature Amplitude Modulation	p. 16
QPSK	Quadrature Phase Shift Keying	p. 16
R-CA	Routing based Channel Assignment	p. 41
RB-CA	Rate-Based Channel Assignment	p. 50
RCA	Routing and Channel Assignment	p. 51
RTS/CTS	Request to Send/Clear to Send	p. 9
SIFS	Short Inter-Frame Space	p. 9
SIR	Signal-to-Interference Ratio	p. 42
SNR	Signal-to-Noise Ratio	p. 16
SSCH	Slotted Seeded Channel Hopping	p. 47
SSID	Service Set Identity	p. 6
STATIC	Static channel assignment	p. 104
TBTT	Target Beacon Transmission Times	p. 13
TIM	Traffic Indication Map	p. 12
TOPP	Train of Packet Pairs	p. 37
VAP	Virtual AP	p. 23
WDS	Wireless Distributed System	p. 23
WLANs	Wireless Local Area Networks	p. 5
WMN	Wireless Mesh Network	p. 6

Contents

Abstract I
Declaration II
Acknowledgements III
Abbreviations List IV
Contents VII
List of Tables XI
List of Figures XII
Chapter 1 Introduction 1
1.1 Problem Statement 1
1.2 Objective and contributions 2
1.3 Thesis Organization 3
Chapter 2 Background 5
2.1 IEEE 802.11 Wireless networks 5
2.1.1 Infrastructure network 5
2.1.2 Ad-Hoc network 6
2.1.3 Wireless mesh network 6
2.1.4 Self-organizing networks 7
2.1.5 Single-radio and multi-radio network 8
2.2 IEEE 802.11 protocol 8
2.2.1 The CSMA/CA Mechanism 9
2.2.2 Inter-frame spacing 9
2.2.3 Back-off mechanism 10
2.2.4 Frame structure 11
2.2.5 Management frame information elements 12
2.3 Channel information 14
2.4 Transmission rate 15
2.5 Capacity of a network 17

2.6 MAC Bandwidth Components and the Access Efficiency Factors 18
2.7 Madwifi wireless driver 22
2.7.1 The architecture of Madwifi 22
2.7.2 Beacon transmission mechanism in Madwifi 24
2.7.3 Channel change process in Madwifi 24
2.7.4 Adaptive rate algorithms in Madwifi 26
2.8 Conclusion 27
Chapter 3 Literature review 29
3.1 Bandwidth estimation methods 30
3.1.1 Passive bandwidth estimation methods 31
3.1.2 Active bandwidth estimation methods 35
3.2 Multi-channel network with unmodified IEEE 802.11 MAC 39
3.2.1 Static channel assignment and dynamic channel assignment 40
3.2.2 Centralized channel assignment 42
3.2.3 Distributed channel assignment 44
3.3 Novel multi-channel MAC protocols 46
3.3.1 Split phase MAC protocol 46
3.3.2 Common hopping MAC protocol 47
3.3.3 Multiple rendezvous MAC protocol 47
3.3.4 Dedicated control channel MAC protocol 48
3.4 Compound channel assignment method 49
3.4.1 Multi-rate involved channel assignment method 49
3.4.2 Game theory based channel assignment method 50
3.4.3 Joint routing and channel assignment method 51
3.4.4 Traffic load aware channel assignment method 55
3.5 Conclusion 56
Chapter 4 Passive bandwidth estimation based on access efficiency 59
4.1 Available bandwidth estimation method 59
4.1.1 Motivation 59
4.1.2 Available bandwidth estimation-scenario 1 63

4.1.3 Available bandwidth estimation-scenario 2 64
4.1.4 Available bandwidth estimation-scenario 3 65
4.1.5 Experimental testbed setup 66
4.2 Analysis and results of the passive available bandwidth estimation 67
4.2.1 Monitoring station becomes congested first 68
4.2.2 One of the neighbour stations becomes congested first 69
4.2.3 Impact of the other factors 71
4.3 Conclusion 71
Chapter 5 Autonomous channel selection algorithm based on neighbour forcing 72
5.1 An autonomous channel selection algorithm based on neighbour forcing 72
5.1.1 Structure of the algorithm 73
5.1.2 Neighbour forcing module 76
5.2 Simple C++ simulator 82
5.2.1 Description of the simulator 83
5.2.2 Structure of the C++ simulator 84
5.2.3 Performance metric for the simulated algorithms 86
5.3 Results of the simulation and discussions 87
5.3.1 Successful reassignment ratio 88
5.3.2 Average number of theoretical successful channel assignment 91
5.4 Modifications to the Madwifi driver 92
5.4.1 Modifying the beacon transmissions in Madwifi 93
5.4.2 Implementation of the channel switch mechanism in Ad-hoc mode of Madwifi 95
5.5 Experimental testbed setup 97
5.5.1 Structure of the testbed 97
5.5.2 Packet transmission and reception module 99
5.5.3 Congestion status checking module 101
5.5.4 Channel monitoring module 101
5.5.5 Channel changing module 102
5.5.6 Predicting module 103
5.5.7 Neighbour forcing module 103

5.5.8 Performance metrics of the channel selection algorithms 104
5.6 Experimental results and discussion 105
5.6.1 Average One-way packet delay of the proposed channel selection algorithm 108
5.6.2 Congestion time of the proposed channel selection algorithm 116
5.7 Benefit of the proposed channel selection algorithm 120
5.8 Limitations of the autonomous channel selection algorithm 121
5.9 Conclusion 121
Chapter 6 Summary and Conclusion 123
6.1 Main achievements of the thesis 124
6.2 Future work 125
References 129
Appendix A 148
Appendix B 149
Appendix C 150
Appendix D 153

List of Tables

Table 2-1 Standardized values for the information element ID number 12

Table 3-1 Passive bandwidth estimation algorithms 33

Table 3-2 Algorithms for the Linear Programming Problem 52

Table 5-1 Successful reassignment ratio for the channel switching algorithms 116

List of Figures

Figure 2-1 IEEE 802.11 management frame structure 11

Figure 2-2 Information element structure 12

Figure 2-3 Structure of the Channel Switch Information Element (CSIE) 13

Figure 2-4 The IBSS DFS element 14

Figure 2-5 The Wi-Fi channels available in the 2.4 GHz band 15

Figure 2-6 The theoretical throughput against S/N for different modulation formats 16

Figure 2-7 The IEEE 802.11 basic access mechanism 19

Figure 2-8 Structure of the Madwifi driver 23

Figure 2-9 Flowchart of channel change in Madwifi 25

Figure 2-10 Parameters of Channel Switch Information Element (CSIE) 26

Figure 3-1 Packet dispersions P_{in} and P_{out} during the probe packet transmission process 37

Figure 4-1 MAC bandwidth components where two stations contend for the medium 61

Figure 4-2 Bandwidth components during bandwidth estimation (scenario 1) 63

Figure 4-3 Bandwidth components during bandwidth estimation (scenario 2) 65

Figure 4-4 Bandwidth components during bandwidth estimation (scenario 3) 66

Figure 4-5 Testbed used for bandwidth estimation 67

Figure 4-6 Available bandwidth and throughput of the two stations (Scenario 1) 69

Figure 4-7 Available bandwidth and throughput of the two stations (Scenario 2) 70

Figure 5-1 Structure of the channel selection algorithm with neighbour forcing 74

Figure 5-2 Bandwidth components of the prediction process 76

Figure 5-3 Relationship between the MAC bandwidth components 79

Figure 5-4 MAC bandwidth components during the neighbour forcing process 80

Figure 5-5 Structure of the simple C++ simulator 84

Figure 5-6 Relationship between the performance parameters 86

Figure 5-7 Successful reassignment ratio under different traffic loads (4 stations) 88

Figure 5-8 Successful reassignment ratio of different traffic load (5 stations) 90

Figure 5-9 Ratio of successful reassignments (5 stations) 91

Figure 5-10 Ratio of successful reassignments (6 stations) 92
Figure 5-11 Modified beacon transmission process in the Madwifi driver 95
Figure 5-12 Structure of the experimental testbed 98
Figure 5-13 Structure of channel selection algorithm 99
Figure 5-14 Average packet delay before and after congestion 106
Figure 5-15 Offered traffic loads of the three stations in the experimental 107
Figure 5-16 Offered traffic load and actual traffic load of three stations (Scenario 1) 109
Figure 5-17 Delay of the stations when find a channel with sufficient available bandwidth 110
Figure 5-18 Offered traffic load and actual traffic load of three stations (Scenario 2) 112
Figure 5-19 Delay of the three stations when no channel has sufficient free bandwidth 114
Figure 5-20 Delay of the stations when the neighbour forcing module is triggered 115
Figure 5-21 PDF of the delay of all three stations 117
Figure 5-22 PDF of the delay related to congestion 119

Chapter 1 Introduction

IEEE 802.11 wireless networks have become very popular in the public, military, business and home sectors. All the stations in an IEEE 802.11 network communicate in the 2.4 or 5 GHz ISM frequency bands. These frequency bands are divided into channels. Generally, there are 3 non-overlapping channels in the license-free 2.4 GHz frequency band and 12 in the 5 GHz frequency band. A station needs to select an operating channel before it can transmit its packets. Unfortunately, the capacity experienced by a station is not fixed. A station becomes saturated when it cannot win a sufficient number of transmission opportunities to satisfy its traffic load requirement. When saturation occurs, the packets which cannot be transmitted will be stored temporarily in a transmit queue. The depth of the transmit queue will increase until it reaches its capacity. Additional packets that arrive in the transmit queue cannot be accommodated and are dropped and hence are lost. This represents a congestion condition and it can lead to a large packet delay and catastrophic packet loss.

1.1 Problem Statement

An IEEE 802.11 wireless network is a contention-based network where all the network stations share a common channel medium. The stations must compete with each other in order to win a sufficient number of transmission opportunities for their packets. As a result of the contention-based nature of channel access, the capacity experienced by a station is not fixed. Consequently, it is not possible to always ensure that a station has sufficient capacity to satisfy its load requirements. When congestion occurs, the station needs to be assigned another channel to satisfy its traffic load requirement. There are many ways to assign channels to stations. For example, static channel assignment and dynamic channel assignment are two popular methods used to assign channels to stations.

Static channel assignment [1] assigns channels to the stations permanently (or at least on a long-term basis). The benefit of this approach is that no further action is required by the network operator after the channel assignment has been performed. However, when congestion occurs, the stations with a static channel assignment do not change the channel. It cannot solve the congestion problem when it arises.

Dynamic channel assignment [2] assigns channels to the stations adaptively according to the traffic load and network topology. It reassigns the channel once congestion occurs. However, under certain traffic conditions, there may be no channel that has sufficient available bandwidth to satisfy the traffic load requirement and therefore it fails to reassign the channels.

Centralized channel assignment [3] collects the load information of all the stations in the network. With an increase in the scale of network, it becomes increasingly difficult to gather all the necessary information. The contention from neighbour stations means that the station cannot win a sufficient number of transmission opportunities and saturation occurs which can cause congestion. Congestion in a wireless mesh network is a neighbourhood phenomenon [4], so the best way to solve the neighbourhood congestion should be under a distributed approach. The distributed channel assignment algorithms [5] [6] exhibit a greater robustness following the failure of a few stations compared to the centralized channel assignment algorithms as they reassign the channels based upon local traffic load information.

1.2 Objective and contributions

To satisfy the traffic load of all the stations using the available channels, the channel assignment algorithm has to know whether any successful channel assignments exist or not. In this thesis, a passive bandwidth estimation method is introduced to estimate the available bandwidth. If the traffic load requirement is greater than the available bandwidth, there must be some stations that will become saturated or congested when these stations start to transmit their packets. This method has been implemented in a *predicting* module to determine the number of successful channel assignments that exist

under current traffic load requirements. If no successful channel assignment exists, it is not necessary to reassign the channels. If there exists at least one successful channel assignment, a distributed channel selection algorithm based on neighbour forcing has been developed to reassign the channels once the station becomes congested. The performance of this algorithm was simulated and validated by a C++ simulator and in a 7 station experimental testbed.

The main contributions of this thesis are:

- A novel method for passive bandwidth estimation designed to check the congestion status of a channel assignment.
- A channel selection algorithm based on neighbour forcing designed to reassign the channels once the stations become congested.
- A C++ simulator developed to validate the feasibility of the algorithm.
- An experimental testbed configured to validate the performance of the algorithm in terms of the average one-way packet delay and the aggregate congestion time.

1.3 Thesis Organization

This thesis is organized as follows:

Chapter 2 presents the background to wireless networks and channel assignment. The basic mechanism used in the packet transmission process of the IEEE 802.11 protocol is also presented. An overview of channel and rate selection mechanisms is introduced. The concept of access efficiency is described in this chapter and the open source wireless device driver Madwifi is also introduced.

Chapter 3 provides an overview of the bandwidth estimation and channel assignment algorithms. Two major bandwidth estimation methods are described in the first section: Passive bandwidth estimation and Active bandwidth estimation. In the second section, a number of channel assignment algorithms working with the unmodified MAC protocol are presented. Multi-channel MAC protocols are presented in the third section. In the last section, an overview of channel assignment algorithms combined with multi-rate,

game theory and routing are described.

Chapter 4 describes a passive bandwidth estimation method based on the concept of access efficiency in the first section. It is used to predict the congestion status of a given channel assignment. An experimental testbed is configured to validate the performance. The results of the passive bandwidth estimation show that the passive bandwidth estimation methods can accurately estimate the available bandwidth.

Chapter 5 presents the details of the channel selection algorithm based on neighbour forcing in the first section. In the second section, a C++ simulator is developed to investigate the probability of the channel selection algorithm to successfully reassign channels once congestion occurs. The results show that neighbour forcing algorithm has a higher probability to successfully reassign the channels once stations become congested. An experimental testbed is configured in forth section. Based on the outcomes of the predicting module, the proposed channel selection can successfully reassign the channels when the congested station cannot find a channel which has sufficient available bandwidth. The results for the average one-way packet delay show that the proposed channel selection algorithm can reduce the incidence of congestion. In the last two sections, the advantages and disadvantages of the *neighbour forcing* method are listed and discussed.

To validate the feasibility and the successful reassignment ratio of the channel selection algorithm, a simple C++ simulator is described in the third section. In the fourth section, the modifications to the beacon transmission process in the Ad-Hoc mode of the Madwifi driver are described. The two-stage beacon transmission process is combined with the proposed channel switch mechanism. A 7 station experimental testbed is described in the last section which is used to validate the performance of the proposed channel selection algorithm.

Chapter 6 summarizes the details of the proposed channel selection algorithm. The contributions of the thesis are also listed. Finally, some suggestions regarding possible future research work are also presented.

Chapter 2 Background

This chapter presents a background to WLAN networks and channel assignment. Because this thesis is mainly concerned with how to select a channel in a wireless network, we will introduce the main types of wireless networks in the first section. In the second section, some of the mechanisms in IEEE 802.11 MAC protocol are presented. In the third and fourth sections, some factors which have an impact on the capacity, such as channel frequency and rate selection mechanism will be introduced. At the end of this chapter, we discuss some other techniques that could be combined with channel assignment, for example multi-rate mechanism and game theory.

2.1 IEEE 802.11 Wireless networks

With the on-going reduction in hardware costs more and more different kinds of Wireless Local Area Networks (WLANs) have been deployed [7]. For example, most homes now use wireless routers to provide access the Internet for devices like laptops, tablets and smart phones. There are two major kinds of network: the infrastructure network and the Ad-Hoc network [8].

2.1.1 Infrastructure network

At present, most WLANs operate in the infrastructure mode where a central Access Point (AP) is used. The transmission range of an AP defines the area of the basic service. All the data communication is relayed through the AP which requires that all the wireless enabled devices are within range of the AP, but no restriction is placed on the distance between wireless devices themselves. The AP has the responsibility to manage the connections of the network. It transmits several beacon frames each second to announce the presence of the WLAN. A beacon frame consists of a MAC header, frame body and Frame Check Sequence (FCS). In the variable-length frame body, the beacon

frame includes network parameters such as timestamps, time interval between beacon frames and Service Set Identity (SSID) etc. This information is utilized to maintain the operation of the network and to broadcast the properties of this network.

2.1.2 Ad-Hoc network

Another popular type of IEEE 802.11 wireless network is the Ad-Hoc network which is sometimes referred as an independent basic service set (IBSS) or an Ad-Hoc BSS. An Ad-Hoc network typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other Ad-Hoc network devices within the transmission range. Wireless devices in an Ad-Hoc network communicate directly with each other and all of them have responsibility to maintain the connectivity of the network [9]. The first member of the Ad-Hoc network will send out beacon frames periodically. Other members will receive the network parameters (such as SSID and beacon interval) and decide to join the network. All the members in an Ad-Hoc network must periodically transmit beacon frames if they don't receive beacon frames from other members within a short random delay period after the beacon is supposed to have been sent.

2.1.3 Wireless mesh network

The IEEE 802.11s standard defines how wireless devices can interconnect to create a WLAN mesh network [10]. A wireless mesh network (WMN) is a communications network made up of radio nodes organized in a mesh topology. It consists of three types of radio nodes: Mesh Point (MP), Mesh Portal (MPP) and Mesh Access Point (MAP). A MP supports a Peer Link Management protocol which is used to discover neighbouring nodes and to keep track of the neighbour information. The neighbour discovery is only limited to nodes which are in the transmission range of an MP. If the destination node is out of the range of a MP, the Hybrid Wireless Mesh Protocol (HWMP) is implemented on a MP to support the neighbour discovery. HWMP is a hybrid protocol as it supports two kinds of path selection protocols – proactive and on-demand protocols. Instead of

an IP address, HWMP uses a MAC address for routing even though it acts like a routing protocol. MPPs are connected to both the mesh network and Internet so the users can access the Internet through these gateway functional MPPs. The MPPs must have at least two interfaces to provide the gateway functionality.

A MAP is the traditional AP augmented with mesh functionality so it can serve as an AP and be a part of the mesh network at the same time.

A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating in effect an ad hoc network) [11]. This distributed feature brings many advantages to WMNs such as low installation costs, easy network maintenance, robustness, and reliable service coverage.

2.1.4 Self-organizing networks

A system is self-organized if it is organized without any external or central dedicated control entity. In other words, the individual entities interact directly with each other in a distributed peer-to-peer fashion. Interaction between the entities is usually localized. But self-organization is more than just distributed and localized control. It is about the relationship between the behaviour of the individual entities (at the local level) and the resulting structure and functionality of the overall system (at the global level). In self-organized systems, the application of rather simple behaviour at the local level leads to sophisticated organization of the overall system. This phenomenon is called emergent behaviour [12]. Another important characteristic of self-organized systems is their adaptability with respect to changes in the system or environment. In fact, the entities continuously adapt to changes in a coordinated manner, such that the system always reorganizes as a reaction to different internal and external triggers for change.

The authors of [13] also introduce an autonomous network reconfiguration system (ARS) which allows a multi-radio WMN (MR-WMN) to autonomously reconfigure its local network settings such as channel, radio, and route assignment for real-time recovery from link failures. ARS also includes a monitoring protocol that enables a

WMN to perform real-time failure recovery in conjunction with the planning algorithm.

2.1.5 Single-radio and multi-radio network

Depending on the number of radios per node used to utilize multiple channels, a network can be categorized as a single-radio network or a multi-radio network. The major advantage of a single-radio network is low price. However, it requires an efficient way to assign the channel to avoid interference and maintain the network connectivity because all IEEE 802.11 devices are half-duplex and cannot receive and transmit packets simultaneously. The multi-radio network can increase the network capacity [14]. It can transmit and receive on different channels simultaneously because it can assign non-overlapping channels to each radio. Multi-radio networks are considerably less sensitive to link failure or deactivation than their single-radio counterparts [15].

The theoretical underpinnings of capacity maximization in multi-radio wireless mesh networks have been extensively studied [14]. These solutions require network-wide coordinated packet scheduling in order to successfully operate which can make them impractical. Even a practical capacity maximization algorithm is difficult to achieve, however it has great potential to increase the capacity of network.

2.2 IEEE 802.11 protocol

Like any other IEEE 802.x protocol, the IEEE 802.11 protocol covers the MAC layer and Physical layers. The MAC layer provides a variety of functions that support the operation of IEEE 802.11-based WLANs. It manages and maintains communications between IEEE 802.11 stations (radio network cards and wireless interface) by coordinating access to a shared radio channel and utilizing protocols that enhance communication over a wireless medium. The IEEE 802.11 MAC layer uses an IEEE 802.11 Physical layer, such as IEEE 802.11b [16], IEEE 802.11g [17] and IEEE 802.11a [18] to perform the tasks of carrier sensing, transmission, and receiving of IEEE 802.11 frames.

The IEEE 802.11 standard defines two forms of medium access, distributed coordination function (DCF) and point coordination function (PCF). DCF is mandatory and based on the CSMA/CA (carrier sense multiple access with collision avoidance) protocol. The important aspects of the IEEE 802.11 relevant to this these will be discussed further in the next few sections.

2.2.1 The CSMA/CA Mechanism

Like Ethernet, IEEE 802.11 uses a carrier sense multiple access (CSMA) scheme to control access to the transmission medium. However, collisions waste valuable transmission capacity, so rather than the collision detection (CSMA/CD) employed by Ethernet, the IEEE 802.11 standard uses a distributed access scheme based upon collision avoidance (CSMA/CA).

CSMA/CA access is provided by the DCF. Another coordination function PCF is used to provide a contention-free service. In DCF, a station desiring to transmit senses the medium, if the medium is busy (i.e. some other station is transmitting) then the station will defer its transmission to a later time, if the medium is sensed free then the station is allow to transmit. To avoid collisions, nodes use a random back-off after each frame, with the first transmitter seizing the channel. In some circumstances, DCF may use the Request to Send/Clear to Send (RTS/CTS) technique to further reduce the possibility of collisions.

2.2.2 Inter-frame spacing

Inter-frame spacing is used to coordinate access to the transmission medium. The IEEE 802.11 standards use four different inter-frame time intervals to create different priority levels for different types of traffic. Three are used to coordinate the medium access, while another is used to deal with transmission errors.

Short inter-frame space (SIFS) is used for the highest-priority transmission such as RTS/CTS frames and positive acknowledgments, i.e. ACKs. *PCF inter-frame space (PIFS)* is used by the PCF during contention-free operation. *DCF inter-frame space*

(DIFS) is the minimum medium idle time for contention-based services. A station may have immediate access to the medium if it has been free for a period longer than the *DIFS*. *Extended inter-frame space (EIFS)* is not a fixed interval and is only used when there is an error in frame transmission.

A station must wait for an interval of *DIFS* to elapse before it can start to transmit a data frame. A medium idle interval of *SIFS* is required before any transmission of RTS/CTS frame and ACK frame. This waiting time represents an overhead of the MAC layer.

2.2.3 Back-off mechanism

Back-off is a well known method to resolve contention between different nodes which want to access the medium. The method requires each node to choose a random number between 0 and a given number called the *Contention Window (CW)*. The node must then wait for this number of time slots to elapse before accessing the medium, all the nodes must continually check the medium to determine if a medium busy condition has occurred. It must temporarily halt its back-off counter in this case.

The IEEE 802.11 standard defines an Exponential Back-off Algorithm. That must be executed in the following cases:

1) If when the node senses the medium before the first transmission of a packet and the medium is busy.

2) After each retransmission, and

3) After a successful transmission.

The only case when this mechanism is not used is when the node decides to transmit a new packet and the medium has been free for more than DIFS.

The Exponential Back-off algorithms will double the contention window when a transmission has failed and reset it to the initial value of *CWmin* following a successful transmission. A large *CW* results in a long duration to access the medium when there are only a few active stations in the system (although a large *CW* can lead to a lower collision rate). A small *CW* can enhance the channel utilization but the number of

collisions could increase quickly if a small *CW* is used for many active stations. Much research has shown that a change in the contention window size may degrade the network performance [19] [20]. They focus on adjusting the contention window size adaptively to fit the system status. Xu [21] compares the performance of different back-off functions for the multiple access protocol in an IEEE 802.11 WLAN. They found that linear and polynomial back-offs with appropriate parameter settings can improve upon the binary exponential back-off specified in the IEEE 802.11 WLAN standards, in terms of throughput, access delay statistics and packet drop rate. *Pause Count Back-off* (*PCB*) [22] observes the number of back-off counter pauses during the channel access contention and sets the appropriate contention window based on estimated results.

2.2.4 Frame structure

There are three main frame types defined in the IEEE 802.11 standard. Data frame are the pack horses of IEEE 802.11 responsible for transporting data from node to node. Control frame are used in conjunction with data frames to perform clearing operations, channel acquisition and carrier-sensing maintenance functions, and positive acknowledgment of received data. Management frames perform supervisory functions; they are used by stations to join and leave wireless networks and move associations from access point to access point.

For the purposes of this thesis, we can ignore the data and control frames instead concentrate on the management frames.

IEEE 802.11 management frames have a common structure as shown in Figure 2-1.

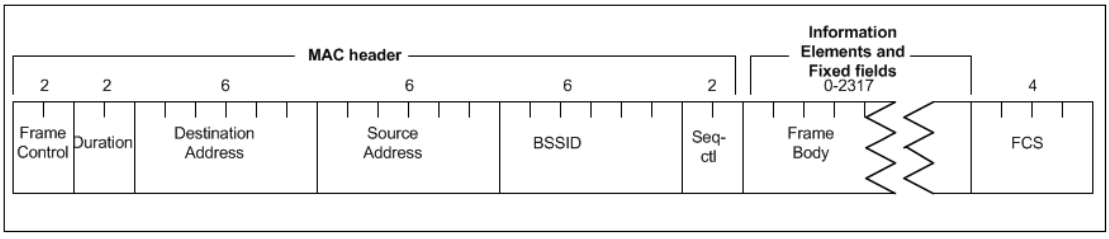


Figure 2-1 IEEE 802.11 management frame structure

The MAC header is the same in all management frames; it does not depend on the frame

subtypes. Some management frames use the frame body to transmit information specific to the management frame subtype.

The length of management frame is variable. Most of the data contained in the frame body uses fixed-length fields called fixed fields and variable-length fields called information elements.

2.2.5 Management frame information elements

Information elements are variable-length components of management frames as shown in Figure 2-2. A generic information element is tagged with an ID number, a length, and a variable-length component.

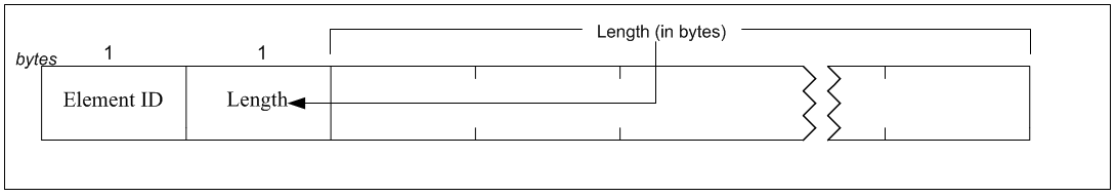


Figure 2-2 Information element structure

New information elements can be defined by newer revisions to the IEEE 802.11 specification.

Table 2-1 Standardized values for the information element ID number

Element ID	Name
0	Service Set Identity (SSID)
1	Supported Rates
2	FH Parameter Set
3	DS Parameter Set
4	CF Parameter Set
5	Traffic Indication Map (TIM)
6	IBSS Parameter Set
7-15	Reserved; unused
16	Challenge text
17-31	Reserved for challenge text extension
32-255	Reserved; unused

The IEEE 802.11h standard defined a new information element called the *Channel Switch Information Element* (CSIE) for the infrastructure network which is shown in Figure 2-3.

Element ID	Element length	Channel switch mode	New channel number	Channel switch count

Figure 2-3 Structure of the Channel Switch Information Element (CSIE)

The Element ID and Length field is the same as in other management frame elements. The Channel Switch Mode field indicates any restrictions on transmission until a channel switches. An AP in a BSS or a station in an IBSS sets the Channel Switch Mode field to either 0 or 1 on transmission. A Channel Switch Mode set to 1 means that the station in a BSS to which the frame containing the element is addressed shall transmit no further frames within the BSS until the scheduled channel switch. A station in an IBSS may treat a Channel Switch Mode field set to 1 as advisory. A Channel Switch Mode set to 0 does not impose any requirement on the receiving station.

The New Channel Number field is set to the number of the channel to which the station is moving. The Channel Switch Count field either shall be set to the number of *target beacon transmission times* (*TBTTs)* until the station sending the Channel Switch Announcement element switches to the new channel or shall be set to 0. A value of 1 indicates that the switch will occur immediately before the next TBTT. A value of 0 indicates that the switch will occur at any time after the frame containing the element is transmitted.

The Channel Switch Announcement element is included in Channel Switch Announcement frames and may be included in Beacon frames, and Probe Response frames. The decision to switch to a new operating channel in an infrastructure BSS is made only by the AP in infrastructure network.

The channel switch in Ad-Hoc network is more complex because of the following reasons:

1) There is no central node to coordinate the channel switch. If stations make independent decisions to switch channel, there is a problem where all stations announce a switch to different channels if several of them make the decision simultaneously.

2) There is no association protocol to exchange the channel switch information and it is difficult to determine the number of nodes in one channel.

3) Beaconing is a shared process in IBSS network. It cannot guarantee the synchronization after the channel switch.

IEEE 802.11h [23] defined another information element called *IBSS DFS element* that is used in the Ad-Hoc network. This is shown in Figure 2-4.

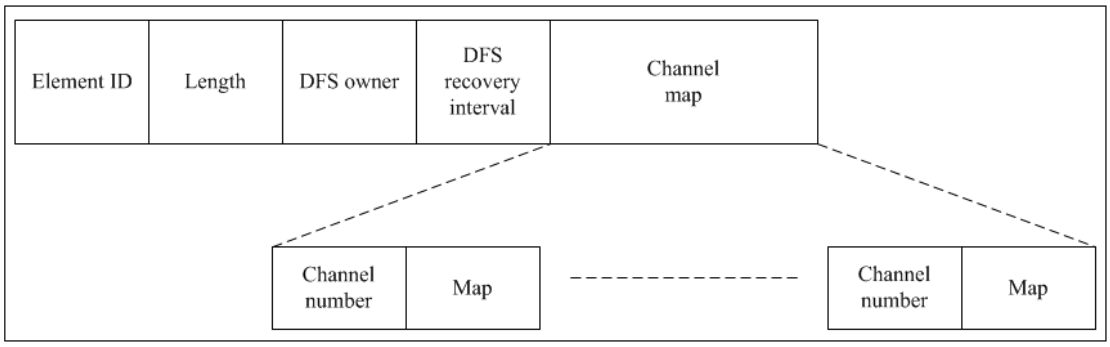


Figure 2-4 The IBSS DFS element

The *Length* field is variable. The Dynamic Frequency Selection (DFS) Owner field is set with the individual IEEE MAC address of the station that is the currently known DFS Owner in the IBSS. The DFS Recovery Interval field indicates the time interval that is used for DFS owner recovery, expressed as an integral number of beacon intervals. The DFS Recovery Interval value is static throughout the lifetime of the IBSS and is determined by the station that starts the IBSS.

The Channel Map field shown in Figure 2-4 contains a Channel Number field and a Map field for each channel supported by the station transmitting the IBSS DFS element.

2.3 Channel information

All the nodes in IEEE 802.11 network communicate in the license-free 2.4 or 5 GHz ISM frequency bands. These frequency bands are divided into channels. Figure 2-3 shows a graphical representation of the Wi-Fi channels in the 2.4 GHz frequency band, which is divided into 13 or 14 (depending on the regulatory regime) channels spaced 5 MHz apart with channel 1 centred on 2.412 GHz and channel 13 on 2.472 GHz.

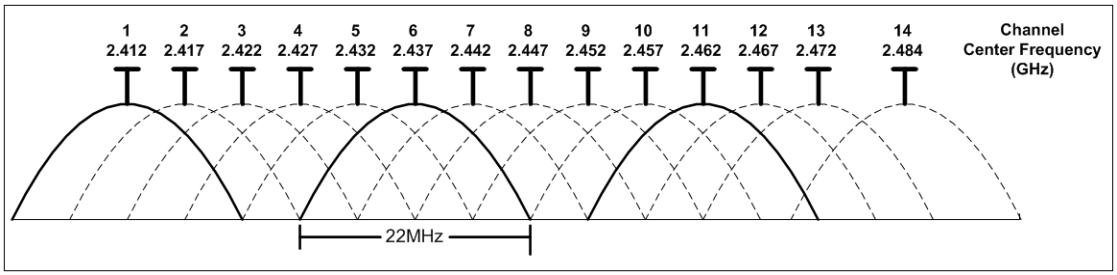


Figure 2-5 The Wi-Fi channels available in the 2.4 GHz band

In addition to specifying the channel centre frequency, IEEE 802.11 also specifies a spectral mask defining the permitted power distribution across each channel. The mask requires the signal be attenuated to a minimum of 30 dB below its peak amplitude at ±11 MHz from the centre frequency of a channel which is effectively 22 MHz wide. When two radio transmitters operating on the non-overlapping channels, in theory they will not interference with each other. One consequence of the channel separation is there are only three non-overlapping channels of the 2.4 GHz frequency band, typically channel 1, 6 and 11. In this thesis, the word channel refers to a non-overlapping channel. The IEEE 802.11a standard utilizes the 5 GHz frequency band and has 12 non-overlapping channels, 8 for indoor transmission and 4 for point-to-point transmission. The gap between central frequencies of neighbour channels is 20 MHz.

2.4 Transmission rate

The transmission rate is related to the modulation technique and coding scheme used. The original version of the IEEE 802.11 standard defined Frequency Hopping (FH) and Direct Sequence (DS) PHYs, but they were only capable of data rates up to 2 Mbps. The IEEE 802.11b standard added another physical layer. It uses the same MAC as all the other physical layers and is based on direct-sequence modulation. However, it enables a transmission rate up to 11 Mbps. The IEEE 802.11b PHY is also known as the high-rate, direct sequence (HRDS) PHY. By using different modulation schemes, IEEE 802.11b devices can transmit at 4 data rates, i.e. 1, 2, 5.5 and 11 Mbps.

IEEE 802.11g standard is the extension of IEEE 802.11b. It broadens IEEE 802.11b's data rate to 54 Mbps using Orthogonal Frequency Division Multiplexing (OFDM)

technique which is also utilized by IEEE 802.11a in the 5 GHz frequency band. The OFDM PHY uses a mixture of different modulation schemes to achieve data rates ranging from 6 Mbps to 54 Mbps. There are eight rates with the OFDM PHY: 6 and 9, 12, 18, 24, 36, 48 and 54 Mbps. These eight rates are divided into four rate tiers which are related to four different modulation schemes. The first one is Binary Phase Shift Keying (BPSK), second one is Quadrature Phase Shift Keying (QPSK), third one is 16-QAM (Quadrature Amplitude Modulation), and the last one is 64-QAM. The details of the PHY and modulation schemes are beyond the scope of this thesis and are not described here.

Higher transmission rates mean shorter transmission times. A shorter transmission time allows for more transmission opportunities which mean potentially higher throughputs. However, a higher transmission rate can also result in a higher packet error rate because the receiver may not be able to decode the packet correctly as a consequence of a lower Signal-to-Noise Ratio (SNR). Figure 2-6 shown the theoretical result of the bit error rate against Signal-to-Noise Ratio [24] when the channel contains Additive White Gaussian Noise (AWGN). With an increase in the transmission rate, the minimum required S/N to maintain a given error performance is increased.

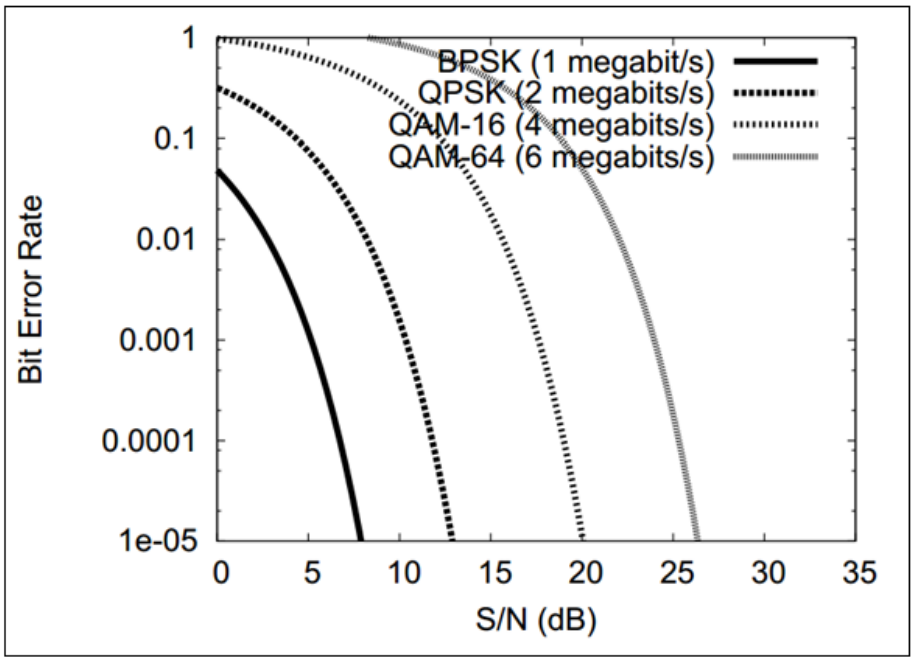


Figure 2-6 The theoretical throughput against S/N for different modulation formats

Due to the fast changes of SNR observed on the wireless channels, adaptive rate algorithms were developed to realise higher throughputs. Auto Rate Fallback (ARF) [25] was the first published rate adaptive algorithm and was used in WaveLan II devices which implemented the IEEE 802.11 DSSS standard. In ARF, the sender uses a higher transmission rate after a fixed number of successful transmissions at a given rate. It also will decrease the transmission rate after 1 or 2 consecutive failures. Because the channel capacity can change quickly, utilizing a fixed threshold of maximum retransmission number as the metric to make decisions to change to a higher transmission rate can lead to a poor performance. Adaptive ARF (AARF) [26] changes the threshold in ARF adaptively. It also increases the threshold used in ARF from 10 to 40 or 80 successful transmissions at a given rate. This algorithm acts like a low-pass filter. It doesn't change the transmission rate for short term changes in channel conditions.

The most widely used open source wireless driver Madwifi implements three different adaptive rate algorithms: *Onoe*, *AMRR* (Adaptive Multi Rate Retry) and *Sample rate*. The details of these three algorithms will be introduced later.

2.5 Capacity of a network

Alzate [27] offers a set of definitions for capacity, bandwidth and available bandwidth. These concepts are related to the idea of a communications link between a pair of nodes. In this thesis, the word link means a one-hop sender-receiver pair. The word path means multiple hops from a traffic source to a traffic sink. For a single hop link, the link capacity equals the maximal transmission rate achievable at physical layer. It doesn't consider the upper layer protocols. For a multi-hop path, the end-to-end capacity is highly depend on the single-hop link capacity and defined as:

$$C^{path} = \min_{i=1,\dots h} \frac{1}{\sum_{j=1,\dots n_1} 1 \Big/ C_{i,j}^{link}} \tag{2.1}$$

Here h is the number of spatial channels and n_1 is the number of hops. $C_{i,j}^{link}$ is the link capacity when j^{th} link select i^{th} spatial channels. When considering the power

constraint, protocol overhead and interference, the maximum throughput that an IEEE 802.11 single-hop link can attain is about half that of its transmission rate. This is the physical upper limit of a single channel network. When *n* identical randomly located nodes each capable of transmitting at *W* bits per second and each with randomly chosen destination, the capacity of single-channel is $\Theta(\frac{W}{\sqrt{n \log n}})$ bits per second under an idealized non-interference protocol [28]. In [14] and [29], the theoretical lower and upper bounds on the multi-channel network capacity are derived. When the nodes density is increased, it is not possible to support all the nodes' traffic requirement on one channel. As a result multiple-channel networks have been attracting much attention. In [30], the author characterized the impact of number of channels and interfaces per node on the network capacity. In [31], the results indicate that there is a significant scope for designing aggressive routing protocols that utilize the network capacity better to improve routing performance.

Many researchers focus on exploiting the use of multiple channels simultaneously to increase the network capacity [14] [32]. The problem of utilizing multiple channels is concerned with how to combine the channels and radios without creating additional interference. The reason is that the capacity is impacted not only by other stations but also by the traffic pattern itself. When we calculate the capacity of a channel, we need to consider the traffic pattern. In chapter 4, we will introduce a novel available bandwidth estimation algorithm which considers the traffic loads of all the stations.

