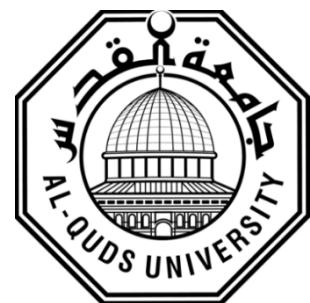


**Deanship of Graduate Studies
Al-Quds University**



Channel Selection in 802.11 WLAN

Ibrahim Ahmad Abed Allah Alwawi

M.Sc. Thesis

Jerusalem-Palestine

1438 / 2017

Channel Selection in 802.11 WLAN

Prepared By:

Ibrahim Ahmad Abed Allah Al-wawi

B.Sc.: Computer Engineering

Polytechnic of Palestine, Palestine

Supervisor: Dr. Murad Abusubaih

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Electronic and Computer Engineering/Department of Electronic and Computer Engineering/ Faculty of Engineering/ Graduate Studies

Jerusalem-Palestine

1438 / 2017

**Deanship of Graduate Studies
Al-Quds University
Electronics and Computer Engineering**



Thesis Approval

Channel Selection 802.11 WLAN

Prepared By: Ibrahim Ahmad Abdallah Al-wawi

Registration No: 21012161

Supervisor: Dr. Murad Abusubaih

Master thesis submitted and accepted, Date: 10.05.2017

The name and signatures of examining committee members are as follows:

8/6/2017

- | | | |
|----------------------|---------------------|------------|
| 1- Head of committee | Dr. Murad Abusubaih | Signature: |
| 2- Internal Examiner | Dr. Ali Jamoos | Signature: |
| 3- External Examiner | Dr. Mutamed Khatib | Signature: |

Jerusalem-Palestine

1438 / 2017

Declaration:

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: *Ibrahim A. A. Al-wawi*

Ibrahim Ahmad Abdallah Al-wawi

Date: 8 / 6 / 2017

Acknowledgments

All praise be to Allah for many blessings...

All the love and appreciation to my family especially my wife, children for their support, Encouragement, and Patience.

Many thanks are given to my thesis supervisor Dr. Murad Abusubaih for his appreciated and valuable help; indeed, his directions, guidance and continuing support helped me so much in accomplishing this thesis.

My deep gratitude goes to all my friends, especially at Al-Quds University.

I'd like to thank all whoever shared me their feelings, encouragement and support.

Abstract

Wireless Local Area Networks (WLANs) based on the IEEE 802.11 standard are becoming the dominant access technology for enterprises, homes, universities and hotspots. Access Points (APs) are densely deployed to provide internet access anytime everywhere to their clients. The dense deployment of APs can introduce interference with each other if they select their channel in unplanned manner, which in turn degrade the performance. This is due to limited number of orthogonal channels that 802.11 supports which requires the assignment of same channel to multiple APs that are close to each other. The problem has increased with appearance of new mechanisms in 802.11n and 802.11ac devices e.g. channel bonding. The main research topic which is focused upon in the literature to mitigate the interference is the channel selection.

This thesis addresses the channel selection problem. A new metric for interference measurement and characterization is proposed, every node follows especial mechanism to make their measurements. Then, algorithms for dynamic channel selection based on the proposed metric are developed. The algorithms assign channels to cells in a way that minimizes interference among interfering cells. Results of a simulation, that demonstrate the superiority of our approach in minimizing the interference, is presented hence maximizing throughput.

Keywords: Wireless Local Area Networks (WLANs), Channel Selection, Interference, Basic Service Set (BSS), IEEE 802.11, Media Access Control (MAC), Channel Bonding, Access Point (AP), WLAN Controller (WLC), Channel Overlapping, OPNET.

Table of Contents

Declaration:	Error! Bookmark not defined.
Acknowledgments	iv
Abstract	v
Table of Contents	vi
Table of Figures.....	viii
Table of Tables.....	x
Chapter 1 Introduction.....	1
1.1 Motivation	1
1.2 Related Work.....	2
1.3 Approach	5
Chapter 2 Background.....	7
2.1 IEEE 802.11 Standard: Overview	7
2.1.1 IEEE 802.11 Protocol Architecture	8
2.1.2 IEEE 802.11 Series.....	9
2.2 IEEE 802.11 Standard: Topology.....	15
2.2.1 802.11 Components.....	15
2.2.2 802.11 Network Topology.....	15
2.2.3 802.11 Network management.....	17
2.3 802.11 MAC.....	18
2.4 Channel Allocation.....	19
2.5 Interference Model	21
2.6 Channel Overlapping Factor.....	21
Chapter 3 Channel Selection	23
3.1 Proposed Measurement Based Interference Approach.....	23
3.2 Channel assignment Algorithms.....	26
3.2.1 Centralized Channel Assignment.....	27

3.2.2 Decentralized channel assignment	28
Chapter 4 Simulation Environment and Model.....	30
4.1 Environment.....	30
4.1.1 OPNET Modeler Version	31
4.1.2 OPNET Modeler Editors	31
4.1.3 OPNET Modeler Simulation Browser.....	37
4.1.4 Statistics.....	39
4.1.5 Radio Transceiver Pipeline.....	42
4.2 Parameters	46
4.2.1 Network objects.....	46
4.3 Experiments.....	52
4.3.1 Application Configuration.....	52
4.3.2 Profile Configuration.....	54
4.3.3 Object Configuration.....	57
4.3.4 Modified code.....	61
4.3.5 Statistics.....	61
Chapter 5 Scenarios, Simulation Results and Discussion	62
5.1 Scenarios and Results.....	62
5.1.1 Scenario One (Increasing Number of nodes)	62
5.1.2 Scenario Two (Increasing Number of Nodes – Different Distribution)	66
5.1.3 Scenario Three (Increasing Load of Nodes).....	69
Chapter 6 Conclusions and Future work	73
6.1 Conclusions	73
6.2 Future work	74
Bibliography	75
Appendices	78
Appendix A	78
98.....	الملاخص

Table of Figures

Figure 1-1:Hidden Interference Problem.....	3
Figure 2-1: Open Systems Interconnection Reference Model.....	8
Figure 2-2: Physical layer and Data Link layer.....	9
Figure 2-3: MIMO Transmission	12
Figure 2-4: 40 MHz Spectral Mask and 20 MHz Mask	13
Figure 2-5: Ad hoc BSS	16
Figure 2-6: Infrastructure Basic Service Set.....	17
Figure 2-7: Using NAV for Virtual Carrier Sensing	19
Figure 2-8: 802.11 Channels in 2.4 GHz ISM Band	20
Figure 2-9: 802.11 Standard Spectral Mask	21
Figure 3-1: Four WLAN Cells	25
Figure 3-3: Algorithm Centralized Channel Assignment.....	28
Figure 4-1: Project Editor.....	32
Figure 4-2: Object Palette.....	32
Figure 4-3: Node Editor.....	33
Figure 4-4: Process Editor.....	34
Figure 4-5: BSS_INIT state.....	35
Figure 4-6: Enter and Exit Executive	36
Figure 4-7: Types of States.....	36
Figure 4-8: Code Blocks.....	37
Figure 4-9: Simulation Result Browser	38
Figure 4-10: Global Statistics.....	40
Figure 4-11: Node Statistics	41
Figure 4-12: Link Statistics	41
Figure 4-13: Radio Transceiver Pipeline.....	42

Figure 4-14: Wlan_wkstn_adv node	47
Figure 4-15: Wlan_ethernet_slip4_router	47
Figure 4-16: Ethernet4_slip8_gtwy_adv	48
Figure 4-17: Ethernet_server_adv	48
Figure 4-18: Application Configuration	49
Figure 4-19: FTP Table	50
Figure 4-20: Profile Configuration.....	51
Figure 4-21: Profile Attributes	51
Figure 4-22: BSS identifier & Channel Number	52
Figure 4-23: 10 MB (Mega Byte) file size	53
Figure 4-24: 5 KB (Kilo Byte) file size.....	54
Figure 4-25: Heavy Download Profile Settings	55
Figure 4-26: Light Download Profile Settings	57
Figure 4-27: Application: Supported Services Attribute	58
Figure 4-28: Applications: supported profiles attribute	61
Figure 5-1: Scenario One (Increasing Number of nodes).....	63
Figure 5-2: Throughput Results of Scenario One - Increasing Number of Nodes (40,65,85,105 and 117).....	65
Figure 5-3: Scenario Two (Increasing Number of nodes – Different Distribution)	66
Figure 5-4: Throughput Results of Scenario Two - Increasing Number of Nodes (40,65,85,105 and 117) - Different Distribution Compared with Scenario one	68
Figure 5-5: Scenario Three (Increasing Loads of 65 nodes)	70
Figure 5-6: Throughput Results of Scenario Three - Increasing Loads of 65 Nodes	72

Table of Tables

Table 2-1: IEEE 802 Standards	7
Table 2-2: 5GHZ NON-OVERLAPPING CHANNEL AVAILABILITY AT VARIOUS CHANNEL WIDTHS	14
Table 3-1: measured power and activity for interfering signal.....	26
Table 4-1: Heavy download profile settings.....	56
Table 4-2: Access Point Attributes.....	59
Table 4-3: Workstation Attributes.....	60
Table 5-1: Increasing number of Nodes “FTP traffic received (MEGABYTES/SEC)”	64
Table 5-2: Increasing number of Nodes “FTP traffic received (MEGABYTES/SEC)”	67
Table 5-3: Number Of Repetitions	69
Table 5-4: Increasing Loads of 65 Nodes “FTP traffic received (MEGABYTES/SEC)”.....	71

Chapter 1

Introduction

1.1 Motivation

Due to several advantages such as mobility, ease and speed of deployment, flexibility and cost, wireless technologies are encroaching on the traditional networks (wired networks). The most successful wireless networking technology is IEEE 802.11 standard [Gas02].

Recently, IEEE 802.11 based wireless local area networks (WLANs) or Wi-Fi provide an infrastructure for wireless Internet access and have been widely deployed in homes, universities, airports, enterprises, etc. due to its ease of installation, easily available, low cost equipment's and unlicensed frequency band of operation. Access Points (APs) are densely deployed to provision anytime anywhere internet access, and WLAN clients associate with a nearby AP to access the network. An AP and its associated clients are referred to as a basic service set (BSS) or cell. If more BSSs exist in the same infrastructure, the system is called extended service set (ESS) which may fall into two categories: a single ESS managed by one particular administrator, or multiple ESSs each managed by different administrator [Elw12][Gon13][Mur12][Chi10][Bad13][Sri09].

IEEE 802.11 WLAN is attractive network solutions due to the users communicate with each other using frequency spectrum without wires and their limitations. Transmission between APs and their associated clients do not use single frequency. Instead, the frequencies are divided into channels. The number of allowable channels are varies from country to country due to each country's regulations on radio spectrum allocation [Gas02] [Rob07].

To improve users' connectivity, a high density of access points (APs) is deployed. Which means that access points (APs) which have shared medium may provide harmful interference to each other, leading to degrade the service quality. Since this service depends on the contention level from other co-channel neighbouring cells. In other words, radios that belong to two nodes, operating on shared medium would interfere with each, this is so-called channel interference. In order to mitigate this interference, some techniques such as load balancing, power control, and channel selection are proposed. An efficient use of available frequency spectrum (channel selection) is an efficient technique to mitigate channel interference, hence improve service quality.

1.2 Related Work

For channel auto configuration, APs traditionally use Least Congested Channel Search (LCCS). In LCCS, each AP monitors packet transmissions on its channel operation. When it detects transmissions from other APs, it scans for an alternative less-congested channel. LCCS suffers from two drawbacks, firstly, it cannot deal with hidden

interference problem, where clients associated to two different APs interfere with each other, while the AP do not interfere among themselves as shown in Figure 1-1.

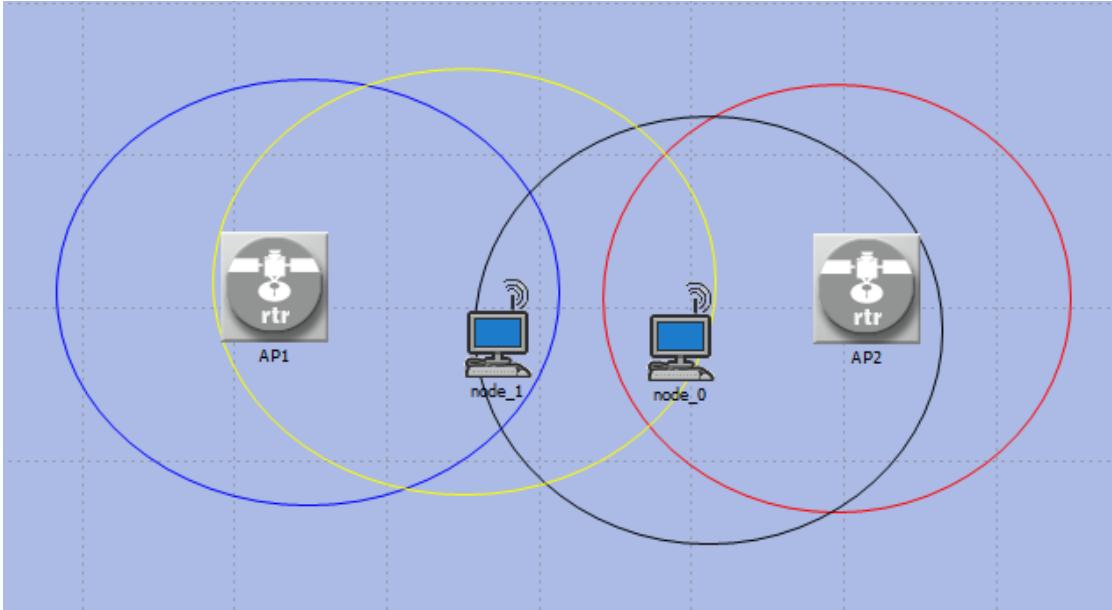


Figure 1-1:Hidden Interference Problem

Secondly, lacks the ability to improve channel reuse based on client distribution [Zho13]. In the last few years several strategies for channel assignment and improving performance are proposed in the literature. Specifically, in [Mish05], on the contrary of LCCS, authors take advantage of channel reuse created by client distributions. The channel assignment problem is modelled as a weighted graph coloring problem in which the APs are the vertices of the graph. An interference between APs is represented by an edge in the graph. Every edge has a weight associated with it, indicates the importance of using different colors (channels) for the corresponding vertices (APs) that are connected by an edge. In [Yue10], they proposed a distributed algorithm termed CACAO (Client-assisted Channel Assignment Optimization). In CACAO, an AP makes use of traffic information by its clients to make channel assignment decision. They consider only non-overlapping channels. Each AP independently chooses the

channel assignment to minimize the objective function locally. On the contrary of [Mish05], they take into account the real traffic load of both APs and clients. In [Rob07], they design a dynamic channel assignment algorithm that minimizing channel interference between APs. They develop a mathematical model that captures the amount of interference between overlapping channels for IEEE 802.11 WLAN systems. They take into account transmit power, distance between APs, attenuation loss and overlapping channels. They looked at interference from APs' perspective. Also, in this proposed algorithm AP changes its own channel in a greedy manner. Authors in [Elw12] built on [Rob07] algorithm's, but they find a global solution in the sense that the objective function minimizes the total interference at all APs by modelling the problem as a mixed integer linear program whose solution is a global minimum. In [Hai08], Authors proposed a mathematical model to assign channels to the APs based on maximizing the total SIR at the user level, hence improve throughput. They applied their channel assignment after network achieves a balanced state for balanced load distribution. They looked at interference from clients' perspective. Zhou et al in [Zho13] investigate the problem of partially overlapping channel assignment to improve of 802.11 WLAN based on SINR. They deduce a direct relationship between maximizing system throughput and minimizing total interference. minimizing total interference problem is formulated as directed weighted interference graph. Then a greedy algorithm is proposed. Authors in [Yang13] proposed a distributed dynamic channel assignment algorithm for maximizing the total weighted SINR by using partially overlapping channels. They looked at interference from clients' perspective. Authors in [Cui11] derived an interference model that takes into account both adjacent channel separation and the physical distance of the two nodes employing adjacent channels. Then they proposed a MICA (Minimum Interference for Channel Allocation)

algorithm to minimize total interference. Bokhari and Zaruba in [Bok13] considered partially overlapping channels (POCs) for efficient spectrum utilization. Their interference model take into account both adjacent channel interference and the corresponding physical distance between mesh nodes. They design a channel assignment algorithm termed as i-POCA which enables the use of smart ants for assigning POCs to radios. Their algorithm may applied in a centralized manner or distributed manner. Authors in [Zha11] proposed a channel assignment algorithm for centralized control WLAN with multiple APs (WMAPN) to maximize overall system throughput and achieve the fairness among APs. They treat channel assignment problem as the weighted graph coloring problem which jointly consider the weight of vertexes and edges termed as (CAOTR). Xu et al in [Xu11] considered both channel assignment and user association problem in 802.11 wireless networks, with the objective of balancing the traffic load across different APs. in [Gong13] authors proposed a distributed channel assignment algorithm in multi-rate 802.11n WLANs aiming at maximizing the network throughput namely Throughput-Maximizing Channel Assignment (TMCA). Their proposed algorithm take into account a new channel bonding and a frame aggregation mechanisms of 802.11n WLAN. Also they considered multi-rates clients. To determine the interference among all clients they proposed a measurement-based protocol.