2.6 MAC Bandwidth Components and the Access Efficiency Factors

The basic access mechanism of IEEE 802.11 based on the CSMA/CA mechanism and the back-off mechanism is shown in Figure 2-7. This diagram shows the scenario of a single station transmitting packets on the network. A set of time intervals in the packet transmission was introduced in [33]. *Busy time* corresponds to the transmission of frames and their positive acknowledgments (at least in the case of data and management frames). The complement of the *busy time* is the *idle time*. A station that has a data or

management frame waiting to be transmitted can use the idle time to win an access opportunity to the medium in order to transmit the frame. This time period is denoted as the *access time*. If the station does not have any data or management frames to be transmitted, the idle time can be viewed as being unused and hence available to the other stations. It is denoted as *free time*.

Summing up all the busy intervals and idle intervals (over a measurement period or some preset time period of interest) can indicate the busy and idle status of medium.

$$T_{busy} = \sum_{i} T_{busy}^{i} \tag{2-2}$$

$$T_{idle} = \sum_{i} T_{idle}^{i} \tag{2-3}$$

Here T_{busy}^{i} and T_{idle}^{i} are the durations of the i^{th} busy and idle intervals within the measurement period.

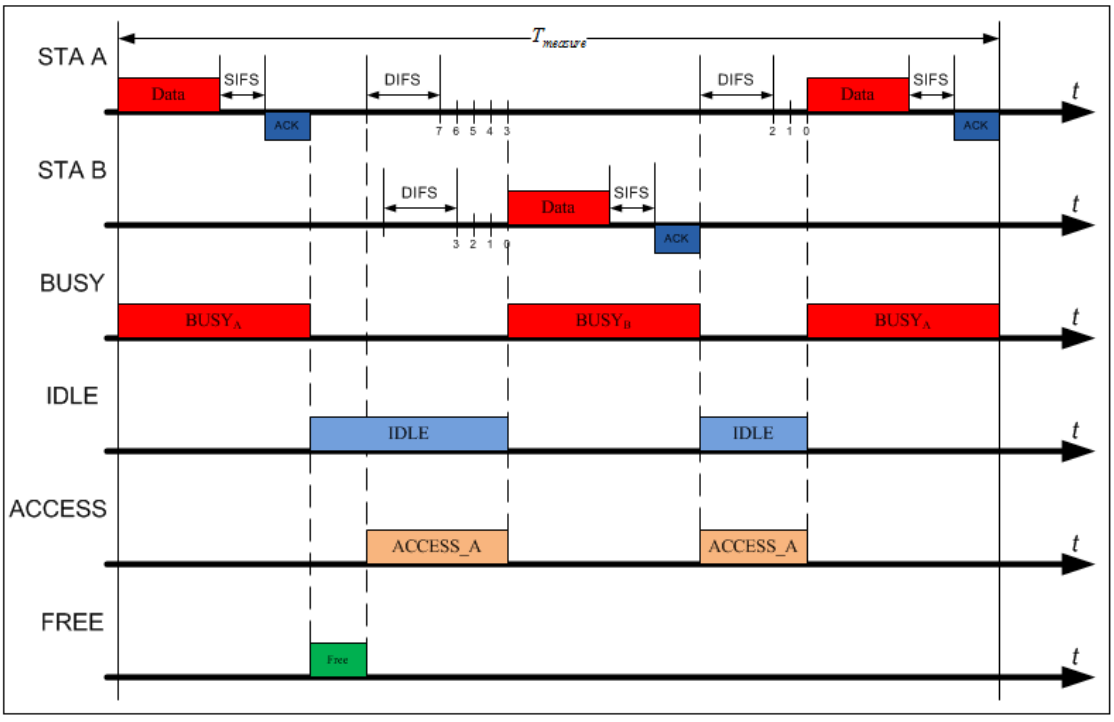


Figure 2-7 The IEEE 802.11 basic access mechanism

Combining the time intervals with the transmit rate, [33] introduces the normalized bandwidth components of BW_{busy} and BW_{idle} as follows:

$$BW_{busy} = \frac{T_{busy}}{T_{busy} + T_{idle}} \tag{2-4}$$

$$BW_{idle} = \frac{T_{idle}}{T_{busy} + T_{idle}} \tag{2-5}$$

Where obviously,

$$BW_{busy} + BW_{idle} = 1 \tag{2-6}$$

The load bandwidth BW_{load}^{k} corresponds to the normalized bandwidth used by a station *k* when it transmits its packets. In the single station scenario, BW_{load} and BW_{busy} will be identical. But in the multi-station scenario, because of the collisions of multiple stations, the relation between BW_{busy} and BW_{load} becomes:

$$BW_{busy} = \sum_{k} BW_{load}^{k} + BW_{collision} \tag{2-7}$$

Here BW_{load}^{k} denotes the portion of the bandwidth used by station *k* in transmitting its traffic load. $BW_{collision}$ is the bandwidth lost due to collisions when multiple stations transmit packets at the same time. The number of retransmission packets is used to calculate $BW_{collision}$.

Similar to the single-station scenario, in a multiple-station scenario the idle bandwidth of each station is composed of two bandwidth components: an access bandwidth BW_{access} and free bandwidth BW_{free}. The access bandwidth denotes the portion of the line rate bandwidth used to contend for access opportunities and the free bandwidth denotes the remaining unused idle bandwidth. The relationship between the two is expressed as:

$$BW_{idle} = BW_{free}^{k} + BW_{access}^{k} \tag{2-8}$$

To calculate the three major bandwidth components, it is required to know how to calculate the time intervals corresponding to T_{busy}, T_{idle} and T_{access}.

The busy time of the i^{th} packet can be calculated from the packet size and transmission rate using:

$$T_{busy}^{i} = \frac{Data_length}{Rate} + SIFS + \frac{ACK_length}{Basic_rate} \tag{2-9}$$

Here the basic rate depends on the physical layer. It is 1 Mbps in IEEE 802.11b and 6Mbps in IEEE 802.11g or IEEE 802.11a. The *Data_length* is the packet size in bits and the *Rate* is the PHY transmission rate in bps which can be obtained from the *radiotap* header which is added by Madwifi when it receives a packet.

The access time depends on a large number of factors, i.e. the time used to defer (i.e. waiting DIFS or EIFS) denoted as $T_{defering}$ and the time used to decreasing the back-off timer denoted as $T_{backoff}$. Because most of these factors are random times, it is make sense to consider the average time used to access the medium.

$$\overline{T}_{access} = \overline{T}_{deferring} + \overline{T}_{backoff}$$

Where $\overline{T}_{defering}$ is the average value of $T^{i}_{defering}$ which is the time used in defer to a busy medium for packet the i^{th} *packet* and $\overline{T}_{backoff}$ is the average value of $T^{i}_{backoff}$ which is the time used in backing off for packet i^{th} *packet*.

Similar to the busy bandwidth and idle bandwidth, the access bandwidth can be calculated using

$$BW_{access} = \frac{T_{access}}{T_{busy} + T_{idle}} \tag{2-10}$$

Where,

$$T_{access} = \overline{T}_{access} \times number_of_frame \tag{2-11}$$

Here the *number_of_frame* is the number of packets which are successfully transmitted during the measurement period.

Based on the concept of bandwidth components, [34] define a metric called *access efficiency* to describe the efficiency with which a station is accessing the medium.

$$\eta_a = \frac{BW_{load}}{BW_{access}} \tag{2-12}$$

This metric indicates the efficiency with which the station accesses the medium. A station with a larger access efficiency can support a larger load and also has a larger free bandwidth. By using the access efficiency, we will introduce a novel bandwidth estimation algorithm to help us to select channels more accurately. It not only considers

the bandwidth components of the channel but also the access efficiency of each station operating on the channel. The detail of this algorithm will be described in chapter 4.

2.7 Madwifi wireless driver

In this section, we will introduce the Madwifi open source wireless driver for the reason that all the operations of the wireless interface card can be controlled through the wireless driver. Different manufacturers have developed different wireless driver [35] to manage their hardware. According to the hardware utilized in the experiment, we will introduce the wireless driver developed by Atheros which is known as Madwifi.

There are three wireless card drivers *Madwifi* [36], *ath5k* [37], *ath9k* [38] that have been developed by Atheros. Madwifi stands for Multiband Atheros Driver for WIFI, which is one of the most widely used WLAN drivers available for Linux users today. It is stable and has an established user base. The driver itself is open source but depends on the proprietary Hardware Abstraction Layer (HAL) that is available in binary form only. We use the stable release v0.9.4-r4133. In this thesis, we have modified the code of the driver in order to implement some special functions which will be described later.

Ath5k is open source and does not depend on the HAL, but it only supports part of the chipset. Unfortunately, the wireless card we use is not included. *Ath9k* is still under development.

2.7.1 The architecture of Madwifi

The Madwifi driver was written in C language and includes four main modules. Figure 2-8 shows the Madwifi structure. *HAL* is the lowest level module which is the closed source API between the hardware and the device driver. *HAL* acts as a wrapper around hardware registries. *ath* module is the layer which is called by the *net80211* module and reads or writes the hardware register through the *HAL*. It includes the Atheros network hardware dependent functions such as hardware initialization and interface configuration. The *net80211* module implements the interface to the network device and

supports a wide range of operation mode such as station, AP, ad-hoc, monitor and Wireless Distributed System (WDS). Other control functions are implemented as separate module. For example, rate adaption, sync scan are implemented as loadable modules [39].

The Madwifi driver supports multiple APs and concurrent AP/Station mode operation on the same device. The devices are restricted to using the same underlying hardware and thus are limited to coexisting on the same channel and using the same physical layer features. Each instance of an AP or station is called a Virtual AP (or VAP). Each VAP can be in AP mode, station mode, Ad-hoc mode and monitor mode. Every VAP has an associated underlying base device which is created when the driver is loaded. There is no way to change the operation mode directly, the only way to change the operation mode is to destroy the old VAP and create a new VAP in the target operation mode.

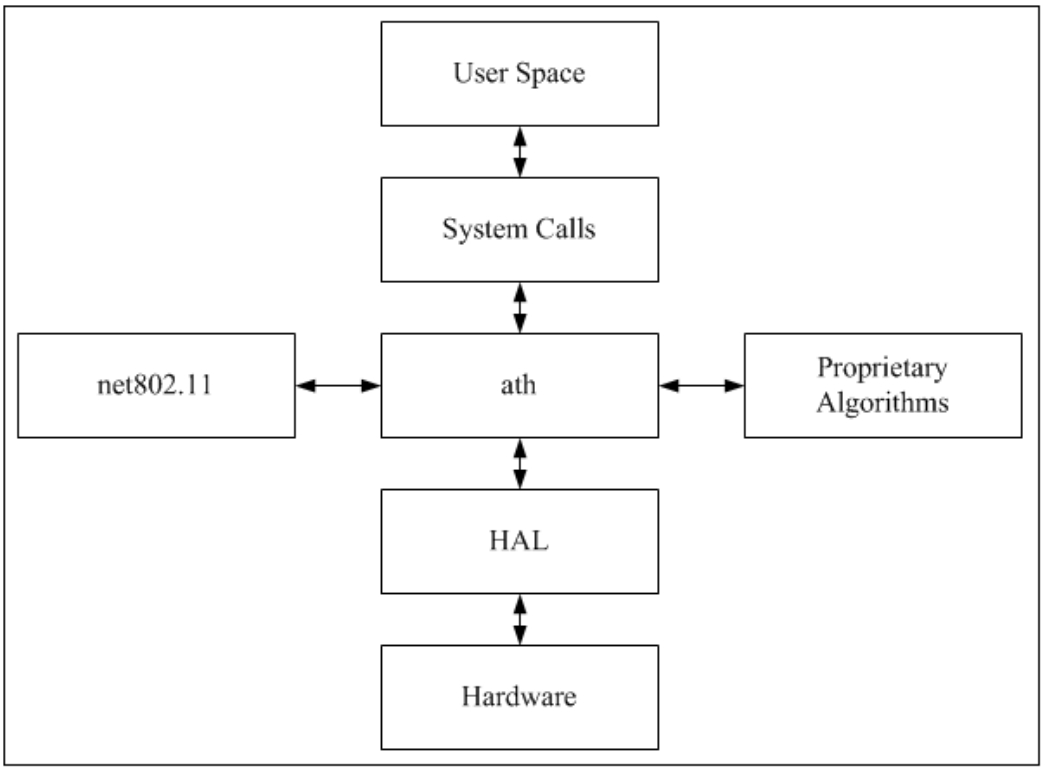


Figure 2-8 Structure of the Madwifi driver

2.7.2 Beacon transmission mechanism in Madwifi

In an infrastructure network, all the beacon frames are transmitted by the AP. However, in Ad-Hoc networks, all the nodes have the responsibility to transmit beacon frames. After receiving a beacon frame, each station waits for the beacon interval and then sends a beacon if no other station does so after a random time delay. This ensures that at least one station will send a beacon, and the random delay rotates the responsibility for sending beacons.

The *initial*, *allocate* and *update* functions of the beacon frame is implemented in the *net802.11* module. These functions are referenced by the *ath* module. Beacon frames are transmitted periodically to announce the presence of a network. All the nodes also synchronize with each other by using beacon frames. Most the beacon management is controlled by device's firmware or device's microcontroller to ensure this time accuracy. When the VAP is created by the user, all the beacon information is initialised in the *net80211* module and transmitted to the firmware. Without any changes to the beacon information, the beacon frame will be sent out continuously. When the upper layers change some of the network parameters such as transmission rate or channel, it needs to stop the beacon transmission process and initialise the beacon information again.

Beacon messages are triggered by *HAL*. When it is time to send a beacon, the *HAL* issues an interrupt, then the function *ath_beacon_send()* is invoked to send the beacon message. The beacon messages are directly passed to the *HAL* to transmit.

2.7.3 Channel change process in Madwifi

Madwifi uses a number of tools to configure the VAP. In this thesis we focus on the channel change process. Figure 2-9 shows the flowchart of channel change in the Madwifi driver.

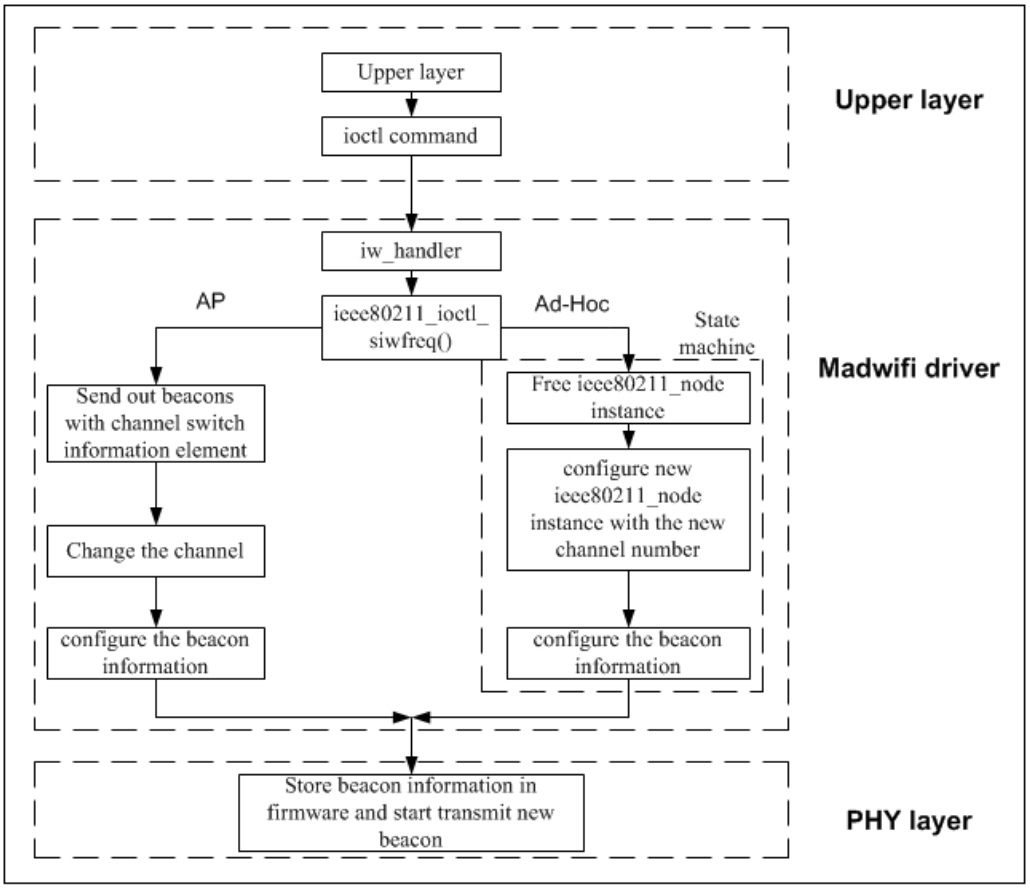


Figure 2-9 Flowchart of channel change in Madwifi

Once the driver recognizes the channel change requirement from an upper layer, it will use its own *iw_handler* function to read the channel change parameter. The function *ieee80211_ioctl_siwfreq* will be called once the channel change operation has been confirmed. Earlier in section 2.2.6, we saw that the beacon transmission mechanism is different in infrastructure and IBSS networks. There are two branches in this function.

If the operating mode of the VAP is the AP mode, it will announce the change of channel to the stations associates with it by sending a couple of beacon frames including *CSIE* (as shown in Figure 2-10). The VAP will change its operating channel and configure the beacon frame with the new channel.

Once a station receives a beacon frame with *CSIE*, it will parse the information element. The station will not change the channel immediately for the reason of security. It will change the channel when the number of beacon frames with *CSIE* reaches *TBTT*.

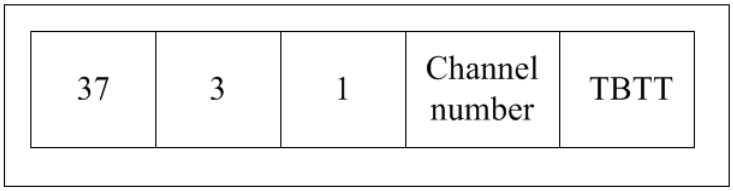


Figure 2-10 Parameters of Channel Switch Information Element (CSIE)

If the operating mode is Ad-Hoc mode, it will change the channel directly because it doesn't have the responsibility to announce the neighbour nodes with the channel change. The only thing it needs to do is to renew the table of neighbour nodes and configure the beacon frame with the new channel.

In chapter 4, we will introduce the details on how to modify the beacon transmission process in Ad-hoc mode to maintain the connectivity between the sender and receiver.

2.7.4 Adaptive rate algorithms in Madwifi

Madwifi includes three adaptive rate algorithms: *Onoe, AMRR* and *Sample.*

Onoe [40] is a credit based Rate Control Algorithm where the values of the credit is determined by the frequency of successful, erroneous and retransmissions accumulated during a fixed invocation period of 1000ms. If less than 10 percent of the packets need to be retransmitted at a particular rate, *Onoe* keeps increasing its credit point until the threshold value of 10 is reached. At this point the current transmission rate is increased to the next available higher rate and the process is repeated with a credit score of zero. A similar logic holds for deducting the credit score and moving to a lower bit rate for failed packets.

AMRR [26] uses a Binary Exponential Back-off technique to adapt the length of the sampling period used to changes in the values of bit-rate and transmission count parameters. It uses probe packets and depending on their transmission status to adaptively changes the threshold value.

The default rate adaptive algorithm selected by Madwifi is *Sample Rate* [24]. It decides the transmission rate to use based on the past history of the performance. It uses a smoothing window technique to keeps a record of the number of successive failures, the average transmission time, number of successful transmits and the total transmission

along with the destination for each transmission rate. When a station starts to send packets, it will select the highest transmission rate. Sample Rate will continue to transmit with that transmission rate until it experiences 4 successive failures. It will decrease the transmission rate until it finds a rate which is capable of transmitting packets. Every ten packets, Sample Rate will select a random rate from the set of transmission rate that may have a better performance than the current transmission rate to transmit one packet.

Sample Rate uses channel information feedback from the wireless interface card to calculate the number of successive failures, the average transmission time, the number of successful transmission and the total transmission along with the destination of the previous ten seconds. If it finds a transmission rate through a sampling process that has a better performance than the current one, it will select this transmission rate.

2.8 Conclusion

In this chapter, we have described the basic topics related to wireless networks and the IEEE 802.11 standard. Different types of networks utilize different mechanisms to manage the operation of the network and to maintain network connectivity. The IEEE 802.11 standard uses beacon frames to announce the presence of a network. However in infrastructure network and Ad-hoc network, the beacon transmission process is different. In infrastructure network, the AP assumes the responsibility to maintain the network connectivity by transmitting beacon frames periodically. All of the associated stations will maintain their connectivity after receiving beacon frames with channel switch information element. In Ad-Hoc network, because there is no central station to maintain the connectivity, all the members in this network will send out beacon frames after the beacon interval plus a random delay since the last time it received a beacon frame. At the end of this chapter, we introduced the most widely utilized wireless driver for Linux systems called Madwifi. This driver is open source so it can be modified to implement the functions we required such as channel switch process which is the basic function of the channel assignment algorithms. We will introduce the channel assignment

algorithms in the next chapter.

Chapter 3 Literature review

In this chapter, we will present how other researchers have sought to utilize multiple channels in order to transmit packets more efficiently and reliably. Before making the decision of channel selection, the nodes need to have accurate information about the channels, e.g. the available bandwidth on each channel. There are two methods used to obtain the information about the channel: Passive and active methods. Passive methods [41] do not need to transmit any packets. These involve configuring the wireless card into the monitor mode and then sniffing all the packets in each channel and calculating the time utilized by the packet transmissions. Active methods usually involve transmitting some probe packets to the destination node and calculating the delay or available bandwidth [42]. Once the nodes have a map of available bandwidth for all the channels, it is still not easy to select which channel is the best channel to transmit packets on because the medium is shared by multiple nodes. The contention and interference can still be important factors when selecting the channel. In section 3.2, we will discuss some channel assignment algorithms. The centralized channel assignment algorithm is implemented on nodes called “central” nodes which have the responsibility to assign the channels to all the nodes [3]. All the other nodes will send special frames to the central node. These frames deliver the information about the traffic load and the number of neighbour nodes for each node. The central node will assign channels to nodes in order to increase the capacity of network or minimize the total interference or minimize the delay. The distributed channel assignment [43] does not need any centralised nodes to select the channel. It collects information from the neighbour nodes and makes a decision regarding channel selection without considering the capacity and interference of the whole network directly. All the nodes in the network are implemented with the same algorithm and they make the decision independently. Some other researchers modify the MAC protocol to utilize multiple channels in order to increase the capacity of the network. In section 3.3, we will discuss a number of

different multi-channel MAC protocols. These protocols modify the medium access mechanism or the handshake mechanism to maintain the connection between the sender and receiver. They also assign a special channel known as dedicated control channel to send and receive control frames and the other channel is marked as the data channel. Based on the layered architecture of most of the devices, each layer has different protocols to control the packet transmission. Different layers can also have different parameters to show the performance of packet transmission. For example, the signal strength, packet number and transmit rate from the PHY layer; packet size from the IP layer. Cross-layer channel assignment algorithms consider all these factors together and also combine channel assignment with routing to maximize the throughput. Some researchers combine channel assignment with game theory where all the nodes are equal and follow the same strategy and make decisions based on the information they know. These two methods will be discussed in section 3.4.

3.1 Bandwidth estimation methods

In this section, we will introduce some bandwidth estimation methods. Because of the contention based nature, the capacity, bandwidth and available bandwidth in wireless network is not fixed. Alzate [27] offers a set of definitions for the capacity, bandwidth and available bandwidth for wireless ad hoc network. If a one-hop link is completely available for one station, the expected value of the link bandwidth of a C-bps link transmitting *L*-bit long packets is defined as:

$$E\left[BW^{link}(l)\right] = \frac{L}{\frac{L}{C} + E[T]} \tag{3-1}$$

Where *L* is the packet size, *C* is the PHY transmission rate in bps, *T* is the time required to get and release the transmission medium at that link.

If the link was shared with some other nodes, the concept of link available bandwidth is a more precise metric to describe the bandwidth. It represents the mean bandwidth available to a link *x* in a network during the interval $(t-\tau, t]$ and is defined as

$$E\left[ABW^{link_x}(L)\right] = \frac{L}{\frac{L}{C_x} + E\left[T_x\right]}\left[1 - \max_{i \in V_x} \sum_{j \in L_i} \sum_{k=l}^{\infty} \lambda_{j,k}\left(\frac{k}{C_j} + E\left[T_j\right]\right)\right] \quad (3\text{-}2)$$

Where $\lambda_{j,k}$ is the packet rate of a link including the forward and backward traffics which share the common operating channel; C_x is the x^{th} link capacity and T_x is the time it takes the packet to compete the transmission medium on link *x*. *V* is the set of spatial channels and L_i is the composed of links which have the same channel *i*. The available bandwidth is highly dependent on the competing cross-traffic.

In this thesis, because this research focuses on the MAC layer, we make the following definition:

Definition 3.1: Available Bandwidth. *For a station, the available bandwidth on a channel is defined as the maximum MAC layer throughput of the station that it could transmit on the channel without any stations being congested.*

There are two main methods used to measure the available bandwidth. The first one is passive method. The nodes operate in the monitor or promiscuous mode and calculate different metrics to indicate the quality and capacity of the channel such as retry rate, delay and jitter indicate the quality of this channel and throughput, transmission rate and bandwidth to indicate the capacity of the channel. The second method is to use active methods to measure the channel bandwidth. The sender or receiver sends out a series of packets and records the time intervals between each packet. The interval serves as a good metric to indicate the bandwidth of channel.

3.1.1 Passive bandwidth estimation methods

The passive bandwidth estimation methods do not consume any bandwidth and can provide a more precise estimation. However, it is hard to predict what will happen to the stations sharing the same channel when its own traffic joins in the competition for access. The previous passive bandwidth estimation methods usually estimate the

maximum achievable bandwidth for traffic control. These algorithms focus on the packet transmission process. It calculates the time used to transmit packets and the time when there are no stations utilizing the transmission medium. Figure 2-7 shows the atomic operation of the IEEE 802.11 MAC mechanism. The relation between the five time components: measurement time $T_{measure}$, free time T_{free}, busy time T_{busy}, idle time T_{idle} and access time T_{access} which can be expressed as in [33]:

$$T_{measure} = T_{busy} + T_{idle} \tag{3-3}$$

$$T_{idle} = T_{free} + T_{access} \tag{3-4}$$

Here, T_{free} is the time component may be used to transmit packet.

Figure 2-7 shows the atomic operation of frame transmission of the IEEE 802.11 MAC mechanism. Based on the definition of available bandwidth earlier, most of the researchers denote the capacity of link as a major factor in estimating the available bandwidth. The capacity should be the upper limit of the available bandwidth. Without considering the protocol overhead, the capacity equals the available bandwidth only when there is no other transmitter in the sensing range of the sender or receiver of the candidate link which means the channel is available for this link. However, the capacity is only one factor that determines the available bandwidth. There are many other factors that have an impact on the available bandwidth. Table 3-1 shows the performance metric and factors that the researchers have considered in order to increase the accuracy of the available bandwidth estimation.

Examining these factors, the most frequent factors are time-based factors such as inter-frame space, the time used to transmit data packets, the time used to transmit ACK packets and the back-off time. The next most frequently used factors are probability-based factors such as collision probability and probability of idle period synchronization.

Table 3-1 Passive bandwidth estimation algorithms

Reference	Performance metric	Factors
[44]	Available bandwidth	Excludes the idle periods shorter than DIFS Bandwidth of both ends of a link The synchronization between sender and receiver
[45]	Available bandwidth	Collision probability Collision time and retransmission back-off time
[41]	Available bandwidth	Carrier sense mechanism Idle period synchronization Collision probability Back-off mechanism
Speedo[46]	IP level available bandwidth	Packet size, Data rate, Packet error rate, Signal strength, Channel utilization, Number of active stations
cPEAB[47]	Available bandwidth	Overhead of control message Hidden nodes and packet size
APBE[48]	Available bandwidth	The bandwidth proportion occupied by DIFS and Back-off mechanism Packet collision probability Acknowledgement delay Channel idle time in the measurement period Overhead by the RTS and CTS
PABE [49]	Available bandwidth	Time used to transmit a packets and including DATA, ACK, interframe space, backoff delay and retransmission Considering all packets in the proposed range only.
[50]	Maximum bandwidth	Sender monitor outgoing traffic and record the retry and number of packets successfully transmitted

Algorithm [41] [44] [45] [47] [48] [49] define the available bandwidth as the maximum

MAC layer throughput that can be transmitted between sender-receiver pairs without disrupting any ongoing flow in the network. However, the difference is they consider different factors which have an impact on the accuracy of the bandwidth estimation algorithm. Sarr et al introduce a similar passive bandwidth estimation scheme [44] [45]. In [44], the authors denote the idle time as the total time during which the sniffer node neither emits any frame nor perceives the medium as being busy. It calculates the available bandwidth at both the sender and receiver of a link. The available bandwidth is denoted as:

$$B_{(S,R)} \leq \min(B_S, B_R) \tag{3-5}$$

Here B_S represents the available bandwidth at the sender station *S*. B_R represents the available bandwidth of the receiver station *R*. $B_{(S,R)}$ represents the available bandwidth of the link between *S* and *R*. Because the idle time of the sender station and receiver station could overlap, the algorithm also considers of the probability of synchronization between sender and receiver. In [45], they increase the accuracy by estimating the probabilities of the overlap of the silence periods experienced by the two peers on a link and an estimation of the collision probability on a link. The advantage of this algorithm is that it considers the synchronization of the idle time between the sender and receiver. However, the disadvantage is that it is difficult to calculate the probability of overlapping idle time.

To calculate the IP layer available bandwidth which is the number of IP bits per second, Speedo [46] takes into account information from multiple layers, i.e. the signal strength, packet count and transmission rate from the PHY layer; retry bit from the MAC layer; packet size from the IP layer and also the impact of dynamic rate adaptation algorithm. cPEAB (Cognitive Passive Estimation of Available Bandwidth) [47] focuses on the channel usage ratio, the proportion of DIFS and backoff, packet collision probability, the time to transmit ACK frame and the channel idle time. APBE (Accurate Passive Bandwidth Estimation) [48] considers the following elements in the passive bandwidth estimation: RTS and CTS overhead, the proportion of DIFS and back-off, packet

collision probability, the time to transmit ACK frame and the channel idle time. PABE (Passive Available Bandwidth Estimation) [49] defines the available bandwidth as the maximum transmission throughput between the two neighbour nodes in a certain transmission direction (i.e. from sender to receiver or from receiver to sender), under the condition that the quality of any outgoing flow will not be disrupted. It calculates the available bandwidth by multiplying the link capacity with the available channel idle time ratio. The available channel idle time ratio considers the impact of the transmission range. The traffic in this range will be recognized by the sniffer nodes. In [50], the author pays more attention to the packet transmission itself. They defined the maximum bandwidth as the number of bits successfully delivered divided by the time used to transmit these packets. It monitors all the outgoing packets and records necessary link-layer parameters such as transmission rate, number of successfully transmitted packets and the number of transmission attempts.

All the passive bandwidth estimation methods listed above consider the factors in the packet transmission process. The more factors that are considered, the better the accuracy of the bandwidth estimation can be achieved. The major advantage of these methods is it does not generate any additional traffic.

All the above passive bandwidth estimation algorithms calculate the time of packet transmission. However, they don’t consider the ability of other stations to access the medium. Different stations have different abilities to access the medium, i.e. the higher transmission rate station will suffer throughput degradation when there is one station transmitting packets at a low transmission rate. We propose a novel passive bandwidth estimation algorithm based on access efficiency which not only takes into account its ability but also the ability of other stations to access the medium.

3.1.2 Active bandwidth estimation methods

The active methods usually estimate the bandwidth by transmitting some back-to-back probing packets at different rates and measure the dispersion. Since dispersion between probing packets is highly correlated with channel capacity, it can be used to calculate

the available bandwidth. Figure 3-1 shows the packet dispersions during the probe packet transmission process [51]. The sender sends out some back-to-back probing packets with the dispersion of P_{in} which equals the time prepared to transmit a packet with size L. These packets are transmitted on a channel with capacity C_j and the dispersion change to P_{out}:

$$P_{out} = \max(P_{in}, \frac{L}{C_j}) \tag{3-6}$$

The receiver will send back *ACK* frames with length l for each packet. If the receiver treats all the packets in the same way, the dispersion of these ACK frames A_{in} will be:

$$A_{in} = P_{out} \tag{3-7}$$

If the time slot P_{out} is big enough for a data packet, it is also big enough for an ACK frame, so the dispersion of the *ACK* frames A_{out} will not be changed. So we could calculate this dispersion at the sender side. Because the relation between the three dispersion times is:

$$A_{out} = A_{in} = P_{out} = \max(P_{in,} \frac{L}{C_j}) \tag{3-8}$$

So, if packets are sent only in response to an *ACK*, the sender's packet spacing will exactly match the packet time on the link.

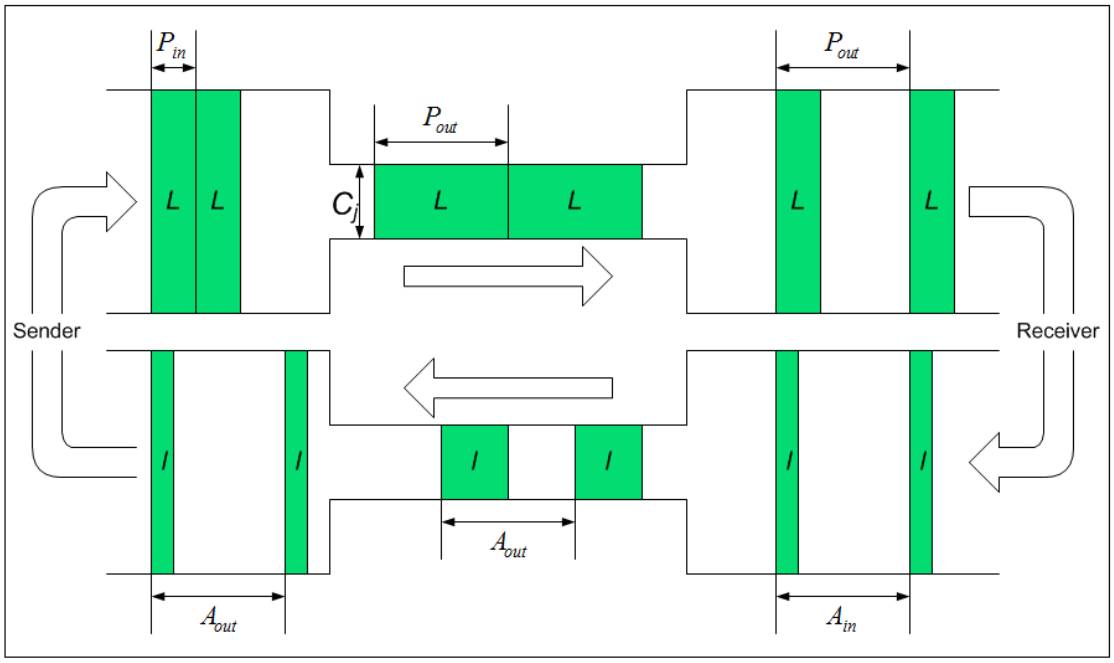


Figure 3-1 Packet dispersions P_{in} and P_{out} during the probe packet transmission process

However, the packet size and transmission rate of probing packets have a large impact on the accuracy of estimation. Smaller size probing packets generates less interference compared to other traffic [52]. Because of the dependence of the link bandwidth on the transmission rate at the PHY layer which is related to the signal strength, the capacity of the link changes frequently. So some researchers utilize a two-stage algorithm to estimate the available bandwidth which will be shown later.

To increase the accuracy of active bandwidth estimation, the difference between probing traffic and data flow needs to be considered. SLOT [53] provides an accurate and fast convergence active method to estimate end-to-end bandwidth. It uses a two-stage method to estimate the available bandwidth. In the first stage, SLOT transmits packets with different probing time and transmission rate in order to discover a more accurate range for the available bandwidth. In the second stage, SLOT measures the available bandwidth similar as TOPP [54] (Train of Packet Pairs) which uses a linear search method to provide an accurate range for the available bandwidth. The main advantage of these active approaches is that they can provide additional traffic information such as the delay, jitter and packet loss of the estimated link. However, the transmission of back-to-back probing packets generates additional extra traffic load on the network

which may cause performance degradations to existing flows. Besides, they can require a long convergence time for the measurements, and produce low accuracy compared with other bandwidth estimation techniques. To decrease the convergence time, WBest (Wireless Bandwidth Estimation Tool) [55] utilizes packet pairs to estimate the WLAN effective capacity which are related to transmit rate in the first stage. In the second stage, WBest sends a packet train at the effective capacity rate to determine the achievable throughput and to infer the available bandwidth. This method avoids the need for a search algorithm to determine the range of the available bandwidth of the link. The first stage is fast to get the effective capacity rate because the number of supported transmission rates is small (in IEEE 802.11b, the number of supported rates is 2, while in IEEE 802.11g and IEEE 802.11a, the number is 8, which was discussed in section 2.4). In [56], the authors halved the time required by the time of the probing process by only sending probe packets at the receiver side. In a homogeneous network, this could be used to estimate the bandwidth effectively. However, in a heterogonous network, this method only measures the bandwidth of the link from sender to receiver not the bandwidth of the both directions and it could not be used to accurately measure the bandwidth.

There are some other ways to estimate the available bandwidth that are neither passive nor active methods. *BART* (Bandwidth Available in Real-Time) [57] uses a *Kalman* filter to estimate the available bandwidth and also the capacity of the bottleneck link. *BART* injects probing packets into the target link and measures the one-way dispersion at the receiver side. The Kalman filter is used to estimate the available bandwidth when the probing rate exceeds the current available bandwidth. Yuan etc [58] developed a novel bandwidth estimation method which is based on a mathematical model that combines a TCP throughput model with an IEEE 802.11 DCF model. Packets should not be transmitted if there are some delayed packets in the queue even if the channel was sensed idle. In [59], it includes queue delay when it estimates the bandwidth because packets cannot be transmitted immediately even if the channel is idle if other packets are queued ahead of this packet. However, it is difficult to calculate the

bandwidth related to the queue delay. These methods estimate the available bandwidth in different ways. However they do not consider the impact of the traffic pattern and transmission rate.

The active bandwidth estimation algorithms discussed above don't focus on the packet transmission process. Otherwise, they utilize probe packets to estimate the available bandwidth. They can be accurate if the probe traffic is similar to the traffic it will transmit. However, because the available bandwidth is related to the transmission rate, the active bandwidth estimation algorithms need to transmit probe packets at all the available transmission rate which will consume more of the precious bandwidth resource. Our proposed bandwidth estimation algorithm does not generate additional traffic. It monitors the available channels and predicts what will happen if the station joins in the channel.

In chapter 4, we will introduce a novel passive available bandwidth estimation method which not only takes into account of the traffic of neighbour stations but also of the traffic itself.

3.2 Multi-channel network with unmodified IEEE 802.11 MAC

Many researchers focus on utilizing off-the-shelf IEEE 802.11 interface cards in multi-channel wireless networks. This method does not change the mechanism of the IEEE 802.11 protocol, so it is easily implemented. Some other researchers introduce new mechanisms into the packets transmission process. This method has a good performance according to simulation results. However, there is no hardware to support these algorithms. In this section, we will introduce the algorithms using multi-channel with an unmodified IEEE 802.11 protocol. In next section, the modified IEEE 802.11 multi-channel protocols will be described.

The current IEEE 802.11 protocol was developed for a single channel. All the stations will contend for the same channel medium. If we want to use multiple channels to increase the capacity of network, it is necessary to change the interface from one channel to another channel. Switching the radio interface from one channel to another

incurs a non-negligible delay. According to [60], the channel switch delay varies from 200 μs to 20 ms; consequently frequent channel-switching may significantly degrade network performance [61].

Depending on the number of channel changes, there are three major ways to assign channels to the nodes: static channel assignment, dynamic channel assignment and hybrid channel assignment.