1.3 Approach

A common shortcoming of many algorithms mentioned earlier for channel assignment is that they do not completely discuss and specify the implementation procedure, others algorithm look at interference from APs' perspective. Further, different algorithms use the distance between wireless APs as an interference measure. They configure APs which are close to each other on different channels. While the fact that accurate distance measurement for indoor Wi-Fi deployments do not exist yet, interference conditions and signal strength depend on many other parameters. Measurement based approaches also faces challenges regarding their implementation. The challenging problem is that a measuring node only observes transmissions sent over the same channel it uses. i.e., it does not see transmissions over other channels. Therefore, the knowledge of signal strength among all node pairs requires a special mechanism [Mur12].

A new metric for interference measurement is proposed. This metric take into account the signal activities from each sending address that interfere with node performs measurements. Then, two algorithms – centralized and decentralized algorithms- are developed for dynamic channel selection based on the proposed metric. The algorithms assign channels to Access points in a way that minimizes interference among interfering cells.

The remainder of this thesis is structured as follows. Chapter 2 introduces the background of IEEE 802.11 standards. The main contribution of this thesis, which is a new metric for interference measurement and proposed algorithms discussed in Chapter 3. Simulation Environment and Model is presented in Chapter 4. In Chapter 5, simulation results and discussion is presented. Chapter 6, contains the conclusion and future work.

Chapter 2

Background

2.1 IEEE 802.11 Standard: Overview

Institute of Electrical and Electronics Engineers (IEEE) create and maintain LAN/MAN Standards Committee (IEEE 802). IEEE 802 deals with Local Area Networks and Metropolitan Area Networks as shown in Table 2-1.

IEEE 802 Standards	
802.1	Bridging and management.
802.2	Logical Link Control
802.3	Ethernet – CSMA/CD Access Method
802.4	Toking Passing Buss Access Method
802.5	Toking Ring Access Method
802.6	Distributed Queue Dual Buss Access Method
802.7	Broadband LAN
802.8	Fiber Optic
802.9	Integrated Service LAN
802.10	Security
802.11	Wireless LAN
802.12	Demand priority Access
802.14	Medium Access Control
802.15	Wireless Personal Access Network
802.16	Broadband Wireless Metro Area Networks
802.17	Resilient Packet Ring

Table 2-1: IEEE 802 Standards

IEEE 802.11 is a technology that allows many electronic devices to exchange data or connect to the internet wirelessly using radio waves.

2.1.1 IEEE 802.11 Protocol Architecture

The Open Systems Interconnection Reference Model (OSI) model, was developed by the International Organization for Standardization which is known as (ISO). OSI model describes the steps to be used for data transfer over a transmission medium from one networked device to another. The OSI model defines the network communications process into seven separate layers as shown in Figure 2-1 [Tektronix].

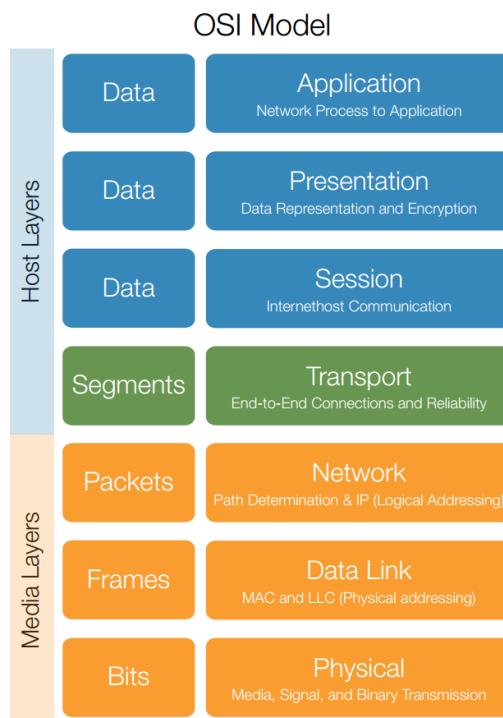
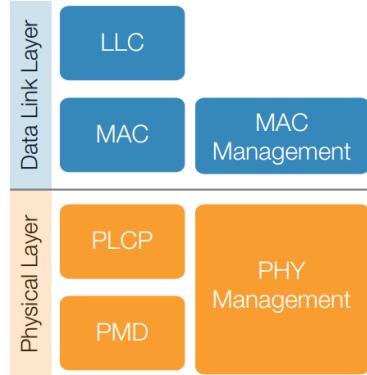


Figure 2-1: Open Systems Interconnection Reference Model



802.11 standard only deals with the two lowest layers of the OSI reference model, the Physical layer and Data Link layer (or Media Access Control layer) as shown in Figure

Figure 2-2: Physical layer and Data Link layer

2-2 [Tektronix].

The MAC layer provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur in the physical layer. It provides access to contention based and contention free traffic on different kinds of physical layers. In the MAC layer, the responsibilities are divided into the MAC sub-layer and the MAC management sub-layer. The MAC sub-layer defines access mechanisms and packet formats. The MAC management sub-layer defines power management, security and roaming services. The Physical Layer defines the electrical and physical specifications for devices. It defines the relationship between a device and a transmission medium. The major functions performed by the physical layer are: Establishment and termination of a connection to a communications medium, contention resolution and flow control and the modulation process [Tektronix].

2.1.2 IEEE 802.11 Series

IEEE has been developing IEEE 802.11 standards which contain specifications for several technologies, covering various speeds and frequencies. There are two unlicensed frequency bands for 802.11. The first one is 2.4 GHz Industrial, Scientific and Medical (ISM) band. The second one is 5 GHz unlicensed National Information Infrastructure (UNII) band.

2.1.2.1 IEEE 802.11 standard

The first version of the standard IEEE 802.11 was released in 1997; operates over 2.4 GHz (ISM) band supporting data rates of 1 Mbps and 2 Mbps. Its specified data rate was to be transmitted via infrared (IR) signals or by either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS) radio signals.

2.1.2.2 IEEE 802.11b

802.11 was rapidly supplemented by the 802.11b in 1999, which supports speeds of 1, 2, 5.5, 11 Mbps using (DSSS) modulation technique. One disadvantage of the 802.11b devices is that they may have interference issues with other products operating in the 2.4 GHz band.

2.1.2.3 IEEE 802.11a

IEEE 802.11a standard has an enhanced speed of up to 54 Mbps using Orthogonal Frequency Division Multiplexing (OFDM) radio technology in the 5 GHz band.

2.1.2.4 IEEE 802.11g

In June 2003, the IEEE 802.11g standard was approved. It works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbps. 802.11g devices are compatible with IEEE 802.11b devices which operate in the same frequency range of 1, 2, 5.5 and 11 Mbps.

2.1.2.5 IEEE 802.11n

802.11n operates in both the 2.4 GHz and the 5 GHz bands. Support for 5 GHz bands is optional. In order to increase data transfer rate significantly, 802.11n supports multiple technologies. These technologies are multiple-input multiple-output (MIMO) and 40 MHz channels (channel bonding) in the Physical layer, and frame aggregation in the MAC layer.

MIMO technology -as shown in Figure 2-3 [Aruba], increases throughput by increasing number of radio transmit and receive chains (up to four transmit and four receive) and it is possible to have a different number of transmit vs. receive chains simultaneously. Spatial streams of data is related to the ability to transmit and receive on multiple radios. They split data into multiple parts and forward them over different radios, and the data takes different paths through the air. MIMO does these improvements with advanced signal processing techniques i.e. Spatial Multiplexing, Space-Time Block Coding and Transmit Beamforming.

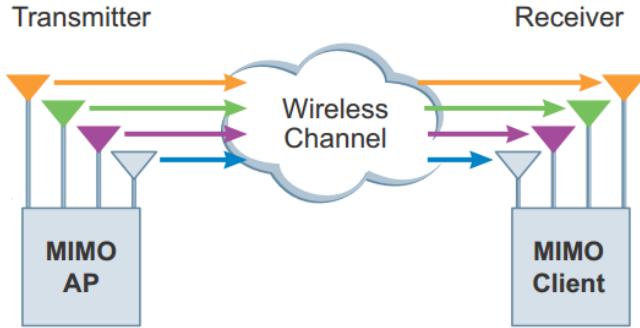


Figure 2-3: MIMO Transmission

The 802.11n allows wireless devices to operate on 40 MHz width channels by doubling their channel width from standard 20 MHz channels. This concept is known as channel bonding. Channel bonding can achieve higher transmission rates. Figure 2-4 Shows the difference between 40 MHz spectral mask and the 20 MHz mask [Aruba]. In traditional SISO (Single Input Single Output) channel bonding leads to a degradation in transmission range and higher interference. On the other hand, with incorporation of the MIMO technology, the problems faced by SISO systems from channel bonding can be mitigated [LARA11]. The data subcarrier count also has increased from the original 48 to 52 subcarriers in 20 MHz channels and 108 subcarriers in 40 MHz channels. This increase means that more data channels are available to carry traffic [Aruba].

In addition to channel bonding and MIMO technologies, a frame aggregation is introduced in 802.11n, where multiple frames are aggregated into a large frame before transmission to enhance MAC efficiency.

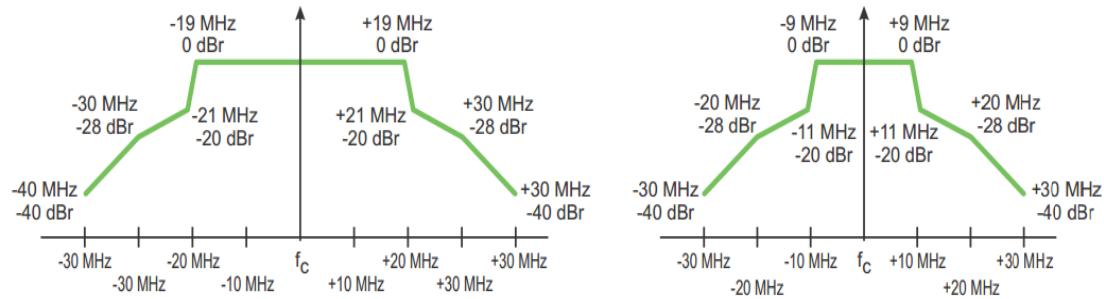


Figure 2-4: 40 MHz Spectral Mask and 20 MHz Mask

2.1.2.6 IEEE 802.11ac

802.11ac (known as Gigabit Wi-Fi) will operate on 5 GHz band, where there are more non-overlapping channels available. Table 2-2 shows the number of available non-overlapping channels in the 5GHz band using various bonding algorithms. It is a faster version of 802.11n. Its speed increases due to three factors [Motorola12]:

- **Channel bonding:** increased from 40 MHz (in 802.11n) up to 80 or 160MHz.
- **Denser modulation:** using 256 quadrature amplitude modulation (QAM), instead of 64 (QAM) in 802.11n.
- **More multiple input, multiple output (MIMO):** Whereas 802.11n stopped at four spatial streams and uses “Single-user” MIMO, where only benefit a single device at a time, 802.11ac goes all the way to eight (double spatial streams) and uses “Multiple-Users” MIMO, which allows multiple streams to be assigned to different clients, increasing the total bandwidth that can be transmitted simultaneously.

Channel Width (MHz)	Number of Channels (US)	Number of Channels (EUROPEAN UNION)
---------------------	-------------------------	-------------------------------------

20	21	16
40	9	7
80	4	3
160	2	2

**Table 2-2: 5GHZ NON-OVERLAPPING CHANNEL AVAILABILITY AT
VARIOUS CHANNEL WIDTHS**

2.2 IEEE 802.11 Standard: Topology

2.2.1 802.11 Components

802.11 networks consist of four major components, which are [Gas02]:

- Distribution system: When several (Access Points) APs are connected to serve a large coverage area, they must communicate with each other to track the movements of mobile stations. The distribution system is responsible of this operation.
- Access Points: Perform several functions. The most important one is the wireless-to-wired bridging function.
- Wireless Medium or Radio Frequency.
- Stations: they are computing devices with wireless network interfaces.

2.2.2 802.11 Network Topology

When a group of stations communicate with each other, they will form a Basic Service Set (BSS). Communications between these stations take place within a fuzzy area (BSS). There three types of BSS:

2.2.2.1 Independent Basic Service Set (IBSS)

Known as (Ad hoc BSSs). Stations communicate directly with each other and thus must be within direct communication range as shown in Figure 2-5.

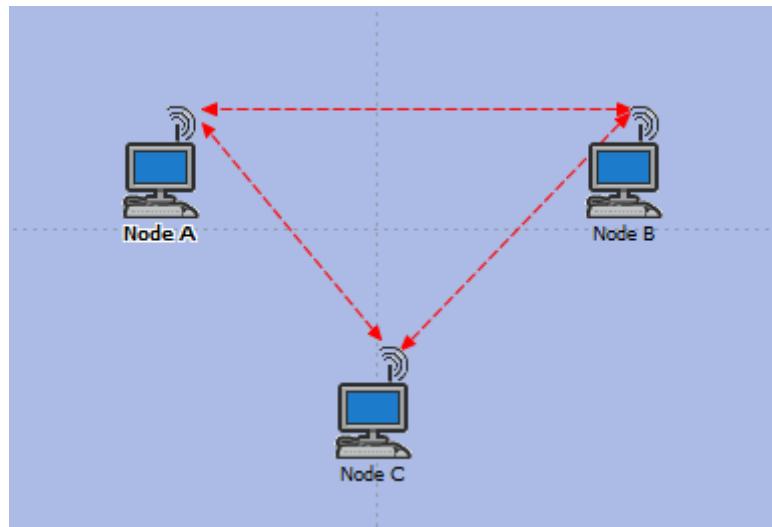


Figure 2-5: Ad hoc BSS

2.2.2.2 Infrastructure Basic Service Set (BSS)

All communications conducted through central Point as shown in Figure 2-6. The central point is known as Access point (AP). When any station needs to transmit data to another station. Sending station will send these frames to an AP, and then an AP will forward these frames to the destination station. In Infrastructure BSS, any station must associate with only one AP. Station will associate to the nearest AP (AP with the strongest received signal strength indicator (RSSI) following the 802.11 protocol [Xu11]. All stations that are associated to the same BSS operate on the same channel, i.e., the AP's channel.

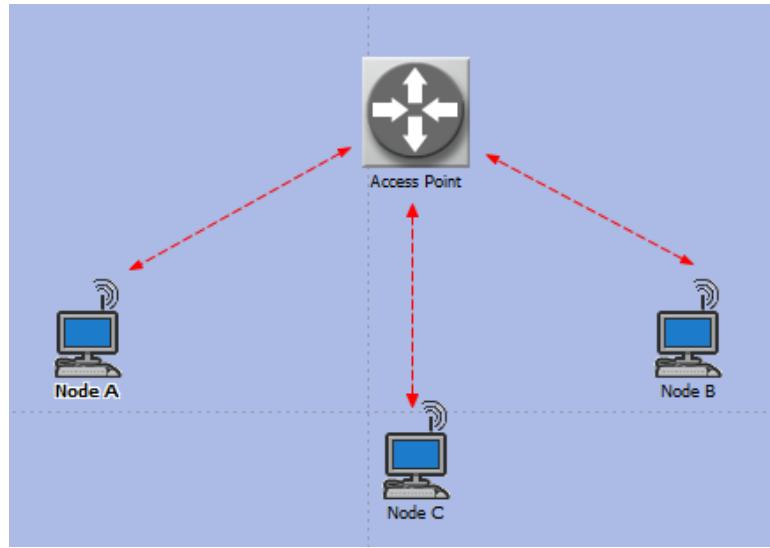


Figure 2-6: Infrastructure Basic Service Set

2.2.2.3 Extended Basic Service Set (ESS)

If Multiple BSSs exist in the same infrastructure, the system is referred to as an extended service set (ESS).

2.2.3 802.11 Network management

Based on how wireless LANs managed, there two types of WLANs:

2.2.3.1 Centralized network

In this type of networks, all APs and associated clients are managed by a central entity.