3.2.1 Static channel assignment and dynamic channel assignment

Static channel assignment assigns channels to the nodes permanently (or at least on a long-term basis). The benefit of this approach is that no further action is required by the network operator after the channel assignment has been performed. However, under certain network load condition, saturation can still occur if there is insufficient capacity available for the stations sharing the same channel. Static assignment strategies are well-suited for use when the interface switching delay is large. In addition, if the number of available interfaces is equal to the number of available channels. The static channel assignment could be well suited. Das et al [62] present four potential metrics can be used in the static channel assignment: 1) Direct maximization of the number of possible simultaneous transmissions in the network; 2) Minimization of the average size of a co-channel interference set; 3) Minimization of the maximum size of any co-channel interference set; 4) Channel diversity which means the difference between the maximum and the minimum number of times that any channel that is used. These four metrics point out the different objectives of the different static channel assignments. Ali [63] proposed a static frequency allocation to maximize capacity that takes into account the dynamic nature of traffic. However it requires that the network operator has a full knowledge of the position of the APs and their transmission range and the spatial traffic distribution. This static channel assignment is not suited for frequently changing traffic loads.

Dynamic channel assignment assigns channels to nodes adaptively according to the traffic load and topology of the nodes. When a node’s traffic load changes and

congestion occurs, it is possible to reassign the channels in order to reduce the incidence of congestion. Most of these dynamic channel assignment algorithms focus on minimizing the interference of the whole network to improve the throughput. In [2], the authors develop a dynamic channel assignment which minimizes the overlapping interference between APs. Each AP will periodically run the channel assignment algorithm to measure the interference and select its own channel which has the lowest interference. However, the way they calculate the overlapping channel interference is rather simplistic as they only consider the overlapping factor and do not consider the traffic on the two channels. Due to the requirement of multiple layer information, cross-layer algorithms could combine channel assignment with routing to minimize the interference.

R-CA (Routing based Channel Assignment algorithm) [65] assigns channels to nodes based on the routing decision. However, because R-CA will wait for an available channel when there is no available channel, this could generate a serious congestion when the nodes have heavy traffic loads or when a large number of nodes is waiting to transmit packets.

Yang et al. [66] propose a distributed collaborative sensing scheme to reduce sensing overhead and energy consumption which could be implemented using conventional IEEE 802.11 hardware with a single radio interface. The interface will select the channel with highest available bandwidth through measurement. However, this mechanism only considers the busy time of the interference when it estimates the available bandwidth and the time used to access the medium is not included.

Hybrid channel assignment methods contain both static and dynamic approaches and are primarily used in multi-radio multi-channel networks. In these networks the control channels are statically assigned and the data channels are dynamically assigned. In [61], a MAC layer module named Interface Management Module (IMM) under a hybrid channel assignment is presented. The IMM manages the multi-channel radio interface based on scheduling algorithms. The MCMR (Multi-Channel Multi-Radio) mesh node fixed interface is primarily used for receiving data from neighbours while its dynamic

interface is dedicated to transmitting data to its neighbouring nodes. The switchable interface is tuned to a channel which may be changed at any time. Thus, if two mesh nodes need to communicate for exchanging data, the switchable interface of the sender node and the fixed interface of the receiver node must be tuned to the same channel. If not, the sender node's switchable interface switches on the channel on which the receiver node fixed interface is tuned. Radio interface coordination for channel switching is handled by CSP (Channel Switch Protocol). However, the disadvantage of this method is it does not consider the capacity of the selected channel. This could result in congestion for all the stations using the channel.

3.2.2 Centralized channel assignment

Depending on whether or not a central node is used to manage the network, the channel assignment algorithms can be divided into distributed channel assignment and centralized channel assignment methods.

Centralized channel assignments typically examine the traffic pattern and the capacity requirements and then make the decision of routing and channel assignments. There usually exists one or more special nodes which could get information from the nodes of the whole network and have the responsibility of assigning channels to all the nodes.

The centralized channel assignment can decrease the interference and improve the aggregate throughput. However, with an increase in the scale of network, it becomes increasingly difficult to gather all the necessary information.

In [67], the channel assignment algorithm finds the appropriate channel and tries to increase the throughput by maximizing the Signal-to-Interference Ratio (SIR) at the user level but not by minimizing the interference. All the users send the SIR information to the associated AP and these APs in turn transmit the information to the central unit (i.e. a wireless or wired server). The central unit will run the channel assignment algorithm to determine which channel is the best for each AP. It requires tight time synchronization between all the APs which is different to achieve in a large scale network.

Ashish etc [3] developed a set of centralized channel assignment, bandwidth allocation, and routing algorithms for multi-channel wireless mesh networks. The first algorithm *Neighbour Partitioning Scheme* performs channel assignment based only on network topology. The second algorithm *Load-Aware Channel Assignment* realises the full potential of proposed architecture by further exploiting traffic load information. Even with the use of just two Network Interface Cards (NICs) per node, the two algorithms improve the network goodput by factors of up to 3 and 8 respectively. This method requires a long iteration time to reduce the difference between the link capacities and their expected load.

MesTiC [68] is a static, centralized channel assignment scheme based on a ranking function that takes into account traffic, number of hops from the gateway and the number of interfaces per node. The link with heavy traffic load, close to the gateway and small number of radios will be assigned to the channel with the least-interference. However, this mechanism doesn't consider the strategy when there is no channel that can satisfy the required load. It also does not consider the ability to recover from failure.

MeshChop (Mesh Channel Hopping) [64] introduced the concept of connected components which are the interface pairs which share the same channel. A centralized channel assignment is used to assign the channel based on capacity requirement. If one of these interfaces changes to another channel, all the other interfaces in the same component will change to the same channel. This method could improve the throughput of a mesh network because it could decrease the interference from co-located wireless networks. The disadvantage of MeshChop is that it doesn't probe the available channel but selects the channel randomly. The disadvantage is that this method cannot find a channel which could satisfy all the traffic load of the component every time.

The centralized channel assignment methods try to improve the throughput by minimizing the interference of the whole network. When the scale of the network increases, it is difficult to get the interference information of all the stations in the network. Another problem of centralized channel assignment methods is the ability to recovery from station failure. The cost of station failure can be high because it might

need to change all the channels of the interface to justify only one failed station. This is the reason why many researchers focus on the distributed channel assignment algorithms which will be introduced in the next section.

3.2.3 Distributed channel assignment

When some nodes in the network drop out, the centralized channel assignment may reassign all the nodes' channel to accommodate the new network topology which could generate much unnecessary channel change costs. The distributed channel assignment methods exhibit a greater robustness following the failure of few nodes. The stations running distributed channel assignment methods should have the ability to check if there are any stations that fail to communicate. It also needs to have the ability to collect the channel and station information of the neighbour stations. The major problem of distributed channel assignment method is how to maintain the network connectivity when one station makes the decision to change the channel. Because the number of channels is larger than the number of interfaces, if one of the interfaces decides to change to another channel, the neighbour stations which connect with this interface will change to the same channel in order to maintain the connectivity. However, this could happen again on the third stations because it also needs to maintain connectivity as well. This is called the *ripple effect*. The *ripple effect* will propagate the channel change of one station to the rest of the network. More seriously, the initial station may change the channel again if the ripple effect reacts to itself. Even if the connectivity of the network is not changed, the throughput still decreases because of the channel change delay exist. Ashish etc [69] propose a multi-channel architecture called *Hyacinth* which requires at least two interfaces on each station to maintain network connectivity. It divides the multi-radio into *UP-NICs* and *DOWN-NICs*. Each WMN node is responsible for assigning channels to its *DOWN-NICs*. This method breaks the collision domain in a single-channel network into multiple collision domains each operating on a different channel. It avoids the ripple effect because the channel change of one interface only impacts on its one-hop neighbours. LCA (Local channel information assisted Channel

Assignment) [70] assigns the channel to multiple interfaces using the local information of the channel. It broadcasts the traffic load information to maintain the connectivity which consumes more bandwidth resources. In [71], a distributed channel assignment algorithm is presented. It requires a wired distributed system to communicate between APs and pass the necessary information to the central AP. However, when some of the APs belong to a different owner, this algorithm cannot be implemented. Our proposed channel selection algorithms could be used in the same network architecture and does not require communication between the APs.

In order to resolve the problem of link failure, [13] presents an *autonomous network reconfiguration system* (ARS) which could autonomously recover from local link failure in order to avoid performance degradation. This system can reconfigure the channel, radio and route assignment for real time recovery from link failure. It periodically monitors the channels to detect the link failure. Once a station detects a link failure, the ARS in the detector stations will trigger the formation of a reconfiguration group which is formed with the mesh routers using the faulty channel. One of the mesh routers will be selected as the leader of this group and will have the responsibility to send requests to a gateway and receive the reconfiguration plan which is selected by the gateway. All the members in the group will execute the corresponding configuration changes. This architecture only requires local reconfiguration changes. However, it still needs to find the possible reconfiguration plans in a centralized station (usually the gateway) which means the same problem in the centralized channel assignment: with the increasing scale of network, it is difficult to collect the information of all the stations in the network.

The major problem of distributed channel assignment is the connectivity. The objective of the proposed algorithm is to realise a fast adaptation to changes and an assignment that could be computed on the APs where the algorithm must work in a timely manner at a low computational cost. The algorithm does not always provide the optimal solution but it is fast and requires little resources. The detail of this algorithm will be introduced in Chapter 4.

3.3 Novel multi-channel MAC protocols

The IEEE 802.11 MAC DCF mechanism is designed to share a single channel among multiple users. Because the current IEEE 802.11 devices are half-duplex, it is difficult to develop a multiple channel protocol. In IEEE 802.11, one station can dynamically switch the wireless device between multiple channels, but it can only transmit on one channel at a time. One station could only listen on one channel at a time and it cannot hear the transmission taking place on a different channel. So it is important to design the protocol to utilize multiple channels to realise a higher performance.

Consequently, many multi-channel MAC protocols are designed to exploit the available channels to enhance the overall throughput. Mo et al. [72] compare some multi-channel MAC protocols and divide them into four categories, namely split phase, common hopping, multiple rendezvous and dedicated control channel. We will discuss each of these in the following sections.

3.3.1 Split phase MAC protocol

The split phase MAC protocol splits the time into fixed duration phases, each comprising a control phase and a data phase. In the control phase, all the nodes switch to a default channel and transmit control frames to negotiate the channel with each other. At the end of the control phase, all the nodes will switch to the negotiated channel and start the normal transmission process. These methods need a separate channel to be the contention channel and the rest of the available channel to be data channel. All the nodes which have packets to transmit will send RTS frame and wait CTS frame in contention channel. After that the sender and receiver will turn to one of the available data channels according to the channel scheduling algorithm. The contention channel is also treated as data channel after the contention period MAP (Multichannel Access Protocol) [73]. The major disadvantage of split phase is that it requires time synchronization between all the nodes in the network. It is difficult to maintain accurate time synchronization in a distributed network even though the beacon frames could be

used to provide for a rough synchronization. MMAC (Multi-channel MAC) [74] assumes that all the channels have the same bandwidth which is physically unrealisable. Another disadvantage is that the control channel could be the bottleneck of the whole network when the traffic load approaches saturation. Because of the separation of contention and transmission on two channels, the average packet delay could increase because all the packets need to wait while the nodes negotiate with each other.

3.3.2 Common hopping MAC protocol

The channel hopping methods allows all the nodes in the network to hop between channels in a common hopping sequence. If two of them decide to communicate with each other, they need to build the control packet handshake on the same channel. After the handshake, they can communicate with each other. During this time, the rest of the network nodes will keep hopping on the common hopping sequence. When the transmission is finished, the sender and receiver need to re-synchronize to the current common channel hop. CHMA (Channel-Hopping Multiple Access) [75] developed a common hopping MAC protocol which could avoid collision by hopping all the nodes not able to exchange data to the next frequency hop. So it does not require carrier sensing or the assignment of unique codes. CHAT (Channel hopping access with Trains) [76] enhances the control handshake of CHMA. It avoids contention by considering the broadcast traffic. The broadcast packets include a special vector which includes the address of receiver and also a sequence number. The major advantage of channel hopping protocols is it only requires one radio per device and all the channel resources can be used to exchange data. The disadvantage is it requires tight synchronization between all the stations in the network.

3.3.3 Multiple rendezvous MAC protocol

Multiple rendezvous MAC protocols utilize multiple device pairs to parallel transmit packets with different channels. However, it requires special coordination to ensure that the device pairs can communicate on the same channel. SSCH (Slotted Seeded Channel

Hopping) [77] utilizes multiple channels with a mechanism called *optimistic synchronization*. In the network, all of the nodes have their own channel hopping sequence. If the sender node knows the channel sequence of receiver, it could switch channel directly to the desired channel. If the sender does not know the channel sequence of receiver node or the information about the receiver node is out of date, it waits a pre-set duration. Because all the nodes hop through the channels, the sender and receiver pair could set up the communication process when the receiver hops to the channel of sender in the pre-set duration. In the multi-hop network, SSCH utilize partial synchronization to maintain the connectivity. The intermediate node maintains synchronization with the sender for half of the time and in the other half of the time it will maintain synchronization with the receiver node. Another example of multiple rendezvous protocol is McMAC (Multi-Channel MAC) [78]. It assumes all the nodes are within one hop of each other. It modifies the packets to include the seed of its own hopping sequence and also the time it will remain on current channel. All receivers will know the neighbours on the same channel. The sender and receiver pairs could negotiate with each other to change to another channel simultaneously.

This protocol overcomes the disadvantage of having a single control channel. Because all control frame handshaking could simultaneously be achieved on multiple channels, it is not necessary to have a control channel. However, it is more complex than the dedicated control channel MAC protocol.

3.3.4 Dedicated control channel MAC protocol

The dedicated control MAC protocol divides the whole frequency band into a single control channel and multiple data channels. Each data channel is assumed to be identical and has the same bandwidth resources. It implements two radios on each device. One of the radios is assigned to the control channel permanently. The other radio is dynamically assigned to the data channels according to the negotiation of the first radio on the control channel. If a node wants to transmit packet to the receiver, it will send a RTS frame and also the channel it wants to use on the control channel. The receiver will send

back a CTS frame if it is possible to receive on the channel the sender required. After these control handshake process have finished, both the sender and receiver will switch the second radio to the negotiated channel [79]. CO-MMAC (Connection-Oriented Multi-channel MAC) [80] utilizes the same mechanism to divide the frequency band into a single control channel and multiple data channels. The difference between DCA and CO-MMAC is the control information exchange on the control channel. DCA utilizes channel usage and a free channel list to assign the channel. However, CO-MMAC combines channel status and neighbour status to select the channel. The dedicated control channel MAC protocols do not need any synchronization between the nodes. However, it requires two or more interfaces per node.

There are still some other multi-channel MAC protocols that do not belong to the above four categories. For example, ODC (On-Demand Switching) [81] is a broadcast based multi-channel MAC protocol. It only requires one radio per device to utilize multiple channels. The nodes will estimate the channel information and change its channel when its traffic share drops below a threshold to prevent unnecessary channel switch. When a node decides to switch channel, it will broadcast its departure from the old channel and its arrival at the new channel.

3.4 Compound channel assignment method

There are some channel assignment algorithms that consider not only the channel information but also the information of other parameters such as the transmission rate, the routing protocol and the action of other stations. In this section, we describe the multi-rate involved channel assignment and also the game theory based channel assignment algorithm. The joint routing and channel assignment algorithms treat the channel selection and traffic distribution together to satisfy the traffic load of all the stations.

3.4.1 Multi-rate involved channel assignment method

In a multi-rate IEEE 802.11 network, transmission over a lower rate link consumes

more time and bandwidth resources as compared with a higher rate link. This generates a *performance anomaly* where the throughput of all stations transmitting at the higher PHY transmission rate is degraded to the level of the lower PHY transmission rate stations [82]. The rate selected by different nodes could be different because of different RSSI on multiple channels. It is important to consider multiple rates when we utilize multiple channels. MesDRCA (Mesh-based Data Rate-Aware Channel Assignment) [83] and DR-CA (Data Rate Adaptive Channel Assignment) [84] assigns the links having identical or comparable data rate with a common channel in order to minimize the usage of channel resources. Because the higher rate links are isolated with lower rate links, it prevents the *performance anomaly* problem. RB-CA (Rate-Based Channel Assignment) [85] separates the higher rate links and lower rate links by considering about the traffic load and transmission rate of possible links. In the network layer, the routing protocol also selects paths based on the transmission rate. Such as the case of a two-hop path in which two links have a transmission rate of 11 Mbps can reach a higher end-to-end throughput than a one hop path which have a lower transmission rate of 2 Mbps. MRMC (Multi-Rate Multi-Channel protocol) [86] utilizes multiple channels and multiple rates in an infrastructure network. The high rate stations will compete with each other on one channel and the low rate stations will compete with each other on another channel. However, this protocol does not solve the problem of how to improve the throughput when one channel is congested and other channels are free to transmit.

3.4.2 Game theory based channel assignment method

Game theory has been used extensively to model strategic decision-making in economics, political science, and other social sciences. Recently, game theory has also been applied to adaptive channel allocation [87] and to access control in single Aloha-like networks [88] [89] [90]. The wireless station is viewed as the player and the choice of channels is viewed as the strategy. The utility functions can account for different metrics such as the interference that a station suffers and the interference other stations will suffer when this station make the choice of channel [87]. A distributed

channel assignment has been modelled with game theory in [91] which is adaptive to the external interference. The utility function refers to a loss function which not only depends on the available bandwidth of the selected channel but also on the switch delay. All the players are selfish in order to occupy the best channel which suffers the least external interference and it is not shared by many neighbouring nodes of the same network. Because there are many constraints when the game theory based channel assignment algorithm is implemented such as the limit capacity of each channel, the limited transmission range of wireless radio, the game theory based channel assignment needs to put all these constraints into the game formulation. However this will increase the complexity of the algorithm.

3.4.3 Joint routing and channel assignment method

To increase the capacity of wireless network, multiple radios were implemented on one device and each radio was assigned to a distinct non-overlapping frequency channel. However, multi-radios create several research challenges. The problem of optimally assigning channels in an arbitrary mesh topology has been proven to be NP-hard [3]. A fundamental problem is the joint channel assignment (CA) and routing problem. Routing selects effective paths for the traffic flow from source nodes to destination nodes, while CA determines the right frequency channel that a radio interface should use. On one hand, CA determines the network connectivity between devices since two radios in each other's transmission range can communicate with each other only when they are assigned a common channel, and this means that CA determines the network topology. It has an impact on link bandwidth. This clearly impacts the routing used to satisfy traffic demands. On the other hand, routing will determine the traffic flow of each link of the whole network. It has an impact on the traffic requirement of each node and also the CA since CA should be dynamically adjusted according to the traffic status. Therefore, routing and CA are tightly coupled [92]. They need to be jointly optimized to achieve the best performance. This is the so-called routing and channel assignment (RCA) problem which is known to be NP-hard [3]. JRCAP (Joint Routing and Channel

Assignment Protocol) [93] defines a density based clustering algorithm for channel allocation which partitions the mesh network into balanced clusters and assigns a fixed and static channel to each cluster. All the nodes will broadcast its density and the node with highest density will be the head of the cluster.

The way to solve this problem would be to consider routing and channel assignment separately. The routing will discover the route for the traffic demand and obtain the load estimation of each link. Then the channel is assigned to the links based on the load estimation. However, this separate method is still not optimal because even with given traffic load, the CA problem is still NP-hard [3]. So the ideal way to maximize the capacity of multi-channel network is to jointly consider routing, channel allocation, interface assignment and network topology [94].

Table 3-2 Algorithms for the Linear Programming Problem

Reference	**Constraint**	**Steps**
[94]	Number of NICs Number of channels Communication range and interference range	i) Logical topology formation ii) Interface assignment iii) Channel allocation iv) Routing
[95]	Inter-path interference Intra-path interference	i) Routing discovery multiple disjoint paths ii) Channel allocation iii) Schedule traffic to reach load balance
[96]	Heterogeneous radio Interference Flow conservation Number of radios Number of channels	i) Solve LP ii) Colouring the schedule graph iii) Schedule the traffic
[97]	Interference Number of radios	i) Solve LP ii) Channel assignment

	Number of channels	iii) Post processing and flow scheduling iv) Interference-free link scheduling
[98]	Transmission flow Resources Interference Dynamic nature of traffic demand	i) Best path for routing the traffic ii) Interference-free channel assignment iii) Efficient schedule for data transmission
[99]	General interference	i) Channel assignment ii) Schedule non-interference links

Many other previous joint routing and channel assignments were modelled as a Linear Programming Problem (LPP) [94] [95] [96] [97] [98] [99]. These algorithms take into account some constraints such as interference, number of channels and number of radios per device. The objective function of such a LPP is to maximize the throughput of network. In order to achieve this target, they need to know *i*) the traffic rate of each link when both sides of the link are assigned a common channel; *ii*) a feasible channel assignment; and *iii*) a feasible interference-free schedule which means there is no simultaneous transmission as happens on the common channel where these links could interfere with each other.

The joint channel assignment method leads to an unclear path for each flow. Flows can be divided into multiple paths. These scenarios can cause difficulty for network management in backbone networks and also incurs a higher computational complexity. There are some ways to decrease the computational complexity of the linear programming problem, but this is beyond the scope of this thesis.

Centralized joint routing and channel assignment algorithms developed their own objective function. CRAFT (Channel and Routing Assignment with Flow Traffic) [100] proposed a distributed, cooperative, computationally efficient and simple to implement algorithm. It jointly optimizes routing and channel assignment by using a properly designed objective function to meet the flow demands of the mesh nodes. The objective

function of CRAFT considers the interference of both the sender and receiver of a link. All the nodes will select the channel which can achieve the maximum the objective function. JRCA (Joint Routing and Channel Assignment protocol) [101] introduces *Route Quality Metric* that utilizes the performance characteristics of data packet transmissions, and effectively captures the effects of intra-flow and inter-flow interference to maximize the probability of success (POS) and minimizes the end-to-end delay of the route. The channel selecting mechanism will select the assignment which has the largest *Route Quality Metric* and routing part will select the route which has the largest *Route Quality Metric*. Layer 2.5 JCAR (Joint Channel Assignment and Routing) [92] introduces the *Channel Cost Metric* (CCM) which is defined as the sum of expected transmission time weighted by the channel utilization over all interfering channels to reflect the interference cost. The word “pattern” was used to denote any specific combined solution of channel selection, interface assignment and routing. Once one of the nodes senses that there is a new pattern with a smaller CCM, it starts to switch channel. The algorithm does not require tight clock synchronization among neighbour nodes and does not need any modification in the current IEEE 802.11 devices and can be applied to other wireless systems such UWB, etc. FCRA (Flow-based Channel and Rate Assignment) [102] aims to minimize the maximum channel utilization. Firstly, it determines the pre-computed flow rates. Secondly, all the possibly channel assignments are found. Finally, it selects the one which has the minimal maximum total utilization. J-CAR (Joint Channel Assignment and Routing protocol) [103] defined a metric called *channel interference index* to select the channel with the least interference. To find the path with the least interference for network load balancing on a global scale, J-CAR employs a length-constrained widest-path routing in which the width means the residual bandwidth.

Distributed joint routing and channel assignment algorithms [104] [105] [106] separate the joint channel assignment and routing problem into two steps: distributed channel assignment and select the best quality route. The aim of JCIAR (Joint Channel allocation, Interface assignment and Routing) [105] is to minimize interference while

satisfying the network connectivity. The routing part of CAR-RECA (joint Channel Assignment and interference-aware QoS Routing algorithms) [104] predefined some disjoint paths under the policy to distribute a commodity fairly on the network. The channel allocation part maintains the topology connectivity combined with interface assignment to improve its ability to scale to large networks. Lin etc. [107] developed a fully distributed algorithm that jointly solves the channel-assignment, packet scheduling and routing problem. Channel assignment is combined with packet schedule to ensure that packets are less likely to be assigned to link-channel pairs that have a smaller capacity.

JRCAP (Joint Routing and Channel Assignment Protocol) [93] utilizes a density based clustering algorithm to balance the traffic demand. It partitions the mesh network into balanced clusters and assigns a fixed and static channel to each cluster. The routing metric MRC (Maximum Residual Capacity) considers channel diversity, data rate and channel load. A separate channel was used between clusters to maintain traffic control.

3.4.4 Traffic load aware channel assignment method

Most of the channel assignment algorithms introduced above focus on how to maximize the throughput or minimize the interference of the network. They aim to build a high capacity high quality network. There are some other researchers who consider the channel assignment to satisfy the traffic load requirement of the individual wireless devices. Rozner et al [116] proposed a static or infrequent channel assignment which considers the traffic pattern of stations and APs. They assume the traffic load requirement of each AP is well known by the centralized controller and the AP with high individual demand is first assigned to non-overlapping channels. However, the network performance is no better than other algorithms with an incomplete traffic information or when the APs cannot communicate with each other to obtain the complete traffic information. LCA [117] proposed a load aware channel assignment algorithm where the AP will select the channel which has the lowest airtime cost which is an approximation of the average per-packet delay. AP and stations will continuously

scan the available channels to obtain the airtime cost value. With an increase in the number of channels or when the traffic load changes dramatically, the process overhead can be unacceptable. In the fifth chapter of this thesis, we propose an autonomous channel selection algorithm which autonomously selects the channel without a central controller in order to reduce the incidence of congestion.

3.5 Conclusion

This chapter introduces other research work in the area of available bandwidth estimation and channel assignment. There are two main methods used to estimate the available bandwidth: Passive methods and active methods. The active methods estimate the capacity of link using the concept of packet dispersion where the dispersion between the packets indicates the capacity of links of both sides of a path. Probe packets are transmitted with different packet sizes and PHY transmission rates to increase the accuracy of estimation.

The next topic presented was the use of channel assignment algorithms to increase the throughput of network by utilizing multiple channels. Depending on whether or not the MAC protocol was modified, multiple channels were used in different ways. Some researchers focus on developing a MAC protocol suited to utilizing multiple channels simultaneously. Some of the protocols separate the channel set into a control channel and several data channels. The control channel is used to maintain the network connectivity and the data channels are used to transmit data frames. Other novel MAC protocols divide the time into small intervals and also separate the control frames and data frames to increase the accuracy of network management. Because these novel MAC protocol methods cannot be directly implemented on the current IEEE 802.11 hardware, some other researchers pay more attention on developing channel assignment algorithms utilizing the standard IEEE 802.11 MAC protocol. These algorithms are divided into three main categories: Static channel assignment, Dynamic channel assignment and Hybrid channel assignment. Static channel assignment combines the channels and radios permanently or for a long time (in the unit of hours or days).

Dynamic channel assignment assigns the channels to the radios depending on the traffic load and channel status. When the current operating channel is highly contended or it cannot satisfy the traffic requirement, it will switch to another available channel. Hybrid channel assignment combines the static and dynamic features together and has been utilized in multi-radio networks. Some of the radios were statically assigned channels to keep the network connectivity and other radios were assigned channels dynamically to suit a dynamic traffic load.

There are also other channel assignment algorithms which combine metrics from other layers such as the transmit rate from PHY layer and routing from the network layer. Game theory based channel assignment algorithms are usually utilized in cognitive networks. All the radios were considered as players and the action of channel selection is defined as the strategies. Minimizing the interference or maximizing the channel utilization is the utility function in these algorithms. However, these algorithms still have many constraints that prevent their implementation in the real networks.

All the previous channel assignment algorithms focus on how to arrange the channel for the wireless radio to maximize the throughput or minimize the interference. Once any of the condition changes (such as a change in the traffic load or a new station joins) the algorithm will be triggered to reassign the channel. The centralized channel assignment has a low efficiency in a large scale network. The distributed channel assignment always changes to the channel with lowest-interference or the channel with largest available bandwidth without considering this channel will satisfy the traffic load or not.

In next chapter, we will introduce a novel available bandwidth estimation which based on passive bandwidth estimation. It collects packet information for all the neighbours and calculates the access efficiency and traffic load of these stations. A dynamic channel selection algorithm is used to select the channel with the help of bandwidth estimation algorithm. This channel selection algorithm operates in distributed manner and all the stations maintain this channel until this channel cannot satisfy the traffic load. If there is no channel that can satisfy the traffic load, it utilizes a “neighbour forcing” method to rearrange the channel of the neighbour stations in order to satisfy the traffic load of all

the stations.

Chapter 4 Passive bandwidth estimation based on access efficiency

The first section of this chapter introduces a new passive available bandwidth estimation method based on the access efficiency. It considers not only the access efficiency of the estimating station itself but also the access efficiency of the other stations with which it shares the same channel. In the second section, an experimental testbed is configured to validate the performance of the algorithm. The results show that this available bandwidth estimation method can be used to predict the congestion status of a given channel assignments.

4.1 Available bandwidth estimation method

This section describes a new passive available bandwidth estimation method used to predict the congestion status of stations under a given channel assignment. It utilizes the concept of access efficiency which is a measure of the ability of a station to access the channel medium. With the available MAC bandwidth information, the station can predict whether it will be successful in joining a new channel, i.e. it determines whether a station will become congested as a result of the channel switch.

4.1.1 Motivation

In section 2.6, we described the MAC bandwidth components framework for analysing the packet transmission process in IEEE 802.11 WLAN networks. *BWbusy, BWaccess, BWidle* and *BWfree* are four bandwidth components that serve to describe the bandwidth utilization on the medium. The concept of access efficiency connects these four components together [33]. This metric indicates the efficiency (in terms of the bandwidth required) of a station in accessing the channel medium. The larger the access efficiency, the more efficiently the station can access the medium. Because stations with

different access efficiencies will have different available bandwidths, when a station contends with its neighbour station for access, the access efficiency of all the stations needs to be considered while estimating the available bandwidth.

In section 3.1, we defined the available bandwidth as the maximum MAC layer traffic load of a station that can be transmitted on a channel without causing saturation either to itself or to other stations which share the same channel.

Therefore, we define saturation and congestion as follows:

Definition 4.1: Saturation. *Saturation occurs when the free bandwidth* BW_{free} *of a station equals zero, i.e. when* $BW_{free} = 0$.

Definition 4.2: Congestion. *Congestion occurs when the number of packets* N_{in} *arriving at the queue per second is larger than the number of packets* N_{out} *transmitted per second and when this condition persists for more than* $T_{confirm} = 10$ *seconds.*

To avoid a false indication of the congestion status, a small value of $T_{confirm}$ should be not selected. On the other hand, a large value of $T_{confirm}$ will cause high packet loss if a prolonged period of congestion occur before detection. However, the issue of how to select the optimal value of $T_{confirm}$ is beyond the scope of this thesis. During the experiment, we found that 10 seconds represented a good trade-off between avoiding false indication of congestion and minimizing the packet loss.

According to these definitions 4.1 and 4.2, if a station can win more transmission opportunities than the number of packets it wants to transmit, there are no packets that need to be stored in the transmit queue. On the other hand, if a station cannot win a sufficient number of transmission opportunities to satisfy the packets that are arriving into the transmit queue, the packets which cannot be transmitted will be stored in the transmit queue. The depth of the transmit queue will increase until it reaches its capacity. This represents a congestion condition as it leads to a large packet delay and

catastrophic packet losses. However, in this thesis, we use these two terms interchangeably to refer the congestion status.

If a channel has only one station, the relationship between the four MAC bandwidth components is

$$BW_{busy} + BW_{idle} = 1 \tag{4-1}$$

$$BW_{free} = BW_{idle} - BW_{access} \tag{4-2}$$

For a channel with multiple stations, the relationship between the four bandwidth components is more complex. The reason is that the time used to gain access the medium is shared between the contending stations.

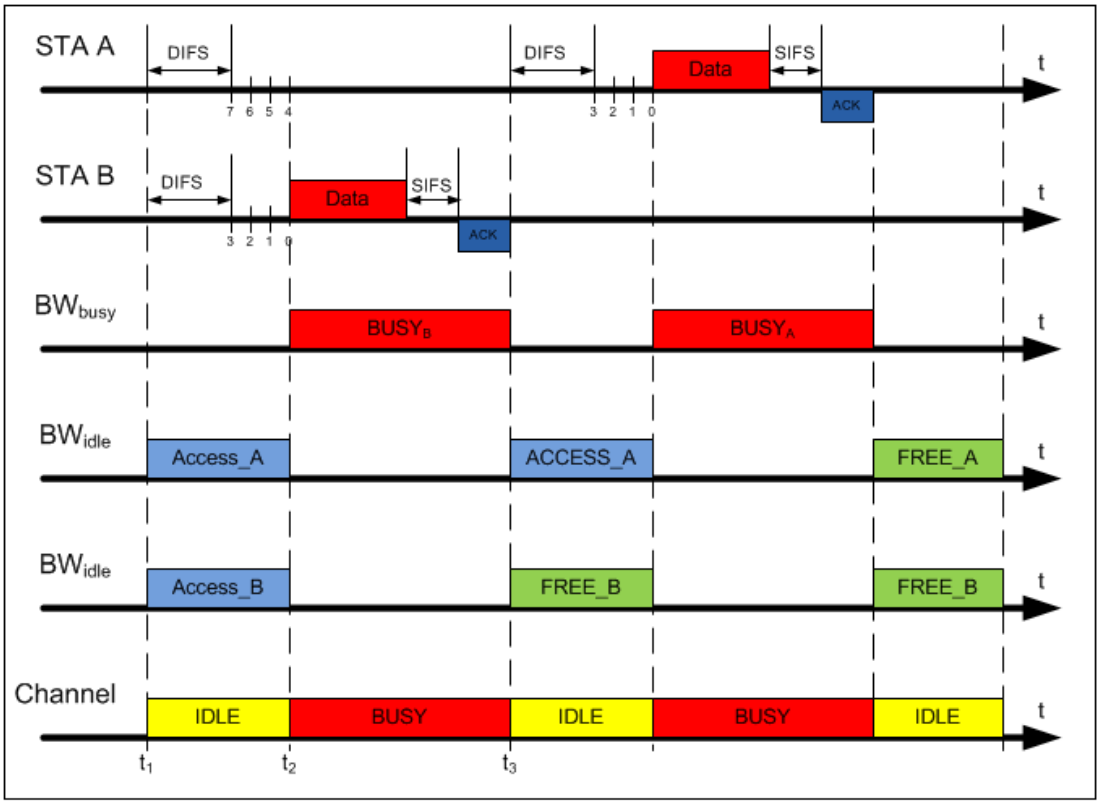


Figure 4-1 MAC bandwidth components where two stations contend for the medium

For example, in Figure 4-1, *Station A* and *Station B* both want to transmit a packet at the same time t_1. However, *Station A* picks a random number of 8 for its back-off counter's initial value and *Station B* picks a random number of 4 for its back-off counter's initial value. *Station B* will transmit its packets when its back-off counter has decremented to zero at time t_2. The back-off process of *Station A* is halted when it finds that the channel

is busy, due to *Station B* commencing its transmission at time t_2. This back-off process of *Station A* will continue once it senses that the medium is idle again at time t_3. Examining the time used to access the medium *Access_A* and *Access_B* shows that it is shared between the stations. Based on this observation, the relationship between BW_{busy}, BW_{access} and BW_{free} of the multiple stations sharing the same channel is:

For Station A:

$$BW_{free}^{A} = BW_{idle} - BW_{access}^{A} \tag{4-3}$$

For Station B:

$$BW_{free}^{B} = BW_{idle} - BW_{access}^{B} \tag{4-4}$$

Because the access bandwidth of these two stations is different, the free bandwidth of the two stations is also different. When one of the stations increases its traffic load, the station with smaller free bandwidth will become saturated earlier. When a station wants to estimate the available bandwidth, it needs to take into account the access bandwidth of all the stations in order to avoid congestion across the network. We will introduce the passive available bandwidth estimation technique in the next two sections based on the order in which stations become saturated when the traffic load equals the available bandwidth.

When the throughput of a station is equal to the available bandwidth, there are three possible scenarios to be considered here: (i) the access bandwidth requirement of the monitoring station is greater than the maximum access bandwidth requirement of other stations which use the same channel; (ii) the access bandwidth requirement of the monitoring station is less than the maximum access bandwidth requirement of other stations which use the same channel and (iii) the access bandwidth requirement of the monitoring station equal to the maximum access bandwidth requirement of other stations which use the same channel. Based of this, we will analysis how to calculate the available bandwidth in the next three sections.

4.1.2 Available bandwidth estimation-scenario 1

In the first scenario, because when the throughput of the estimation station equals the available bandwidth and the access bandwidth requirement of the monitoring station is greater than the minimum access bandwidth requirement of other stations, the monitoring station becomes saturated and other stations are not saturated. We refer to the station which is monitoring the channel as the *estimation station*.

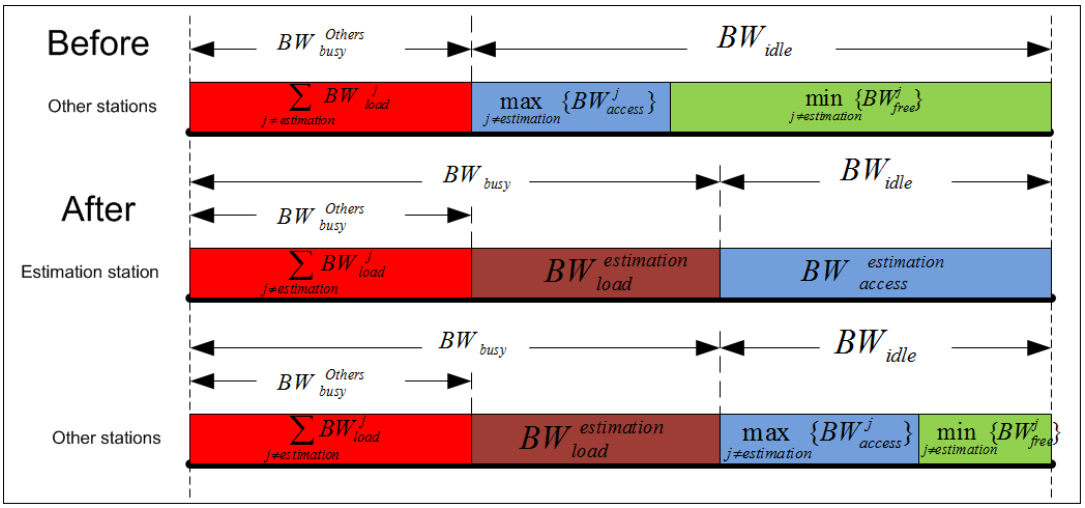


Figure 4-2 Bandwidth components during bandwidth estimation (scenario 1)

Before the estimation station transmits packets on the channel, all the other stations contend for access to the medium and the idle bandwidth BW_{idle} is greater than zero as shown in Figure 4-2. A monitoring program is running to capture all the packets. The busy bandwidth BW_{busy}^{Others} and free bandwidth of each other station BW_{free}^{j} are calculated through formulas (2-4) and (2-8).

The available bandwidth is the maximum MAC layer throughput that it could transmit without any stations becoming congested. As shown in Figure 4-2, because the *estimation station* has a larger access bandwidth requirement than any other stations when the throughput of estimation equals the available bandwidth, it will become saturated before any other stations. When the free bandwidth of the *estimation station* equals zero, the minimum free bandwidth of all other stations $\min_{j \neq estimation}(BW_{free}^{j})$ is still greater than zero which means that the other stations do not experience saturation.

When the *estimation station* becomes saturated, the bandwidth components relationship is:

$$BW_{access}^{estimation} + BW_{busy}^{Others} + BW_{load}^{estimation} = 1 \tag{4-5}$$

Considering the access efficiency of the *estimation station* $ACE^{estimation}$ which can be calculated through formula (2-12):

$$\frac{BW_{load}^{estimation}}{ACE^{estimation}} + BW_{busy}^{Others} + BW_{load}^{estimation} = 1 \tag{4-6}$$

The load bandwidth of the *estimation station* when it is saturated is:

$$BW_{load}^{estimation} = \frac{1 - BW_{busy}^{Others}}{1 + \frac{1}{ACE^{estimation}}} = \frac{ACE^{estimation}}{1 + ACE^{estimation}} \times (1 - BW_{busy}^{Others}) \tag{4-7}$$

Here $BW_{load}^{estimation}$ represents the throughput of the e*stimation station* and $BW_{access}^{estimation}$ represents the access bandwidth requirement of the *estimation station* when the free bandwidth of the *estimation station* equals zero. Here $ACE^{estimation}$ is the access efficiency of the *estimation station.*

The available bandwidth of the *estimation station* is:

$$Available_bandwidth = \frac{ACE^{estimation}}{1 + ACE^{estimation}} \times (1 - BW_{busy}^{Others}) \tag{4-8}$$

4.1.3 Available bandwidth estimation-scenario 2

The second scenario for the available bandwidth estimation is where the access bandwidth requirement of the estimation station is less than the minimum access bandwidth requirement of other stations. The station which has the minimum access bandwidth requirement becomes saturated when the throughput of the *estimation station* equal to the available bandwidth.