In [Zha11] and [Mur12], there exist a WLAN controller (WLC). The main function of this controller is to maintain association information between APs and clients, perform some measurements, and then make channel assignment decisions.

2.2.3.2 Decentralized (Uncoordinated) network

The most popular network in homes and urban areas. It has several properties such as: does not exist a central entity, non-specialists users (administrators), unplanned topology and the APs of different WLANs operate without any communication between themselves [Yue10].

2.3 802.11 MAC

802.11, like Ethernet, uses a carrier sense multiple access (CSMA) scheme to control access to the transmission medium. Due to Collisions waste valuable transmission capacity, 802.11 uses collision avoidance (CSMA/CA) instead of the collision detection (CSMA/CD) employed by Ethernet. If a station has pending traffic, it first senses the medium. If the medium is free, then the station can transmit. While if the medium is busy, then the station should wait until the medium becomes free [Gas02].

CSMA/CA is provided by distributed coordination functions (DCF) and point coordination function (PCF). DCF ensures that the medium is free before transmitting. It uses a random back off after each frame to avoid collisions. Also, DCF uses the CTS/RTS clearing technique to further reduce the possibility of collisions. PCF provides contention-free services in special station i.e. Access point. Virtual carrier sensing is used to determine if the medium is available. CSMA/CA is provided by a timer that indicates the amount of time the medium will be reserved, this timer called (NAV) Network Allocation Vector as shown in Figure 2-7 [Gas02]. The NAV is carried in the frame headers on the RTS and CTS frames.

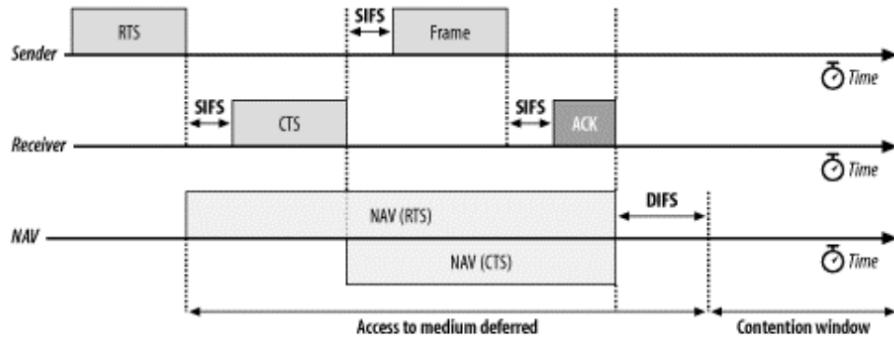
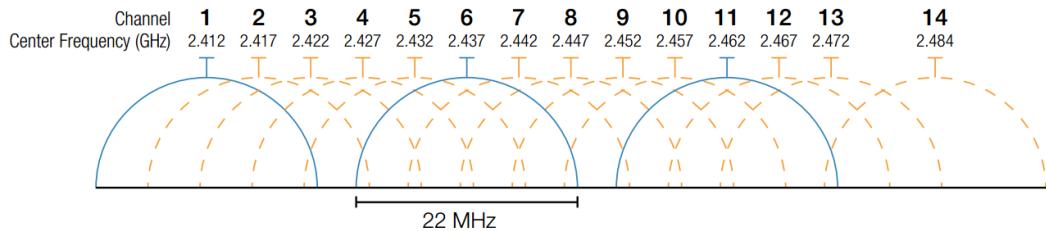


Figure 2-7: Using NAV for Virtual Carrier Sensing

2.4 Channel Allocation

802.11 WLANs operate in two unlicensed frequency. The first one is 2.4 GHz Industrial, Scientific and Medical (ISM) band. Another one is 5 GHz unlicensed National Information Infrastructure (UNII) band. These unlicensed frequencies are regulated by a regulatory body in each country, e.g. the Federal Communications Commission (FCC) in the USA. For example, 802.11b standard defines 14 channels, only 1 through 11 are permitted in the US, while 14 channels are permitted in Japan. 802.11 (b/g) standards use 2.4 GHz frequency band in ISM band, while 802.11 (a/ac) standards utilize 5 GHz frequency band. 802.11 (n) standard use both (2.4 and 5) GHz frequency bands. Each of these spectrums is sub-divided into channels with a centre

frequency and bandwidth. The 2.4 GHz band is divided into 14 channels spaced 5 MHz apart, beginning with channel one, which is centred on 2.412 GHz as shown in Figure 2-8 [Tektronix], only three channels are non-overlapping (orthogonal / non-interfering)



which are 1,6 and 11. Each channel that is approximately 20 MHz wide. 802.11n uses 20 MHz or 40 MHz (which known as Channel Bonding) to boost throughput. 80.11 ac supports 80 MHz and 160 MHz bandwidth additionally.

Figure 2-8: 802.11 Channels in 2.4 GHz ISM Band

802.11 standard specifies a spectral mask which defines the permitted power distribution across each channel as shown in Figure 2-9 [Tektronix].

Figure 2-8: 802.11 Channels in 2.4 GHz ISM Band

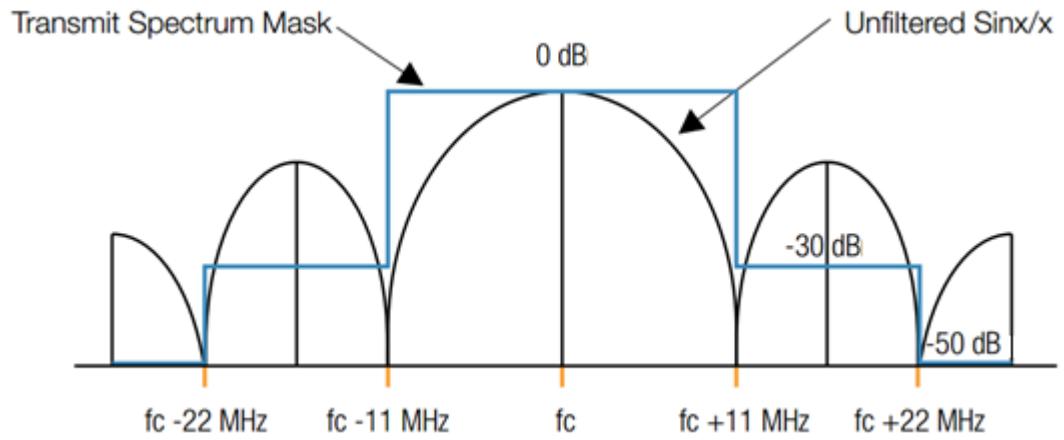


Figure 2-9: 802.11 Standard Spectral Mask

From Figure 2-8 and Figure 2-9, we note that radio frequency energy is going into adjacent channels. This forms what is known as overlapping channels.

2.5 Interference Model

As section 2.4, there is amount of energy going into adjacent channels causing what is known as Interference. Interference in 802.11 WLANS may prohibit two nodes from sending packets, despite none of the signals interferes the receiver of the other. On the other hand, an interfering signals may damage a packet being received, if its power is strong enough relative to the desired signal power at its receiver [Mur12]. There are different type of interference:

- **Co-channel Interference:** Refers to the fact that two nodes operating on the same channel would interfere with each other, if they are within the interference range of each other.
- **Adjacent Channel Interference:** Occurred when two nodes are configured to partially overlapping channels.
- **Self-Interference:** When a node is equipped with multiple radios in mesh network.

2.6 Channel Overlapping Factor

When we use overlapping (non-orthogonal) channels in assignment, the degree of overlapping between channels should be calculated. The degree of overlapping between

channels known as overlapping channel interference factor or channel overlapping factor. In literature, empirical measurement is performed in [Mish05] to find out channel overlapping factor which is called interference factor (I-factor) in their paper. In [Akl07] and [Hai08], they model overlapping channel interference by defining an overlapping channel interference factor as $O(m, n)$ as in Equation 2-1.

$$O(m, n) = \begin{cases} (1 - |C_n - C_m| \times 1/5), & O(m, n) \geq 0 \\ 0, & O(m, n) < 0 \end{cases} \quad \text{Equation 2-1}$$

When two APs n and m are using overlapping channels, The interference will increase depend on $O(m, n)$ where C_n is the channel assigned to AP n , C_m is the channel assigned to AP m , 5 is the number of channels between two consecutive non-overlapping channels (1,6,11). In [Zho12], authors simulate 802.11b signal using MATLAB and calculate channel overlapping factor according to modulated signal.

Chapter 3

Channel Selection

3.1 Proposed Measurement Based Interference Approach

A common shortcoming of many algorithms proposed for channel assignment in the literature is that they use the distance between wireless APs as an interference measure. They configure APs which are close to each other on different channel. In addition to fact that accurate distance measurement for indoor Wi-Fi deployments do not exist yet, interference conditions and the signal strength does depend on many other parameters.

Measurement based approaches also face challenges regarding their implementation. The challenging problem here is that a measuring node only observes transmissions over same channel it uses. Therefore, the knowledge of signal strength among all node pairs requires a special mechanism.

In this thesis, a measurement based approach is developed. In this approach, wireless nodes (stations and their APs) measure signal activities from each sending address. In order for a node to perform measurements over channels other than the one it operates on, it switches the operating channel and starts measurements. However, the switching and measurement take place during long back-off periods (which are expected during high interference conditions). Further, measurements over same channel are distributed over non-contiguous small time intervals. We refer to this measurement protocol as

non-disruptive measurement, i.e. measure while operate. The main feature of this protocol is that, it does not require a node to perform measurements over all channels consecutively. Thus, it gradually builds up measurement report. For real implementation, the 802.11k reports can be used for the delivery of measurements to the APs. In order to avoid simultaneous measurements by nodes, a management node (i.e. WLAN controller in Enterprises) successively instructs each AP to provide measurement results. Through 802.11k reports, an AP instructs each associated node to perform and report measurements.

Suppose that we have an area covered by N APs. Each AP serves a WLAN cell. The APs are managed by a WLAN controller. During measurement period of (T), each node records the cell ID, power level, the length and the physical transmission rate of all captured packets that received from all neighbouring cells. Then the node (k) summarizes the measured power and activity from all neighbouring cells as shown in Equation 3-1:

$$I_{kn} = \left(\frac{1}{T}\right) \sum_{i=1}^{Gn} \frac{P_i \times L_i}{R_i} \quad \text{Equation 3-1}$$

Where:

G_n : Number of packets are captured from cell n.

P_i : Received Power level in (watts).

L_i : length in (bits).

R_i : physical transmission rates of frames (i) in (bit/s).

Then, an AP serving a cell m computes the impact of cell n as in Equation 3-2:

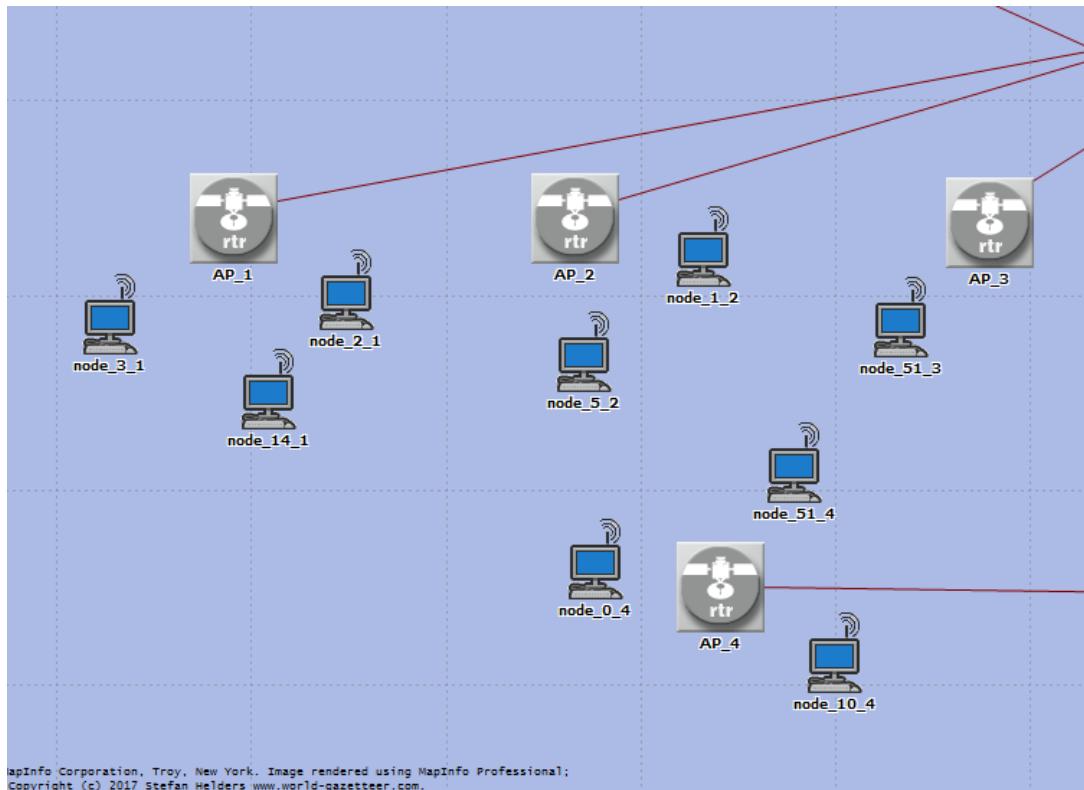
$$I(m, n) = \sum_{k=1}^{N_m} I_{km} \quad \text{Equation 3-2}$$

Where:

N_m : Number of nodes that were able to capture transmissions from cell n .

Then, each AP reports this information to the WLAN controller.

For further illustration, Suppose that we have four cells as shown in Figure 3-1.



Where:

AP_m: Access Point serving a cell (m).

Node_{k_m}: Node (k) in cell (m).

During measurement period of (T), each node in a cell (1) records the cell ID, power level, the length and the physical transmission rate of all captured packets that received from all neighbouring cells (2,3 and 4) as shown in Table 3-1.

Cell ID (where cell_id \neq 1)	P_i (watt)	L_i (bit)	R_i (bit/s)
4			
3			
2			
4			
⋮	⋮	⋮	⋮
2			
4			

Table 3-1: measured power and activity for interfering signal

3.2 Channel assignment Algorithms

In a dense WLAN, it is required to assign orthogonal channels to neighbouring APs to avoid interference among WLAN cells. Hence, it is nontrivial to assign channels in WLANs based on the legacy 802.11 a/b/g devices. The problem becomes more challenging with devices based on the new 802.11n/ac standards, due to the channel bonding mechanism, which uses multiple non-overlapping channels together. Our proposed algorithm can be applied in centralized and decentralized WLANs.

3.2.1 Centralized Channel Assignment

A heuristic approach based on the interference measurement metric proposed in section 3.1 is developed. The key idea is that the WLAN controller sorts the cells based on the interference level they measure. Non-overlapping channels are firstly assigned to APs that are mostly impacted. Then, it starts assigning channels to the remaining cells trying to minimize interference from the cells which already assigned channels on the cell under consideration. The problem can be simplified as follows:

Let A represent the set of cells already assigned channels. AP n is assigned channel c that minimizes interference (F) as in Equation 3-3:

$$F = \min \sum_{m \in A, m \neq n} I(m, n) O(m, n) \quad \text{Equation 3-3}$$

Where:

$I(m, n)$: impact of cell n on cell m.

$O(m, n)$: overlapping channel interference factor, calculated as Equation 2-1.

A formal description of the algorithm is shown in Figure 3-2:

```

AP_list={};
Request measurements reports from APs;
for  $m = 1; m \leq N; m++$  do
|   for  $n = 1; n \leq N; n++$  do
|   |    $T_I(m) = \sum_{n=1}^N I(m, n)O(m, n);$ 
|   end
end
Sort cells in descending order according to  $T_I(m)$ ;
Assign all non-overlapping channels K to the first K APs in the sorted
list;
Update AP_list;
for  $i = K+1; i \leq N; i++$  do
|    $min = infinity;$ 
|   for  $c = 1; c \leq C; c++$  do
|   |   Channel[APi] = c;
|   |    $Z = \sum_{m \in AP\_list} (I(i, m) + I(m, i))O(i, m);$ 
|   |   if  $Z < min$  then
|   |   |    $min = Z;$ 
|   |   |   channel = c;
|   |   |   end if
|   |   end
|   |   Channel[APi] = c;
end

```

Figure 3-2: Algorithm Centralized Channel Assignment

3.2.2 Decentralized channel assignment

In this approach, each AP with help of the associated clients measures the interference level experienced on every channel. Then, it selects the channel that resulted least measured interference. It should be noted here that this simplified approach can also be used in managed WLANs, where the WLAN controller may instruct a cell to find out a less interfered channel in case of bad communication conditions on the current

channel due to interference. Particularly, during a measurement period of (T), each node (k) in a cell (m) computes the interference level over channel c as in Equation 3-4:

$$I_k(c) = \left(\frac{1}{T}\right) \sum_{i=1}^{G_c} \frac{P_i \times L_i}{R_i} \quad \text{Equation 3-4}$$

Where:

G_c : Number of frames captured on channel c from all neighbouring cells.