As shown in Figure 4-3, before the *estimation station* transmits packets on the channel, all the other stations contend for access to the medium. The idle bandwidth BW_{idle} is greater than zero.

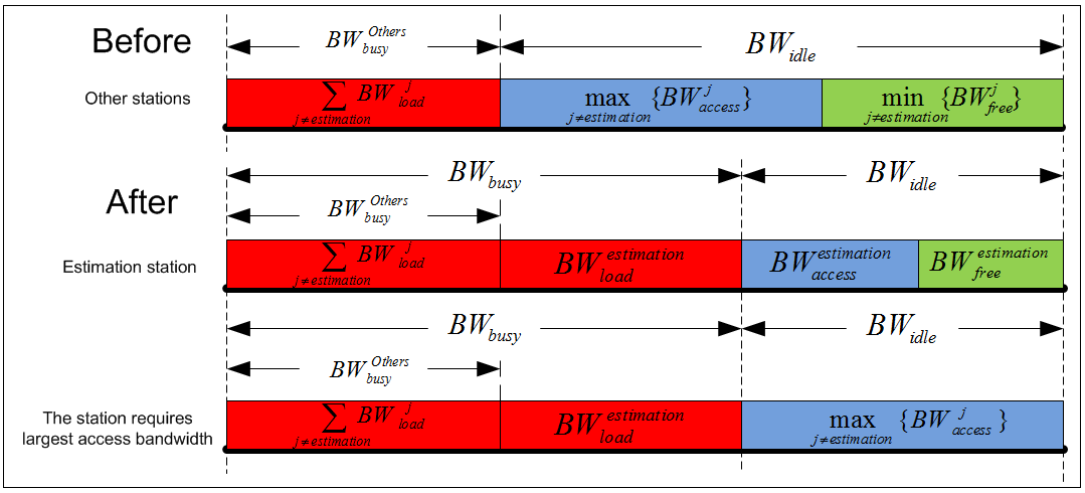


Figure 4-3 Bandwidth components during bandwidth estimation (scenario 2)

When the *estimation station* starts to transmit its traffic load on this channel, because the access bandwidth requirement of the *estimation station* is less than the maximum access bandwidth requirement of other stations, the free bandwidth of the *estimation station* is still greater than zero when one of the other stations becomes saturated.

Comparing the first bar and the third bar in Figure 4-3, the available bandwidth can be calculated as:

$$Available_bandwidth = BW_{load}^{estimation} = \min_{j \neq estimation} \left\{ BW_{free}^{j} \right\} \tag{4-9}$$

Here, $\min_{j \neq estimation} \left\{ BW_{free}^{i} \right\}$ represents the minimum free bandwidth of all the other stations which are operating on the channel.

The most challenging aspect in estimating the available bandwidth in scenario 2 is to establish the minimum free bandwidth as it needs to calculate and compare all the free bandwidth components of other stations one by one.

4.1.4 Available bandwidth estimation-scenario 3

The third scenario for the available bandwidth estimation is wheret the access bandwidth requirement of the estimation station is equal to the minimum access bandwidth requirement of other stations. When the throughput of the *estimation station* equals that of the available bandwidth, the estimation station and the station with the minimum access bandwidth requirement become saturated at the same time.

As shown in Figure 4-4, before the *estimation station* transmits packets on the channel, all the stations contend for access to the medium. The idle bandwidth BW_{idle} is greater than zero.

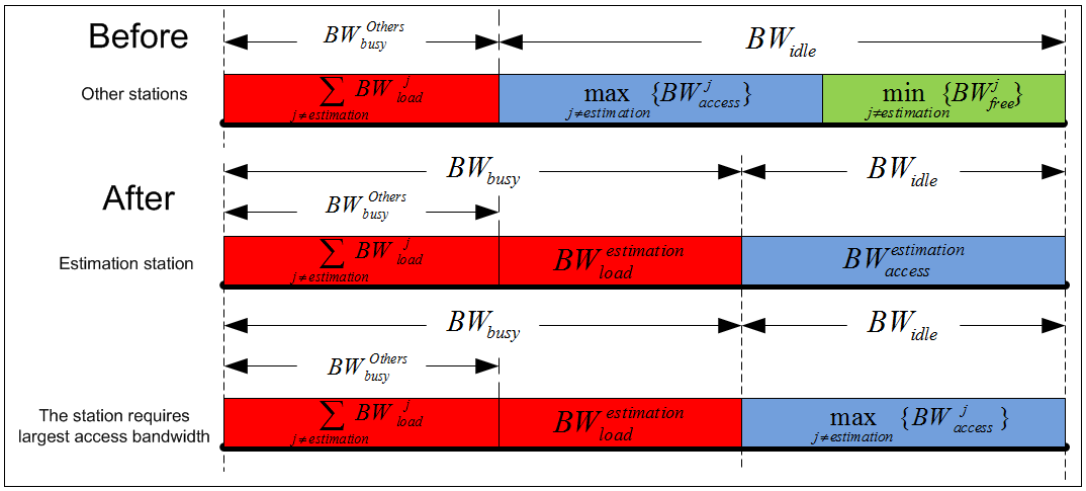


Figure 4-4 Bandwidth components during bandwidth estimation (scenario 3)

When the *estimation station* starts to transmit packets on this channel, because the access bandwidth requirement of the *estimation station* equals to the maximum access bandwidth requirement of other stations, the free bandwidth of the *estimation station* and one of the other stations will be zero and both of these two stations become saturated.

Comparing the first bar and the third bar in Figure 4-4, the available bandwidth can be calculated use the same formula (4-9). So during the experiment, we only present the first two scenarios.

4.1.5 Experimental testbed setup

A 5 station experimental testbed was configured to validate this new passive bandwidth estimation method. In this experimental testbed, two stations are implemented as traffic *Senders*, the other two stations are implemented as traffic *Receivers*.

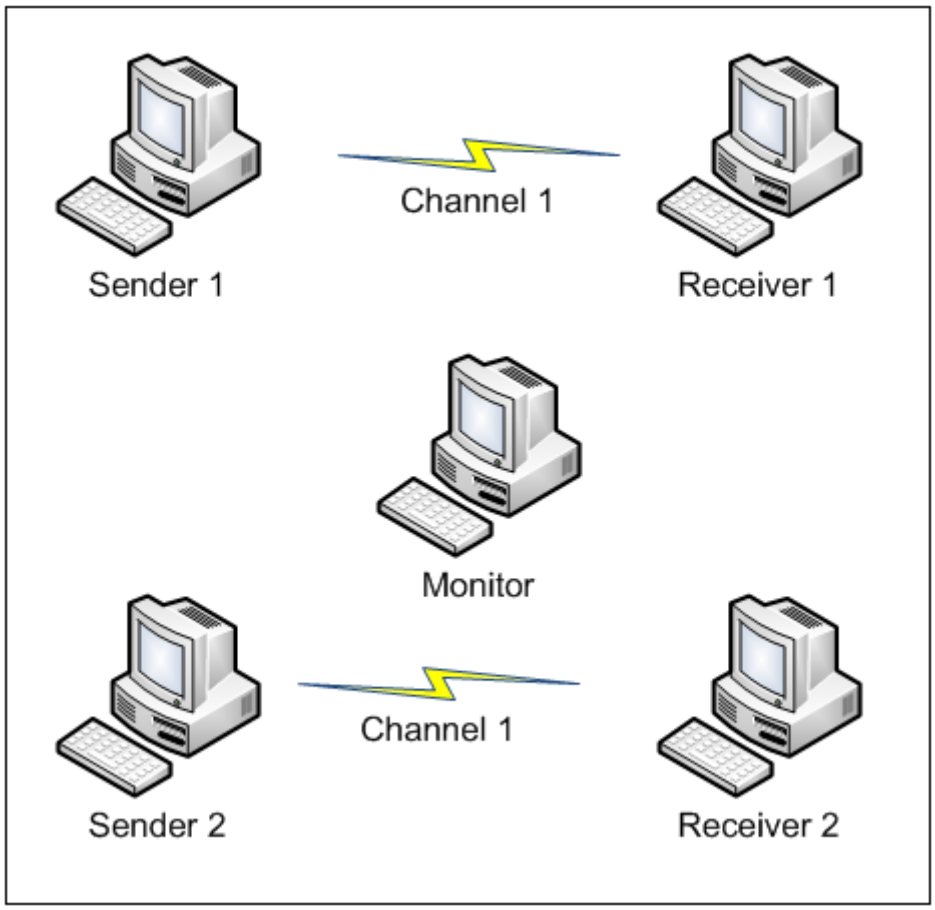


Figure 4-5 Testbed used for bandwidth estimation

As shown in Figure 4-5, *Sender 1* sends traffic to *Receiver 1* and *Sender 2* wants to transmit packets to *Receiver 2*. Before *Sender 2* transmits packets, it needs to estimate the available bandwidth of the channel. In order to demonstrate the performance of the bandwidth estimation method, a *monitor* PC is used to capture all the traffic loads from these four stations for analysis. The results from this testbed are presented in next sections.

4.2 Analysis and results of the passive available bandwidth estimation

In this section, the results of the passive available bandwidth estimation method are presented. We use the testbed shown in Figure 4-5 to investigate the difference between the actual maximum throughput and the estimated available bandwidth. Based on which station becomes congested first, we illustrate the performance in the next two sections.

4.2.1 Monitoring station becomes congested first

The *Sender 1* in Figure 4-5 transmits packets to *Receiver 1* at a constant packet rate of 500 pps. The PHY transmission rate is 12 Mbps and the packet size is 440 Bytes. The operating channel is 36. *Sender 2* wants to transmit packets to *Receiver 2*. It needs to monitor the available bandwidth of channel 36 before it transmits its packets. The monitoring program is running to capture all the packets transmit on that channel. The packet length and PHY transmission rate can be read from the header of each packet. The time used to transmit each packet is calculated through formula (2-9). The load bandwidth of each station BW_{load}^{k} is the bandwidth used by a station *k* when it transmits its packets. In this experiment, the offered traffic load of *Sender 1* is 1.75 Mbps, the access efficiency of *Sender 1* is 3.54 which is calculated through formula (2-12). Here because there is only one other station, $BW_{busy}^{Others} = BW_{load}^{1} = 0.17$ through the monitoring module of *Sender 1*. According to formula (4-8), the normalized available bandwidth of *Sender 2* is $Availableb_Bandwidth = \frac{3.54}{1+3.54} \times (1-0.17) = 0.6472$. Because the PHY transmission rate is 12Mbps, the estimated available bandwidth is $0.6472 \times 12Mbps = 7.7662Mbps$ which is the red line be shown in Figure 4-6. To validate the accuracy of the available bandwidth estimation method, *Sender 2* transmits packets to *Receiver 2* with a different packet rate. The PHY transmission rate is 12 Mbps and the packet size is also 440 Bytes. The packet rate of *Sender 2* is increased by 100 pps every 50 seconds.

Figure 4-6 shows the traffic loads of the two stations when *Sender 2* increases its offered traffic load. Both stations can satisfy its offered traffic load when $t < 450$ seconds, because the throughput of *Sender 2* is less than the estimated available bandwidth. However, when $t = 450$ seconds, the offered traffic load of *Sender 2* is greater than the estimated available bandwidth, it becomes saturated and it cannot win any more transmission opportunities, i.e. there is no further increase in its actual traffic load. This means that the maximum throughput of *Sender 2* on channel 36 is about 7.76

Mbps. The result shows that the passive bandwidth estimation can accurately estimate the available bandwidth when the monitoring station is the first station to be congested.

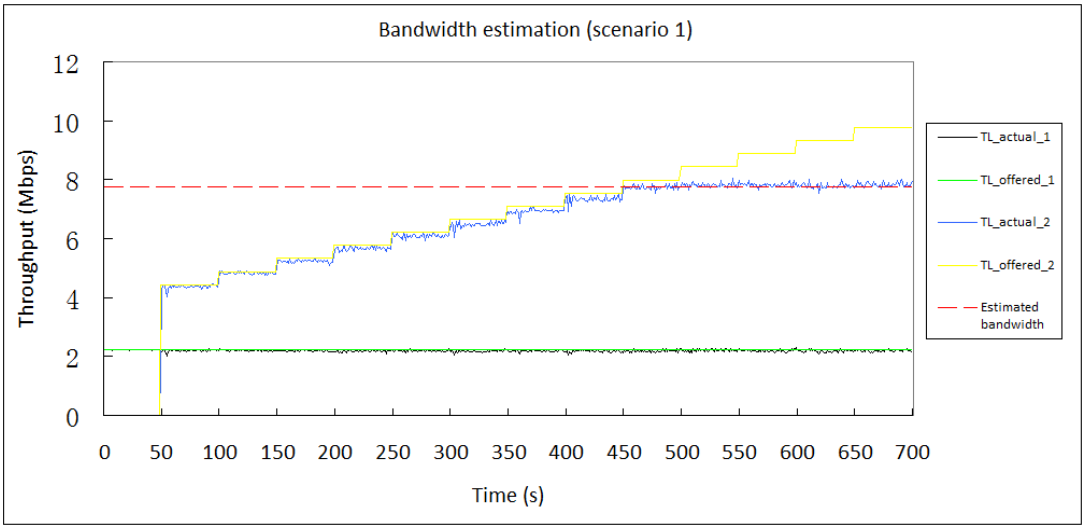


Figure 4-6 Available bandwidth and throughput of the two stations (Scenario 1)

4.2.2 One of the neighbour stations becomes congested first

In another scenario, *Sender 1* transmits packets with a higher traffic load. *Sender 1* transmits packets to *Receiver 1* with a packet rate of 700 pps. The PHY transmission rate is 12 Mbps and the packet size is 1200 Bytes. The access efficiency is 8.72. According to the *monitoring* module of *Sender 2*, the load bandwidth $BW_{load}^{1} = 0.6$. Because there is only *Sender 1* transmit packets, the busy bandwidth $BW_{busy}^{Others} = 0.6$, the minimum free bandwidth calculated with formula (2-8) is $1 - 0.6 - \frac{0.6}{8.72} = 0.3311$. *Sender 2* transmits packets to *Receiver 2* and the PHY transmission rate is 12 Mbps and packet size is 1200 Bytes. According to formula (4-9), the normalized available bandwidth of *Sender 2* on channel 36 is 0.318. Because the PHY transmission rate is 12 Mbps, the estimated available bandwidth is $12 * 0.3311 = 3.9732$ Mbps which is the read link is shown in Figure 4-7.

To validate the performance of the bandwidth estimation, the packet rate of *Sender 2* is increased by 50 pps every 50 seconds. When *Sender 2* does not transmit any packets, only *Sender 1* transmits packets on the channel, the actual traffic load equals its offered

traffic load which is about 7 Mbps. Figure 4-7 shows that if the offered traffic load of *Station 2* is greater than the estimated available bandwidth, the other station (i.e. *Sender 1*) cannot win a sufficient number of transmission opportunities to satisfy its offered traffic load when $t > 300$ seconds and consequently it becomes congested.

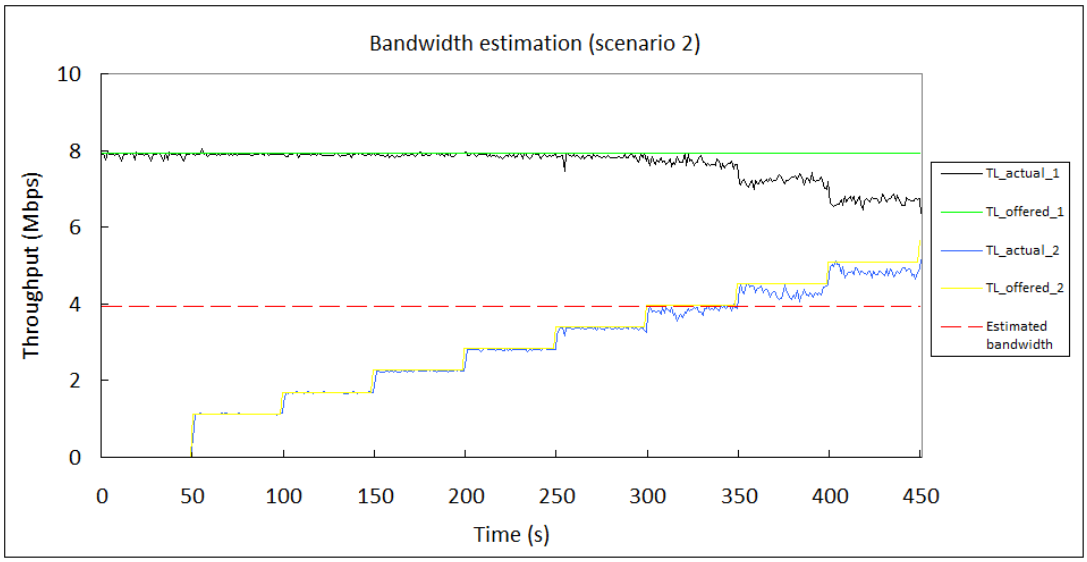


Figure 4-7 Available bandwidth and throughput of the two stations (Scenario 2)

As shown in Figure 4-7, because *Sender 1* transmits at a constant packet rate, the traffic load of *Sender 1* remains constant when the offered traffic load of *Sender 2* is less than the estimated available bandwidth.

When $t = 300$ seconds, the offered traffic load of *Sender 2* exceeds the estimated available bandwidth, we can see that the actual traffic load of *Sender 1* starts to decrease even though the actual traffic load of *Sender 2* still increases, which means *Sender 2* can still win more transmission opportunities even if the *Sender 1* has become congested.

These two results clearly show that this novel passive bandwidth estimation method can accurately estimate the available bandwidth. It does not use probe packets and so it avoids incurring an overhead. This method will be used in the channel selection algorithm to predict the theoretical existence of a successful channel assignment. If we know the traffic loads of all the stations, we can use the same bandwidth estimation formulas to estimate the available bandwidth of all stations. If one of the free bandwidths is less than or equal to zero, then this station will become congested when all the stations start to transmit packets. If the free bandwidth of all the stations on the

available channels is greater than zero, this assignment will be considered to be a successful channel assignment.

4.2.3 Impact of the other factors

There are many factors that have impact on the accuracy of the passive bandwidth estimation algorithm. When there are hidden nodes present it will overestimate the available bandwidth because the monitoring station cannot successfully receive the packets transmit from the hidden nodes. A possible solution is to consider the available bandwidth on the receiver side. The available bandwidth is the smaller of the available bandwidth on the sender and the available bandwidth on the receiver. When the number of active links increases, the accuracy of the available bandwidth estimation method will decrease because the number of retransmission increases. Because the passive available bandwidth estimation algorithm has already considered the packet size and PHY transmission rate, the traffic load with different packet size and PHY transmission rate have little impact on its accuracy.

4.3 Conclusion

In the first section of this chapter, we described a passive method to estimate the available bandwidth. This passive bandwidth estimation method is based on the concept of access efficiency. In the second section, based on which station becomes congested when the traffic load equals the available bandwidth, three scenarios are presented to show the passive available bandwidth estimation algorithm. This method does not transmit probe packets. Instead, it monitors the available channels to obtain the busy bandwidth and free bandwidth of each channel. It also calculates the load bandwidth and access efficiency of each station. The experimental results show that this available bandwidth estimation method correlates well with the actual maximum MAC layer throughput that can be successfully transmitted without causing congestion in any stations. This method is used not only to estimate the available bandwidth but also it could be used to determine the congestion status of a given channel assignment.

Chapter 5 Autonomous channel selection algorithm based on neighbour forcing

In this chapter, an autonomous channel selection algorithm based on neighbour forcing is introduced where all the stations operate in an autonomous manner. The congested station will change the channel once it finds a channel that has sufficient available bandwidth or it will start the neighbour forcing process when there exists at least one theoretical successful channel assignment. In the second section, a simple C++ simulator is described which is used to validate the theoretical feasibility of the proposed channel selection algorithm. The results of the simulation are presented in the third section. In the fourth section, we introduce the modifications to the beacon transmission process implemented in the Madwifi driver in order to implement the channel switching module. In the fifth section, an experimental testbed is described which is used to validate the performance of the autonomous channel selection algorithm. The results from the experimental testbed will be presented in the sixth section. Because the purpose of the proposed channel selection algorithm is not to increase the throughput directly but to minimize the congestion time of the whole network, we use the average one-way packet delay and the congestion time as the performance metrics for the channel selection algorithm.

5.1 An autonomous channel selection algorithm based on neighbour forcing

In this section, an autonomous channel selection algorithm based on neighbour forcing is introduced. We will analyse the structure of the algorithm first. Then the neighbour forcing module will be introduced.

We consider the follow situation: because in the infrastructure network, the AP will make the decision to change the channel. However, in the Ad-Hoc network, there is no station that has the responsibility to maintain the channel selection process. We focus on

the channel switch process when the stations are configured in Ad-Hoc mode. All the stations are assumed to be within the transmission range of each other.

5.1.1 Structure of the algorithm

We assume that all the stations in the network are autonomous which means that the stations cannot change the channel of other stations on the network directly by sending commands. Based on this assumption, an autonomous channel selection algorithm is developed. This algorithm does not require a central controller to assign channels otherwise each node selects the channel based upon its local traffic load information and the MAC bandwidth components of its neighbour stations on the same channel.

This new autonomous channel selection algorithm includes four modules: ***Monitoring, Channel switching, Predicting and neighbour forcing***. The structure is shown in Figure 5-1.

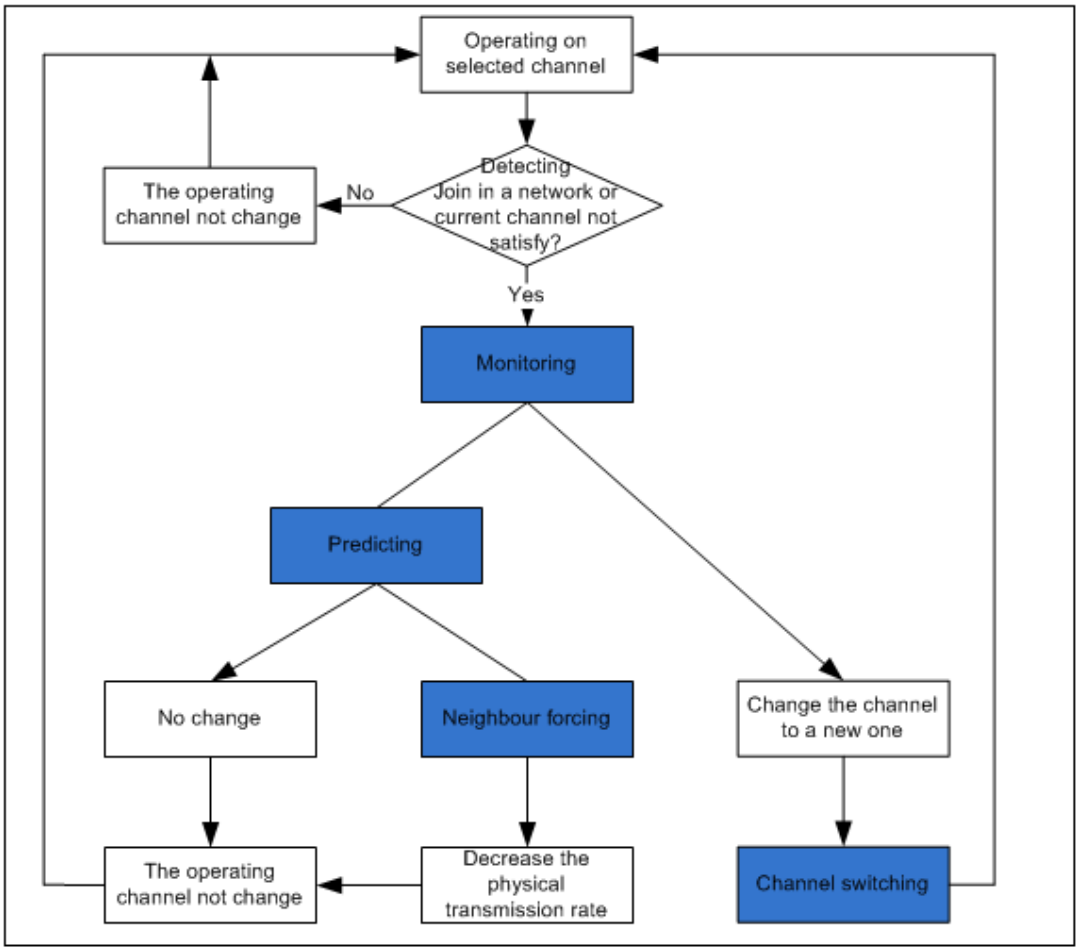


Figure 5-1 Structure of the channel selection algorithm with neighbour forcing

The *monitoring* module is used to obtain the information on the MAC bandwidth components of its neighbour stations. When a station confirms that it has become congested, the *monitoring* module will be triggered. This station will switch the WLAN adapter card into the monitor mode and monitor all the available channels to obtain the MAC bandwidth components, i.e. it will determine the number of neighbour stations, the bandwidth usage and the access efficiency of the stations. In the experiment, we monitor each channel for one second and this time is denoted as $T_{monitor}$.

With a larger value of $T_{monitor}$, the *monitoring* module can collect more packets to more provide for a precise calculation of the traffic load requirement of each station. However, the longer the time spending on monitoring the channel, the longer the time the station cannot transmit packets because of the half-duplex nature of IEEE 802.11 devices. During the experiment, because two available channels are used and also the traffic load

requirement does not change frequently, one second is a suitable duration to monitor the channel in order to generate an accurate measurement of the traffic load requirement.

In the *monitoring* module, a sniffer programme is used to monitor all available channels one by one. Based on the monitoring results, a station will calculate bandwidth components and access efficiency through formulas (2-4) to (2-12). The traffic load it wants to transmit which is denoted as the *offered traffic load* in this thesis. The traffic load of a station that can be successfully transmitted is denoted as *actual traffic load* in this thesis. The congested station will select the channel which has sufficient available bandwidth to satisfy its offered traffic load. However, sometimes there is no channel that has sufficient free bandwidth to satisfy its offered traffic load. In this situation, most of the distributed channel selection algorithms will select the channel that has the least traffic load or the channel which has the lowest interference level even though the free bandwidth is less than the offered traffic load which will produce congestion. The proposed channel selection algorithm does not use these strategies. Instead, when there is no channel that has sufficient free bandwidth, it will run the *predicting* module to determine whether there exists at least one successful channel assignment to satisfy the offered traffic load of all the stations or not.

The *predicting* module is used to determine if there exists at least one successful channel assignment under the offered traffic loads of all stations. We define a *successful channel assignment* as follows:

Definition 5.1: Successful channel assignment – *a channel assignment under where there is no station that becomes congested.*

For a network with N stations and M channels, the number of possible channel assignments is M^N. If there exists such a successful channel assignment under the offered traffic loads of all the stations, we denote this channel assignment as a *theoretical* successful channel assignment. There can be multiple theoretical successful channel assignment under a specified offered traffic load.

5.1.2 Neighbour forcing module

In this section, we will describe the autonomous channel selection algorithm with neighbour forcing.

As shown in Figure 5-2(a), there are three stations in the network which contend for 2 channels: channel 1 and channel 2. *Station A* is assigned to channel 1 and *Station B* is assigned to channel 2. *Station C* needs to select a channel to transmit its traffic load requirement.

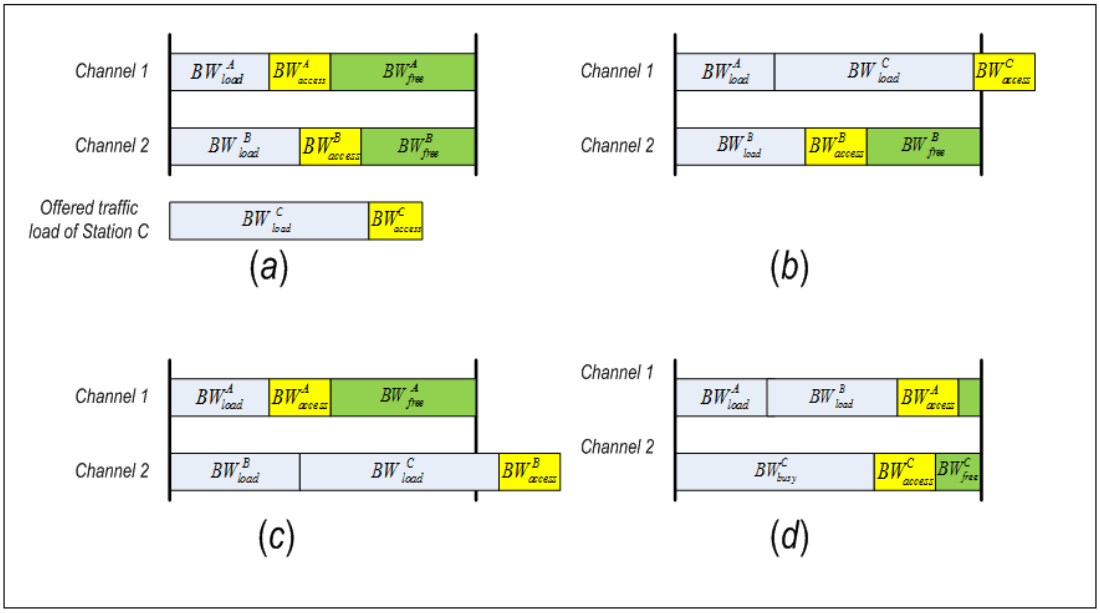


Figure 5-2 Bandwidth components of the prediction process

Because the free bandwidths BW_{free}^{A} and BW_{free}^{B} are greater than zero, *Station A* and *Station B* can transmit their offered traffic loads without saturation (and congestion) before *Station C* starts its transmission. When *Station C* wants to transmit packets, it needs to make the decision to select either channel 1 or channel 2. However, the *monitoring* module of *Station C* finds that both channels do not have sufficient free bandwidth. If *Station C* selects the least used channel, i.e. channel 1 as shown in Figure 5-2(b), the free bandwidth of *Station C* will be less than zero which means *Station C* becomes saturated under this channel assignment. A similar situation applies when *Station C* selects channel 2, as shown in Figure 5-2(c). The free bandwidth of *Station B* is less than zero which means that *Station B* becomes saturated under this channel

assignment. These two channel assignments are unsuccessful channel assignments because there is at least one station that becomes saturated under these two channel assignments. The *predicting* module of Station C will find that if *Station A* and *Station B* are assigned to channel 1, and *Station C* is assigned to channel 2, then all the three stations can transmit their offered traffic load without saturation as shown in Figure 5-2(d).

In the proposed channel selection algorithm, the *predicting* module will only be triggered when the *monitoring* module cannot find a channel that has sufficient available bandwidth. Because all the stations are within the transmission range of each other, the *predicting* module checks all the possible N^M channel assignments one by one where *N* is the number of available channels and *M* is the number of stations. It can determine the successful channel assignment exists or not. The problem of hidden nodes does exist however this has been left to future research in this area.

If there exists at least one theoretical successful channel assignment, the next step is to determine the strategy for reassigning the channels. There are three possible outcomes from the *monitoring* module and *predicting* module: (i) There exists no theoretical successful channel assignment; (ii) There is one channel which has sufficient free bandwidth for the congested station; (iii) There is no channel which has sufficient free bandwidth but there exists at least one theoretical successful channel assignment. Each outcome results in a different switching strategy being employed.

1) *There exists no theoretical successful channel assignment* – Here the *predicting* process has determined that there exists no theoretical successful channel assignment. Consequently, no further action will be taken, i.e. the station does not attempt to change its own channel or to force its neighbours to change their channels.

2) *There is one channel that has sufficient free bandwidth for the congested station*– Here the *monitoring* module has identified another channel capable of accommodating the offered traffic load of the congested station therefore it triggers the *channel changing* module to change its channel. Firstly, it sends out a number of special beacon frames containing the *channel switch information element* that announces to its receiver

that a channel change is imminent. These beacon frames ensure that the receiver can maintain its connectivity with the station. The receiver will change its channel once it receives a beacon frame containing the *channel switch information elemen*t. Secondly, the congested station switches to the new channel and starts to transmit its packets on the new channel.

3) *There is no channel that has sufficient free bandwidth but there exists at least one theoretical successful channel assignment* – Here the *predicting* module has identified that there is at least one theoretical successful channel assignment that can avoid congestion. However, this involves the other stations having to change their channel, in addition to the congestion station itself changing channel. As all the stations operate autonomously, a station cannot directly instruct its neighbour stations to change channel. Therefore, the mechanism employed here involves deliberately driving the neighbour nodes into congestion in order to force them to switch to another channel. The development of this *neighbour forcing* module has been the main challenge and effort of this work.

In the *neighbour forcing* module, a station does not change the channels of its neighbour stations directly. Instead, it forces its neighbours into congestion in order to get them to change their channels by purposely reducing its own PHY transmission rate.

Because the time required to transmit a wireless frame is:

$$T_{frame} = T_{preamble} + T_{PLCP} + \frac{8 \times Length}{Rate} \tag{5-1}$$

Here $T_{preamble}$ is the time used to transmit the preamble bytes which is depends on the standard used, i.e. in IEEE 802.11a $T_{preamble} = 20\mu s$ and in IEEE 802.11b $T_{preamble} = 96\mu s$. T_{PLCP} is the time used to transmit the PLCP header. In IEEE 802.11a and IEEE 802.11g $T_{PLCP} = 4\mu s$; in IEEE 802.11b $T_{PLCP} = 48bits$ when short preamble is used. $Length$ is the total frame length which includes the MAC header. $Rate$ is the PHY transmission rate of the packets in units of Mbps. If the station transmits packets at a lower PHY transmission rate, the load bandwidth will increase because it needs more time to transmit a frame.

Figure 5-3 shows the relationship between the MAC bandwidth components of the congested station when the transmission rate of the congested station is changed. We assume that the traffic load and access efficiency of other stations remains the same during the neighbour forcing process.

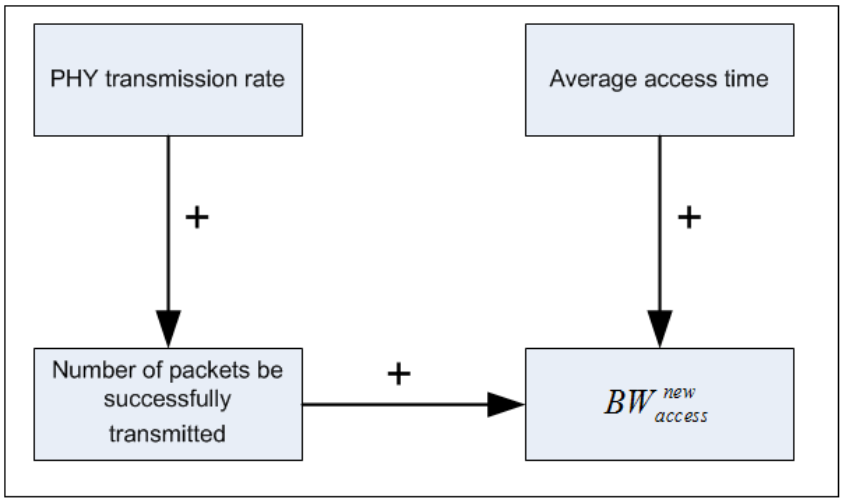


Figure 5-3 Relationship between the MAC bandwidth components

As shown in equation (2-9), the transmission time of a packet is negatively correlation with the PHY transmission rate. The number of packets successfully transmitted will decrease when the PHY transmission rate is reduced.

The average access time of the station depends on the number of stations contending for access the medium. The PHY transmission rate reduction has little impact on the average access time because the number of stations remains the same. However, because the number of packets being successfully transmitted decreases, the access bandwidth requirement of this station will decrease according to equation (2-11). We denote this as the *new access bandwidth* BW_{access}^{new} in Figure 5-3.

If a station transmit packets with higher PHY transmission rate cannot satisfy its traffic load requirement, it also cannot satisfy its traffic load requirement with a lower PHY transmission rate. Therefore, it becomes saturated when it decreases its PHY transmission rate.

According to equation (2-8), because all the station experience the same idle bandwidth, if a neighbour station has an access bandwidth requirement greater than BW_{access}^{new}, that

neighbour station become saturated when the congested station reduces its PHY transmission rate. In this case, the *neighbour forcing* module has worked successfully.

After a period of $T_{forcing}$, the station which reduces it PHY transmission rate will increase its transmission rate back to its original PHY transmission rate and resume transmitting its offered traffic load. Here $T_{forcing}$ is the time *Station C* transmits packets at a lower PHY transmission rate. It should be longer than the time $T_{confirm}$ which is used to confirm the congestion status.

If there is no station which has a smaller access bandwidth than the new access bandwidth of the congested station after it changes its PHY transmission rate, the *neighbour forcing* module fails to force its neighbours into congestion because when $BW_{free}^{congested}$ equals zero, the BW_{free} of other stations is still greater than zero which means they can still win a sufficient number of transmission opportunities to satisfy their traffic load. They cannot be forced into congestion and therefore the channel selection algorithm will also fail to reassign the channels to stations in the network.

Figure 5-4 shows the MAC bandwidth components during the *neighbour forcing* process.

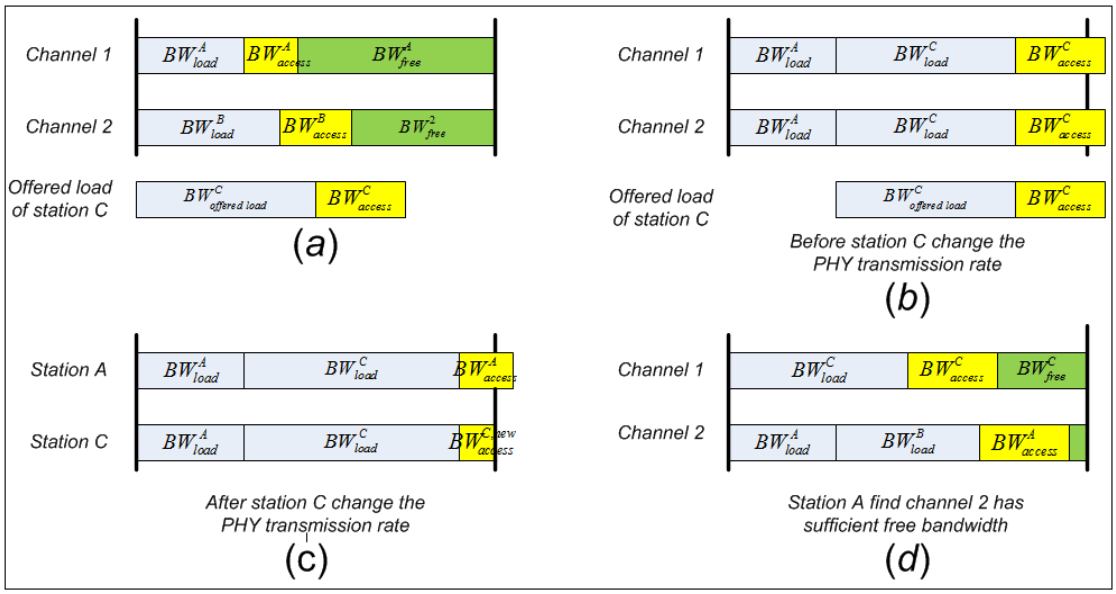


Figure 5-4 MAC bandwidth components during the neighbour forcing process

There are 3 stations in the network: *Station A*, *Station B* and *Station C*. The MAC

bandwidth components of each station are shown in Figure 5-4(a). *Station A* is assigned to channel 1 and *Station B* is assigned to channel 2. *Station C* wants to join the network and it finds that the channels don't have sufficient available bandwidth as shown in Figure 5-4(b). However, the *predicting* module of *Station C* discovers that there exists at least one successful channel assignment with the condition that *Station A* or *Station B* changes its channel. Because all the stations are autonomous, *Station C* cannot instruct *Station A* or *Station B* to change channel.