P_i : Received Power level in (watts).

L_i : length in (bits).

R_i : physical transmission rates of frames (i) in (bit/s).

Then, AP computes the overall interference from each channel as Equation 3-5:

$$I(c) = \sum_{i=1}^{N_n} I_i(c) \quad \text{Equation 3-5}$$

Where:

N_m : Number of nodes associated with AP m.

Then AP selects the channel that provides minimum $I(c)$.

Chapter 4

Simulation Environment and Model

4.1 Environment

Simulation provides a way to create, modify, and study the network behaviours of proposed designed before implementing it in real world. OPNET simulator is an example simulator used to simulate and design networks. OPNET stands for Optimized Network Engineering Tools, and was created by OPNET Technologies, Inc., which was founded in 1986. It provides corresponding product modules to address the following aspects of communications networks (Application performance management, Planning, Engineering, Operations and Research and development). The most popular of these product modules which is used for research and development is OPNET Modeler. OPNET Modeler is an event based simulation program has several advantages: includes GUI interface, comprehensive library of network protocols and models, graphical interface to view the results, availability of documentation for the user to develop the network models and Easy for understanding the network behaviour in various scenarios [Lu12].

4.1.1 OPNET Modeler Version

OPNET Modeler Version: 14.5.A PL8 (Build 7808 32-bit) copyright © 1986-2008 is used. Microsoft Visual C++ 2010 version 10.0.30319.1 RTMRel copyright © 2010 is used as compiler for OPNET Modeler to compile C/C++ based simulation models.

4.1.2 OPNET Modeler Editors

There are many editors we can deal with it in OPNET Modeler. We focus on some editors that we use in this thesis e.g. Project Editor, Node Editor and Process Editor. OPNET models also have a three-layer structure: network model, node model, and process model.

4.1.2.1 Project Editor

Simulation projects and scenarios can be managed by the Project Editor, as shown in Figure 4-1. Project Models can be created in Project Editor. Many network objects such as network devices, nodes and links from Object Palette as shown in Figure 4-2 can be dragged in this editor.

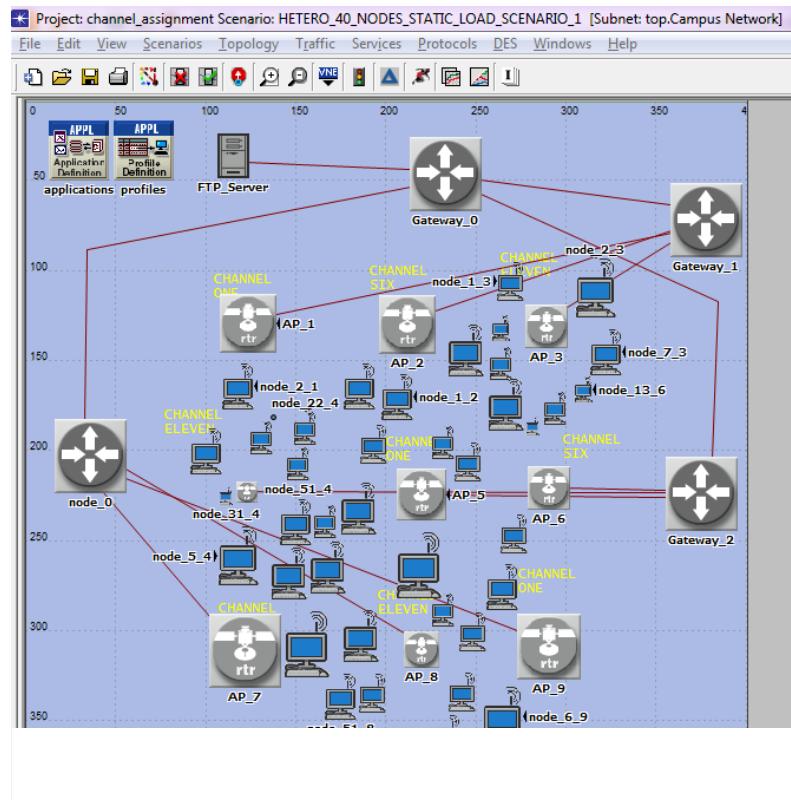
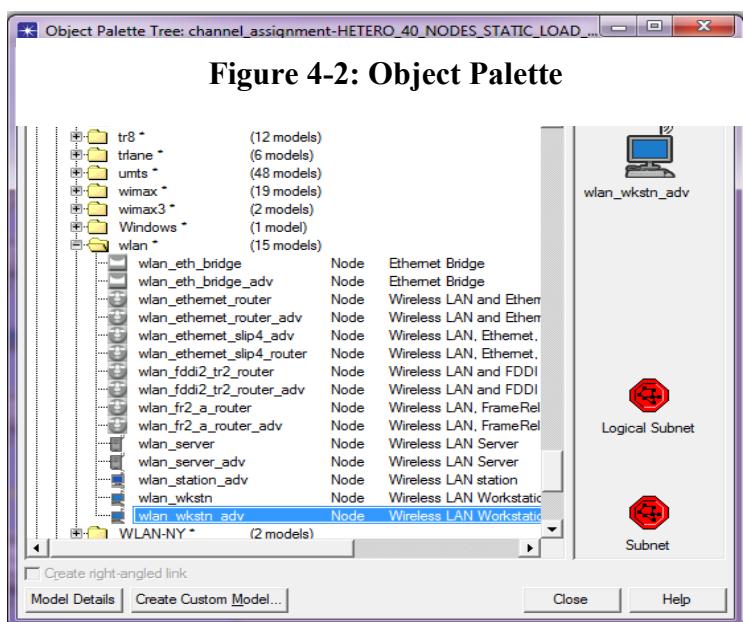


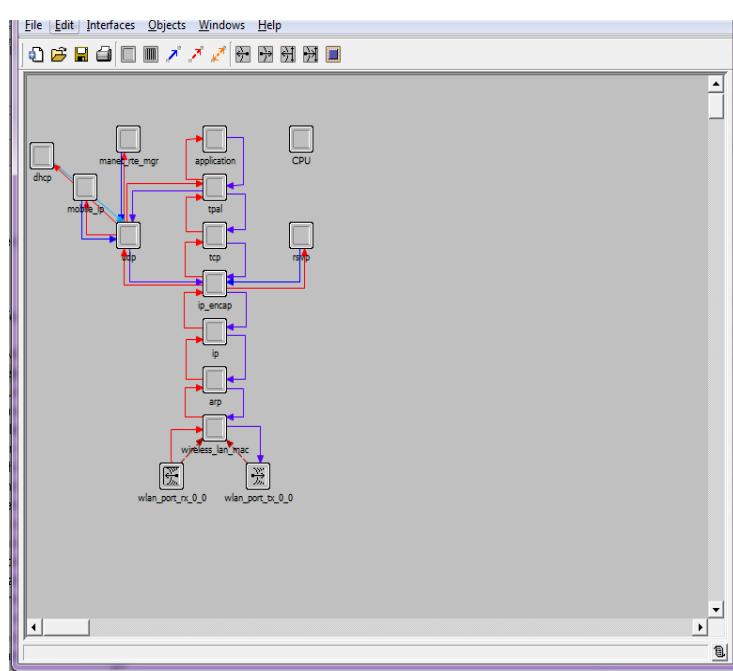
Figure 4-1: Project Editor



4.1.2.2 Node Editor

An internal structure (module) of a device or node is created and edited in this editor, as shown in Figure 4-3. It shows the node model for wireless workstation. Each node is composed of several modules. These modules are generally separated by logical functionalities and are able to communicate with each other via packet streams and statistic streams. Each module represents a particular functionality of that node. The modules include processor module, queue module, transceiver module, antenna module, and external system module. Modules can be used to transmit packets, receive packets, process data, store data, and route packets, etc. [Zheng12].

Figure 4-3: Node Editor



4.1.2.3 Process Editor

As mentioned in previous sections, each module represents a particular functionality. This functionality can be implemented in the Process editor, as shown in Figure 4-4. In this editor, proposed algorithms and protocols can be written. Figure 4-4 shows the process model for wireless workstation. Each process has state transition diagrams (STDs). In STDs, the process is composed of several states. States can transition between each other if certain conditions are triggered as shown in Figure 4-4.

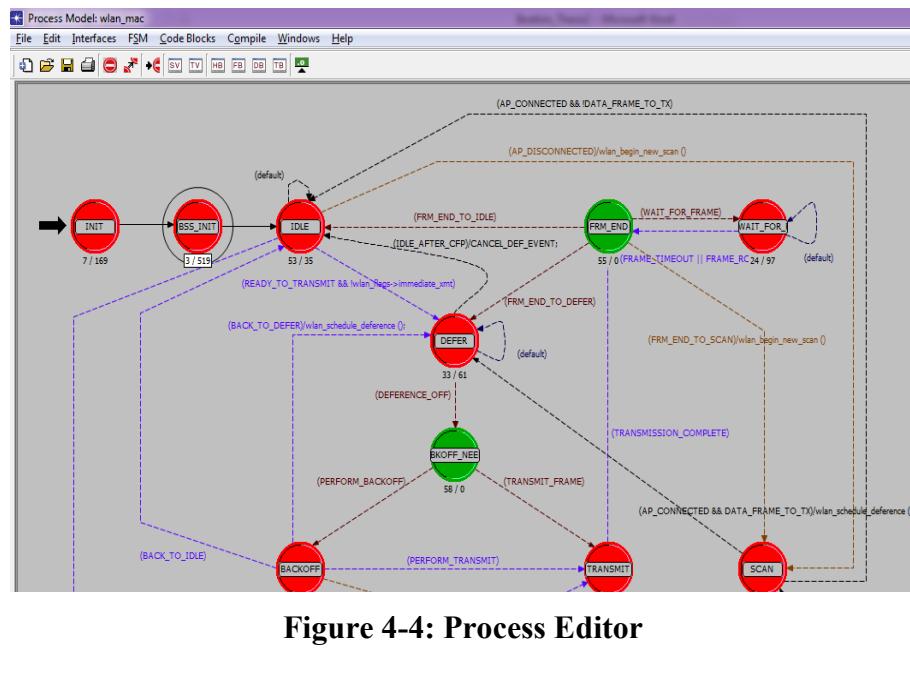


Figure 4-4: Process Editor

C/C++ codes can be written within a state to perform some operations when double clicked. Figure 4-5 is popped up, when we double click on BSS_INIT state. OPNET Modeler also has other editors like link editor, packet format editor, ICI editor, PDF editor and probe editor.

```

1  /* Obtain the values assigned to the various attributes */
2  op_ima_obj_attr_get (my_objid, "Wireless LAN Parameters", &wlan_params_comp_attr_objid, OPC_OBJTYPE_GENERIC, 0
3  params_attr_objid = op_topo_child (wlan_params_comp_attr_objid, OPC_OBJTYPE_GENERIC, 0
4
5  /* Determining the final MAC address after address resolution. */
6  op_ima_obj_attr_get (my_objid, "Address", &integer_mac_address);
7  my_address = integer_mac_address;
8
9  /* Update our own process registry record with the final address */
10 /* information. */
11 oms_pr_attr_set (own_process_record_handle, "address", OMSC_PR_INT64, my_address, OPC_
12
13 /* Destroy the global list of BSS IDs that was used while assigning */
14 /* channels to BSSs. */
15 wlan_bss_id_list_manage (bss_id, phy_type, "destroy_list");
16
17 /* Since the frequency band of the channels of all radio nodes are */
18 /* determined, refresh our potential receiver group to exclude the */
19 /* receiver channels that have non-overlapping bands with ours, */

```

Figure 4-5: BSS_INIT state

```

49 WLAN_STD_DOOR_REGISTER (bss_id, my_address, ap_flag, my_objid, priv_cridr_flag);
50
51 /* Get a handle to the information record of our own BSS, which */
52 /* also contains the STA list of our BSS. */
53 my_bss_info_ptr = wlan_bss_info_get (bss_id);
54
55 /* If we are operating in a bridge/switch node, indicate that our */
56 /* address is not a valid destination address, since the station */
57 /* models may choose their destination addresses randomly from the */
58 /* address pool. */
59 if (wlan_flags->bridge_flag)

```

4.1.2.3.1 States

Each state has two executive blocks, enter executive are invoked when entering a state and exit executive are invoked before exiting a state as shown in Figure 4-6. There are three types of states; initial state, forced state and unforced state, as shown in Figure 4-7.

- **Initial state:** the place where execution begins in a process.

- **Forced state:** does not allow a pause during the process.
- **Un-forced state:** the code will be executed in the enter executive and pause until a further interrupt is triggered by the transition condition.

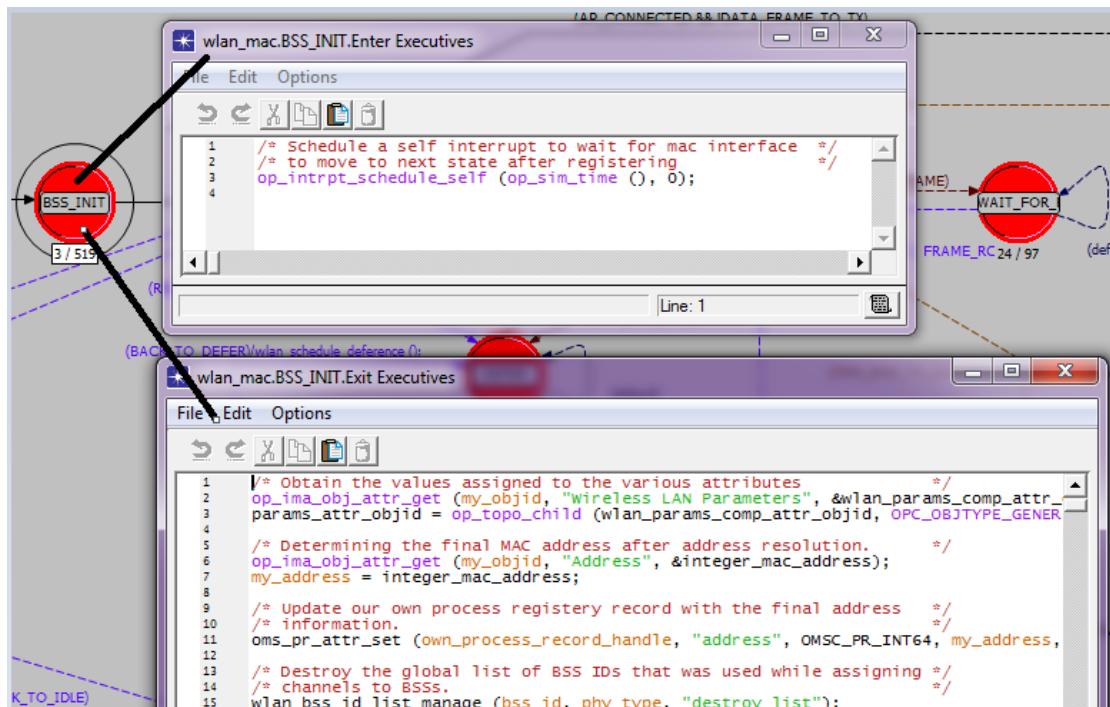


Figure 4-6: Enter and Exit Executive

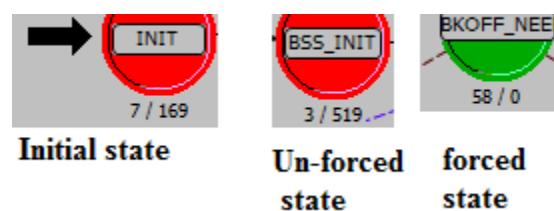


Figure 4-7: Types of States

4.1.2.3.2 State transitions

States can transition between each other if certain conditions are triggered.

4.1.2.3.3 Code

Code can be placed in several places: states (enter and exit executive blocks), SV, TV, HB, FB, DB, TB code blocks, which are located in toolbar of process model as shown in Figure 4-8. SV stands for state variable which keeps valid between state transitions. TV stands for temporary variable which is only valid within a state. HB stands for header block which is similar to a C/C++ header file which can be used to include other header files and define macros, etc. FB stands for function block, which contains definitions of functions. DB stands for diagnostic block, which contains diagnostic statements that print out diagnostic information to the standard output device [Lu12].

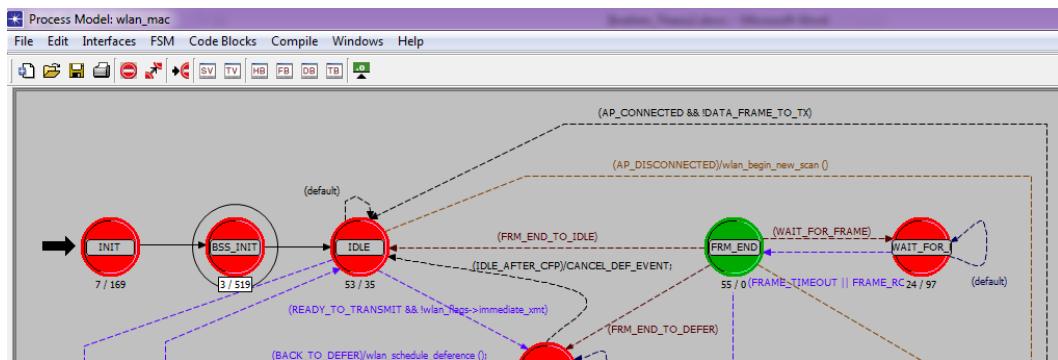
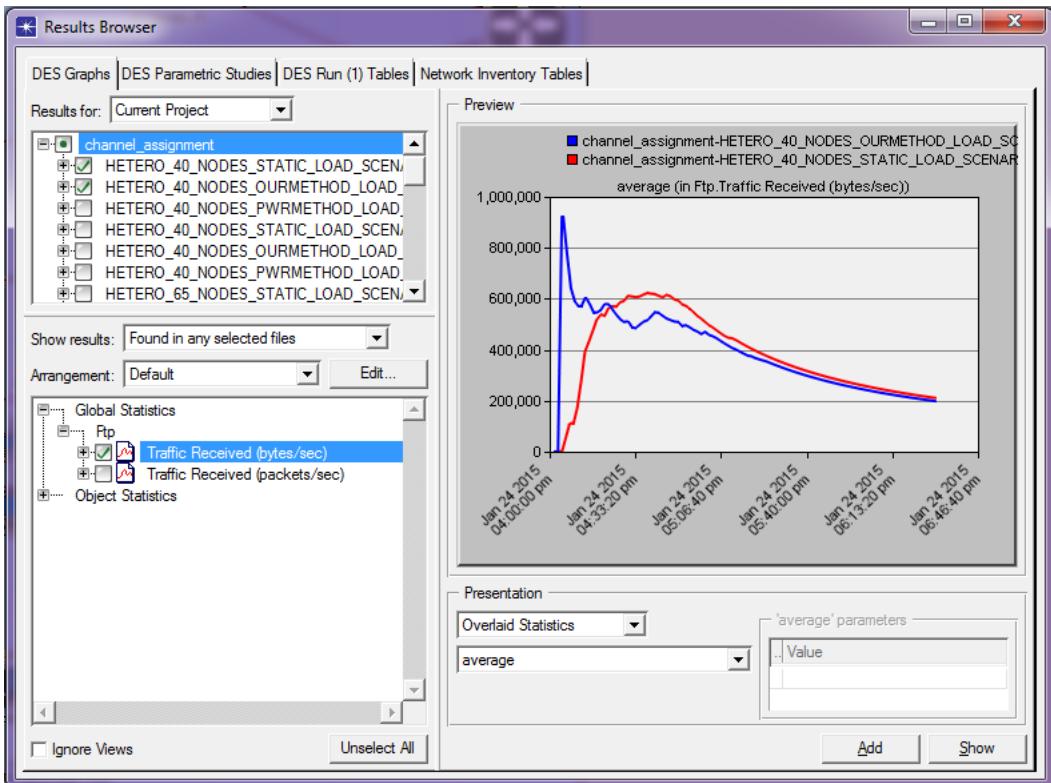


Figure 4-8: Code Blocks

4.1.3 OPNET Modeler Simulation Browser

When we have several scenarios, we can view and compare all simulation results through Simulation Results Browser as shown in Figure 4-9. There are lots of statistical tools allow us to view and compare the statistical results in different scales such as



logarithm, reciprocal, time average, sample sum etc. We can also export these results into a spread sheet for further processing.

Figure 4-9: Simulation Result Browser

4.1.4 Statistics

There are different types of statistics are provided by OPNET as Global statistics, Node statistics, Module statistics and link statistics. Different types of statistics are shown in Figure 4-10 to Figure 4-11.

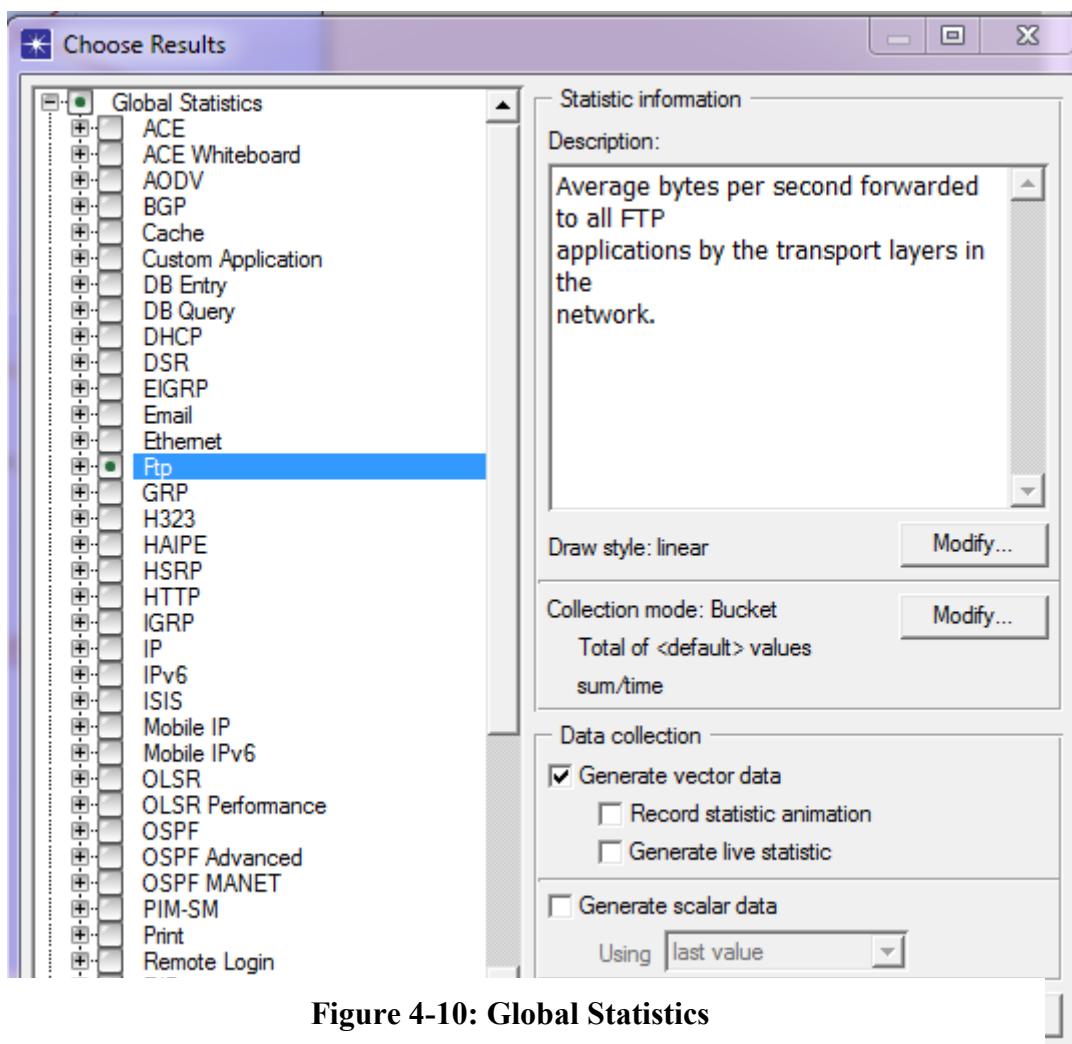


Figure 4-10: Global Statistics

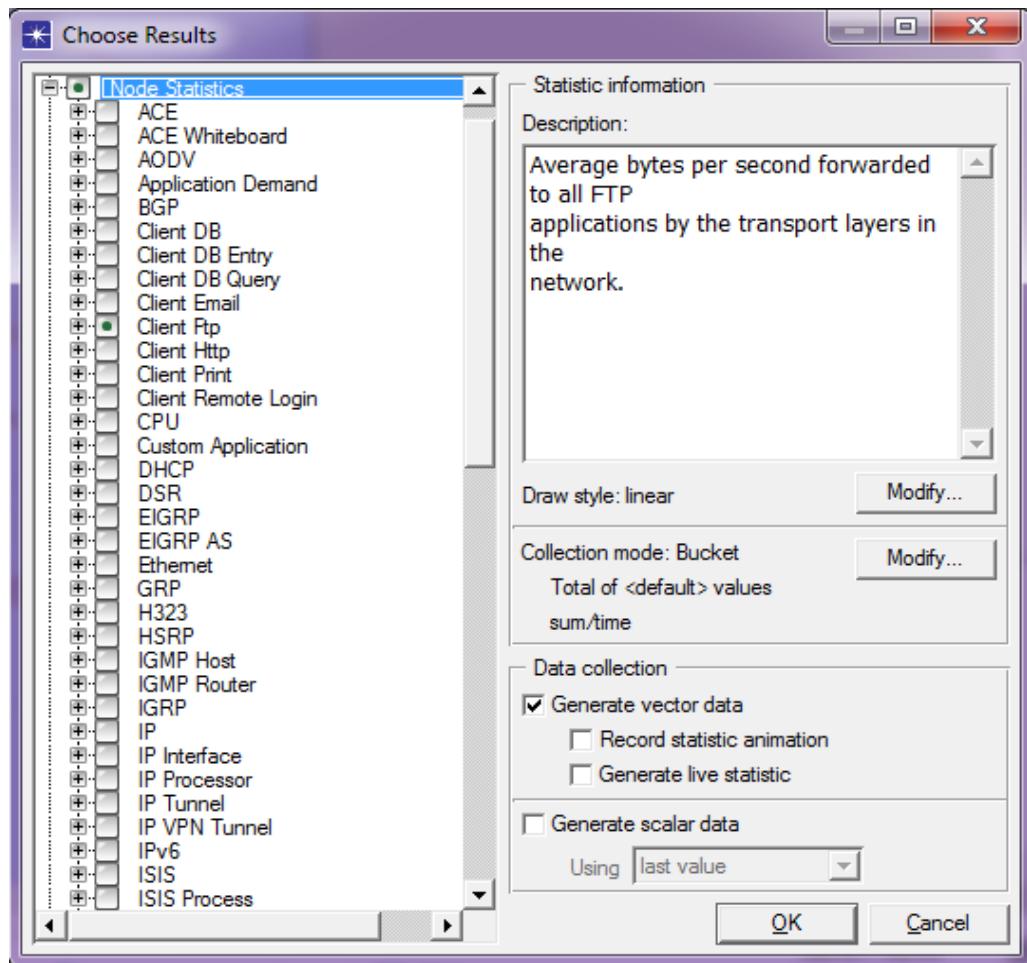


Figure 4-11: Node Statistics

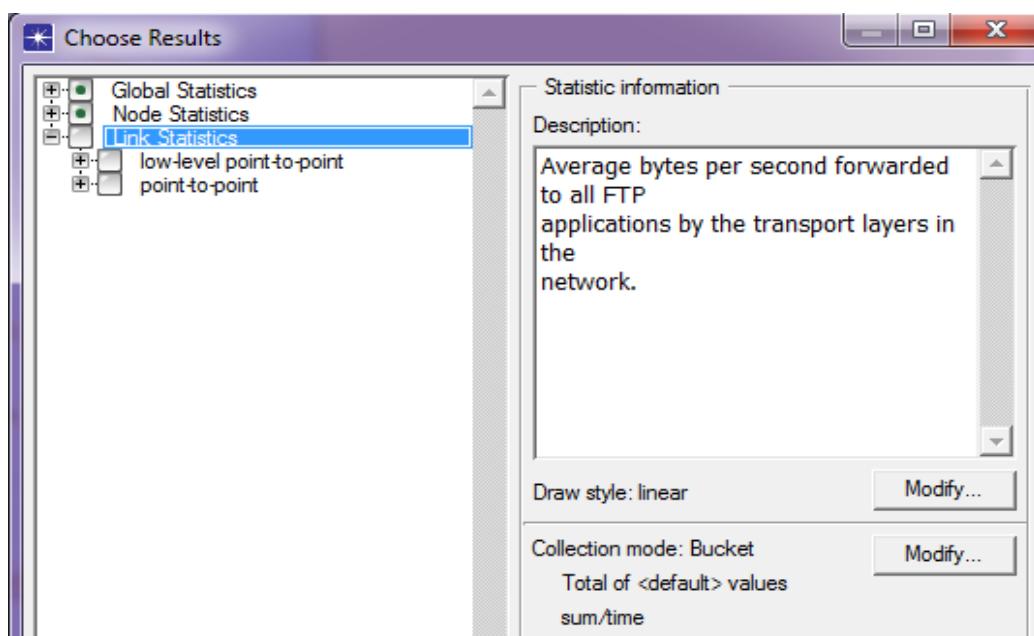
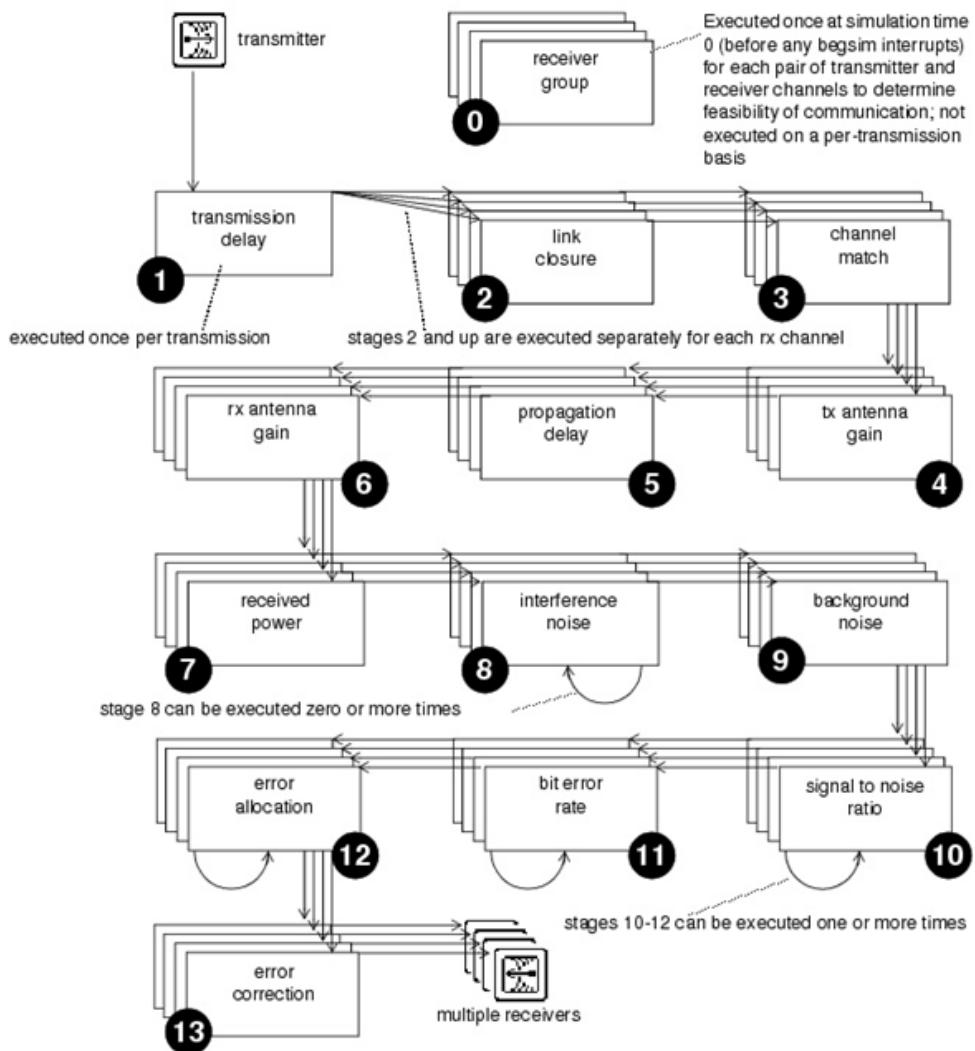


Figure 4-12: Link Statistics

4.1.5 Radio Transceiver Pipeline

OPNET Modeler uses radio transceiver pipeline stages to model wireless transmission of packets. Radio transceiver pipeline consists of fourteen stages, most of which must be executed on a per-receiver basis whenever a transmission occurs. Radio transceiver pipeline is shown in Figure 4-13.