The channel selection algorithm deployed on *Station C* makes the decision to drive its neighbour stations into congestion in order to force a channel change. *Station C* will reduce its PHY transmission rate. According to the performance anomaly mechanism [85], the throughput of all the other stations which transmit packets at a higher PHY transmission rate will experience a shortage of available bandwidth and will be forced into congestion.

Figure 5-4(c) shows the MAC bandwidth components after *Station C* reduce its PHY transmission rate. Because *Station C* transmits at a low PHY transmission rate, the number of packets successfully transmitted by *Station C* will decrease significantly, the access bandwidth of *Station C* will decrease significantly. The load bandwidth of *Station C* will increase until the free bandwidth of *Station C* equals zero.

For station C:

$$0 = BW_{free}^{C} = 1 - BW_{load}^{A} - BW_{load}^{C} - BW_{access}^{C,new} \tag{5-2}$$

For station A:

$$BW_{free}^{A} = 1 - BW_{load}^{A} - BW_{load}^{C} - BW_{access}^{A} \tag{5-3}$$

After the PHY transmission rate of *Station C* is reduced, if the access bandwidth $BW_{access}^{C,new}$ is less than the access bandwidth of *Station A* BW_{access}^{A}, the free bandwidth of *Station A* BW_{free}^{A} will be less than zero which means that *Station A* will be forced into saturation. *Station A* will call its *monitoring* module after it confirms that it has become saturated and it will find that channel 2 has sufficient free bandwidth to satisfy its traffic load. It will change to channel 2. *Station A* and *Station B* will share channel 2 and both

of these two stations can transmit their offered traffic loads without saturation.

Finally, as shown in Figure 5-4(d), the free bandwidth of all three stations is greater than zero. *Station A* and *Station B* can transmit their packets on channel 2 without saturation. *Station C* can transmit its packets on channel 1 without saturation.

In the network, all the stations are running the same algorithm and periodically they check their saturation status. Once a station confirms that its operating channel cannot provide a sufficient number of transmission opportunities to satisfy its offered traffic load, the station will initiate the *monitoring* module to collect MAC bandwidth information from all the available channels. Because all the stations are autonomous, they make decision based on the local information it has monitored, the neighbour forcing mechanism may generate another channel change or initiate another neighbour forcing.

A simple C++ simulator was developed which will be described in the next section in order to demonstrate the feasibility and the performance of the proposed channel selection algorithm.

5.2 Simple C++ simulator

There are many available wireless simulators such as ns-2, ns-3, and OPNET etc can be used. However, because the purpose of the simulation is only to demonstrate the theoretical feasibility of the channel selection algorithm, it was not considered worthwhile to expend considerable time and effort becoming familiar with a complicated simulator. Instead, we developed a simple C++ simulator to demonstrate the theoretical feasibility of the channel selection algorithm. In this simulator, we do not include the details of the MAC protocol. Otherwise, we focus on the channel switch algorithm in terms of the actions that the congested station will take when the traffic load requirement is changed; what will happen to the bandwidth components when the access efficiency is changed etc.

The simulator was found to produce sets of results that were broadly consistent with the experimental results in terms of the successful reassignment ratio.

5.2.1 Description of the simulator

In this simulator, there are two classes: *class Tnode* and *class Tchan*. The *Tnode* class includes four members: *station ID, associated operating channel, busy bandwidth* and *access efficiency*. The *Tchan* class includes the busy bandwidth, free bandwidth, the largest access bandwidth and the list of stations which are operating on this channel. The busy bandwidth of each channel equals the sum of the load bandwidths of all the stations which are operating on this channel.

$$BW_{busy}^{j} = \sum_{i} BW_{load}^{i,j} \tag{5-4}$$

Here BW_{busy}^{j} represents the busy bandwidth of channel j and $BW_{load}^{i,j}$ represents the load bandwidth of station i which is operating on channel j.

For all the stations operating on a channel, the station with the largest access bandwidth is the station with the smallest free bandwidth which means that the first station to become saturated will be the station with the largest access bandwidth.

There is a node vector *vector <Tnode*> NodeVector* which includes all the stations in the network. The node vector will be updated when the traffic load of station has been changed.

There is also a channel array *Tchan Array_chan[NUM_CHAN]* where *NUM_CHAN* represents the number of channels which are available during the simulation. The capacity of the channel is normalized to 1.0 if there is no station using that channel.

We developed a separate traffic generator to generate the input file to the simulator which will be introduced later. The input to the simulator is a sequence of numbers that includes *station ID, load bandwidth* and *access efficiency.* If the station with the same ID is included in the *NodeVector*, the simulator will change the load bandwidth and access efficiency of that station. If there is no station that has the same ID this means that this is a new station joining the network. Therefore this station needs to select a channel in order to transmit its packets without causing congestion.

We run the simulator in a sequence of *simulation cycles*. At the beginning of each simulation cycle, the simulator will read a new sequence of data from the input file and

the whole channel selection process will start. The simulation cycle will finish when the *predicting* module confirms that there exists no theoretical successful channel assignment or that the free bandwidth of all the stations is greater than zero which means all the stations can transmit packets without saturation.

5.2.2 Structure of the C++ simulator

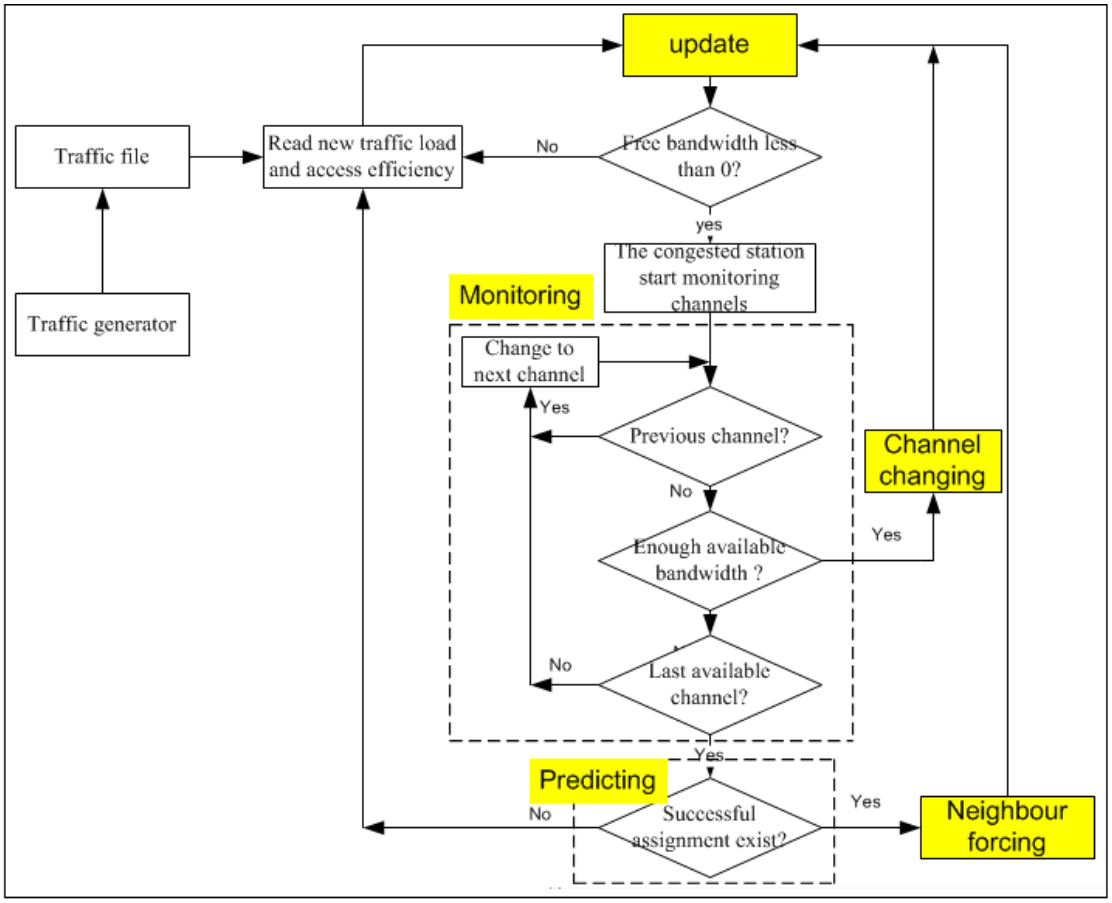


Figure 5-5 Structure of the simple C++ simulator

Figure 5-5 shows the structure of the C++ simulator. Before the simulation, the *traffic generator* module will generate a traffic file. Each line of this file includes three parameters: *station ID*, *traffic load* and *access efficiency*. The traffic load is a random number generated under a Poisson distribution with mean value λ. It represents the bandwidth requirement or the offered traffic load of this station. We will discuss the results for different λ values in section 5.3. The range of λ was calculated from experiment with different packet sizes and transmission rates. The typical range of

access efficiency value is shown in *Appendix A*. The values of access efficiency in this simulation are selected from the range 2 to 14.

The *update* module checks the bandwidth components of each channel. If the free bandwidth of all the stations is greater than zero, there is no saturation in the network and the simulator will start a new simulation cycle. If there is one station that has a free bandwidth less than zero, the next step is to determine the ID of the saturated station. In the simulation, all the saturated stations will be considered to have become congested and hence one station will start the monitoring process once its free bandwidth $BW_{free} \leq 0$.

The *monitor* model checks all available channels to compare the available bandwidth with the offered traffic load of the congested station. If a channel is found where the available bandwidth is greater than the offered traffic load, the congested station will change to this channel. Otherwise, it will move on to monitor the next channel.

If all the channels have been monitored and no channel was found to have sufficient free bandwidth, the *predicting* module will be triggered. This module will check all the possible channel assignments to determine if there exists at least one theoretical successful channel assignment that can satisfy the offered traffic loads of all the stations. If such a channel assignment exists, the station needs to force its neighbour to change the channel. In this case the *neighbour forcing* module will be triggered. If no such a theoretical successful channel assignment exists, then there is no way to satisfy the entire offered traffic load in which case we do not change the channel of any stations. A new simulation cycle will start.

In the simulator, the operation of the *neighbour forcing* module is straightforward. It only needs to change the access efficiency to the largest value which has been set to 14. The load bandwidth of the congested station will be increased until its free bandwidth reaches zero with the new access efficiency value.

At the end of each simulation cycle, the simulator will record the result of simulation and the result of the *predicting* module. These recorded results are later analysed to determine the performance of the algorithm.

5.2.3 Performance metric for the simulated algorithms

With each simulation cycle, there are two possible results: the first one is where the available channels cannot satisfy the offered traffic loads of all the stations. The other result is where the network channel can satisfy the offered traffic loads of all the stations. Figure 5-6 shows the relationship between those outcomes and the number of successful and failed channel assignments.

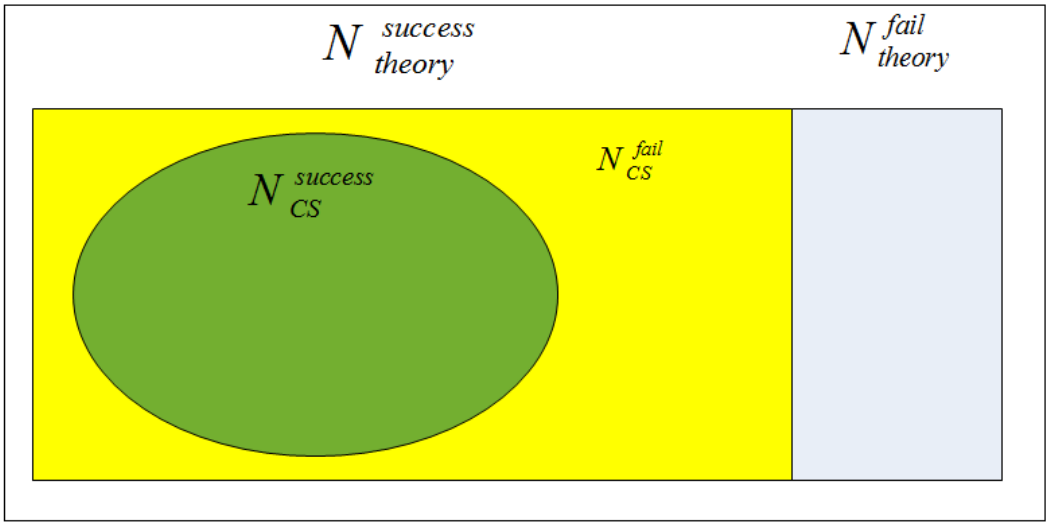


Figure 5-6 Relationship between the performance parameters

For the whole simulation, $N^{success}_{theory}$ represents the number of simulation cycles where at least one theoretical successful channel assignment exists. The N^{fail}_{theory} represents the number of simulation cycles where there exists no theoretical successful channel assignment. Under different traffic patterns the values of $N^{success}_{theory}$ and N^{fail}_{theory} will also be different.

The $N^{success}_{CS}$ represents the number of simulation cycles where the channel selection algorithm can achieve a successful channel reassignment to satisfy the traffic loads of all the stations. The N^{fail}_{CS} represents the number of simulation cycles where the channel selection algorithm cannot assign the channels to stations to satisfy the corresponding traffic load but there exists at least one theoretical successful channel assignment. The

relationship between the last four parameters is shown in Figure 5-6.

The ratio between $N_{theory}^{success}$ and N_{CS}^{succes} provides the successful reassignment ratio for the channel selection algorithm.

$$\eta = \frac{N_{CS}^{success}}{N_{theory}^{success}} \tag{5-5}$$

The higher the value of η the more efficient the channel selection algorithm is in finding a successful channel assignment when congestion occurs.

The algorithm successful reassignment ratio η is used to investigate its performance in the simulation. The performance of channel selection algorithm is different under different traffic loads, different number of channels and different average number of stations per channel. This performance will be presented and discussed in section 5.3.

In the simulation, we also focus on the feasibility of the proposed channel selection algorithm. As described previously, because all the stations are running this algorithm autonomously, a channel change or a neighbour forcing may generate another channel change or neighbour forcing. We refer to this as the *ripple effect* and in the simulation we record the number of channel changes and number of neighbour forcing for every simulation cycle. The result will be presented and discussed in next section.

In the simulator, we also recorded the number of theoretical successful channel assignment under different traffic loads. This result will be presented and discussed in next sections.

5.3 Results of the simulation and discussions

The simulator described last section is used to demonstrate the feasibility of the proposed channel selection algorithm. The traffic load of each station in the simulator is generated by a Poisson distribution generator which randomly changes the traffic load and access efficiency. The station which will change its traffic load is also randomly selected from the list of stations. We assume there is only one station changing its traffic load and access efficiency at given time.

In the next three sections, we will present the performance of the proposed channel selection algorithm under different traffic conditions and different numbers of channels and stations. During each simulation, the traffic of each station has the same distribution. We run multiple simulations with different mean values of bandwidth requirement to investigate the performance.

5.3.1 Successful reassignment ratio

There are 3 channels available during the simulation. The capacity of each channel is normalized to 1.0 if there is no station using that channel. Because the normalized mean bandwidth requirement of each station is the same, the total traffic load of all the stations will increase with the mean value of traffic load. We increase the mean λ value of the normalized bandwidth requirement from 0.33 to 0.46 in steps of 0.01. For each normalized bandwidth requirement λ, the Poisson distributed generator generates 100,000 different numbers. Each bar in Figure 5-7 and 5-8 is the successful reassignment ratio according to the 100,000 traffic load requirement.

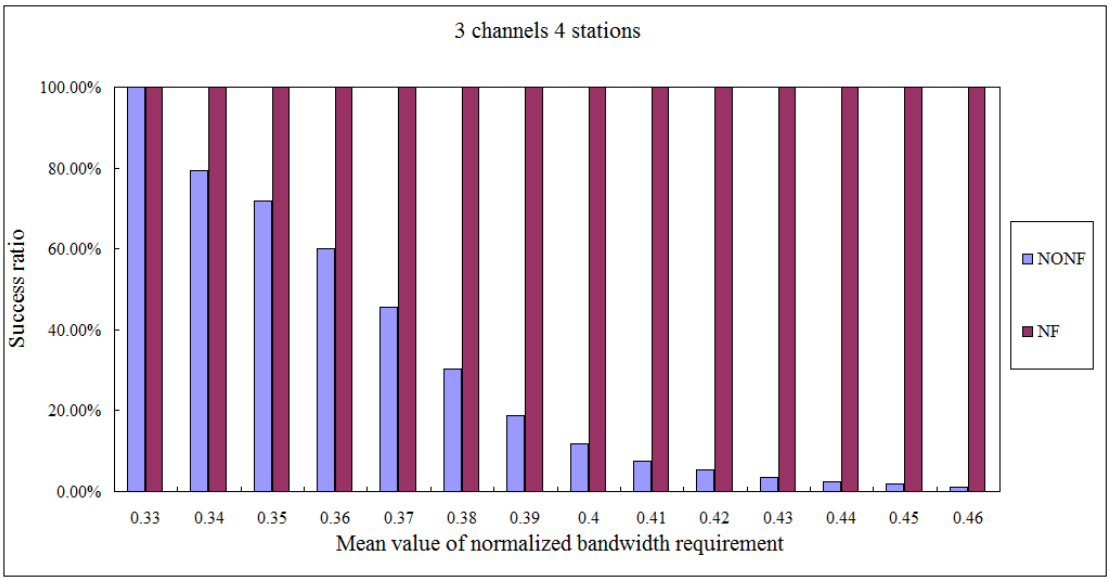


Figure 5-7 Successful reassignment ratio under different traffic loads (4 stations)

Figure 5-7 shows the successful reassignment ratio for different traffic loads under the two channel selection algorithms, i.e. *NONF* and *NF*. There are 4 stations operating on these 3 channels. This figure shows that the successful reassignment ratio of the *NF* algorithm outperforms the *NONF* algorithm for all traffic loads which indicates that the

proposed channel selection algorithm has a greater ability to solve the congestion problem.

Figure 5-7 also shows that the successful reassignment ratio of the *NF* algorithm remains at 100% under all mean values of the offered traffic load. It successfully reassigns the channels to the stations without producing congestion (under the condition that there exists at least one theoretical successful channel assignment).

The reason is that the congested station does not need to share the channel with other stations if there exists at least one theoretical successful channel assignment when the number of stations is one greater than the number of channels. Because the number of stations is only one greater than the number of channels, there is only one channel that contains two stations. The congested station must be one of these two stations. If the congested station needs to share the channel with other stations according to the theoretical successful assignments, the *channel monitoring* module will find a channel that has sufficient free bandwidth to satisfy its traffic load. If the congested station does not need to share the channel with any other stations according to all the theoretical successful assignments, there must be two other stations that can share a channel without congestion. When the *neighbour forcing* module is triggered, the congested station forces all its neighbour stations into congestion and the neighbour stations must find another channel that has sufficient free bandwidth.

The successful reassignment ratio of the *NONF* algorithm will decrease with the increase in the traffic load requirement. The reason is that this algorithm fails to reassign the channels when it cannot find a channel that has sufficient free bandwidth to satisfy the offered traffic load. With the increase in the traffic load requirement, a congested station has a lower probability of finding a channel that has a sufficient free bandwidth to satisfy its traffic load.

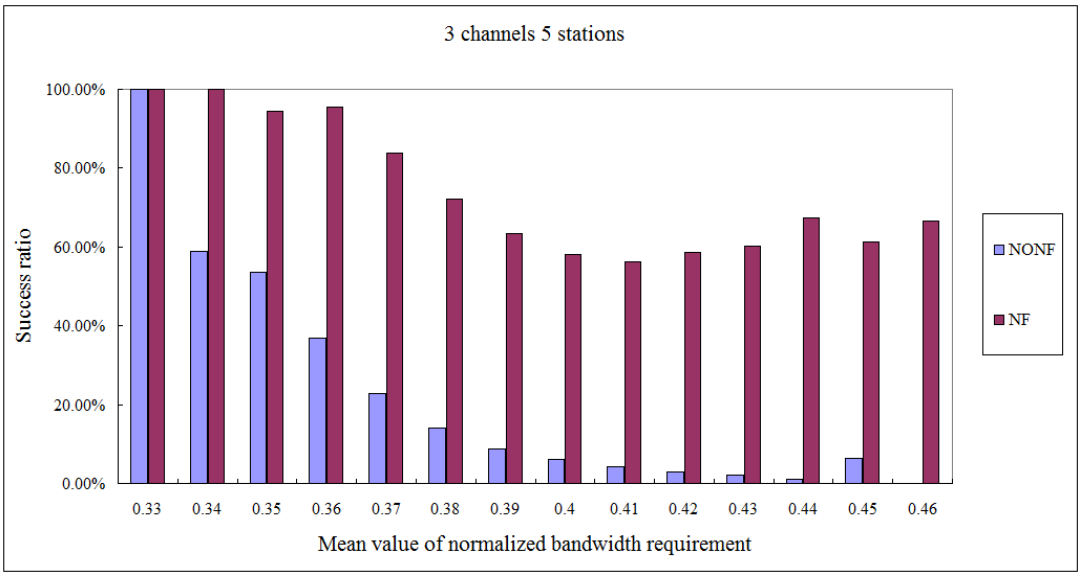


Figure 5-8 Successful reassignment ratio of different traffic load (5 stations)

Figure 5-8 shows the successful reassignment ratio of the network with 3 channels and 5 stations. It also shows that the *NF* algorithm has a better performance than the *NONF* algorithm under all offered traffic loads.

Figure 5-8 also shows that the successful reassignment ratios of both of the algorithms decreases with the increase in the traffic load of each station.

The reason why the ratio of successful reassignment decreases with the increase in the traffic load for the *NONF* algorithm is that it has a lower probability of finding a channel that has sufficient free bandwidth to satisfy its traffic load under higher traffic loads.

The *NF* algorithm fails when the congested stations needs to share a channel with at least one other station in all of the possible theoretical successful channel assignments. However, because the congested station that reduces its PHY transmission rate will force all its neighbour stations into congestion, the algorithm fails to achieve any of the theoretical successful channel assignments. This situation usually happen when the number of theoretical successful channel assignments is small, typically 1 or 2. When the mean traffic load increases, the number of theoretical successful channel assignments will decrease which will be shown in next section.

5.3.2 Average number of theoretical successful channel assignment

In the simulation, we also recorded the number of theoretical successful channel assignments under different traffic loads. Figure 5-9 shows the ratio of successful reassignments against the number of theoretical successful channel assignments for the *NF* channel selection algorithm. There are 3 available channels and 5 stations in the simulation. It shows that the successful reassignment ratio increases with the number of theoretical successful channel assignments under different mean traffic loads. In other words, the *NF* algorithm has a larger probability to successfully reassign the channels when the number of theoretical successful channel assignments is large. As shown in Figure 5-9, when the number of theoretical successful channel assignments is greater than 9, the *NF* channel selection algorithm has a successful reassignment ratio of 100% in reassigning the channels.

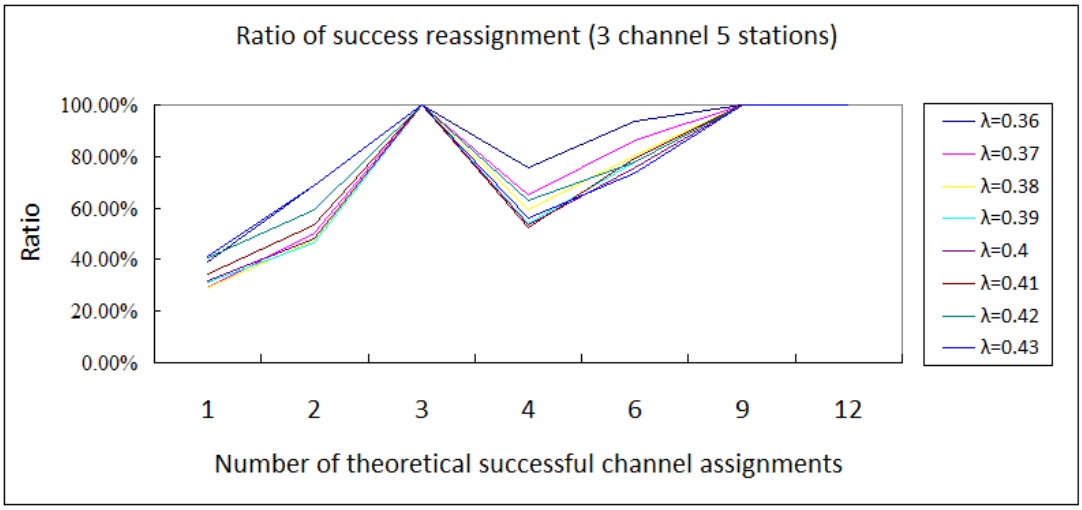


Figure 5-9 Ratio of successful reassignments (5 stations)

The reason why the proposed channel selection algorithm has such a successful reassignment ratio of 100% in reassigning the channel when there are 3 theoretical channel assignments is because the congested station needs to be assigned to a separate channel. The other 4 stations will share the other 2 channels and each channel needs to be assigned two stations.

Figure 5-10 presents the ratio of successful reassignment when there are 3 available channels and 6 stations in the simulation. The ratio increases when the number of

theoretical successful channel assignment increases for all the mean offered traffic loads.

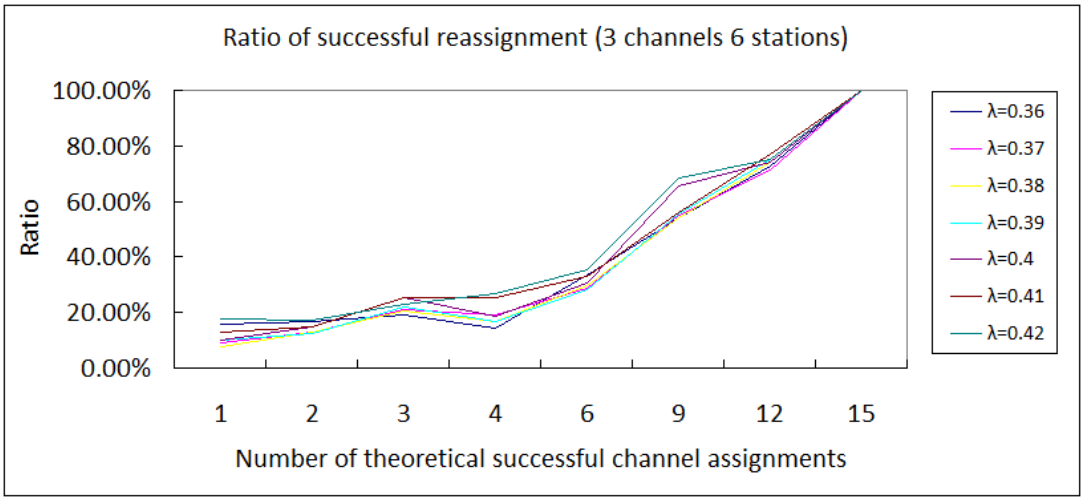


Figure 5-10 Ratio of successful reassignments (6 stations)

Figure 5-9 and Figure 5-10 show the ratio of successful reassignments against the number of theoretical successful channel assignments. It shows that for all the mean traffic loads, the successful reassignment ratio increases with the number of theoretical successful channel assignments. This result suggests that if the number of theoretical successful channel assignments is small, (i.e. there exists only one or two theoretical successful channel assignments according to the *predicting* module) the proposed channel selection algorithm should avoid starting the neighbour forcing process because there is a high probability that it cannot reassign the channels successfully.

5.4 Modifications to the Madwifi driver

In section 2.2.5, we described the channel switching process in the Infrastructure and Ad-Hoc modes. However, the open source Madwifi driver doesn't implement the same mechanism in the Ad-Hoc mode. Because there is no central station to control the channel switch in Ad-Hoc mode, it is more complicated to implement the channel change in Ad-hoc network. The IEEE 802.11h standard introduced a new management information element called the *IBSS DFS information element* to support channel change in Ad-Hoc networks. However, this mechanism is difficult to implement and its implementation has been omitted in Madwifi driver.

We have implemented a simple channel switch process for use in infrastructure networks in our experimental testbed.

5.4.1 Modifying the beacon transmissions in Madwifi

Information elements in management frame are used to announce the existence of a network. A management frame includes multiple information elements such as the SSID, support rate, traffic indication map and parameters for channel and network. IEEE 802.11h defines two information elements used to support the channel changing: *channel switch information element* and *IBSS DFS (Dynamic Frequency Selection) information element*.

The *channel switch information element* is used in infrastructure networks and is included in beacon frames. The AP which needs to change channel will transmit some beacon frames containing *channel switch information elements* to all the stations associated with it. The stations which receive the beacon frames with the *channel switch information element* will change the channel within the time indicated in the *channel switch information element*.

As its name implies, the *IBSS DFS information element* is intended to support channel change in an Ad-Hoc network. The station which needs to change channel is denoted as the *DFS owner*, it has the responsibility to provide the coordination for the channel switch process. It will send out beacon frames containing the *IBSS DFS information element*. If a station detects the *IBSS DFS information element* and wants to attempt a channel switch following the DFS owner, the station shall broadcast one or more *Measurement Report frames* [23] indicating the presence of the station. The DFS owner will select the channel according to the station information it has collected.

For various reasons arising from the complexity in channel switching in IBSS networks, we do not use the channel switch service as defined in IEEE 802.11h for our experiments. Instead, we will make some modifications in the operation of the Ad-hoc mode, as follows:

1) *Implementing a similar beacon transmission process as used in an infrastructure network.*

2) *Only the traffic sender can send out the beacon frames containing the channel switch information element.*

3) *The traffic receivers change the channel immediately after they receive the beacon frames containing a channel switch information element. It does not transmit any frames to inform the channel switch to other stations.*

These modifications make it simpler to maintain the connectivity between two stations in an Ad-hoc network.

In Section 2.7.3 we described the beacon frame transmission mechanism in infrastructure and Ad-Hoc networks. In Ad-Hoc network, all the stations have the responsibility to transmit beacon frames. If a station does not receive any beacon frames within a random time delay, it needs to send out beacon frames to announce the channel change to maintain network connectivity. Before we make any modifications to the beacon frame, we describe the beacon transmission process in the Ad-hoc network.

When an Ad-Hoc mode VAP is created, it will recognize the configuration of the network and the network information such as SSID, operating channel, beacon interval and transmit power etc. from the user layer if it is the first station of this network or it will read the network information from the beacon frame when it joins an existing Ad-Hoc network. If it receives beacon frames within a random time interval after the previous beacon was received, it will read the timestamp from the beacon frame and synchronize its clock with the frame sender. Otherwise it will send a beacon frame with its own timestamp in order that its neighbours can synchronize with it.

In order to send out beacon frames at any time when the upper layer requests it, we modified the beacon transmission process which is shown in Figure 5-11.

We combined the channel change mechanism with the beacon transmission process and

this will be introduced in the next section.

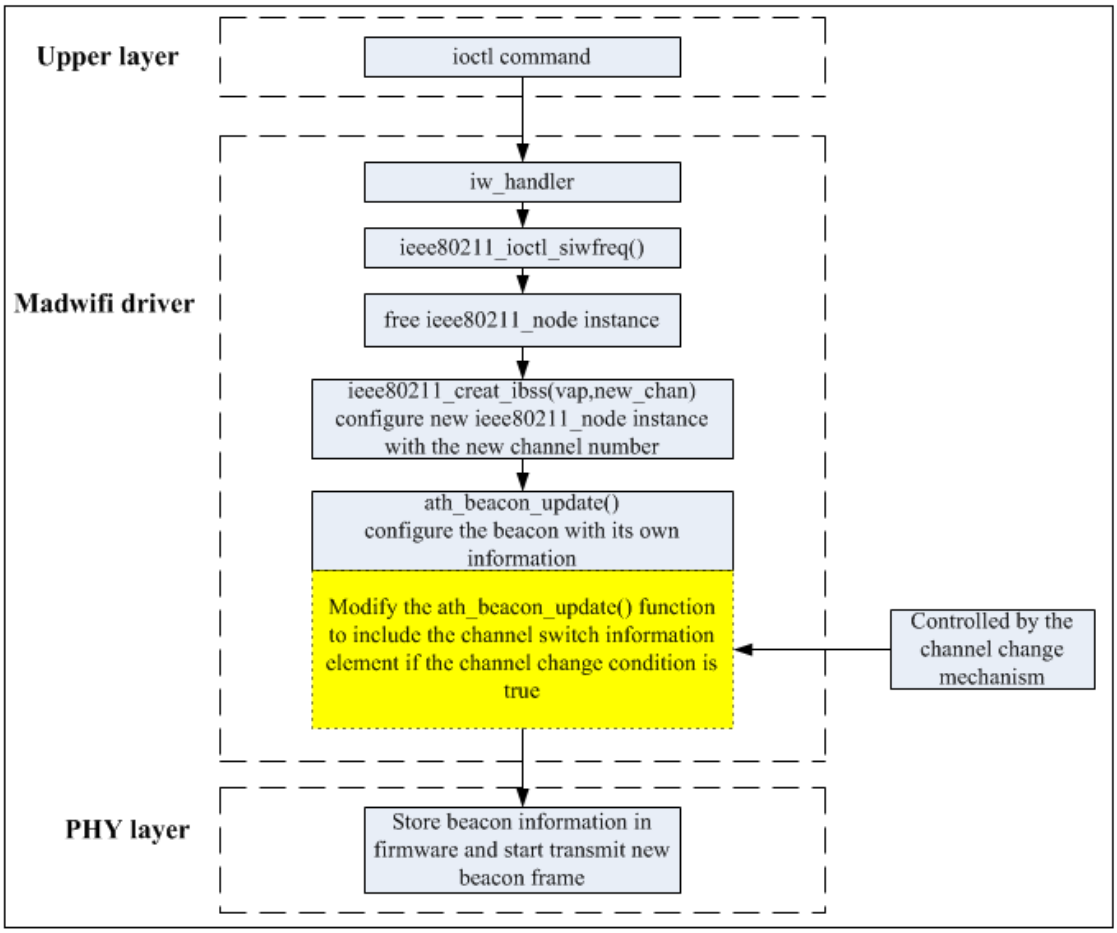


Figure 5-11 Modified beacon transmission process in the Madwifi driver

5.4.2 Implementation of the channel switch mechanism in Ad-hoc mode of Madwifi

IEEE 802.11h defines different channel switch mechanisms for the Ad-hoc mode and infrastructure modes. However, due to the complexity of this IBSS DFS mechanism, the Madwifi driver does not implement this mechanism by default. When a station wants to change the operating channel, it needs to take actions through the process described in Figure 2-9. Madwifi utilizes *iw_handler* to receive and parse the *ioctl* command such as *set* or *get* the transmit rate, the transmit channel, the SSID and the transmit power etc. Because the scope of this thesis is channel selection mechanism, we follow the channel change process as shown in Figure 5-11.

Because we need to update the beacon frame when it is required with the channel switch command and it cannot continue to send the beacon frames containing channel switch

information element after the channel switch process, we utilize a two-step process to update the beacon frame.

Step 1: *When one station decides to change the channel, it will not change the operating channel immediately otherwise it will follow the operation of the state machine. This is different from the normal channel switch process. When it creates a new ibss with ieee80211_creat_ibss, it does not use the desired channel but the current operating channel. In the ieee80211_beacon_update function, it also includes the channel switch information element if this is the first time receiving the channel switch command. After it finishes the beacon update process, it returns to the RUN status again and the station will send beacon frames containing the channel switch information element.*

Step 2: *After a few seconds, the upper layer will issue another command to change the channel. This time is required to ensure that the receiver changes the channel. In the experiments, we use a fixed duration of 2 seconds to ensure that the receiver could receive at least one beacon frame with the channel switch information element. When it receives the command to change the channel at the second time, Madwifi will change the channel in the normal way and send out beacon frames without the channel switch information element.*

According to [60], the channel switch overhead varies from 200 μs to 20 ms. However, this delay is only the hardware overhead on the sender side. When it is necessary to make sure the receiver can receive at least one beacon frame with channel switch information element, especially when the channel utilization is high, we use 2 seconds to guarantee the connectivity between the sender and receiver. In the further research, we should investigate the channel switch overhead to increase the performance of our algorithm. This two-step process sends out beacon frames on the previous channel which could inform the receiver to change the channel. The receiver will decide to follow this change or not after it receives the first beacon frame containing the *channel*

switch information element. During the experiment, the receiver will change the channel after it has received a beacon frame containing the *channel switch information element*.

5.5 Experimental testbed setup

In this section we will introduce the experimental testbed used to validate the autonomous channel selection algorithm. The testbed includes a number of stations which are divided into two types: *Sender* and *Receiver*. Each pair of sender and receiver PCs constitutes a link. Each link will attempt to operate on a channel to satisfy the traffic requirements of the sender. Firstly we will introduce the structure of the testbed. On each station, the process of packet transmission and reception will be described next. The *congestion status monitoring module*, *channel monitoring module*, *channel changing module* and *predicting module* will be presented separately. We also make a minor modification to the traffic generator to calculate the packet transmission delay and number of packets successfully transmitted per second.

5.5.1 Structure of the testbed

This experimental testbed consists of 7 PCs. Each PC runs Fedora 12. The version of Linux kernel is 2.6.32-175.fc12.i686. Every PC was implemented with the modified Madwifi wireless driver. The original version of Madwifi driver is 0.9.4-r1433. Each PC includes two interfaces: one is a PCI wireless interface, the other is an Ethernet interface. A Netgear WAG 511 dual band wireless PC adapter was used for the wireless PCI interface.

In the network, 6 PCs are placed in a rectangular area. For the purpose of accurately calculating the packet delay and to avoid generating any additional interference, we used another PC as the *manager* to control the other 6 PCs through the wired Ethernet port. The *manager* PC maintains the synchronization of the 6 PCs and triggers the packet transmission process of each PC. Figure 5-12 shows the structure of the testbed.

The wireless adapters are configured to transmit at a fixed power. The RTS/CTS mechanisms were disabled. The adapters operate in a "pseudo-IBSS" mode in which

they send no other management messages. Each data packet consists of 24 bytes of IEEE 802.11 header and 4 bytes of a frame check sequence (FCS).

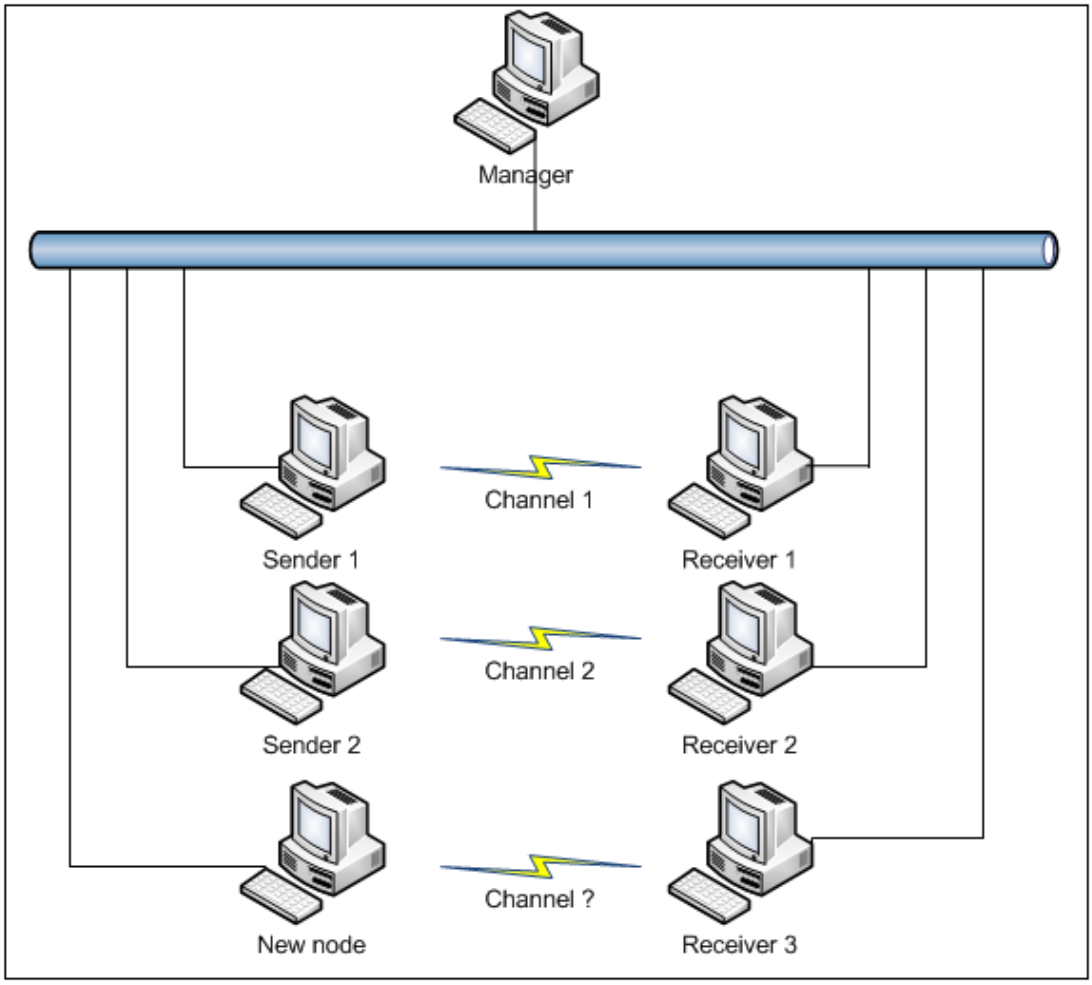


Figure 5-12 Structure of the experimental testbed

On each PC, we implement the following modules: *Packet transmission/receive module*, *congestion status checking module*, *channel monitoring module*, *channel changing module*, *predicting module* and *neighbour forcing module.* Figure 5-13 shows the structure of channel selection algorithm with these modules. We will describe each module separately over the next 6 sections.