4.1.5.1 Stage 0 (receiver group)

Figure 4-13: Radio Transceiver Pipeline

The purpose of this stage is to create an initial receiver group for each transmitter channel. Possible reasons for not including a receiver channel in the receiver group are:
Distinct frequency bands (do not overlap) or Physical separation.

4.1.5.2 Stage 1 (transmission delay)

It is executed just once per transmission. It is invoked to calculate the amount of time required for the entire packet to complete transmission. The total time taken to transmit a packet.

4.1.5.3 Stage 2 (closure)

It is responsible for determining if communication between the transmitter and receiver is possible, based on physical considerations like obstacles, surface of the earth etc.

4.1.5.4 Stage 3 (channel match)

It classifies the current transmission for a particular receiver channel as: firstly, valid transmissions where the packet may be accepted and forwarded to next modules for further processing. Secondly, Noise where the transmission can generate the interference at the receiver. Thirdly, ignore Transmissions.+

4.1.5.5 Stage 4 (tx antenna gain)

The purpose of the transmitter antenna gain stage is to compute the gain provided by the transmitter's associated antenna, based on the direction of the vector leading from the transmitter to the receiver.

4.1.5.6 Stage 5 (Propagation Delay)

Determine the time taken by the packet to travel from source to destination.

4.1.5.7 Stage 6 (Receiver Antenna Gain)

This is the first stage at the receiver side. Compute the gain provided by the receiver's associated antenna, based on the direction of the vector leading from the receiver to the transmitter.

4.1.5.8 Stage 7 (Receiver Power)

The purpose of this stage is to calculate the received power of arriving packet's signal (in watts). Computation of received power depends on many factors like transmission power, distance separating transmitter and receiver, transmission frequency, transmitter and receiver antenna gain.

4.1.5.9 Stage 8 (Interference Noise)

The purpose of this stage is to account for the interactions between transmissions that arrive concurrently at the same receiver channel.

4.1.5.10 Stage 9 (Background Noise)

The purpose of this stage is to represent the effect of all noise sources such as commercial radio, amateur radio, or television etc.

4.1.5.11 Stage 10 (Signal-to-Noise Ratio)

The purpose of this stage is to compute the current average power SNR for the arriving packet. SNR of the packet is important in determining receiver's ability to correctly receive the packet's content.

4.1.5.12 Stage 11 (Bit Error Rate)

BER stage is executed for all the valid packets for which SNR stage is executed. The purpose of the BER stage is to derive the probability of bit errors during the past interval of constant SNR.

4.1.5.13 Stage 12 (Error Allocation)

This segment might be the entire packet, if no changes in bit error probability occur over the course of the packet's reception. Bit error count estimation is usually based on the bit error probability (obtained from stage 11) and the length of the affected segment.

4.1.5.14 Stage 13 Error Collection

It is invoked when a packet completes reception, immediately after the final return of the error allocation stage, with no simulation time elapsing in between [Doc08].

4.2 Parameters

4.2.1 Network objects

4.2.1.1 Wlan_wkstn_adv node

This node model represents a wireless workstation, shown in Figure 4-14, with client-server applications running over TCP/IP and UDP/IP. It supports one underlying WLAN connection at 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps.



Figure 4-14: Wlan_wkstn_adv node

4.2.1.2 Wlan_ethernet_slip4_router

This is a wireless LAN based router with one Ethernet and 4 SLIP interfaces, shown in Figure 4-15.

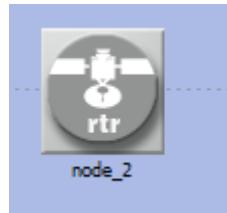


Figure 4-15: Wlan_ethernet_slip4_router



4.2.1.3 Ethernet4_slip8_gtwy_adv

This node model, shown in Figure 4-16, represents an IP-based gateway supporting four Ethernet hub interfaces, and eight serial line interfaces. IP packets arriving on any interface are routed to the appropriate output interface based on their destination IP address.

Figure 4-16:Ethernet4_slip8_gtwy_adv

4.2.1.4 Ethernet_server_adv

This model, shown in Figure 4-17, represents a server node with server applications running over TCP/IP and UDP/IP. It supports one underlying Ethernet connection at 10 Mbps, 100 Mbps, or 1Gbps.



Figure 4-17: Ethernet_server_adv

4.2.1.5 10Gbps_Ethernet_adv

The 10Gbps_Ethernet_adv duplex link represents an Ethernet connection operating at 10Gbps. The connected MACs must always operate in a full-duplex mode as the 802.3ae standard does not support half-duplex operation.

4.2.1.6 Application Configuration

This model, shown in Figure 4-18, specifies applications using available application types. There are several applications Database, Email, FTP, Http, Print, Remote login, Video Conferencing and voice. FTP application has Parameters shown in Figure 4-19; the client always downloads one file per session. The server may change for each session.



Figure 4-18: Application Configuration

The dialog box is titled '(Ftp) Table'. It contains a table with the following data:

Attribute	Value
Command Mix (Get/Total)	100%
Inter-Request Time (seconds)	exponential (360)
File Size (bytes)	constant (10000000)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

At the bottom are buttons for 'Details', 'Promote', 'OK', and 'Cancel'.

Figure 4-19: FTP Table



4.2.1.7 Profile Configuration

This node can be used to create user profiles, shown in Figure 4-20. These user profiles can be used by different nodes in the network. The applications defined in the "Application Configuration" objects are used by this object to configure profiles. "Application Configuration" object must be created before using this object. The attribute of the profiles shown in Figure 4-21.

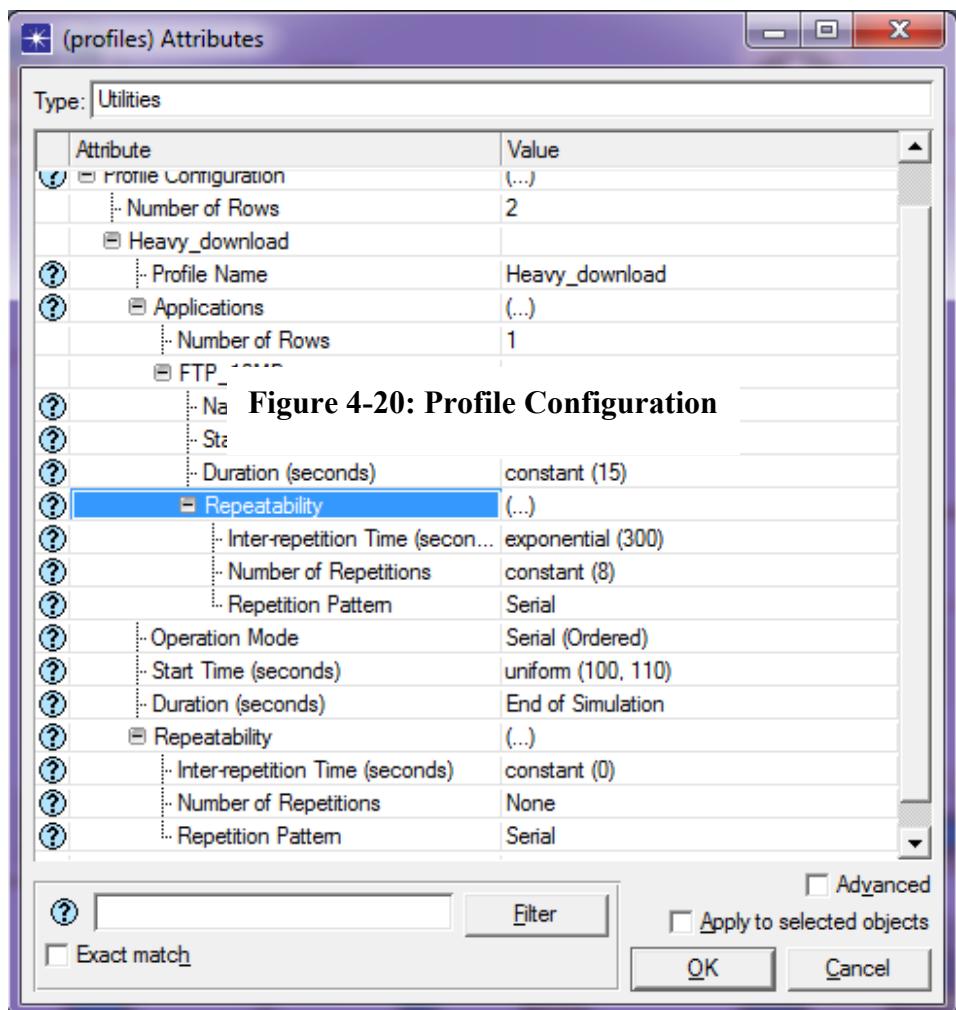


Figure 4-21: Profile Attributes

4.3 Experiments

In the evaluation experiments, simulation topology is comprised of nine cells located in 400 X 400 m². Each cell has an AP that serves a group of nodes. An AP and its associated nodes have same BSS identifier and channel number settings as shown in Figure 4-22.

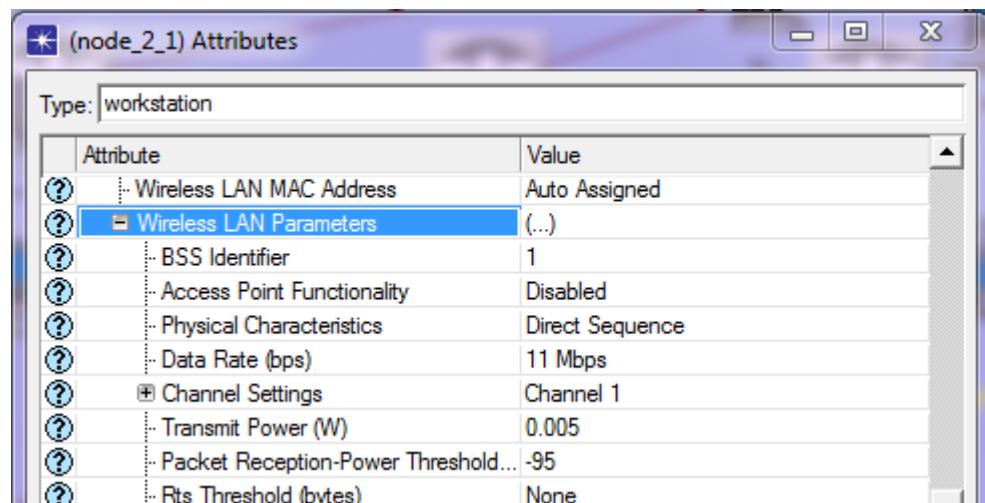
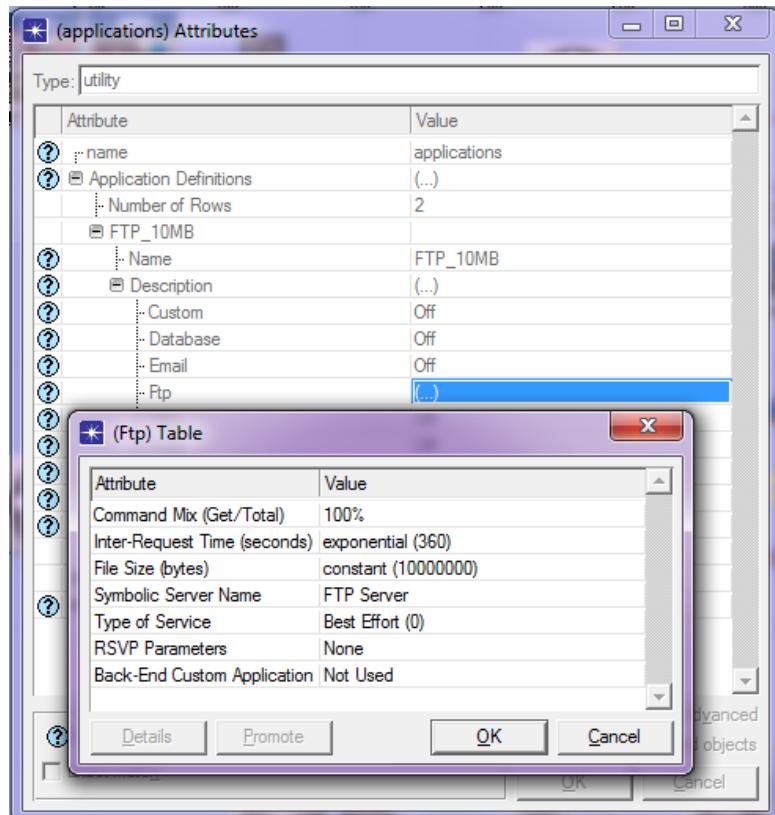


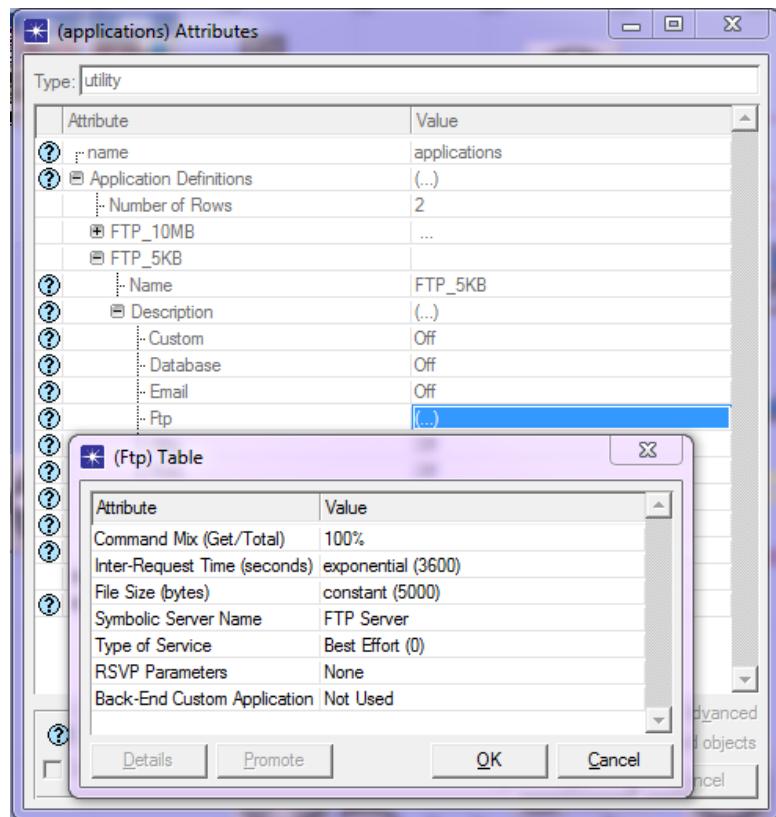
Figure 4-22: BSS identifier & Channel Number

4.3.1 Application Configuration



To model different loads, two FTP (File Transfer Protocol) applications are used, one of them has 10 MB (Mega Byte) file size, as shown in Figure 4-23, and the other has 5 KB (Kilo Byte) file size, as shown in Figure 4-24.