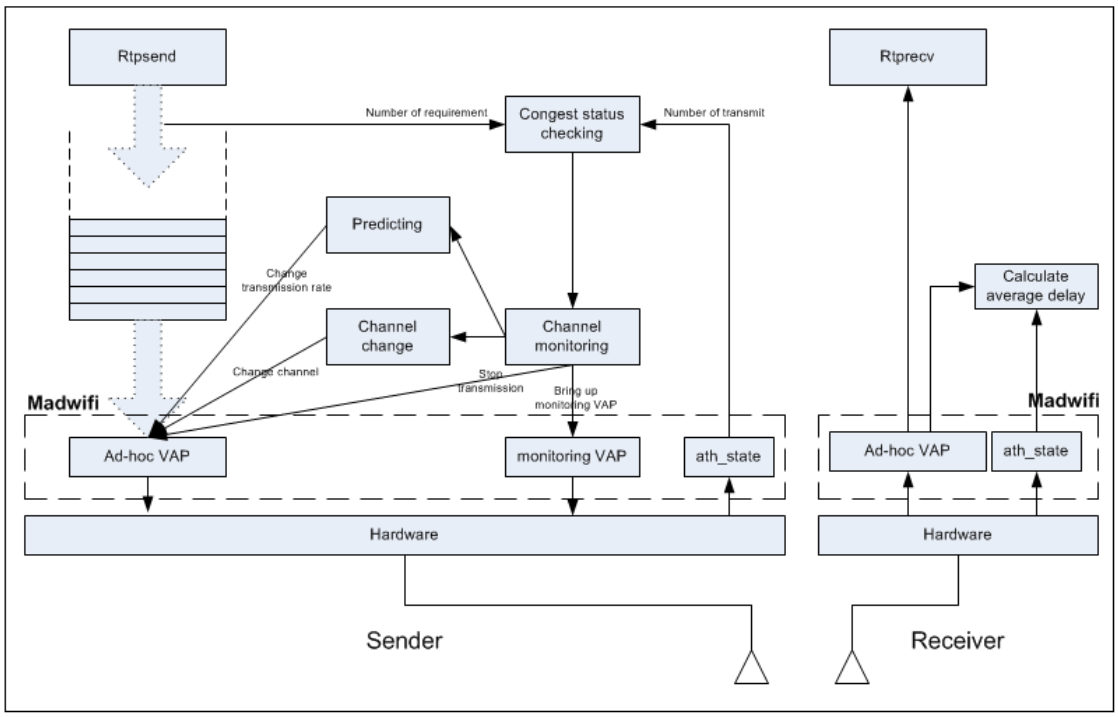


Figure 5-13 Structure of channel selection algorithm

5.5.2 Packet transmission and reception module

We use the *rtptools-1.18* traffic generator [108] to generate the offered traffic load. It includes a number of small applications that can be used to transmit RTP data. *rtpplay, rtpsend, rtpdump* and *rtptrans* are four major applications of *rtptools*.

Rtpdump listens on the specified address and port pair for RTP and RTCP packets and dumps a processed version to *outputfile* if specified or *stdout* otherwise.

rtpplay reads RTP session data, recorded by *rtpdump –F dump* from either the *file* or stdin, if *file* is not specified, sending it to network address and port with a time-to-live (*ttl*) value which is specified with the command.

rtpsend sends an RTP packet stream with configurable parameters. The RTP or RTCP headers are read from a file which is generated by hand, a script program or *rtpdump* (format “ascii”).

rtptrans transmits RTP/RTCP packets arriving from one of the addresses to all other addresses. Addresses can be a multicast or unicast.

As shown in Figure 5-13, we utilize *rtpsend* and *rtpdump* to transmit and receiver RTP

packets. The command of *rtpsend* is:

rtpsend [-a] [-l] [-s *sourceport*] [-f *file*] *destination/port* [/ttl]

At the sender side, *rtpsend* transmits packets in the file which is generated by another script programme. Each line of this file includes: the length of packet load, the sequence number and an 8 bytes timestamp plan to transmit this packet. These parameters indicate the offered traffic load used to validate the channel selection algorithm. *Appendix B* shows the details of the traffic file.

At the receiver side, *rtpdump* listens on the specified port. The command is:

rtpdump [–F *format*] [-t *duration*] [-x *bytes*] [-f *file*][–o *outputfile*] *address/port*

To validate the performance, at the receiver side, we modified the *rtpdump* source code to calculate the packet delay and number of packet it received in the previous second. When this packet is received by the receiver, *rtpdump* reads the timestamp. The difference between these two timestamps is defined as the delay.

Definition 5.2: Packet delay. *The packet delay used in this thesis is defined as the time difference between the moment a packet was created by the* ***rtpsend*** *at the sender side and the moment the packet was received by the* ***rtpdump*** *at the receiver side.*

The delay includes the time waiting in the transmit queue T_{queue}, the time used to win access of the channel T_{access}, the on-air packet transmitting time $T_{transmitting}$, the time waiting in the receive queue $T_{recceive}$.

$$Delay = T_{queue} + T_{access} + T_{transmitting} + T_{receive} \quad (5\text{-}6)$$

Here $T_{receive}$ depends on the CPU processing rate of the receiver station and it is assumed that it will not change during this experiment. Where T_{access} depends on the contention level and it will increase when the station is congested. However, the impact of the change on the delay can be omitted because it is small when compared with T_{queue}. Here $T_{transmitting}$ depends on the PHY transmission rate and the frame size. It will

not change when the station become congested but will increase when the PHY transmission rate is decreased. The T_{queue} exhibits the largest change when the station become congested. We record the delay of every packet and calculate the average packet delay in every second. The PDF of the average packet delays will be presented in section 5.6.

5.5.3 Congestion status checking module

The concept of saturation and congestion were defined in section 4.1.1. If the free bandwidth of a station reaches zero, the station cannot win any more transmission opportunities to transmit its offered traffic load. As a result of this, the number of packets stored in the transmit queue will grow and will eventually reach its capacity and overflow. By comparing the number of packets arriving at the queue N_{in} and the number of packets transmitted N_{out} (which can be obtained from the *athstat* tools in the Madwifi tool set), it can determine if congestion is occurring. We calculate the difference between N_{in} and N_{out} using:

$$N_{queue} = N_{in} - N_{out} > 10 \tag{5-7}$$

Because there are some packets are stored in the queue in the hardware and some packets are retransmitted, we use this threshold condition of 10 packets is to avoid the error when calculating the number of packets being successfully transmitted. According to the definition 4.2, when this condition persists for a period of $T_{confirm} = 10$ seconds, the station is considered to be experiencing congestion. Once the congestion is confirmed by the *congestion status checking* module, this station will trigger the *channel monitoring* module which will be described in the next section.

5.5.4 Channel monitoring module

Once a station becomes congested, it needs to start the channel selection process to find a suitable channel where it can win a sufficient number of transmission opportunities.

This channel selection process is based on the channel information obtained through the *channel monitoring* module.

As shown in Figure 5-13, on the sender side, we create two VAPs in the Madwifi driver. The first one is an *Ad-hoc* VAP which is used to transmit packets. Another one is the *monitor* VAP which is used to monitor the medium. These two VAPs are not active simultaneously because wireless card is half-duplex and it cannot transmit and receive packets simultaneously.

Once the station is confirmed to have become congested, it will bring down the Ad-Hoc VAP and all the packets will be stored in the transmit queue. The monitor VAP will be brought up to collect packet and channel information. This VAP will calculate the MAC bandwidth information on all available channels in order to select a “suitable” channel. Because we do not need to select the “best” channel, once we find a channel which has sufficient free bandwidth to satisfy the traffic load of the congested station, the channel selection process will be finished and the congested station will change to that channel.

The *channel monitoring* module records the channel of each station, the packet size, the PHY transmission rate, the number of packets each second and the number of retransmission packets each second. It also calculates the load bandwidth and the access efficiency of each station, the busy bandwidth and idle bandwidth of each channel. This information will be used by the *channel changing* module and the *successful channel assignment predicting* module which will be described in the next two sections.

5.5.5 Channel changing module

This *channel changing* module implements the mechanism to select channel based on the result of the *channel monitoring* module. The channel which has sufficient free bandwidth will be selected as the operating channel. The station will not directly change the channel because it needs to maintain the connectivity between the sender and receiver. In this *channel changing* module, we implement the channel switch mechanism described in section 5.1 The sender will transmit a number of beacon frames containing the *channel switch information element* for few seconds before it changes the

channel. The receiver will change the channel immediately after it receives the beacon frame with channel switch information element.

If the result of *successful channel assignment predicting* module indicates that this station needs to force its neighbour into congestion, it will reconfigure the Ad-Hoc VAP to use the lowest PHY transmission rate of the IEEE 802.11 protocol, i.e. 1 Mbps in IEEE 802.11b or 6 Mbps in IEEE 802.11a and IEEE 802.11g.

If the result of the *successful channel assignment predicting* module indicates that there exists no successful channel assignment, the station will turn down the monitor VAP and turn up the Ad-Hoc VAP to transmit packets with the previous configuration.

5.5.6 Predicting module

When the *channel monitoring* module has monitored all the available channels and has determined that none of these channels has sufficient free bandwidth, the *predicting* module will be activated. This module will check all the possible channel assignments one by one until a successful channel assignment has been found. The time spent on executing this module will increase with the number of channels and number of stations. However, how to reduce the time spent on executing this module is out of the scope of this thesis. In the simulation and experimental testbed, because the number of stations and the number of channel is less than 6, we will check all the possible channel assignments.

5.5.7 Neighbour forcing module

The *neighbour forcing* module is the main part of the proposed channel selection algorithm. The aim of this module is to reassign the channel once a theoretical successful channel assignment has been found by the *predicting* module. Because all the stations are autonomous, it cannot directly change the channel of other stations. In this module, the congested station will reduce its PHY transmission rate to force its neighbour stations into congestion. Its neighbour stations will then independently start their own channel selection process to find a channel to satisfy their offered traffic load.

5.5.8 Performance metrics of the channel selection algorithms

In order to analyze the performance of the channel selection algorithm, we utilize the average one-way packet delay as the performance metric which is defined in section 5.5.2. The purpose of the proposed channel selection algorithm is to minimize the congestion time of the whole network. Once congestion occurs, packets will be stored (or even discarded) in the queue until the wireless interface adapter wins a sufficient number of transmission opportunities. Because we do not change the packet transmission mechanism, the packet delay is impacted by the ability of the station to win transmission opportunities. If the channel still has free bandwidth for this station to transmit its packets, the delay will remain at a low level, typically less than 5 ms. Otherwise the delay will increase to a higher level if congestion occurs.

We compare the result between a static channel assignment algorithm and a dynamic channel assignment without neighbour forcing. In this thesis, we use *STATIC* to represent the static channel assignment, use *NONF* to represent the dynamic channel assignment without neighbour forcing and use *NF* to represent the dynamic channel assignment with neighbour forcing.

All the stations which have been implemented with the *STATIC* algorithm will select the channel with sufficient available bandwidth based on the monitoring result when it starts to transmit packets in the first time. This *STATIC* algorithm does not change the operating channel during the experiment. The *NONF* algorithm does not have the neighbour forcing process, it changes channel only if it finds a channel that has sufficient free bandwidth. The delay associated with the static channel assignment and dynamic channel assignment methods also are also recorded.

We also record the congestion status of each station every second for the duration of the experiment. To analyze the congestion time, we define it as follows:

Definition 5.3: Congestion time *is defined as the number of seconds when there is at least one station that is congested.*

The congestion time is used to characterize the congestion status of the network. The channel selection algorithm fails if there are one or more stations that have become

congested. We will compare the congestion time of the three algorithms in next section.

5.6 Experimental results and discussion

To demonstrate the performance of the proposed channel selection algorithm, we have implemented the channel selection algorithm in an experimental testbed

Once a station becomes saturated, it cannot win any more transmission opportunities because there is not enough available bandwidth. The incoming packets will be stored in the transmit queue at the MAC layer. Here they will wait to be transmitted until the station wins transmission opportunities or they may be discarded. Figure 5-14 shows the relationship between the average one-way packet delay, the offered traffic load and the actual traffic load. In the experiment, there are 2 stations transmitting packets on one channel and the PHY transmission rate of both stations is 54 Mbps. One of the stations transmits with a constant throughput of 5 Mbps. This is denoted as the background traffic load. The other station increases its offered traffic load from 5.78 Mbps (i.e. 500 pps with a 1468 bytes packet size). It increases its packet rate by 20 pps every 100 seconds. The pink line presents the offered traffic load *TL_offered* every second. The offered traffic load increases over the entire test duration of 16,000 seconds. The blue line is the actual traffic load *TL_actual* which equals the offered traffic load until t = 10,000 seconds. However, it cannot win a sufficient number of transmission opportunities to transmit the offered traffic load after t = 10,500 seconds. This is the reason the actual traffic load remains about 30 Mbps when t > 10,500 seconds. The red line is the average one-way delay of a packet transmitted from sender to receiver.

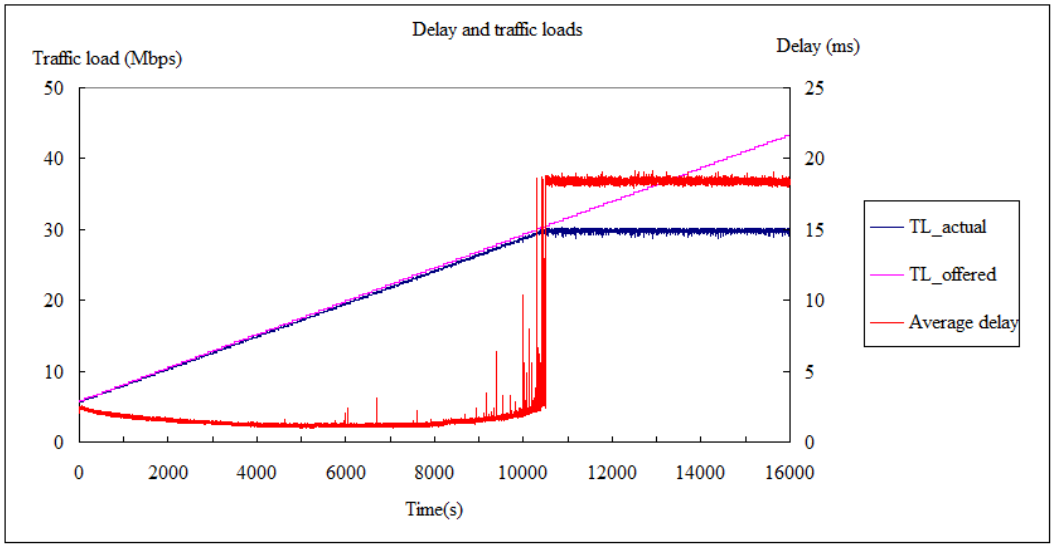


Figure 5-14 Average packet delay before and after congestion

The average one-way packet delay remains at 2 ms if the offered traffic flow is less than 30 Mbps because the free bandwidth $BW_{free} > 0$. This station can win a sufficient number of transmission opportunities to satisfy the offered traffic load. However, the delay increases dramatically to 18 ms if the offered traffic load exceeds 30 Mbps after *t* = 10,500 seconds. Because the offered traffic load is larger than the bandwidth this station can win on this channel, the rest of the packets will be stored in the transmit queue or perhaps discarded. The queue waiting time T_{queue} will increase dramatically.

If one station becomes congested, the high one-way delay will become unacceptable to applications such as the voice and video transmission. It needs to find a channel which has sufficient free bandwidth in order to transmit packets without congestion. This is the purpose of the proposed channel selection algorithm. We focus on the average one-way packet delay and the congestion time for all the stations in the network.

The proposed channel selection algorithm does not run continually to avoid wasting precious computing resources. It is triggered when a station is confirmed to be congested by the *congestion status checking* module described in section 5.5.3. Every second, the *congestion status checking* module checks the difference between the number of packets coming into the queue and the number of packets which are successfully transmitted. To avoid the impact of the environment noise such as

interference from non-IEEE 802.11 devices, a station will be confirmed as congested only when this situation persists for more than 10 seconds. This 10 seconds time duration is defined as $T_{confirm}$. It can avoid unnecessary monitoring and channel selection with a large number of $T_{confirm}$. However, a larger value of $T_{confirm}$ means the station needs to remain in a congestion status for a longer time which is contrary to the purpose of the proposed channel selection algorithm. If $T_{confirm}$ is small, the proposed channel selection algorithm can reduce the congestion time but it may initiate an unnecessary channel selection.

During all the experiments, we assumed that all the stations did not change their offered traffic load frequently. Because all the theoretical successful channel assignments are calculated from the monitored MAC bandwidth components, if the offered traffic load changes every second, other stations cannot predict the available bandwidth based on the monitoring methods. The station cannot obtain the theoretical successful channel assignment based on inaccurate channel information.

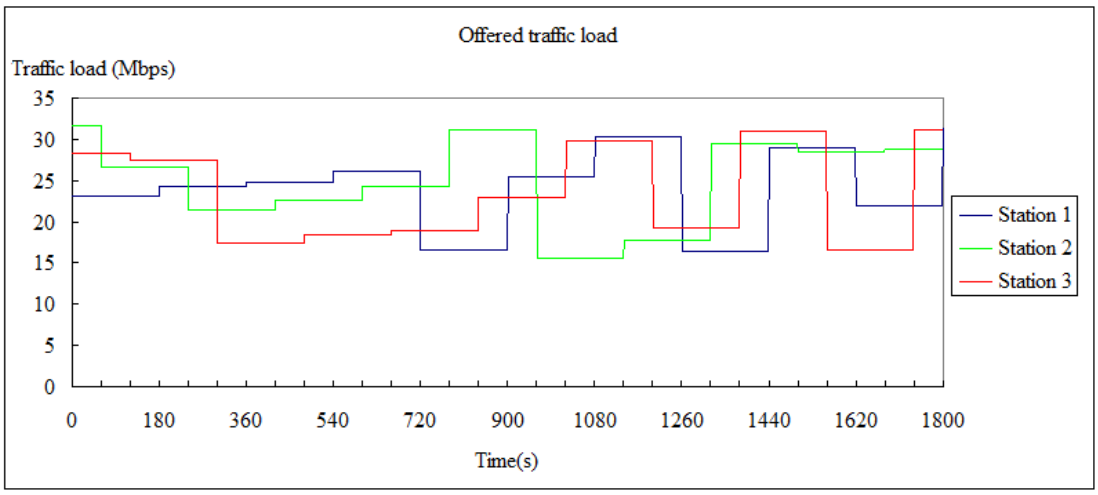


Figure 5-15 Offered traffic loads of the three stations in the experiment

Figure 5-15 shows the offered traffic load of three *Senders* we used during the experiment. We describe each 60 second interval as a "cycle", so T_{cycle} = 60 seconds. At the beginning of each cycle, one of the three *Senders* will change its traffic load and it will maintain the traffic load for the next 60 seconds. This type of traffic load pattern

will be used in all the experiments. The results for the traffic patterns for different cycle times are presented in *Appendix C*.

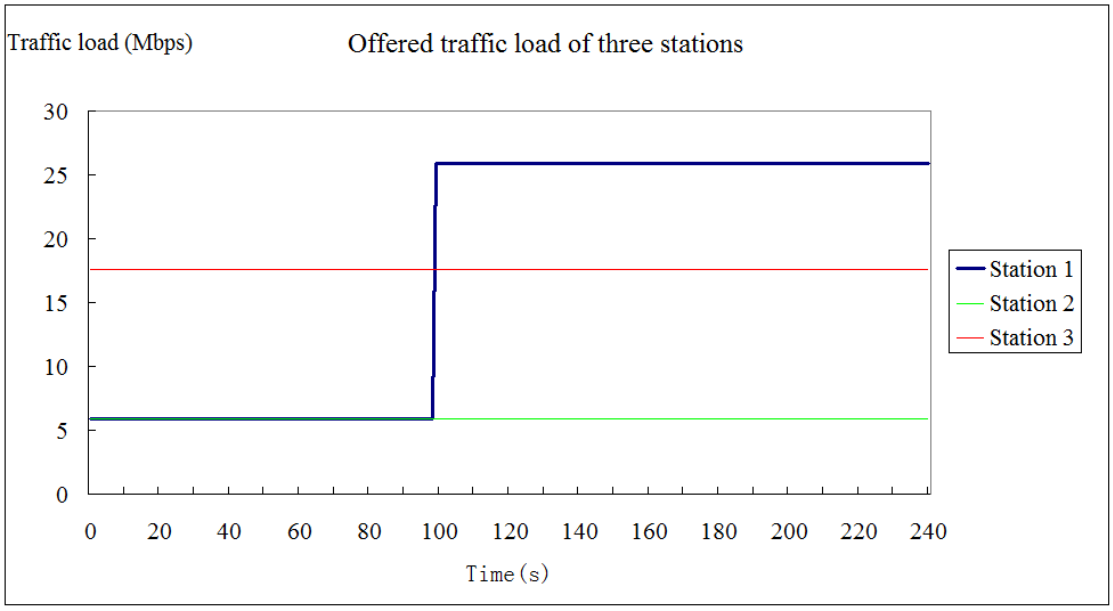
5.6.1 Average One-way packet delay of the proposed channel selection algorithm

The purpose of the experiment is to show the delay performance of the proposed channel selection algorithm. There are three different outcomes according to the *channel monitoring* module. In the next two sections, we will investigate the average one-way packet delay of the different scenarios.

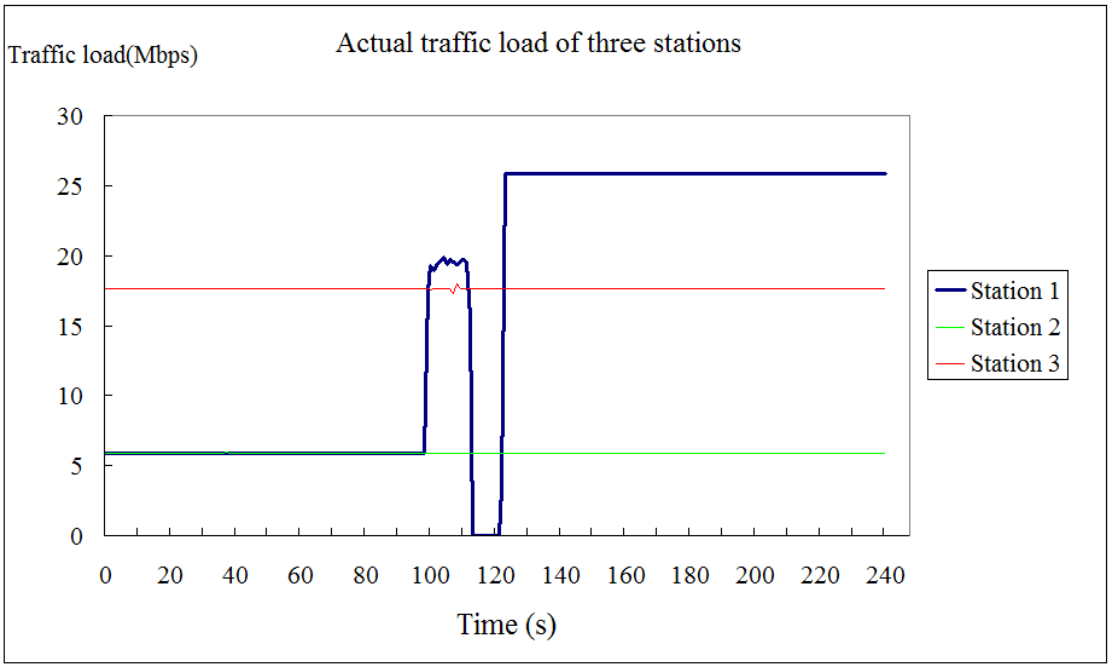
5.3.1.1 Finding a channel that has sufficient free bandwidth

The first scenario is where the congested station finds a channel that has sufficient free bandwidth. In this scenario, The *NF* algorithm and the *NONF* algorithm take the same action — they change the operating channel of the congested station to a channel which has sufficient free bandwidth.

During the experiment, each of the three stations transmits packets on one of the 2 available channels. *Station 1* and *Station 3* transmit packets on the same channel and *Station 2* transmits packets on the other channel. As shown in Figure 5-16(a), *Station 2* and *Station 3* maintain their offered traffic load during the whole cycle. *Stations 1* transmits an offered traffic load of 5.8 Mbps when t < 100 seconds. It increases its traffic load to 26 Mbps when t = 100 seconds.



(a) Offered traffic load



(b) Actual traffic load

Figure 5-16 Offered traffic load and actual traffic load of three stations (Scenario 1)

Figure 5-19(b) shows the actual traffic load of the three stations. Comparing the offered traffic loads shows that both algorithms can successfully reassign the channels to the three stations to satisfy their offered traffic loads. These two algorithms require 10 seconds to confirm that the station is congested and another 8 seconds to change the channel.

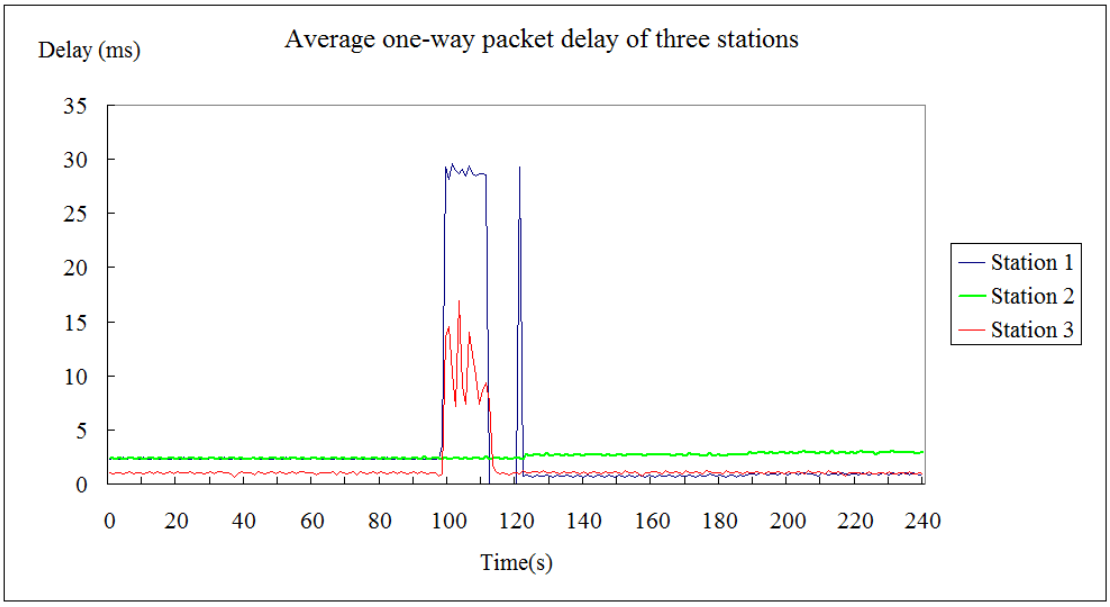


Figure 5-17 Delay of the stations when find a channel with sufficient available bandwidth

Figure 5-17 presents the average one-way packet delay before and after the channel selection when the monitoring result indicates that there exists at least one channel that has sufficient free bandwidth. It shows that if the station transmits packets on a channel which has sufficient available bandwidth, the average one-way packet delay remains at about 3 ms. However, the delay will increase dramatically from 3 ms to 28 ms once the station becomes congested. The average packet delay of other stations on the same channel such as *Station 3* will also increase slightly because the contention increases under heavy traffic loads. After about 10 seconds, the *Congestion status checking module* of *Station 1* will confirm that this station is congested. The *channel monitoring module* of *Station 1* is triggered when t = 112 seconds. It requires approximately 10 seconds to obtain the MAC bandwidth information on the two available channels. When t = 120 seconds, it changes its operating channel to the channel which has sufficient free bandwidth. Afterwards, all three stations can transmit their packets without congestion as confirmed by the average one-way packet delay of the three stations decreasing to a lower level.

The reason why the average one-way packet delay of *Station 1* increases to 30 ms when t = 120 seconds is that the station turns off the transmitting VAP to monitor the channel information and all the packets will be stored in the buffer. When the station changes to

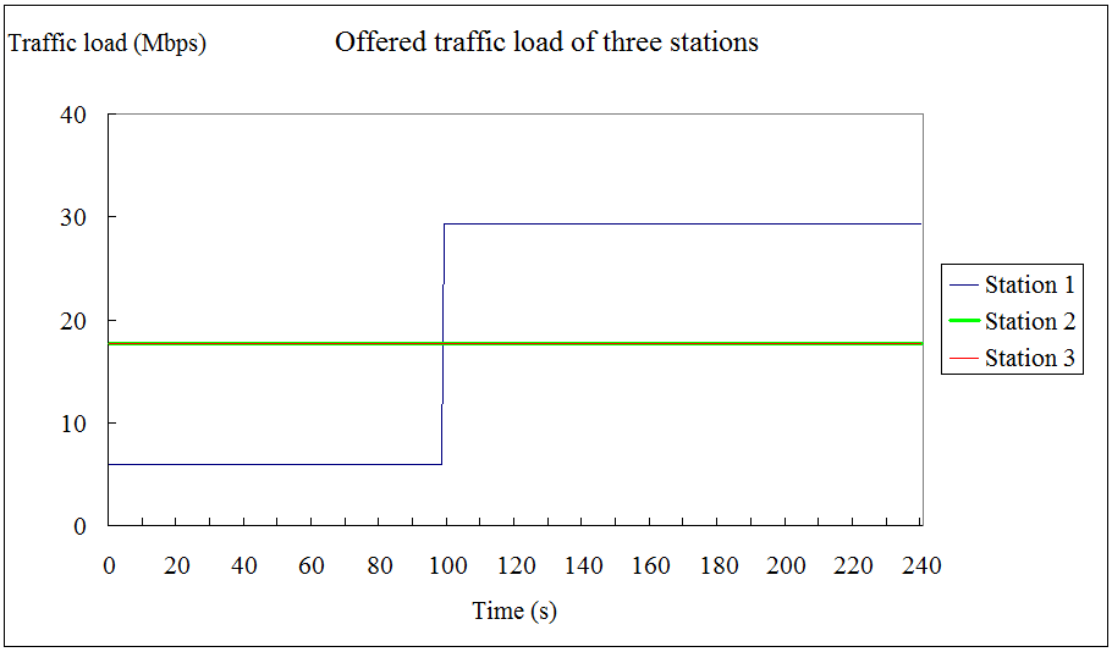
a new channel that has sufficient free bandwidth, it will flush out all the packets stored in the transmit queue and as a consequence the one-way packet delay will temporarily increase to 30 ms until all the packets in the output queue have been successfully transmitted.

In this situation, both the *NONF* algorithm and *NF* algorithm can successfully reassign the channels to avoid congestion. The congestion time will decrease also.

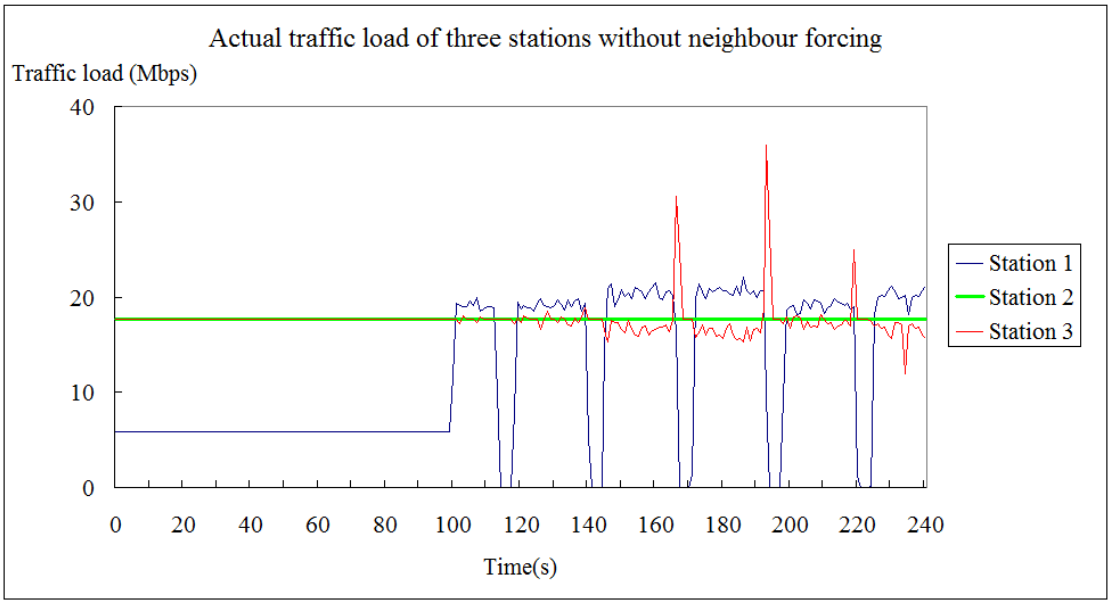
5.3.1.2 No channel has sufficient free bandwidth

The second scenario is where the *channel monitoring* module of the congested station cannot find a channel that has sufficient free bandwidth. In this situation, the *NONF* algorithm and *NF* algorithm take different actions if the *predicting* module determines that there exists at least one theoretical successful channel assignment.

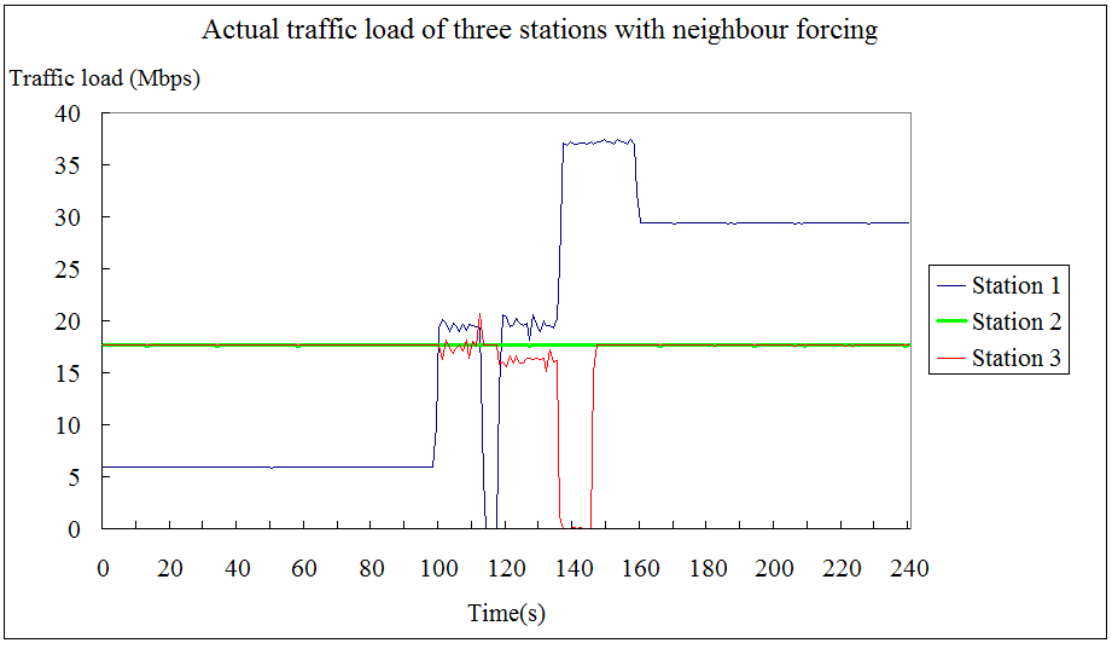
There are 3 stations transmitting packets on 2 available channels. *Station 1* and *Station 3* share the same channel and *Station 2* transmits packets on the other channel. As shown in Figure 5-18 (a), the offered traffic load of *Station 2* and *Station 3* remains the same during the 240 seconds test duration. *Station 1* will change its offered traffic load when *t* = 100 seconds from 5.8 Mbps to 29.3 Mbps.



(a) Offered traffic load



(b) Actual traffic load of the algorithm without neighbour forcing



(c) Actual traffic load of the algorithm with neighbour forcing

Figure 5-18 Offered traffic load and actual traffic load of three stations (Scenario 2)

As shown in Figure 5-18(b), the actual traffic load of all three stations equals their offered traffic load in the first 100 seconds. However, when *Station 1* changes the offered traffic load at t = 100 seconds, the actual traffic load of *Station 1* increases to 19 Mbps which is less than its offered traffic load of 29.3 Mbps. The operating channel of *Station 1* cannot support the offered traffic load of *Station 1* and *Station 3* without congestion. The *channel monitoring* module of *Station 1* cannot find a channel that has

sufficient free bandwidth and no channel change occurs. *Station 1* will transmit packets on the previous channel. Comparing the offered traffic load and actual traffic load, the *NONF* algorithm fails to reassign the channel under this scenario. The reason for the impulse in the actual traffic load of Station 3 at t = 167 seconds, 194 seconds and 220 seconds is that because its neighbour *Station 1* doesn't transmit packets when it starts the monitoring process. *Station 3* will flush out the packets stored in its transmit queue.

The *NF* algorithm takes different actions. It will trigger the *predicting* module to determine the number of theoretical successful channel assignments. *Station 1* will start the neighbour forcing process if there exists at least one theoretical successful channel assignment. The actual traffic load of the *NF* algorithm is shown in Figure 5-18(c). It shows that the *NF* algorithm can successfully reassign the channels of all three stations to transmit packets without congestion after about 50 seconds.

In the first 100 seconds of the experiment, the actual traffic load of all the three stations equals their offered traffic load. The actual traffic load of *Station 1* increases to 19 Mbps when *Station 1* changes its offered traffic load to 29 Mbps. Because the actual traffic load of *Station 1* is less than the offered traffic load, *Station 1* confirms that it becomes congested 10 seconds later. It requires about 5 seconds to obtain the channel information and then takes the decision to initiate the neighbour forcing process to force its neighbour *Station 3* into congestion when t = 118 seconds.

The actual traffic load of *Station 3* decreases during t = 118 seconds to t = 136 seconds because it cannot win a sufficient number of transmission opportunities when *Station 1* transmits packet with a lower PHY transmission rate. When t = 136 seconds, *Station 3* initiates the monitoring process to obtain the channel information because the *congestion status checking* module of *Station 3* confirms the congestion. About 10 seconds later, the *channel monitoring* module of *Station 3* discovers that the other channel has sufficient free bandwidth. *Station 3* changes its operating channel when t = 146 seconds and after that *Station 3* can win a sufficient number of transmission opportunities on the new channel.