Figure 4-23: 10 MB (Mega Byte) file size

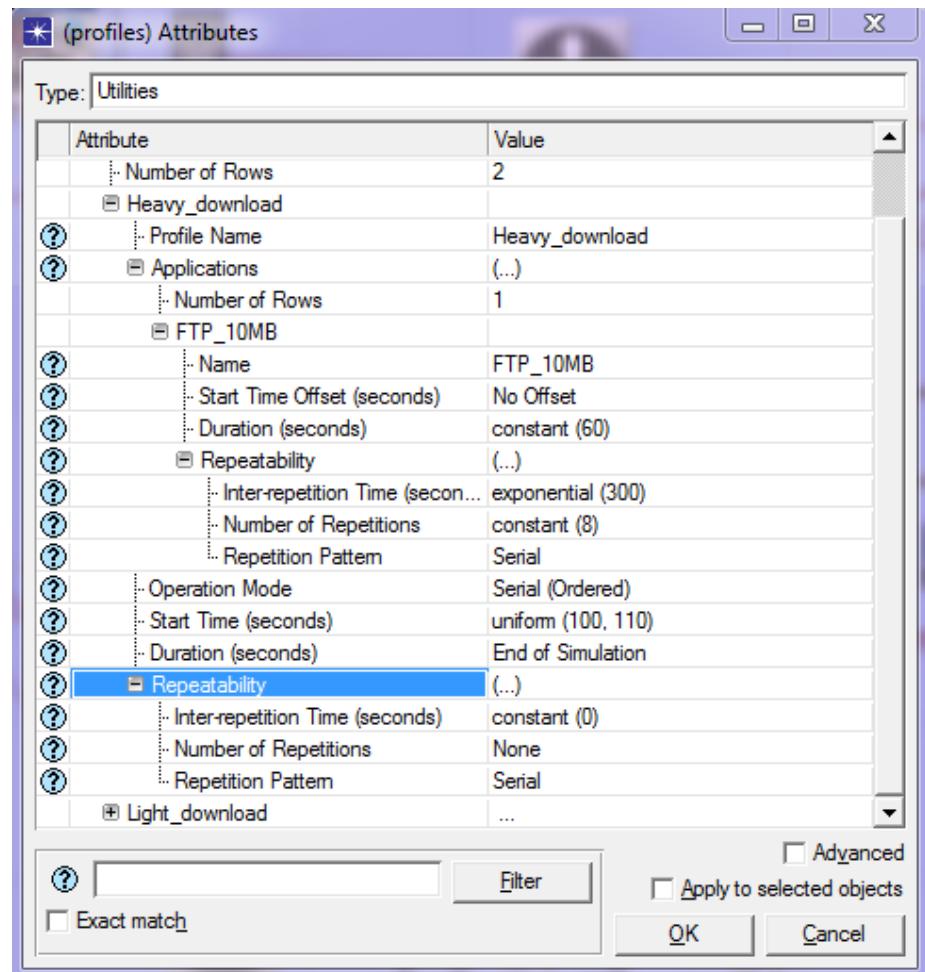


4.3.2 Profile Configuration

We have two profiles in our simulation, heavy download and light download.

4.3.2.1 Heavy Download Settings

Figure 4-24: 5 KB (Kilo Byte) file size



Heavy download profile settings shown in Figure 4-25. We illustrate these settings in

Table 4-1.[Doc08].

attribute	value	Attribute description
Name	FTP_10MB	Name of the application configured in this profile. The application should be defined in the 'Application configuration' object.
Duration (seconds)	Constant(60)	The maximum amount of time allowed for an application session before it aborts. This is often used as a timeout.
Number of repetitions	Constant(8)	Specifies the distribution name and arguments to be used for generating random session counts. "Unlimited" will allow new sessions to be created up until the "Application Stop Time" specification.
Repetition pattern	Serial	Defines when the next session of the application will start.
Operation mode	Serial (ordered)	Defines how applications will start. Serial (Ordered) - They can start one after each other in an ordered manner (first row to last row). Serial (Random) - They can start one after each other in a random manner. Simultaneous - They can start all at the same time.
Inter-repetition time (seconds)	Constant(0)	Defines when the next session of the profiles will start.

Table 4-1: Heavy download profile settings

4.3.2.2 Light Download Settings

Light download profile settings are shown in Figure 4-26.

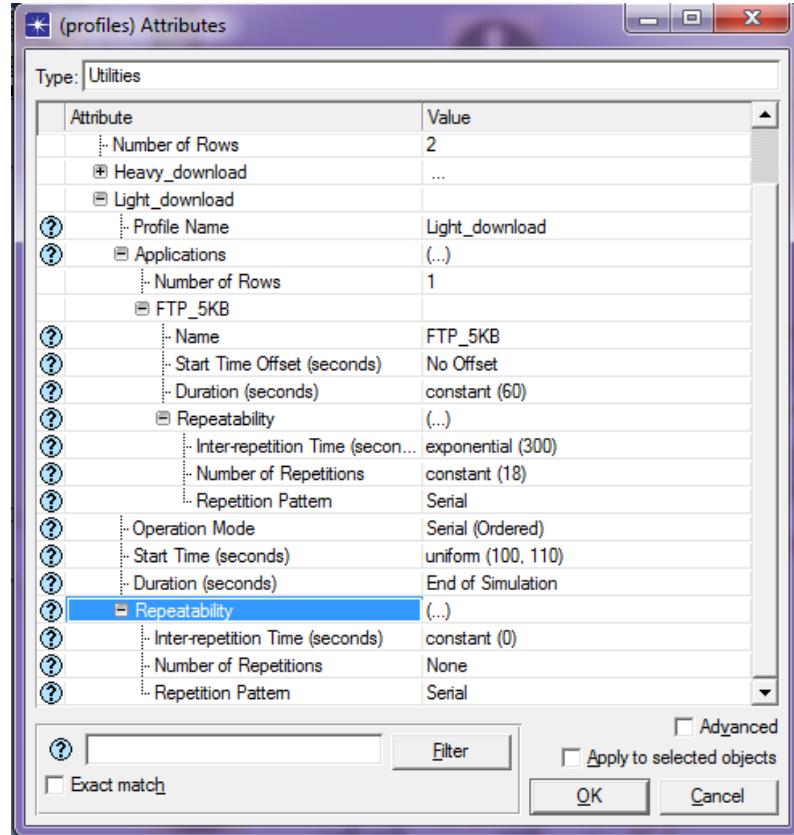


Figure 4-26: Light Download Profile Settings

4.3.3 Object Configuration

4.3.3.1 FTP Server configuration

In “application: supported Services” attribute, we configure two applications (FTP_5KB) and (FTP_10MB), as shown in Figure 4-27. This attribute is configured to start and setup services for various applications at this server. Clients can send traffic to this server for only those applications which are supported by this attribute.

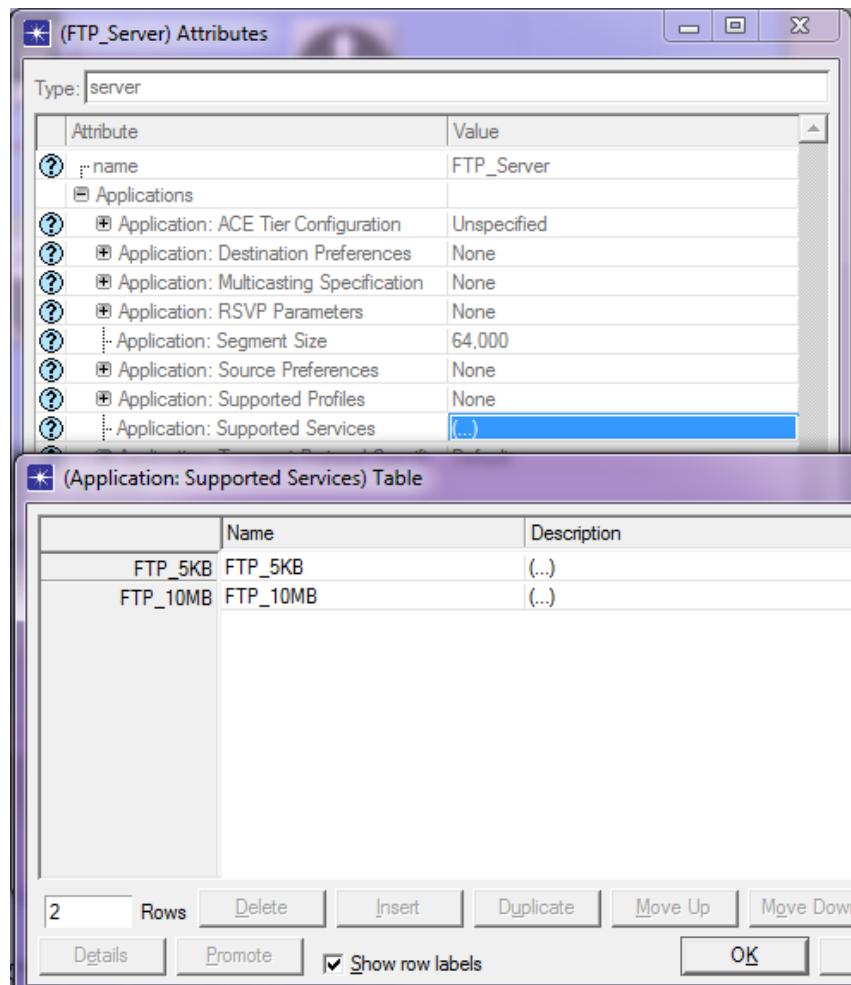


Figure 4-27: Application: Supported Services Attribute

4.3.3.2 wlan_ethernet_slip4_router (Access Point)

It serves as an AP and has two interested attributes as shown in Table 4-2.

attribute	value	Attribute description
BSS	BSS vary from one to nine	This attribute identifies the BSS to which the WLAN MAC belongs.
Channel settings	1,6 and 11	Specifies the frequency band that will be used by the radio transmitter and receiver connected to the MAC.

Table 4-2: Access Point Attributes

4.3.3.3 wlan_wkstn_adv (node model)

It represents a workstation with client-server applications running over TCP/IP and UDP/IP. It has three interested attributes shown in Table 4-3. To mimic real world applications (heterogeneous applications), we have two types of workstations. The first one downloads a file of 5 KB size, which has a profile (Light_download). The other one downloads a file of 10 MB size, which has profile (Heavy_download). We determine the type of profile as shown in Figure 4-28, through Applications: supported profiles attribute.

attribute	value	Attribute description
BSS	BSS vary from one to nine	This attribute identifies the BSS to which the WLAN MAC belongs. Their BSS should be the same BSS of an AP that serves it.
Channel settings	1,6 and 11	Specifies the frequency band that will be used by the radio transmitter and receiver connected to the MAC. Their Channel

		settings should be the same Channel settings of an AP that serves it.
Applications: supported profiles	Heavy_download or Light_download	Specifies the names of all profiles which are enabled on this node.

Table 4-3: Workstation Attributes

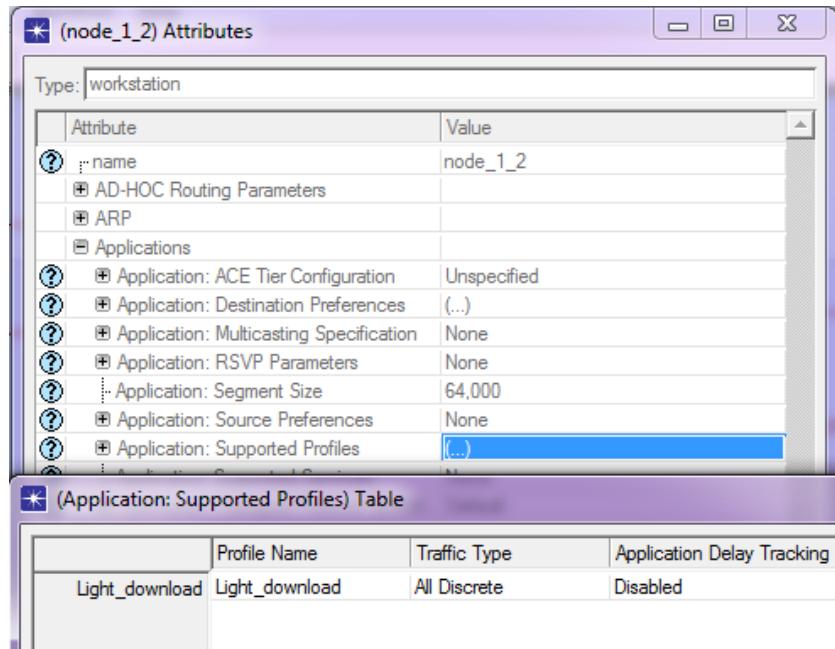


Figure 4-28: Applications: supported profiles attribute

4.3.4 Modified code

To implement our interference measurement proposed approach and channel assignment algorithm, detailed in chapter two, we modify the code in the file “wlan_mac.pr”, the modified code in provided appendix A.

4.3.5 Statistics

The proposed method for channel selection is evaluated by comparing the measured network throughput against static method and power method. The network throughput is the FTP traffic Received in (Bytes/sec) all FTP applications and forwarded by the transport layers in the network.

Chapter 5

Scenarios, Simulation Results and Discussion

5.1 Scenarios and Results

In our simulation model, we have nine cells located in a 400 X 400 m² area. We have three scenarios. In the first and second: We increase the number of nodes while fixing the load. In the third one, we increase the load, while fixing the number of nodes. For each scenario, we applied the proposed, static and power methods [Elw12][Hai08][Zho13][Zho12][Yang13] and [Zha11]. We also run every scenario ten times with seed values chosen randomly.

The proposed method for channel selection is evaluated by comparing the measured network throughput against the static method and the power method. The network throughput is the metric used in the evaluation. It is the FTP traffic forwarded to by the transport layers in the network and received in Bytes/sec by all FTP applications.

5.1.1 Scenario One (Increasing Number of nodes)

We have two kinds of (Heterogeneous) cells according to the downloaded file size, light cells and heavy cells. In the light cells, every node downloads a file of 5 KB size. While in heavy cells, the file size is 10 MB. We increase the number of nodes in the topology

as 40, 65, 85, 105, 117 nodes. We apply proposed, static, and power methods for every scenario so that we have fifteen scenarios. We also run every scenario ten times with different seed values chosen randomly. The type of the cell (light or heavy cell) and its channel which is configured manually (static method) are shown in Figure 5-1.

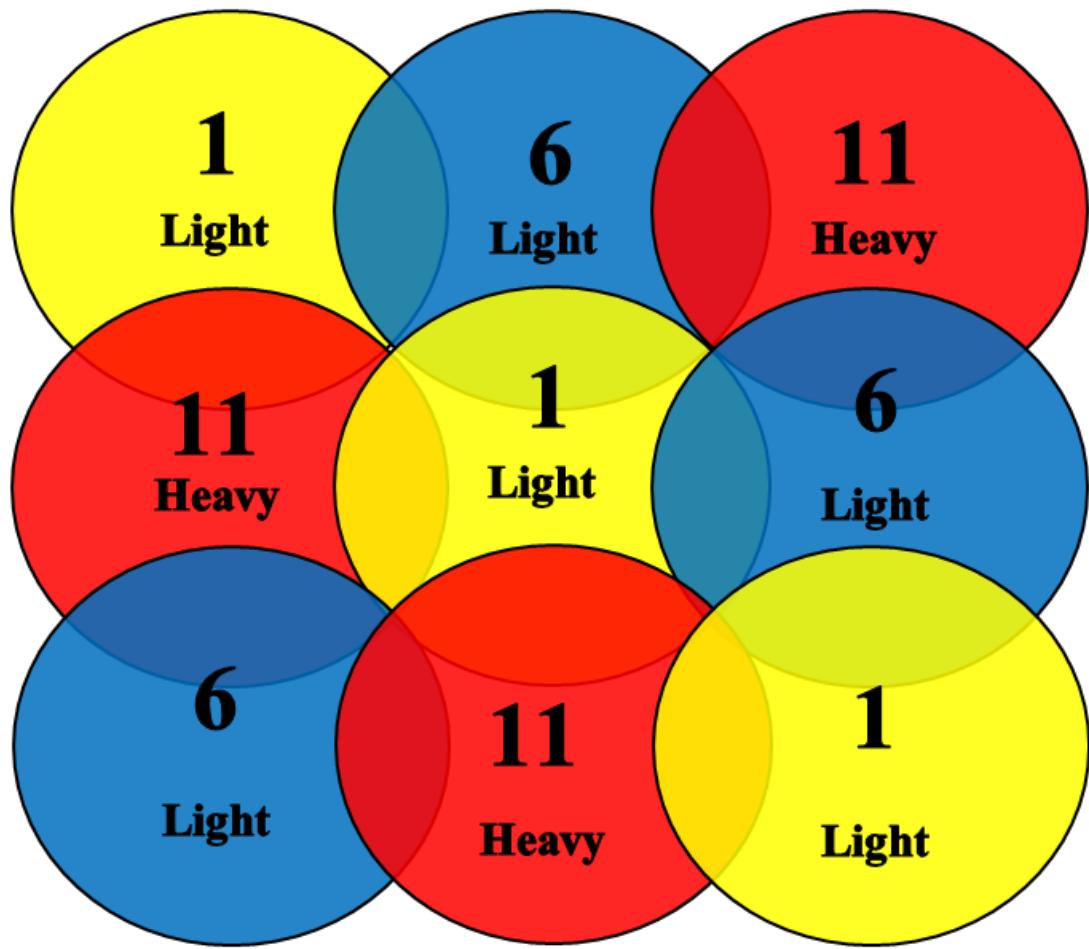


Figure 5-1: Scenario One (Increasing Number of nodes)

5.1.1.1 Results of Scenario One (Increasing Number of Nodes)

The results of scenario one is provided in Table 5-1 and plotted in Figure 5-2. The results reveals that the proposed method outperforms the power and static channel allocation methods [Elw12][Hai08][Zho13][Zho12][Yang13] and [Zha11]. This superiority becomes more clear when the number of nodes is increased. This means that in heterogeneous scenarios, the power method and the static channel allocation method do not guarantee optimal performance. The proposed method tries to utilize the radio resources by mitigating interference conditions.