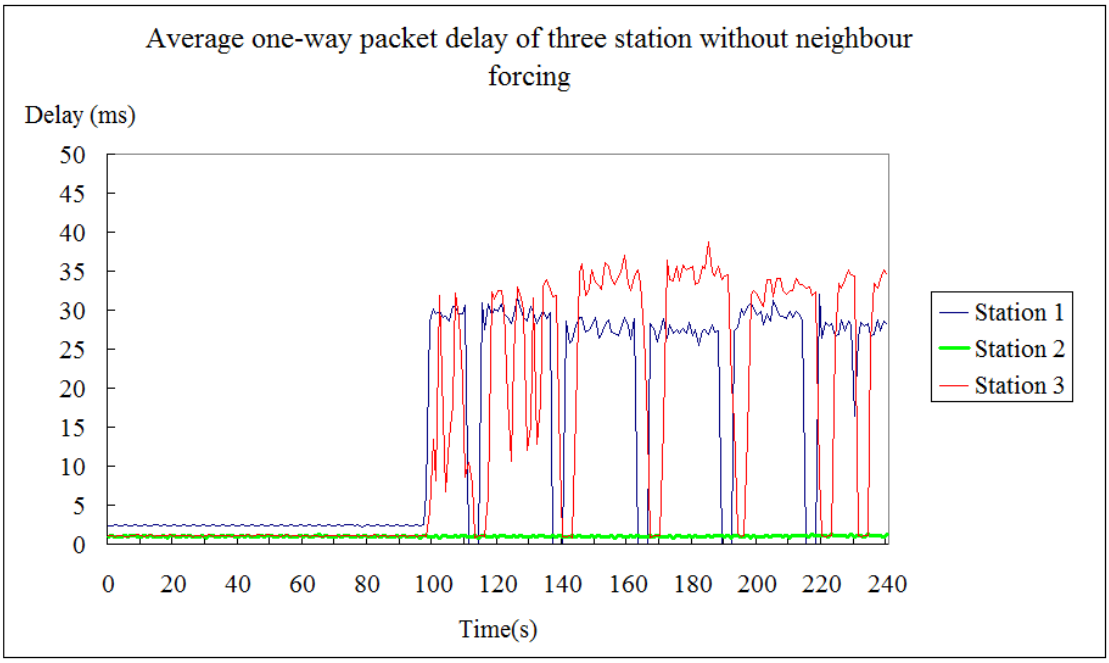


Figure 5-19 Delay of the three stations when no channel has sufficient free bandwidth

Figure 5-19 shows the average one-way packet delay of the *NONF* algorithm. Because the *NONF* algorithm cannot successfully reassign the channel under this situation, the average one-way packet delay of *Station 1* and *Station 3* remains at about 30 ms when *Station 1* increases its offered traffic load at t = 100 seconds. *Station 1* triggers the *channel monitoring* module when the *congestion checking module* of *Station 1* confirms that this station is congested again at t = 137 seconds, 164 seconds, 190 seconds and 215 seconds. However, none of these can successfully reassign the channels.

Figure 5-20 shows the delay of the *NF* algorithm with the same traffic load as shown in Figure 5-18(a).

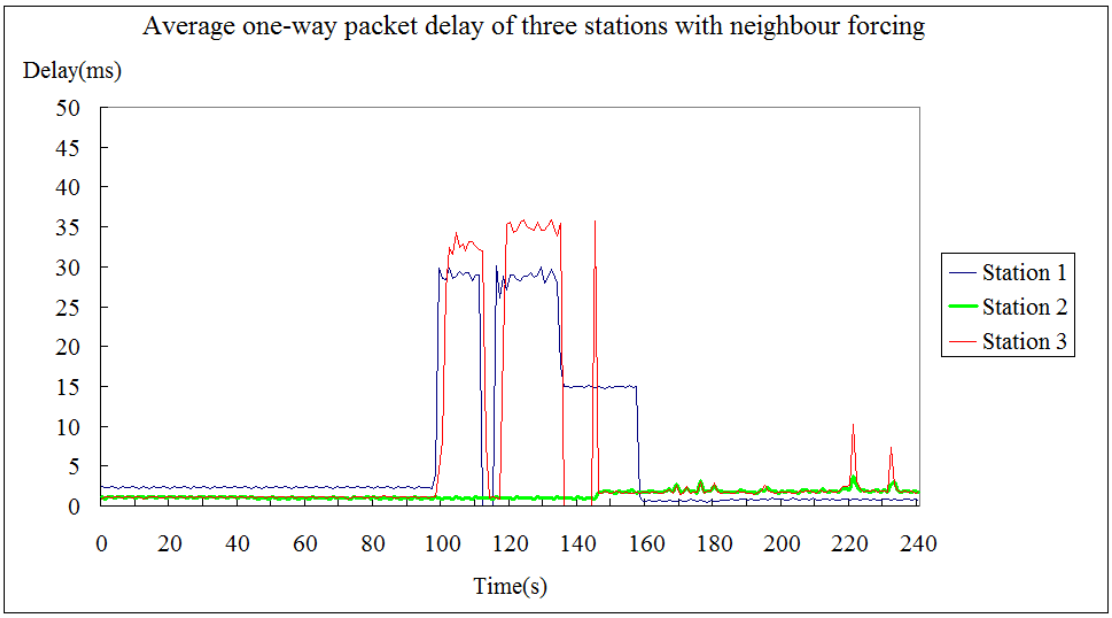


Figure 5-20 Delay of the stations when the neighbour forcing module is triggered

Figure 5-20 shows that the *NF* algorithm can reduce the congestion time under this scenario. It requires about 60 seconds to reassign the channels and the average one-way delay of all the three stations is reduced to about 2 ms.

The reason why the delay of *Station 1* remains at about 15 ms between t = 136 seconds and t = 158 seconds is that *Station 1* needs to flush the queue because many packets have been stored in the output queue when it transmits its packets at a lower PHY transmission rate.

When t = 146 seconds, the average one-way packet delay of *Station 3* increases to 35 ms. The reason is that it needs to flush out the packets stored in the output queue when it changes the channel.

Comparing the delays of the two dynamic channel assignments, which are shown in Figures 5-19 and 5-20, the proposed *NF* channel selection algorithm has a better performance in reducing the congestion time of the stations in the experimental testbed. The percentage improvement depends on how long a station maintains the high traffic load. As shown in Figure 5-19, the congestion persists if the traffic load of the three stations does not change. If the stations in the network don't change the traffic load after t = 160 seconds, the stations implemented with the proposed channel selection algorithm can transmit their packets without congestion.

5.6.2 Congestion time of the proposed channel selection algorithm

In the previous section, we saw that the proposed channel selection algorithm could decrease the congestion time in some scenarios. To analyze the performance under different traffic loads, we calculate the PDF of the average one-way packet delay.

We have calculated the number of possible channel assignments of each cycle. The result is denoted as the *theoretical analysis*. Table 5-1 shows the number of successful channel assignments and failed channel assignments for the *theoretical analysis* and the three channel assignment algorithms (i.e. *STATIC*, *NONF*, *NF*).

Table 5-1 Successful reassignment ratio for the channel switching algorithms

	Number of cycles	Failed channel assignments	Successful channel assignments	Successful reassignment ratio
Theoretical analysis	300	95	205	-
STATIC	300	185	115	0.561
NONF	300	153	147	0.717
NF	300	106	194	0.946

Table 5-1 shows that under the same traffic load, the *STATIC* channel assignment algorithm achieves a successful channel assignment for about half of the cycles. The *NONF* channel assignment algorithm increases the successful reassignment ratio to 71.7% because it can reassign the channels of stations when it finds a channel that has sufficient available bandwidth. The *NF* algorithm increases the successful reassignment ratio to 94.6% because it can reassign the channels of stations when the *predicting* module finds that there exists at least one theoretical successful channel assignment.

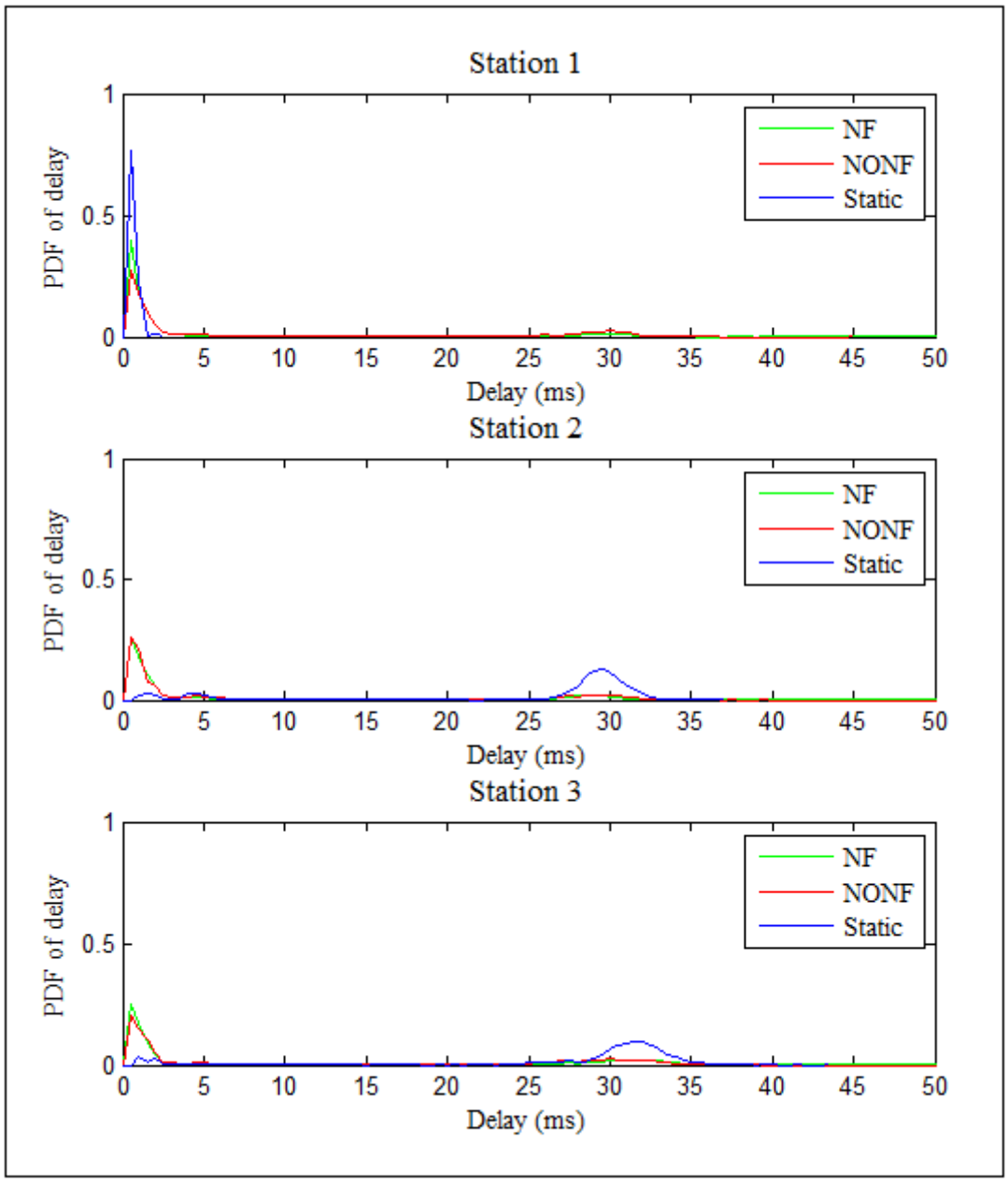


Figure 5-21 PDF of the delay of all three stations

Figure 5-21 shows the PDF of the average one-way packet delay of all three stations. We compare the three channel assignment algorithms: *STATIC*, *NONF* and *NF*. The *STATIC* algorithm does not change the channel once all the stations are assigned a channel before it starts to transmit packets. Because there are three stations that share the two available channels, the station which does not share with the other stations can transmit packets without congestion and the average one-way packet delay is less than 5 ms during the whole experiment. However, the other two stations which share one channel become congested under some of the traffic loads.

The *STATIC* channel assignment doesn't change the channel once it selects a channel to transmit packets. It has the lowest probability of achieving a successful channel assignment especially when the traffic loads of the stations increase. The *NONF* algorithm changes the channel based on the monitoring result. It achieves a successful channel assignment when the congested station finds a channel which has sufficient available bandwidth. The *NONF* algorithm fails when it cannot find a channel that has sufficient available bandwidth even where multiple theoretical successful channel assignments exist. The *NF* algorithm has a higher probability in achieving a successful channel assignment than the *NONF* algorithm based on the simulation results.

Figure 5-22 shows more clearly the PDF of the average one-way packet delay of the three algorithms. The *STATIC* algorithm has the worst performance because it cannot reassign the channel if some of the stations become congested. The *NF* algorithm has a better performance in terms of decreasing the congestion time than the *NONF* algorithm. The reason is that the *NF* algorithm can reassign the channels when there exists a theoretical successful channel assignment but the *NONF* algorithm cannot reassign the channels under this situation.

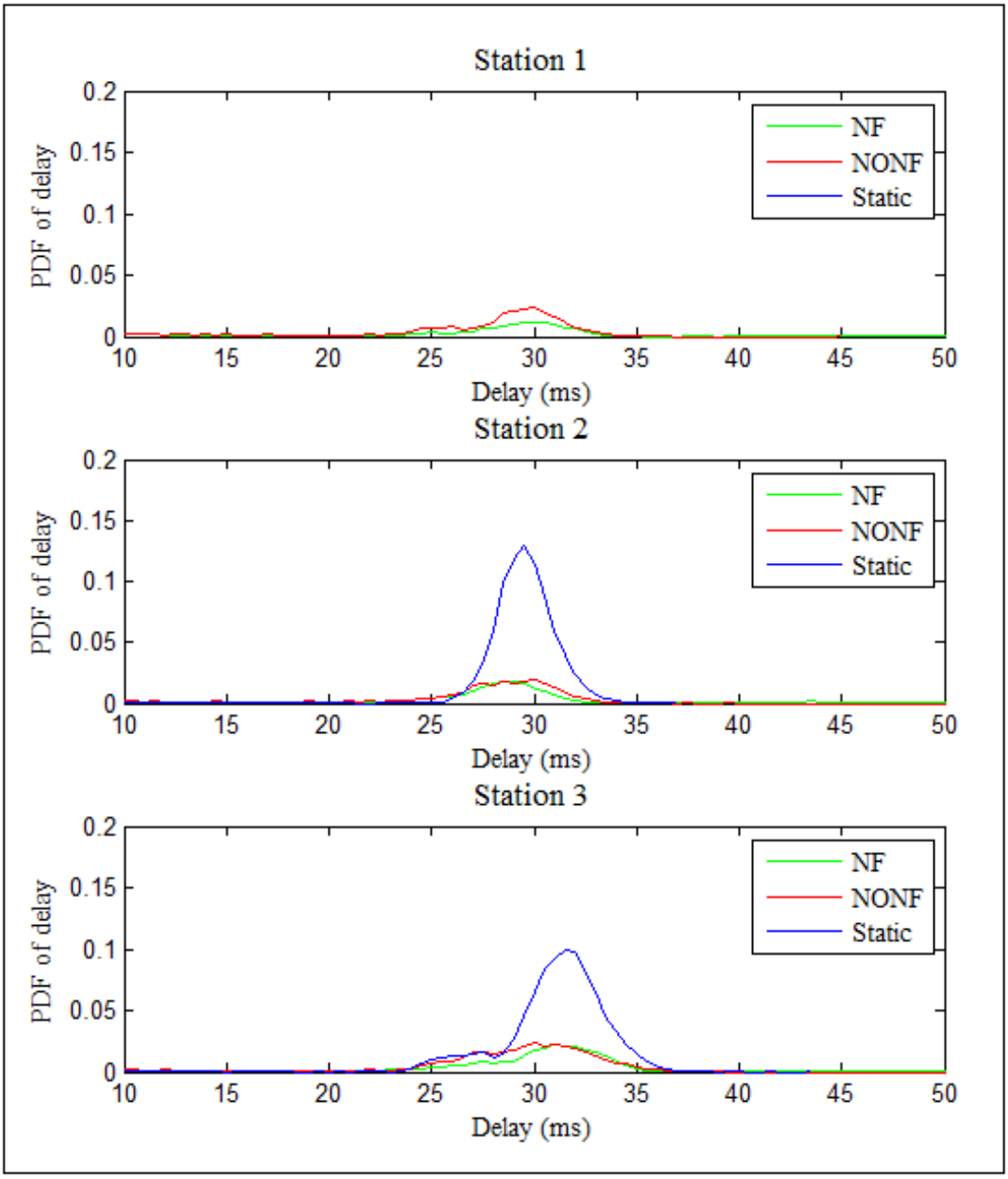


Figure 5-22 PDF of the delay related to congestion

In the simulation, when the number of stations is one greater than the number of channels, the *NF* algorithm has a successful reassignment ratio to 100%. The experiment result shows that there are some cycles where the *NF* algorithm cannot reassign the channels. The reason is that when the estimated available bandwidth is close to or equal to the offered traffic load, the predicting module indicates that no successful channel assignment exists and therefore the *NF* algorithm is not triggered.

The time spent on the channel reassignment process is different. *STATIC* channel

assignment does not change the channel once the channel is assigned. The *NONF* channel assignment needs a few seconds to collect the MAC bandwidth information of each available channel. During the monitoring process, the station cannot transmit packets because the wireless adapter can only support half-duplex operation. Figure 5-8 shows that the station requires 10 seconds to monitor the two available channels. The *NF* channel selection algorithm requires more time to complete the reassignment because the station will transmit packets at a lower rate to force the neighbour stations into congestion. However, because all the stations require a finite time interval to confirm the congestion status, the time spent on neighbour forcing should be longer than the time required to confirm the congestion status.

5.7 Benefit of the proposed channel selection algorithm

a) The proposed channel selection algorithm can successfully reassign the channel when the number of stations is one greater than the number of available channels. It can reassign the channels when there exists at least one theoretical successful channel assignment. The proposed channel selection algorithm fails only when the available bandwidth of one channel of all the possible successful channel assignments is close to zero.

b) If all the stations are implemented with the *NF* channel selection algorithm, they can operate autonomously. They make decisions based upon locally monitored channel information. This feature could be useful for the upper layer mechanisms such as routing protocols and traffic control mechanisms. For example, when one of the hops becomes congested, the station will change the channel then continue to transmit the traffic load without congestion, the routing protocol does not need to update the routing table.

c) The proposed algorithm can be used in networks when they have different SSIDs. Because each station only needs the MAC bandwidth components information and does not need to control other stations, we could deploy this algorithm in stations with different SSIDs. For example, in an urban neighbourhood setting where one household

and all their neighbours use different SSID wireless routers to support the broadband internet services, if all of them are implemented with the proposed channel selection algorithm, they could automatically reassign the channels without having to change the channel of its neighbours' router directly.

d) The proposed channel selection algorithm reduces the ripple effect associated with channel change. Because the channel change occurs only when the congested station finds a channel that has sufficient free bandwidth and the station which triggers the neighbour forcing process does not change its operating channel, the ripple effect of channel change does not occur. The ripple effect associated with neighbour forcing occurs only when the forced neighbour stations cannot find a channel that has sufficient free bandwidth.

5.8 Limitations of the autonomous channel selection algorithm

a) Because the proposed channel selection algorithm needs a few seconds to monitor the available channels and another few seconds to force its neighbour into congestion the algorithm fails when the traffic load of the station changes quickly.

b) When the traffic load of each station is similar, the successful reassignment ratio of proposed channel selection algorithm performance is poor because the station that reduces its PHY transmission rate will occupy all the free bandwidth and all the neighbour stations will be forced into a congestion status which will trigger another channel selection process. However, all the theoretical successful channel assignments show that it needs to share the channel with other stations. It is not possible for this congested station to share the channel with other stations when it reduces its PHY transmission rate to force its neighbour stations into congestion. If the traffic loads of each station are similar, it is more difficult to achieve the theoretical successful channel assignment.

5.9 Conclusion

In this chapter, we introduced an autonomous channel selection algorithm based on

neighbour forcing which dynamically reassigns the channels once the station is confirmed as being congested. Because all the stations are running in an autonomous manner, a station cannot change the channel of its neighbour stations by sending a command. In this channel selection algorithm, if there is no channel that has sufficient available bandwidth but there exists at least one theoretical successful channel assignment, the congested station will reduce its PHY transmission rate to force its neighbour stations into congestion. The congested neighbour stations will then start their own channel selection process to find a channel with sufficient available bandwidth that can satisfy their traffic loads.

In the second section, a simple C++ simulator was developed to validate the feasibility of the proposed channel selection algorithm. The results of the simulator show that under different mean traffic loads the neighbour forcing algorithm has a higher successful reassignment ratio than the dynamic algorithm without neighbour forcing.

In order to implement the channel selection algorithm, we modified the Madwifi driver to implement a two-stage method to transmit this special beacon frame in Ad-Hoc mode. In the first stage, the station transmits beacons frames containing *channel switch information element* to inform its neighbour station of the impending channel switch. In the second stage, it will change to the channel which is indicated in the channel switch information element after it transmits a number of beacon frames. The neighbours will change to the new channel after they receive the first beacon frame with the channel switch information element.

We also implemented the proposed channel selection algorithm in an experimental testbed. These 7 stations were divided into a manager, and three sender and receiver pairs. The manager station is used to maintain the time synchronization in order to calculate the delay accurately. The three sender-receiver pairs contend for two channels. We compare the results for two algorithms: static channel assignment and dynamic channel selection. The results show that the *NF* algorithm has a higher successful reassignment ratio compared to the other channel selection algorithms considered.

Chapter 6 Summary and Conclusion

The IEEE 802.11 wireless network is a contention-based network. All the stations on the same operating channel must contend with each other to win their transmission opportunities. Consequently, the capacity experienced by a station on one channel is not fixed. If a station cannot win a sufficient number of transmission opportunities to satisfy its traffic load, it will experience saturation and possibly congestion. The average one-way packet delay will increase to an unacceptable level. Channel assignment mechanisms can reassign the channels to stations in order to avoid congestion. However, if a station cannot find a channel that has sufficient available bandwidth to satisfy its offered traffic load (even though multiple successful channel assignments may exist), its congestion status cannot be solved.

In this thesis, an autonomous channel assignment mechanism has been introduced. It comprises five main modules: *congestion status checking* module, *channel monitoring* module, *channel changing* module, *predicting* module and *neighbour forcing* module. The *congestion status checking* module operates every second to check the difference between the offered traffic load and the actual traffic load to confirm whether congestion has occurred or not. The *channel monitoring* module is triggered only when congestion is confirmed. The *channel monitoring* module passively estimates the available bandwidth of each channel. Once it finds a channel that has sufficient available bandwidth, the *channel changing* module will change the channel after it sends out several beacon frames containing the channel switch information element. If there is no channel that has sufficient free bandwidth, the *predicting* module is activated to check all the possible channel assignments. Once the theoretical existence of a successful channel assignment has been determined, the *neighbour forcing* module starts to reduce its PHY transmission rate to force its neighbour stations into saturation. The neighbour stations will start their own channel selection process to find another channel to satisfy their offered traffic loads. All stations in the network make their

decisions based only on their own offered traffic load and the local channel information obtaining from their *channel monitoring* module.

A simple C++ simulator was developed to validate the feasibility of the proposed channel selection algorithm. This simulator shows that the successful reassignment ratio of the proposed channel selection algorithm has a better performance than the *NONF* algorithm in successfully reassigning the channels once congestion occurs. In particular, when the number of stations is one greater than the number of channels, the proposed channel selection algorithm can always successfully reassign the channels if there exists at least one successful channel assignment. This simulator shows that the *NF* algorithm has a higher probability to successfully reassign the channels with a larger number of theoretical successful channel assignments.

A 7 station experimental testbed was developed to investigate the average one-way packet delay and the congestion time of the channel selection algorithm. The *STATIC* algorithm cannot reassign the channel once some stations become congested. *NONF* algorithm can successfully reassign the channels if there is a channel that has sufficient free bandwidth. The *NF* algorithm has a higher probability to successfully reassign the channels if there exists at least one theoretical successful channel assignment. The *NONF* algorithm has a better performance than the *STATIC* algorithm to solve the congestion problem because it can reassign the channels of the congested station. However, in some scenarios because there is no channel that has sufficient available bandwidth for the congested station, the *NONF* algorithm cannot successfully reassign the channels even though multiple theoretical successful channel assignments may exist. Of the three algorithms investigated, the *NF* algorithm has the highest probability to successfully reassign the channels once congestion occurs.

6.1 Main achievements of the thesis

The main achievements of this thesis are:

- The development of a passive available bandwidth estimation method that can accurately estimate the available bandwidth. There is no additional traffic overhead

because this algorithm requires no probe packets.

- A *predicting* module checks all the possible channel assignment to determine not only the existence of successful channel assignment but also the number of successful channel assignments. It supports the channel selection mechanism regarding the possible outcomes of the channel reassignment. It prevents the channel selection algorithm from undertaking unnecessary channel reassignments.
- The *NF* channel selection algorithm has a higher successful reassignment ratio than the *NONF* algorithm to reassign the channels. It improves the successful reassignment ratio from 71.7% to 94.6%. In particular, if the predicting module indicates that there exists at least one theoretical channel assignment, the *NF* channel selection algorithm has a 100% reassignment ratio in reassigning the channels to stations when the number of stations is one greater than the number of channels under all traffic load condition.
- The *NF* channel selection algorithm has a higher probability to successfully reassign the channels with a larger number of successful channel assignments in the experimental testbed. When the number of successful channel assignments is greater than 15 according to the *predicting* module, the successful reassignment ratio is 100% because the congested station has a high probability to find a channel that has sufficient available bandwidth.
- All the stations which are implemented with the *NF* algorithm can reassign the channel based only on its local monitored channel information. Because it does not require network wide information, it is easy to implement the *NF* algorithm into actual networks.

6.2 Future work

The proposed channel selection algorithm operates at the MAC layer to reduce the incidence of congestion. It can reassign the channels of stations if one of the stations is confirmed as being congested. There are some other issues that should be addressed in a further investigation as follows:

- The *NF* channel selection algorithm requires only one radio to transmit packets and monitor the channel information. The time spent on the channel switch process includes the monitoring time which increases with the number of available channels. This will generate a series of problems such as a traffic spike when the station's buffer is being flushed after the channel reassignment. A possible method to reduce the time spent on the channel reassignment is to use a multi-radio mechanism [109] [110] [111]. It can configure a dedicated interface card into the monitoring mode and the other interface cards into the Ad-Hoc mode. The monitoring mode interface card will monitor all available channels periodically. Once one of the Ad-Hoc mode interfaces is confirmed as having become congested, the *NF* channel selection algorithm can estimate the available bandwidth immediately without waiting for the channel information.

- The *predicting* module uses a brute-force method to check all the N^M possible channel assignments. However, the number of possible channel assignment can be a very large number with an increase in the number of stations and channels. This may require considerable computational resources. Because the purpose of the *predicting* module is to determine whether there exists at least one successful channel assignment, how to efficiently discover whether there is at least one possible successful channel assignment certain traffic load requirement will be a challenging problem when implementing the channel selection algorithm in a large scale network. There are many restrictions that can be used to reduce the number of possible channel assignments. For example, if the predicting module takes into account that each channel should be assigned at least one station, the number of possible channel assignments that it needs to check is:

$$N^M - \binom{N}{1} \times (N-1)^M - \binom{N}{2} \times (N-2)^M - \cdots - \binom{N}{N-1} \times (N-(N-1))^M \quad (6\text{-}1)$$

This number is smaller than N^M and it will need less time to check all the possible channel assignments.

Another possible method is to order all the stations based on their access bandwidth

requirements. If the station with the largest access bandwidth requirement can share a channel with i other stations, the number of possible channel assignments will be decreased to $(N-1)^{(M-i)}$. Otherwise, if the station with the largest access bandwidth requirement cannot share a channel with other stations, the number of possible channel assignments will be decreased to $(N-1)^{(M-1)}$. This method can more quickly discover the existence of successful channel assignments.

- The *NF* algorithm fails to successfully reassign the channels when all the possible successful channel assignments from the predicting module indicate that the station which reduces its PHY transmission rate needs to share one channel with other stations. However, during the neighbour forcing process, the congested station reduce its PHY transmission rate to the lowest PHY transmission rate of the IEEE 802.11 protocol, i.e. 1 Mbps in IEEE 802.11b or 6 Mbps in IEEE 802.11a and IEEE 802.11g deployments. This action will lead to a problem that the channel selection algorithm fails to successfully reassign the channels when the congested station needs to share one channel with other stations. A possible method for solving this problem is to reduce the PHY transmission rate in small steps, such as reducing to 48 Mbps or 36 Mbps instead of 6 Mbps (in the case of IEEE 802.11a/g networks). According to the neighbour forcing process described in section 4.2.2, all the stations will be forced into saturation if their access bandwidth requirements are larger than the new access bandwidth requirement of the station which is reducing its PHY transmission rate. Reducing the PHY transmission rate in smaller steps can force part of the neighbour stations into saturation and the stations which have a smaller access bandwidth will remain on their channel. In other words, the station which triggers the neighbour forcing will share the channel with other stations after it reduces its PHY transmission rate. This may achieve a successful channel assignment.
- The development of the open source wireless device driver Madwifi has been stopped. It is highly dependent on the proprietary *HAL* [112] which acts as a wrapper around the hardware registers. *Ath5k* [37] is a completely FOSS Linux

driver for Atheros wireless cards. It is based on Madwifi and the *OpenHAL* [113]. It can call hardware functions directly. The *NF* channel selection algorithm uses the difference between the number of packets arriving into the transmit queue and the number of packets successfully transmitted out of the queue to confirm the congestion status. With the help of *OpenHAL*, the depth of transmit queue can be easily obtained from the hardware. Combining the *NF* channel selection algorithm with the *Ath5k* wireless driver could be a more efficient method to obtain the information on the buffer occupancy.

- During the experimental test, we assume that there are no hidden stations present and all stations can hear each other. However, hidden nodes can cause many performance problems, including unfair throughput distribution among flows and throughput degradation etc. [114]. How the bandwidth estimation algorithm and channel selection algorithm will perform when there are hidden stations present should be a further topic for investigation. The bandwidth estimation algorithm needs to be modified to improve the accuracy considering the impact of hidden stations.
- Security is a big challenge in wireless network especially in an autonomous network. Even though IEEE 802.11i [115] implemented as WPA2 specifies security mechanisms for wireless networks. If the attacker floods the network with dummy packets to force other stations which have been implemented with the proposed channel selection algorithm into saturation and then these victim stations will initiate their channel selection process. This will lead to continuous channel switching if there is one such attacking station on each available channel. One possible method is to use an access control list based upon the MAC address, i.e. a white-list of known and trusted network stations and a black-list for unknown stations. The channel selection algorithm will only be triggered when the station on white-list forces it into saturation.

References

[1] Liang Zhou; Xinbing Wang; Wei Tu; Muntean, G.; Geller, Benoit.; "Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 3, pp. 409-419, Apr. 2010.

[2] Akl, R.; Arepally, A.; "Dynamic Channel Assignment in IEEE 802.11 Networks," *Portable Information Devices (PORTABLE '07), IEEE Int. Conf. on*, Orlando, FL, USA, pp. 1-5, May 25-29, 2007.

[3] A. Raniwala. K. Gopalan. T. Chiueh.; "Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks"; *ACM Mobile Computing and Communications Review (MC2R)*, ACM, NY, USA, vol. 8, Issue. 2, pp. 50-65, Apr. 2004.

[4] Sumit Rangwala, Apoorva Jindal, Ki-Young Jang, Konstantinos Psounis, and Ramesh Govindan.; "Understanding congestion control in multi-hop wireless mesh networks," *Mobile computing and networking (MobiCom '08), in Proc. 14th ACM Int. Conf. on*, San Francisco, CA, USA, pp. 291-302, Sept. 14-19, 2008.

[5] Bong-Jun Ko; Misra, V.; Padhye, J.; Rubenstein, D.; "Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks," *Wireless Communications and Networking Conference (WCNC '07)*, Kowloon, Hong Kong, China, pp. 3978-3983, Mar. 11-15, 2007.

[6] Leith, D.J.; Clifford, P.; "A Self-Managed Distributed Channel Selection Algorithm for WLANs," *Modelling and Optimization in Mobile, Ad Hoc and Wireless*

Networks, 4th Int. Symposium on, Boston Massachusetts, USA, pp. 1-9, Apr. 3-6, 2006.

[7] Mehmet S. Kuran, Tuna Tugcu.; "A survey on emerging broadband wireless access technologies," *Computer Networks*, vol. 51, Issue 11, pp. 3013-3046, Aug. 2007.

[8] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1-2793, March 29 2012, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6178212&isnumber=6178210 (Last accessed on 14 Oct. 2013)

[9] Matthew S Gast. 802.11 Wireless Networks: The Definitive Guide, Second Edition, O'Reilly, 2005. ISBN: 0-596-00183-5

[10] IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*, pp. 1-372, Sept. 10 2011, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6018236&isnumber=6018235 (Last accessed on 14 Oct. 2013)

[11] Akyildiz, I.F.; Xudong Wang.; "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23- S30, Sept. 2005.

[12] Prehofer, C.; Bettstetter, C.; "Self-organization in communication networks: principles and design paradigms," *Communications Magazine, IEEE,* vol. 43, no. 7, pp. 78- 85, July 2005.

[13] Kyu-Han Kim; Shin, K.G.; "Self-Reconfigurable Wireless Mesh Networks," *IEEE/ACM Transactions on Networking, IEEE,* vol. 19, no. 2, pp. 393-404, Apr. 2011.

[14] P. Kyasanur and N. Vaidya.; "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," *Mobile Computing and Networking (MobiCom '05), in Proc. 11th Annu. Int. Conf. on,* Cologne, Germany, pp. 43–57, Aug. 28- Sept. 02, 2005.

[15] Basagni, S.; Nidito, F.; Farago, A.; "The multi-radio advantage," *Radio and Wireless Symposium (RWS '09), in Proc. 4th Int. Conf. on,* San Diego, CA, USA, pp. 451-454, Jan. 14, 2009.

[16] Supplement to IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," IEEE Std 802.11b-1999,pp. 1-90, 2000, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=817038&isnumber=17714 (Last accessed on Oct. 14, 2013)

[17] Draft Supplement to Standard [for] Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and

Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz band (Amendment to IEEE Std 802.11, 1999 Edition)," IEEE Std P802.11g/D8.2, 2003, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4040924&isnumber=4040923 (Last accessed on Oct. 14, 2013)

[18] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band, IEEE Std 802.11, 1999, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=815305 (Last accessed on Oct. 14, 2013)

[19] Der-Jiunn Deng; Chih-Heng Ke; Hsiao-Hwa Chen; Yueh-Min Huang.; "Contention window optimization for IEEE 802.11 DCF access control," *Wireless Communications, IEEE Transactions on*, vol.7, no.12, pp.5129-5135, Dec. 2008.

[20] Ksentini, A.; Nafaa, A.; Gueroui, A.; Naimi, M.; "Determinist contention window algorithm for IEEE 802.11," *Personal, Indoor and Mobile Radio Communications (PIMRC '05), IEEE 16th Int. Symposium on*, Berlin, Germany, vol. 4, pp. 2712-2716, Sept. 11-14, 2005.

[21] Dongxia Xu; Sakurai, T.; Vu, H.L.; "An Analysis of Different Backoff Functions for an IEEE 802.11 WLAN," *Vehicular Technology Conference (VTC 2008-Autumn)*, Calgary, BC, pp. 1-5, Sept. 21-24, 2008.

[22] Hao-Ming Liang; Zeadally, S.; Chilamkurti, N.K.; Ce-Kuen Shieh.; "A Novel Pause Count Backoff Algorithm for Channel Access in IEEE 802.11 Based Wireless LANs,", *Computer Science and its Applications (CSA '08), Int. Symposium on,* Wrestpoint Hotel, Hobart, Australia, pp.163-168, Oct. 13-15, 2008.

[23] IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe," IEEE Std 802.11h-2003 (Amendment to IEEE Std 802.11, 1999 Edition, (Reaff 2003)), pp. 0_1-59, 2003, [Online]. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1243739&isnumber=27870 (Last accessed on Oct. 15, 2013)

[24] J.Bricket, “Bit-rate Selection in Wireless Networks,” Master's thesis, Dept. Elec. Eng. Computer Sci., MIT, Cambridge, MA, 2005, [Online]. Available at: http://pdos.csail.mit.edu/papers/jbicket-ms.pdf (Last accessed on Oct. 21 2013)

[25] Kamerman, A. and Monteban, L.; “WaveLAN®-II: a high-performance wireless LAN for the unlicensed band,” *Bell Labs Tech.* vol. 2, issue. 3, pp. 118–133, Autumn 1997.

[26] M. Lacage, M. Hossein and T. Turletti.; “IEEE 802.11 Rate Adaptation: A Practical Approach,” *Modeling, Analysis and Simulation of Wireless and Mobile systems (MSWIM '04), in Proc. 7th ACM Int. Symposium on*, Venice, Italy, pp. 126-134, October 4-6, 2004.

[27] Alzate, M.A.; Pena, N.M.; Labrador, M.A.; "Capacity, bandwidth and available bandwidth concepts for wireless ad hoc networks," *Military Communications Conference (MILCOM '08)*, San Diego, CA, USA, pp. 1-7, Nov. 17-19, 2008.

[28] Gupta, P.; Kumar, P.R.; "The capacity of wireless networks," *Information Theory*, IEEE Transactions on, vol. 46, no. 2, pp. 388-404, Mar. 2000.

[29] Murali Kodialam, Thyaga Nandagopal,; “Characterizing the capacity region in multi-radio multi-channel wireless mesh networks,” *Mobile computing and networking (MobiCom '05), in Proc. 11th Annual Int. Conf. on*, Cologne, Germany, pp. 73-87, Aug. 28-Sept. 2, 2005.

[30] Pradeep Kyasanur, "Multi-Channel Wireless Networks: Capacity and Protocols," Ph.D. dissertation, Computer Science, Graduate College, Univ. Illinois, Urbana Champaign, 2006, [Online]. Available at: https://www.ideals.illinois.edu/bitstream/handle/2142/11233/Multichannel%20Wireless%20Networks%20Capacity%20and%20Protocols.pdf?sequence=2 (Last accessed on Oct. 21 2013)

[31] Gupta, N.; Das, S.R.; "A capacity and utilization study of mobile ad hoc networks," *Local Computer Networks (LCN '01), in Proc. 26th Annual IEEE Conf. on*, Tampa, FL, USA, pp. 576-583, Nov. 14-16, 2001.

[32] Shila, D.M.; Yu Cheng; Anjali, T.; "Capacity of Cooperative Wireless Networks Using Multiple Channels," *Communications (ICC '10), 2010 IEEE Int. Conf. on*, Cape Town, South Africa, pp. 1-5, May 23-27, 2010.

[33] Mark Davis; “A wireless traffic probe for radio resource management and QoS provisioning in IEEE 802.11 WLANs,” *ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, Venezia, Italy, pp. 234-243, Oct. 4-6, 2004.

[34] Davis, M.; Raimondi, T.; "A novel framework for radio resource management in IEEE 802.11 wireless LANs," *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, (WIOPT '05), Third Int. Symposium on*, pp. 139- 147, Apr. 3-7, 2005.

[35] Existing Linux wireless driver, [Online], Available at: http://linuxwireless.org/en/users/Drivers (Last accessed at Oct. 15, 2013).

[36] Madwifi, [Online], Available at: http://madwifi-project.org/ (Last accessed at Oct. 15, 2013).

[37] Ath5k, [Online], Available at: http://wireless.kernel.org/en/users/Drivers/ath5k (Last accessed at Oct. 15, 2013).

[38] Ath9k, [Online], Available at: http://wireless.kernel.org/en/users/Drivers/ath9k (Last accessed at Oct. 15, 2013)

[39] Vipin, M.; Srikanth, S.; "Analysis of open source drivers for IEEE 802.11 WLANs," *Wireless Communication and Sensor Computing (ICWCSC '10), Int. Conf. on*, Chennai, Tamilnadu, India, pp.1-5, Jan. 2-4, 2010.

[40] Onoe, [Online], Available at: http://madwifi-project.org/wiki/UserDocs/RateControl (Last accessed at Oct. 15, 2013).

[41] Sarr, C.; Chaudet, C.; Chelius, G.; Lassous, I.G.; "Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks," *Mobile Computing, IEEE Transactions on*, vol.7, no.10, pp.1228-1241, Oct. 2008.

[42] Ningning Hu and Peter Steenkiste; “Estimating Available Bandwidth Using Packet Pair Probing,” School of Computer Science, Carnegie Mellon University, Pittsbutgh, PA, USA, Tech. Rep. CMU-CS-02-166, 2002, [Online]. Available at: http://www.cs.cmu.edu/~hnn/papers/igi-tr.pdf (Last accessed on Oct. 21, 2013)

[43] Aditya Dhananjay, Hui Zhang, Jinyang Li, and Lakshminarayanan Subramanian;

“Practical, distributed channel assignment and routing in dual-radio mesh networks,” *Special Interest Group on Data Communication (SIGCOMM '09), in Proc. on*, Barcelona, Spain, pp. 99-110, Aug. 17-21, 2009.

[44] Sarr, C.; Chaudet, C.; Chelius, G.; Lassous, I.G.; , "A node-based available bandwidth evaluation in IEEE 802.11 ad hoc networks," *Parallel and Distributed Systems (ICPADS '05), in Proc., 11th Int. Conf. on*, Fuduoka, Japan, vol.2, pp. 68-72, July 20-22, 2005.

[45] Cheikh Sarr; Claude Chaudet; Guillaume Chelius; Isabelle Guerin Lassous; "Improving Accuracy in Available Bandwidth Estimation for IEEE 802.11-based Ad Hoc Networks," *Mobile Adhoc and Sensor Systems (MASS '06), IEEE Third Int. Conf. on*, pp. 517-520, Vancouver, BC, Canada, Oct. 9-12, 2006.

[46] Khan, M.A.Y.; Veitch, D.; "Speedo: Realistic achievable bandwidth in 802.11 through passive monitoring," *Local Computer Networks (LCN '08), 33rd IEEE Conf. on*, Quebec, Canada, pp. 892-899, Oct. 14-17, 2008.

[47] Tursunova, S.; Inoyatov, K.; Young-Tak Kim; "Cognitive passive estimation of available bandwidth (cPEAB) in overlapped IEEE 802.11 WiFi WLANs," *Network Operations and Management Symposium (NOMS '10), 12th IEEE/IFIP on*, Osaka, Japan, pp. 448-454, Apr. 19-23, 2010.

[48] Hyung Joon Park; Byeong-hee Roh; “Accurate Passive Bandwidth Estimation (APBE) in IEEE 802.11 Wireless LANs,” *Ubiquitous Information Technologies & Applications (CUTE '10), in Proc. 5th Int. Conf. on*, Sanya, Hainan, China, pp. 1-4, Dec. 16-18, 2010.

[49] Peng Zhao; Xinyu Yang; Chiyong Dong; Shusen Yang; Bhattarai, S.; Wei Yu; "On an Efficient Estimation of Available Bandwidth for IEEE 802.11-Based Wireless

Networks," *Global Telecommunications Conference (GLOBECOM '11), 54th Annual Conf. on*, Houston, Texas, USA, pp. 1-5, Dec. 5-9, 2011.

[50] Guerin, J.; Glass, S.; Peizhao Hu; Wee Lum Tan; Portmann, M.; "Time-based and low-cost bandwidth estimation for IEEE 802.11 links," *Wireless Communications and Mobile Computing Conference (IWCMC '12), 8th Int. Conf.* on, Limassol, Cyprus, pp. 251-256, Aug. 27-31, 2012.

[51] Jacobson, V.; "Congestion avoidance and control," *Communications architectures and protocols (SIGCOMM '88), in Proc. on*, Stanford, CA, USA, pp. 314-329, Aug. 16-18, 1988.

[52] Ghittino, A.; Di Maio, N.; Di Tommaso, D.; "WiFi network residual bandwidth estimation: A prototype implementation," *Wireless On-demand Network Systems and Services (WONS '12), 9th Annual Conf. on*, Courmayeur, Italy, pp. 43-46, Jan. 9-11, 2012.

[53] Amamra, Abdelaziz; Hou, Kun Mean; "SLOT: A Fast and Accurate Technique to Estimate Available Bandwidth in Wireless IEEE 802.11," *Computer Modeling and Simulation (UKSIM '08), in Proc. 10th EUROS/UKSim Int. Conf. on*, Cambridge Univ., Emmanuel College, Cambridge, UK, pp. 46-51, Apr. 1-3, 2008.

[54] Melander, B.; Bjorkman, M.; Gunningberg, P.; "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," *Global Telecommunications Conference (GLOBECOM '00), in Proc. of*, San Francisco, CA, USA, vol. 1, pp. 415-420, Nov. 27-Dec. 1, 2000.

[55] Mingzhe Li; Claypool, M.; Kinicki, R.; "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks," *Local Computer Networks (LCN '08), 33rd IEEE Conf. on*, Quebec, Canada, pp. 374-381, Oct. 14-17, 2008.

[56] Vinh Dien Hoang; Shao, Z.; Fujise, M.; "A New solution to Estimate the available Bandwidth in MANETs," *Vehicular Technology Conference (VTC '06-Spring)*, In Proc. IEEE 63rd, Melbourne, Australia, vol. 2, pp. 653-657, May 7-10, 2006.

[57] Ekelin, S.; Nilsson, M.; Hartikainen, E.; Johnsson, A.; Mangs, J.-E.; Melander, B.; Bjorkman, M.; "Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering," N*etwork Operations and Management Symposium (NOMS '06), 10th IEEE/IFIP*, Vancouver, Canada, pp.73-84, Apr. 3-7, 2006.

[58] Zhenhui Yuan; Venkatarama.; H.; Muntean, G.-M.; "A novel bandwidth estimation algorithm for IEEE 802.11 TCP data transmissions," *Wireless Communications and Networking Conference Workshops (WCNCW '12), IEEE*, Paris, France, pp. 377-382, Apr. 1-4, 2012.

[59] Qinghui Wang; Ansong Feng; Jingxing Cao; "Available Bandwidth Estimation in IEEE 802.11 Ad Hoc Networks," *Hybrid Intelligent Systems (HIS '09), Ninth Int. Conf. on*, Shengyang, China, vol. 1, pp. 135-137, Aug.12-14, 2009.

[60] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, “Distributed channel management in uncoordinated wireless environments”, *Mobile Computing and Networking (MobiCom '06), in Proc. Twelfth Annual Int. Conf. on*, Los Angeles, California, USA, pp. 170-181, Sept. 23-29, 2006.

[61] Zhioua, G.E.M.; Tabbane, N.; "A load and QoS aware scheduling algorithm for multi-channel multi-radio wireless mesh networks," *Wireless Communications and Networking Conference (WCNC '12)*, Paris, France, pp. 2043-2047, Apr. 1-4, 2012.

[62] Das, A.K.; Vijayakumar, R.; Roy, S.; "WLC30-4: Static Channel Assignment in Multi-radio Multi-Channel 802.11 Wireless Mesh Networks: Issues, Metrics and

Algorithms," *Global Telecommunications Conference (GLOBECOM '06), IEEE 49th Annual conf. on*, San Francisco, FL, USA, pp. 1-6, Nov. 27-Dec. 1, 2006.

[63] Ibrahim, A.; Roberts, J.; "A traffic-oriented approach for channel assignment in WLAN," *21st International Teletraffic Congress (ITC '09)*, Paris, France, pp. 1-8, Sept. 15-17, 2009.

[64] Agrawal, D.; Mishra, A.; Springborn, K.; Banerjee, S.; Ganguly, S.; "Dynamic interference adaptation for wireless mesh networks," *Wireless Mesh Networks (WiMesh '06), 2nd IEEE Workshop on*, Reston, Virginia, USA, pp. 33-37, Sept. 25-28, 2006.

[65] Weifeng Sun; Rong Cong; Feng Xia; Xiao Chen; Zhenquan Qin; "R-CA: A Routing-Based Dynamic Channel Assignment Algorithm in Wireless Mesh Networks," *Ubiquitous Intelligence & Computing and Autonomic & Trusted Computing (UIC/ATC '10), in Proc. 7th Int. Conf. on*, Xi'an, Shanxi, China, pp. 228-232, Oct. 26-29, 2010.

[66] Guang-Hua Yang; Haitao Zheng; Jun Zhao; Li, V.O.K.; "Adaptive Channel Selection Through Collaborative Sensing," I*nternational Conference Communications (ICC '06), IEEE on*, Istanbul, Turkey, vol. 8, pp. 3753-3758, June 11-15, 2006.

[67] Haidar, M.; Al-Rizzo, H.; Yupo Chan; Akl, R.; Bouharras, M.; "Throughput Validation of an Advanced Channel Assignment Algorithm in an IEEE 802.11 WLAN," *Communication Software and Networks (ICCSN '09), Int. Conf. on*, Macau, China, pp. 801-806, Feb. 27-28, 2009.

[68] H. Skalli, S K. Das, L. Lenzini, M. Conti; "Traffic and interference aware channel assignment for multi-radio wireless mesh networks", Dept. Comp. Sci. Eng., IMT,

Lucca, Italy, Tech. 2006, [Online]. Available at: https://www2.imtlucca.it/_documents/publications/publication54-9369_Technical_report_2006.pdf (Last accessed on Oct. 21, 2013)

[69] Ashish Raniwala; Tzi-cker Chiueh; "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," *Computer and Communications Societies (INFOCOM '05), Proc. IEEE 24th Annual Joint Conf. on*, Miami, FL, USA, vol.3, pp. 2223- 2234, Mar. 13-17, 2005.

[70] Sok-Hyong Kim; Young-Joo Suh; "Local Channel Information Assisted Channel Assignment for Multi-Channel Wireless Mesh Networks," *67th Vehicular Technology Conference (VTC Spring '08), Proc. IEEE*, Singapore, pp. 2611-2615, May 11-14, 2008.

[71] E. Garcia Villegas; R.Vidal Ferro; J. Paradells Aspas; "Implementation of a Distributed Dynamic Channel Assignment Mechanism for IEEE 802.11 Networks," *Personal, Indoor and Mobile Radio Communications (PIMRC '05), IEEE 16th Int. Symposium on*, Berlin, Germany, vol. 3, pp. 1458-1462, Sept. 11-14, 2005.

[72] Jeonghoon Mo; Hoi-Sheung Wilson So; Jean Walrand; "Comparison of Multichannel MAC Protocols," *Mobile Computing, IEEE Transactions on*, vol.7, no.1, pp.50-65, Jan. 2008.

[73] Jenhui Chen; Shiann-Tsong Sheu; Chin-An Yang; "A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs," *Personal, Indoor and Mobile Radio Communications (PIMRC '03), 14th IEEE Proc.* on, Beijing, China, vol.3, pp. 2291- 2296, Sept. 7-10, 2003.

[74] Jungmin So and Nitin Vaidya; “Multi-Channel MAC for Ad Hoc Networks:

Handling Multi-Channel Hidden Terminals Using a Single Transceiver," *Mobil Ad Hoc Networking and Computing (MobiHoc '04), Proc. 5th ACM Int. Symposium on*, Tokyo, Japan, May 24-26, 2004.

[75] Asimakis Tzamaloukas and J.J. Garcia-Luna-Aceves; "Channel-hopping multiple access," *Communications (ICC '00), IEEE Int. Conf. on*, New Orleans, Louisiana, USA, vol. 1, pp. 415-419, June 18-22, 2000.

[76] Asimakis Tzamaloukas and J.J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access with Packet Trains for Ad Hoc Networks," *Mobile Multimedia Communications (MoMuC '00), Proc. 7th Int. Work on*, Tokyo, Japan, Oct. 23-26, 2000.

[77] Paramvir Bahl, Ranveer Chandra, and John Dunagan; "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," *ACM Mobile computing and networking (MobiCom '04). Proc. 10th annual Int. Conf. on*, Philadelphia, Pennsylvania, pp. 216-230, Sept. 26 - Oct. 1, 2004.

[78] Hoi-Sheung Wilso So; Jean Walrand; Jeonghoon Mo; "McMAC: A Multi-Channel MAC Proposal for Ad Hoc Wireless Networks," *Wireless Communications and Networking Conference (WCNC '07), Proc. Hong Kong*, China, pp. 334-339, Mar. 11-15, 2007.

[79] Shih-Lin Wu; Chih-Yu Lin; Yu-Chee Tseng; Jang-Laing Sheu; "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," *Parallel Architectures, Algorithms and Networks (I-SPAN '00), Proc. 5th Int. Symposium on*, Dallas/Richardson, TX, USA, pp.232-237, Dec. 7-10, 2000.

[80] Peng-Jung Wu and Chung-Nan Lee; “Connection-oriented multi-channel MAC protocol for ad-hoc networks,” *Computer Communication,* vol. 32, no. 1, pp. 169-178, Jan. 23, 2009.

[81] Papadopouli, Maria and Porwal, Priyank; "On-demand channel switching for multi-channel wireless MAC protocols," *Wireless Conference 2006 - Enabling Technologies for Wireless Multimedia Communications (European Wireless), 12th European*, Athens, Greece, pp. 1-9, Apr. 2-5, 2006.

[82] Martin Heusse; Franck Rousseau; Gilles Berger-Sabbatel; Andrzej Duda; "Performance anomaly of 802.11b," *22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM ‘03)*, San Francisco, CA, USA, vol.2, pp. 836-843, Mar. 30-Apr. 3, 2003.

[83] Sook-Chin Yip; Su-Wei Tan; Teong-Chee Chuah; "Data Rate-Aware Channel Assignment algorithm for multi-rate multi-channel wireless mesh networks," *Wireless and Pervasive Computing (ISWPC '11), 6th Int. Symposium on*, Hong Kong, China, pp. 1-6, Feb. 23-25, 2011.

[84] Niranjan; Sugam Pandey; Aura Ganz; "Design and evaluation of multichannel multirate wireless networks," *Broadband Networks (BroadNets '05), 2nd Int. Conf. on*, Boston, Massachusetts, USA, pp. 558- 564 ,vol. 1, Oct. 3-7, 2005.

[85] Sok-Hyong Kim; Young-Joo Suh; "Rate-Based Channel Assignment Algorithm for Multi-Channel Multi-Rate Wireless Mesh Networks," *Global Telecommunications Conference (GLOBECOM '08), Proc. of*, New Orleans, LA, USA, pp. 1-5, Nov. 30, 2008-Dec. 4, 2008.

[86] T. Kuang, Q. Wu, and C. Williamson, "MRMC: A Multi-Rate Multi-Channel MAC Protocol for Multi-Radio Wireless LANs," *Wireless Networks and Communication*

Systems (WiNCS '05), Proc. 1st Workshop on, Philadelphia, PA, Mar. 23-28, 2005.

[87] Nie Nie and Cristina Comaniciu; "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05), 1st IEEE Int. Symposium on*, Baltimore, Maryland, USA, pp. 269-278, Nov. 8-11, 2005.

[88] Altrnan, E.; Barman, D.; El Azouzi, R.; Jimenez, T.; "A game theoretic approach for delay minimization in slotted ALOHA," *International Conference on Communications (ICC '04), IEEE*, Paris, France, vol. 7, pp. 3999-4003, June 20-24, 2004.

[89] Y. Jin and G. Kesidis, "A Pricing Strategy for an ALOHA Network of Heterogeneous Users with Inelastic Bandwidth Requirements," Tech. Report, Dept. Computer Sci. & Eng., Pennsylvania State Univ., USA, 2002, [Online]. Available at: http://www.cse.psu.edu/~kesidis/papers/finalciss.pdf (Last accessed on Oct. 21, 2013)

[90] MacKenzie, A.B.; Wicker, S.B.; "Game theory and the design of self-configuring, adaptive wireless networks," *Communications Magazine, IEEE,* vol. 39, no. 11, pp. 126-131, Nov. 2001.

[91] Nezhad, M.A.; Cerda-Alabern, L.; "Adaptive Channel Assignment for Wireless Mesh Networks using Game Theory," *Mobile Adhoc and Sensor Systems (MASS '11), IEEE 8th Int. Conf. on*, Valencia, Spain, pp.746-751, Oct. 17-22, 2011.

[92] Haitao Wu; Fan Yang; Tan, K.; Jie Chen; Qian Zhang; Zhensheng Zhang; "Distributed Channel Assignment and Routing in Multiradio Multichannel Multihop Wireless Networks," *Selected Areas in Communications, IEEE Journal on*, vol.24, no.11, pp. 1972-1983, Nov. 2006.

[93] Ghannay, S.; Gammar, S.M.; "Joint routing and channel assignment protocol for multi-radio multi-channel IEEE 802.11s mesh networks," *Wireless and Mobile Networking Conference (WMNC '11), 4th Joint IFIP*, Toulouse, France, pp. 1-8, Oct. 26-28, 2011.

[94] Mohsenian-Rad, A.H.; Wong, V.W.S.; "Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 12, pp. 4432-4440, December 2007.

[95] Bononi, L.; Di Felice, M.; Molinaro, A.; Pizzi, S.; "Joint Channel Assignment and Multi-path Routing for Multi-radio Wireless Mesh Networks," *Distributed Computing Systems Workshops (ICDCS Workshops '09). 29th IEEE Int. Conf. on*, Montreal, Quebec, Canada, pp. 476-481, June 22-26, 2009.

[96] Xiangrui Meng, Kun Tan, Qian Zhang, "Joint Routing and Channel Assignment in Multi-Radio Wireless Mesh Networks," *Communications (ICC '06), IEEE Int. Conf. on*, Istanbul, Turkey, vol. 8, pp. 3596-3601, June 11-15, 2006.

[97] Alicherry, M.; Bhatia, R.; Li Erran Li; "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 1960-1971, Nov. 2006.

[98] Kai Peng; Jing Xu; Wei Liu; Xiaohui Chen; Jian Wu; "Resource Optimization for Uncertain Traffic Demand in Multi-radio Multi-channel Wireless Networks," *Computational Intelligence and Security (CIS '11), Seventh Int. Conf. on*, Sanya, Hainan, China, pp. 139-143, Dec. 3-4, 2011.

[99] Al-Ayyoub, M.; Gupta, H.; "Joint Routing, Channel Assignment, and Scheduling for Throughput Maximization in General Interference Models," *Mobile Computing,*

IEEE Transactions on, vol. 9, no. 4, pp. 553-565, Apr. 2010.

[100] Hong, W.L.W.; Fei Long; Pengye Xia; Chan, S.H.G.; "Distributed Joint Channel and Routing Assignment for Multimedia Wireless Mesh Networks," *Multimedia and Expo (ICME '12), IEEE Int. Conf. on*, Melbourne, VIC, Australia, pp. 404-409, July 9-13, 2012.

[101] Pal, A.; Nasipuri, A.; "JRCA: A joint routing and channel assignment scheme for wireless mesh networks," *Performance Computing and Communications Conference (IPCCC '11), IEEE 30th Int.*, Orlando, Florida, USA, pp. 1-8, Nov. 17-19, 2011.

[102] Avallone, S.; Akyildiz, I.F.; Ventre, G.; "A Channel and Rate Assignment Algorithm and a Layer-2.5 Forwarding Paradigm for Multi-Radio Wireless Mesh Networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 1, pp. 267-280, Feb. 2009.

[103] Hon Sun Chiu; Yeung, K.L.; King-Shan Lui; "J-CAR: An efficient joint channel assignment and routing protocol for IEEE 802.11-based multi-channel multi-interface mobile Ad Hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 4, pp. 1706-1715, Apr. 2009.

[104] Yanbing Liu; Guoping Dan; "Cross-layer design for channel assignment with connectivity maintenance in wireless mesh networks," *Mechatronic Science, Electric Engineering and Computer (MEC '11), Int. Conf. on*, Jilin, China, pp. 1996-2000, Aug. 19-22, 2011.

[105] Ngoc Thai Pham; Won-Joo Hwang; "Joint Disjoint Path Routing and Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks," *Vehicular Technology Conference (VTC 2008-Fall), Calgary, BC, IEEE 68th*, pp. 1-5, Sept.

21-24, 2008.

[106] Xiaojun Lin; Rasool, S.B.; "Distributed and Provably Efficient Algorithms for Joint Channel-Assignment, Scheduling, and Routing in Multichannel Ad Hoc Wireless Networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 6, pp. 1874-1887, Dec. 2009.

[107] Xiaojun Lin and Rasool, S.; "A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-hoc Wireless Networks," *Computer Communications (INFOCOM '07), IEEE 26th Int. Conf. on*, Anchorage, Alaska, USA, pp. 1118-1126, May 6-12, 2007.

[108] Henning Schulzrinne; Ping Pan; Akira Tsukamoto; RTP Tools, [Online], Available at: http://www.cs.columbia.edu/irt/software/rtptools/ (Last accessed at Oct. 16, 2013)

[109] Adya, A.; Bahl, P.; Padhye, J.; Wolman, A.; Lidong Zhou; "A multi-radio unification protocol for IEEE 802.11 wireless networks," *Broadband Networks (BroadNets '04), Proc. First Int. Conf. on*, San Jose, CA, USA, pp. 344-354, Oct. 25-29, 2004.

[110] Krishna Ramachandran, Irfan Sheriff, Elizabeth M. Belding, and Kevin C. Almeroth; “A multi-radio 802.11 mesh network architecture,” *Mobile Network and Applications*, vol. 13, no. 1-2, pp. 132-146, Apr. 2008.

[111] Skalli, H.; Ghosh, S.; Das, S.K.; Lenzini, L.; Conti, M.; "Channel Assignment Strategies for Multi-radio Wireless Mesh Networks: Issues and Solutions," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 86-95, Nov. 2007.

[112] HAL, [Online], Available at: http://madwifi-project.org/wiki/About/HAL (Last

accessed at Oct. 16, 2013)

[113] OpenHAL [Online], Available at: http://madwifi-project.org/wiki/About/OpenHAL?redirectedfrom=OpenHAL (Last accessed at Oct. 16, 2013)

[114] Viral V. Kapadia; Sudarshan N. Patel; Rutvij H. Jhaveri; “Comparative Study of Hidden Node Problem and Solution Using Different Techniques and Protocols,” *Journal of Computing*, vol. 2, no. 3, pp. 65-67, Mar. 2010.

[115] ISO/IEC International Standard - Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements," ISO/IEC 8802-11, Second edition: 2005/Amendment 6 2006: IEEE STD 802.11i-2004 (Amendment to IEEE Std 802.11-1999), pp. c1-178, July 23, 2004. [Online], Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4100091&isnumber=4100090 (Last accessed at Oct. 16, 2013)

[116] Eric Rozner; Yogita Mehta; Aditya Akella and Lili Qiu; “Traffic-aware channel assignment in wireless LANs,” SIGMOBILE Mob'07, Comp., Comm., pp. 43-44, Apr. 2007.

[117] Athanasiou, G.; Broustis, I.; Korakis, T.; Tassiulas, L., "LAC: Load-aware channel selection in 802.11 WLANs," Personal, Indoor and Mobile Radio Communications, (PIMRC'08), IEEE 19th International Symposium on, pp.1-6, Sept. 15-18, 2008.

Appendix A

Access efficiency value with different PHY transmission rate and packet size

Rate(Mbps) / Size(Bytes)	6	12	18	24	36	48	54
100	2.7	1.6	1.2	1.0	0.9	0.8	0.7
200	4.0	2.3	2.7	1.4	1.1	0.9	0.8
300	5.2	2.9	2.1	1.7	1.3	1.2	1.1
400	6.4	3.5	2.5	2.0	1.6	1.3	1.2
500	7.7	4.1	2.9	2.4	1.8	1.5	1.4
600	8.9	4.8	3.4	2.7	2.0	1.6	1.5
700	10.2	5.4	3.8	3.0	2.2	1.8	1.6
800	11.4	6.0	4.2	3.3	2.4	1.9	1.8
900	12.7	6.6	4.6	3.6	2.6	2.1	1.9
1000	14.0	7.3	5.0	3.9	2.8	2.3	2.1
1100	15.2	7.9	5.5	4.2	3.0	2.4	2.2
1200	16.5	8.5	5.9	4.5	3.2	2.6	2.3
1300	17.7	9.2	6.3	4.9	3.4	2.7	2.5
1400	19.0	9.8	6.7	5.2	3.6	2.9	2.6

Appendix B

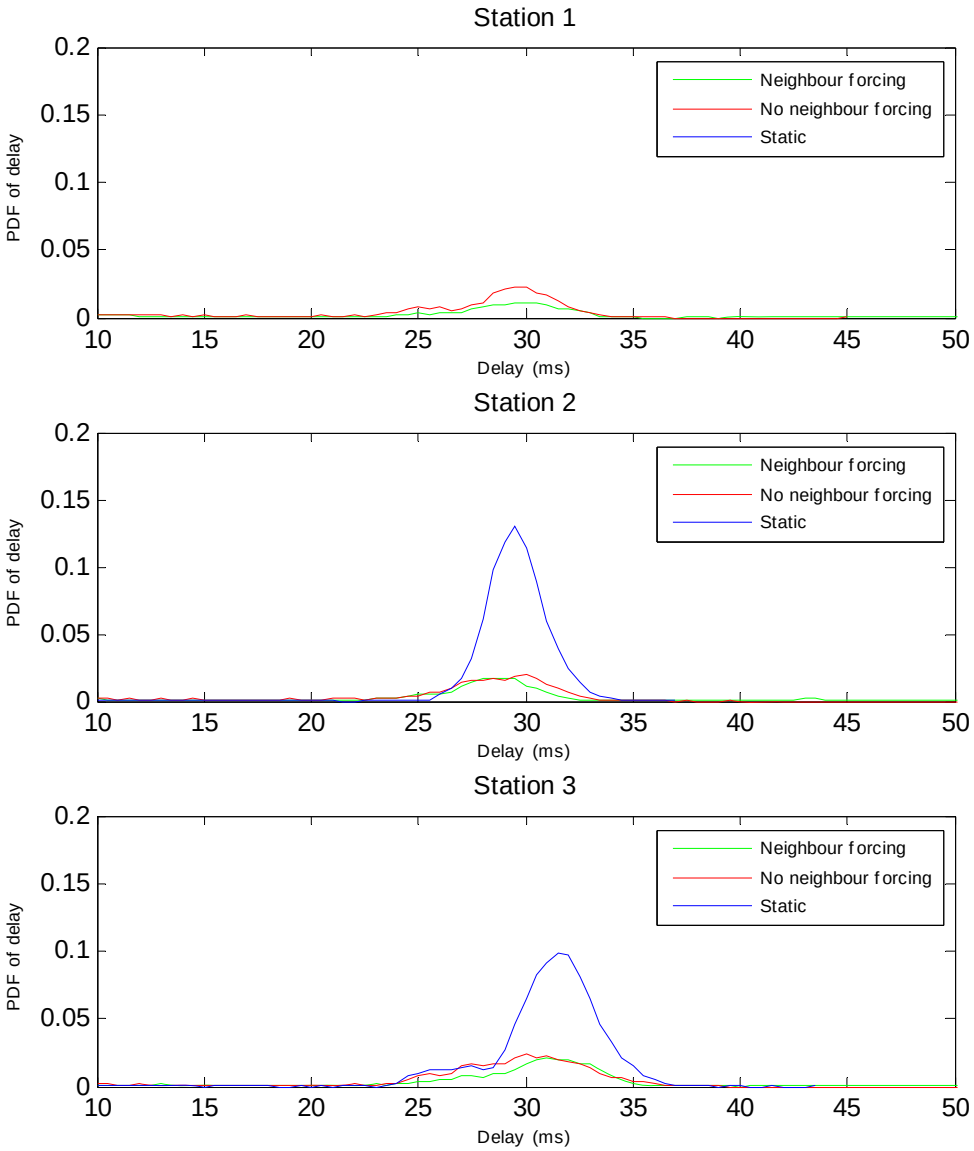
RtpTools traffic file

0.0000	RTP	data=0x0	len=1000	ts=5000	seq=0	pps=500
RTP Timestamp	RTP packet type	Content in packet	Packet length without header	Time flag of each second	Sequence number	Number of packets in next 1 second

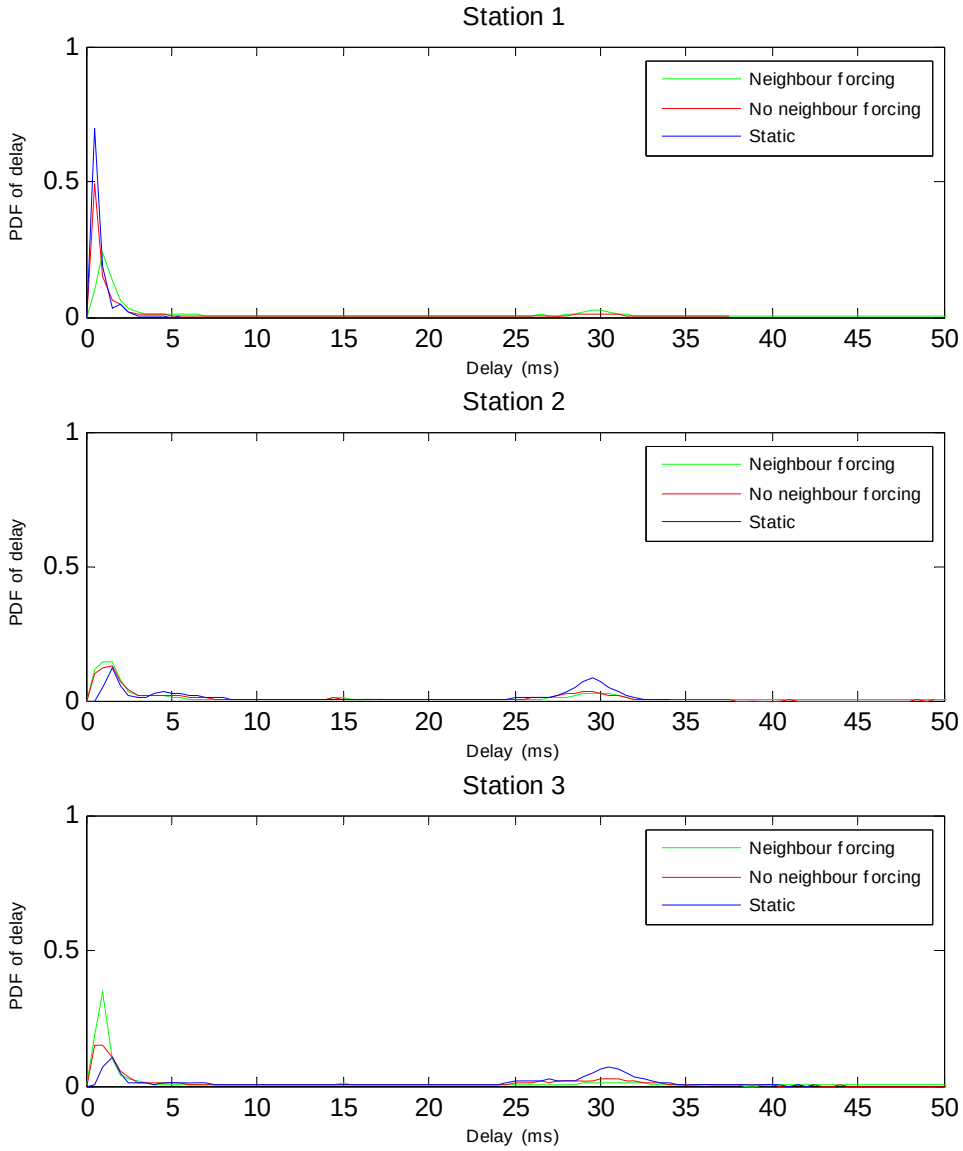
0.0000 RTP data=0x0 len=1000 ts=5000 seq=0 pps=500
0.0020 RTP data=0x0 len=1000 seq=1
0.0040 RTP data=0x0 len=1000 seq=2
0.0060 RTP data=0x0 len=1000 seq=3
0.0080 RTP data=0x0 len=1000 seq=4
0.0100 RTP data=0x0 len=1000 seq=5
0.0120 RTP data=0x0 len=1000 seq=6
0.0140 RTP data=0x0 len=1000 seq=7
0.0160 RTP data=0x0 len=1000 seq=8
0.0180 RTP data=0x0 len=1000 seq=9
0.0200 RTP data=0x0 len=1000 seq=10
0.0220 RTP data=0x0 len=1000 seq=11
. . . .
. . . .
. . . .
0.9920 RTP data=0x0 len=1000 seq=11
0.9940 RTP data=0x0 len=1000 seq=11
0.9960 RTP data=0x0 len=1000 seq=11
0.9980 RTP data=0x0 len=1000 seq=11
1.0000 RTP data=0x0 len=1000 ts=5000 seq=0 pps=500
1.0020 RTP data=0x0 len=1000 seq=11
1.0040 RTP data=0x0 len=1000 seq=11
1.0060 RTP data=0x0 len=1000 seq=11
1.0080 RTP data=0x0 len=1000 seq=11

Appendix C

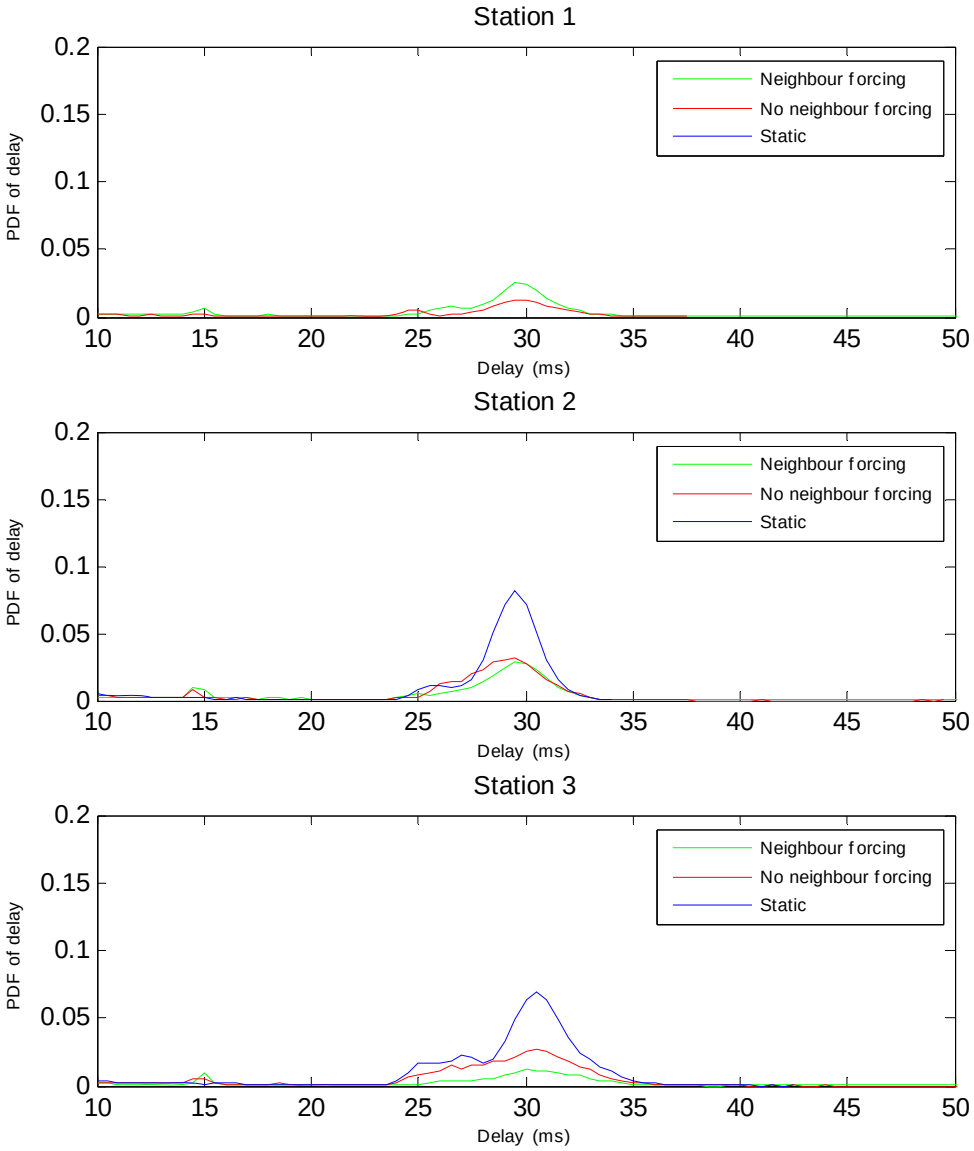
PDF of delay



C1 T_{cycle} = 60 seconds

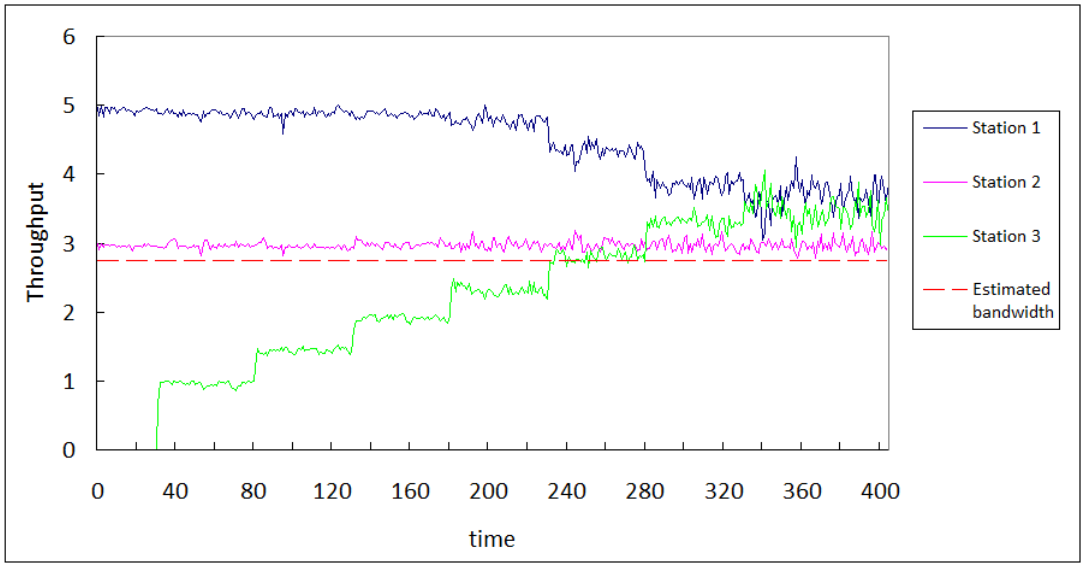


C2 T_{cycle} = 60 seconds

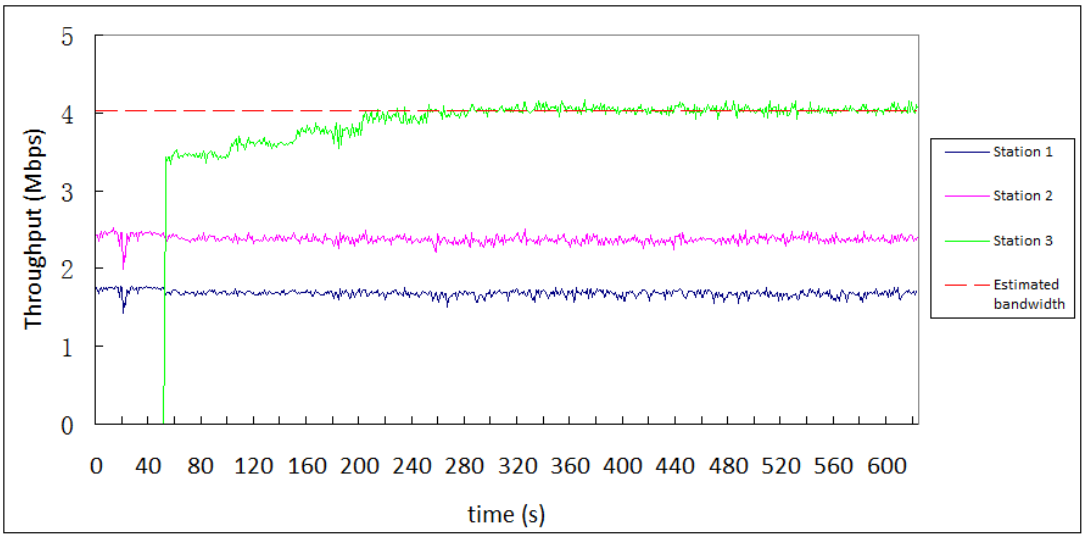


C3 T_{cycle} =100 seconds

Appendix D



There are two stations exist on the channel Station 1 and Station 2. They maintain the traffic load requirement during the experiment and Station 3 needs to estimate the available bandwidth. The Station 1 becomes saturated and when the throughput of Station 3 equals the available bandwidth when t = 230 seconds.



There are two stations exist on the channel Station 1 and Station 2. They maintain the traffic load requirement during the experiment and Station 3 needs to estimate the available bandwidth. The Station 3 becomes saturated and when the throughput of Station 3 equals the available bandwidth when t > 200 seconds.