Number of NODES	STATIC METH. (MB/s)	POWER METH. (MB/s)	PROPOSED METHOD (MB/s)
40	0.363	0.408	0.402
65	0.395	0.405	0.565
85	0.365	0.358	0.607
105	0.213	0.202	0.682
117	0.082	0.079	0.595

Table 5-1: Increasing number of Nodes “FTP traffic received (MEGABYTES/SEC)”

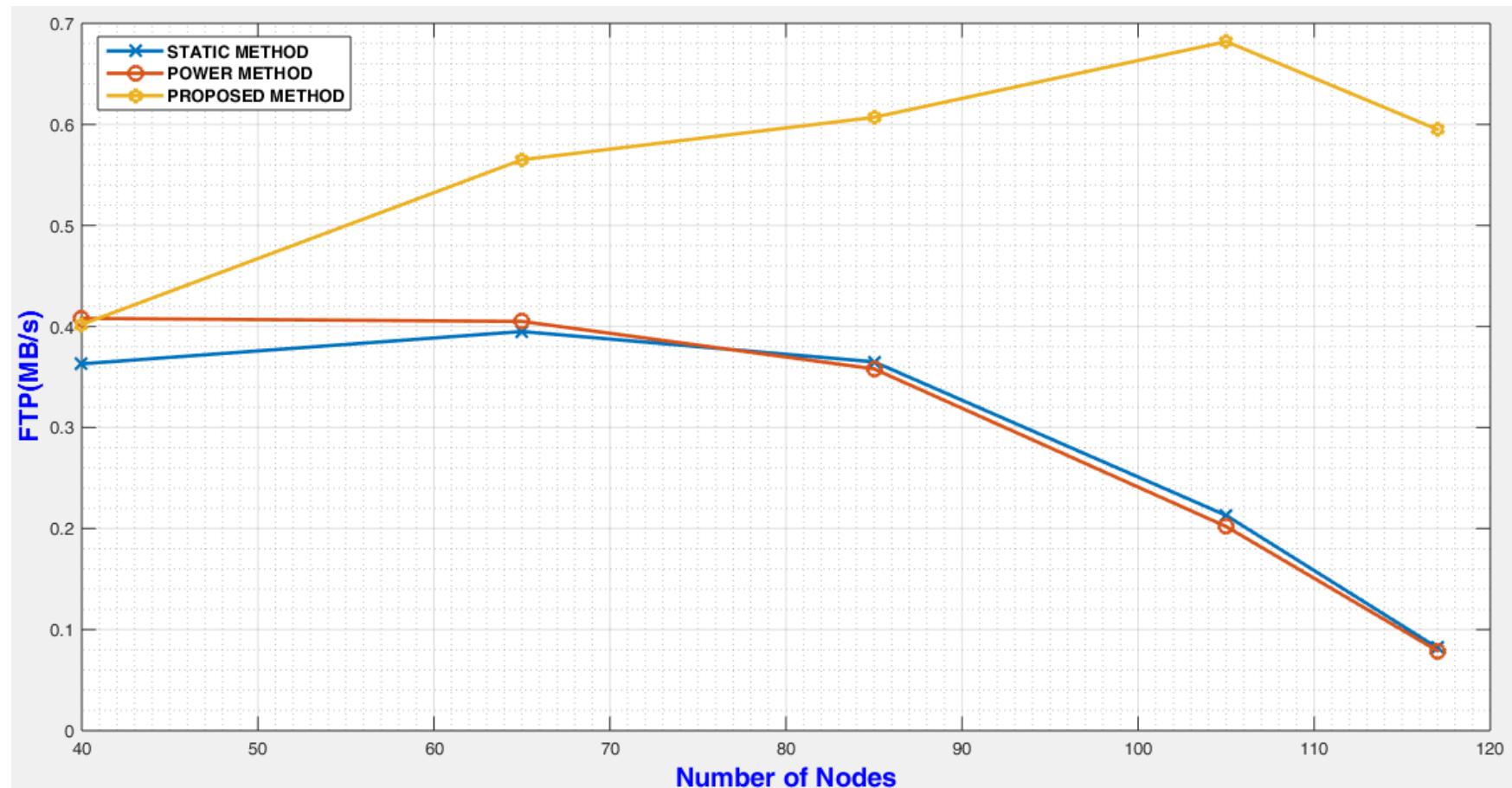


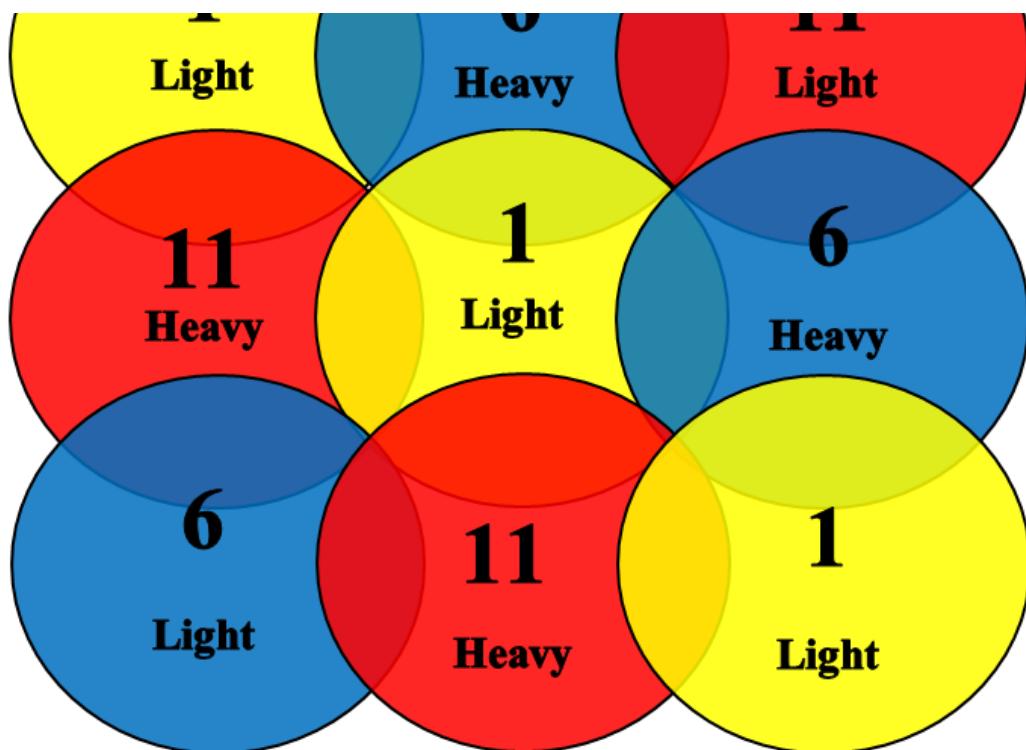
Figure 5-2: Throughput Results of Scenario One - Increasing Number of Nodes (40,65,85,105 and 117)

5.1.2 Scenario Two (Increasing Number of Nodes – Different Distribution)

Here we have a different distribution of the light and heavy cells as shown in Figure 5-3. We apply the proposed, static, and power methods.



Figure 5-3: Scenario Two (Increasing Number of nodes – Different Distribution)



5.1.2.1 Results of Scenario Two (Increasing Number of Nodes – Different Distribution)

The results of scenario Two is provided in Table 5-2 and plotted in Figure 5-4. Clearly, as the difference between cell loads increase, the difference in performance becomes greater with the proposed method.

Number of NODES	STATIC METH. (MB/s)	POWER METH. (MB/s)	PROPOSED METHOD METH. (MB/s)
40	0.471	0.474	0.470
65	0.650	0.671	0.672
85	0.748	0.751	0.810
105	0.738	0.734	0.925
117	0.658	0.662	0.917

Table 5-2: Increasing number of Nodes “FTP traffic received (MEGABYTES/SEC)”

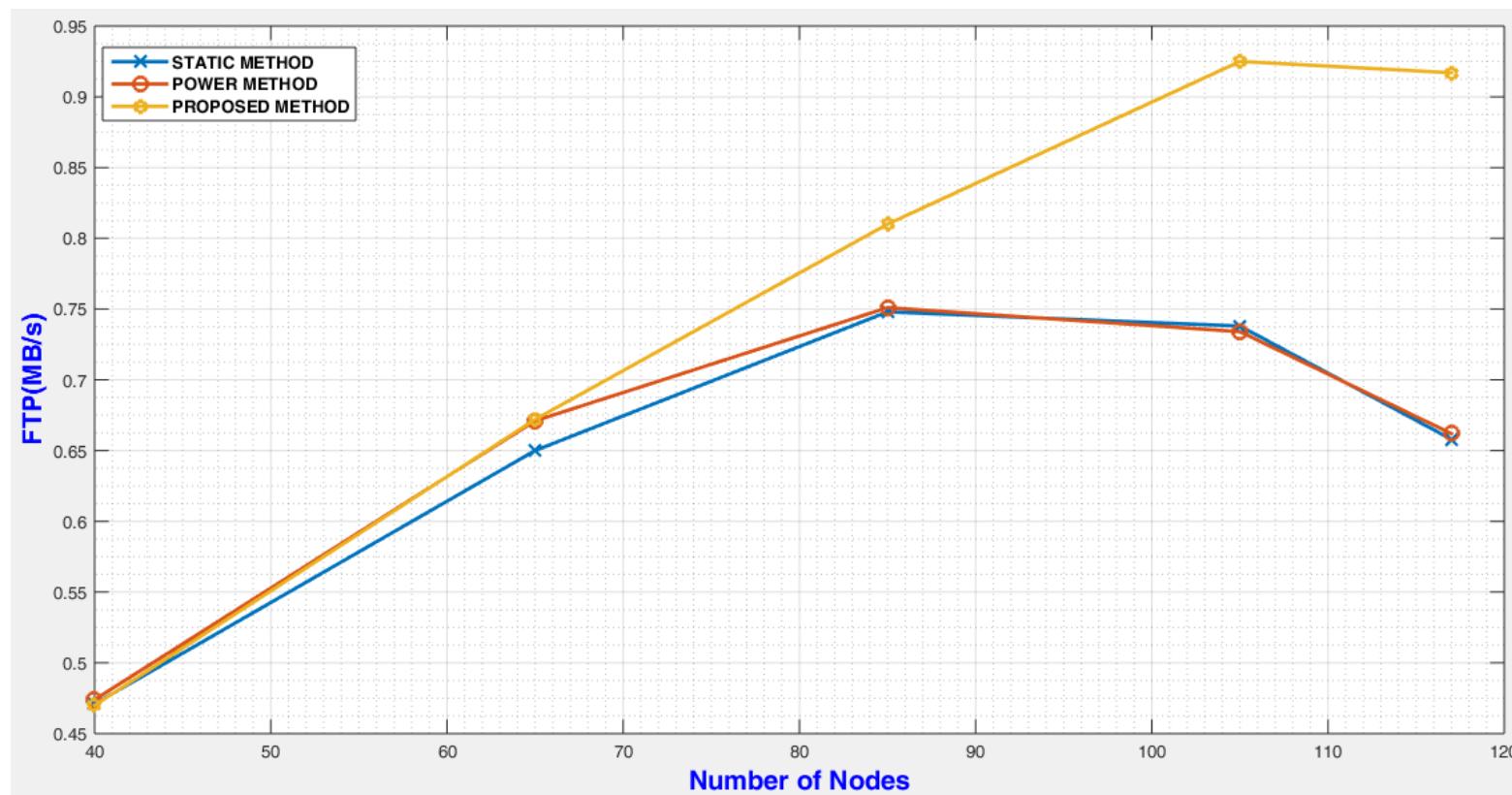


Figure 5-4: Throughput Results of Scenario Two - Increasing Number of Nodes (40,65,85,105 and 117) - Different Distribution Compared with Scenario one

5.1.3 Scenario Three (Increasing Load of Nodes)

In this scenario, we fix the number of nodes to 65 nodes. Then, we increase the load by adjusting attribute “Number of repetitions” in the profile attributes for both light and heavy profiles as shown in Table 5-3Error! Reference source not found.. This attribute is described in Table 4-1. Channels of cells are assigned as shown in Figure 5-5. We apply proposed, static, and power methods for each scenario.

Number Of Repetitions	
High Profile	Light Profile
4	5
8	18
12	60
16	100
20	150
23	350

Table 5-3: Number Of Repetitions

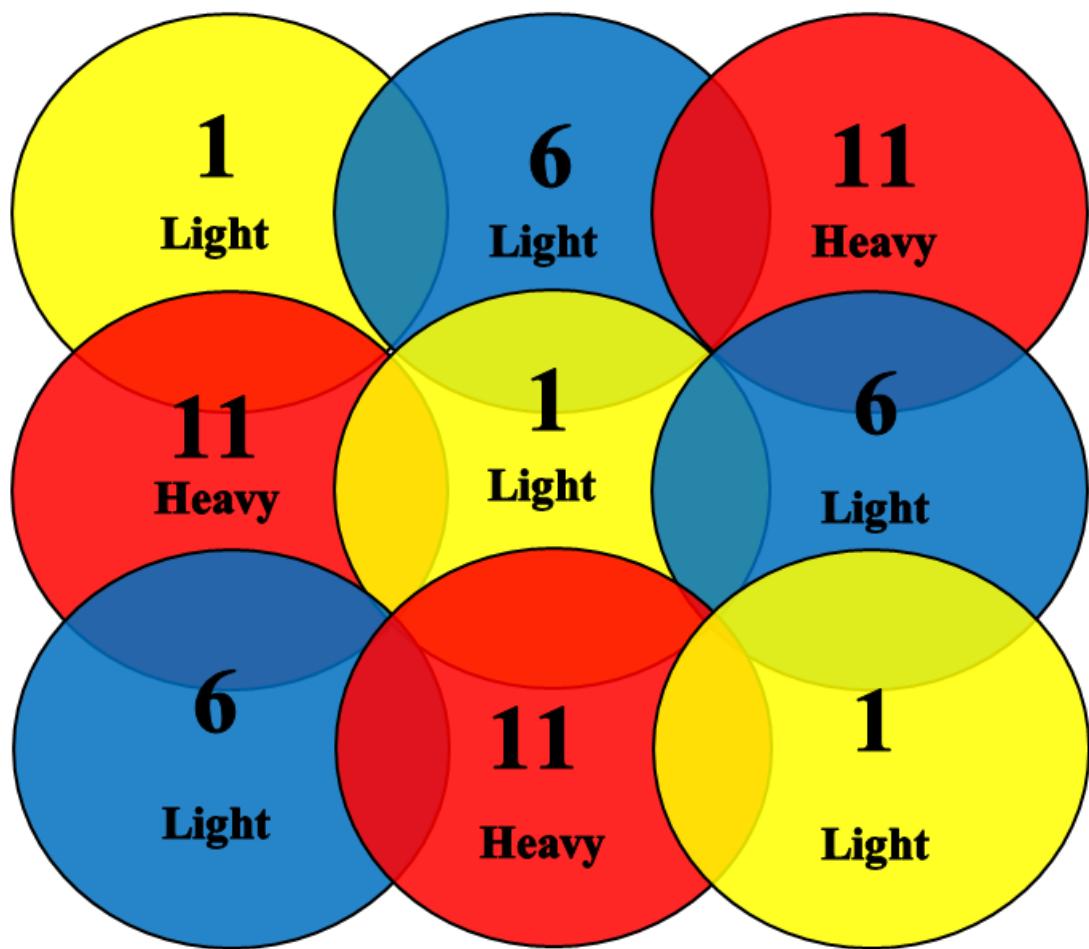


Figure 5-5:Scenario Three (Increasing Loads of 65 nodes)

5.1.3.1 Results of Scenario Three (Increasing Load of Nodes)

The results of scenario Three is provided in Table 5-4 and plotted in Figure 5-6. Clearly, as loads (Number of Repetitions High/Light) of nodes increase, the difference in performance becomes greater with the proposed method compared with static and power methods.

NUMBER OF REPETITIONS HIGH/LIGHT	STATIC METH. (MB/s)	POWER METH. (MB/s)	PROPOSED METHOD METH. (MB/s)
4-5	0.287	0.291	0.396
8-18	0.401	0.399	0.555
12-60	0.444	0.462	0.675
16-100	0.466	0.473	0.745
20-150	0.454	0.464	0.786
23-350	0.471	0.447	0.804

**Table 5-4: Increasing Loads of 65 Nodes “FTP traffic received
(MEGABYTES/SEC)”**

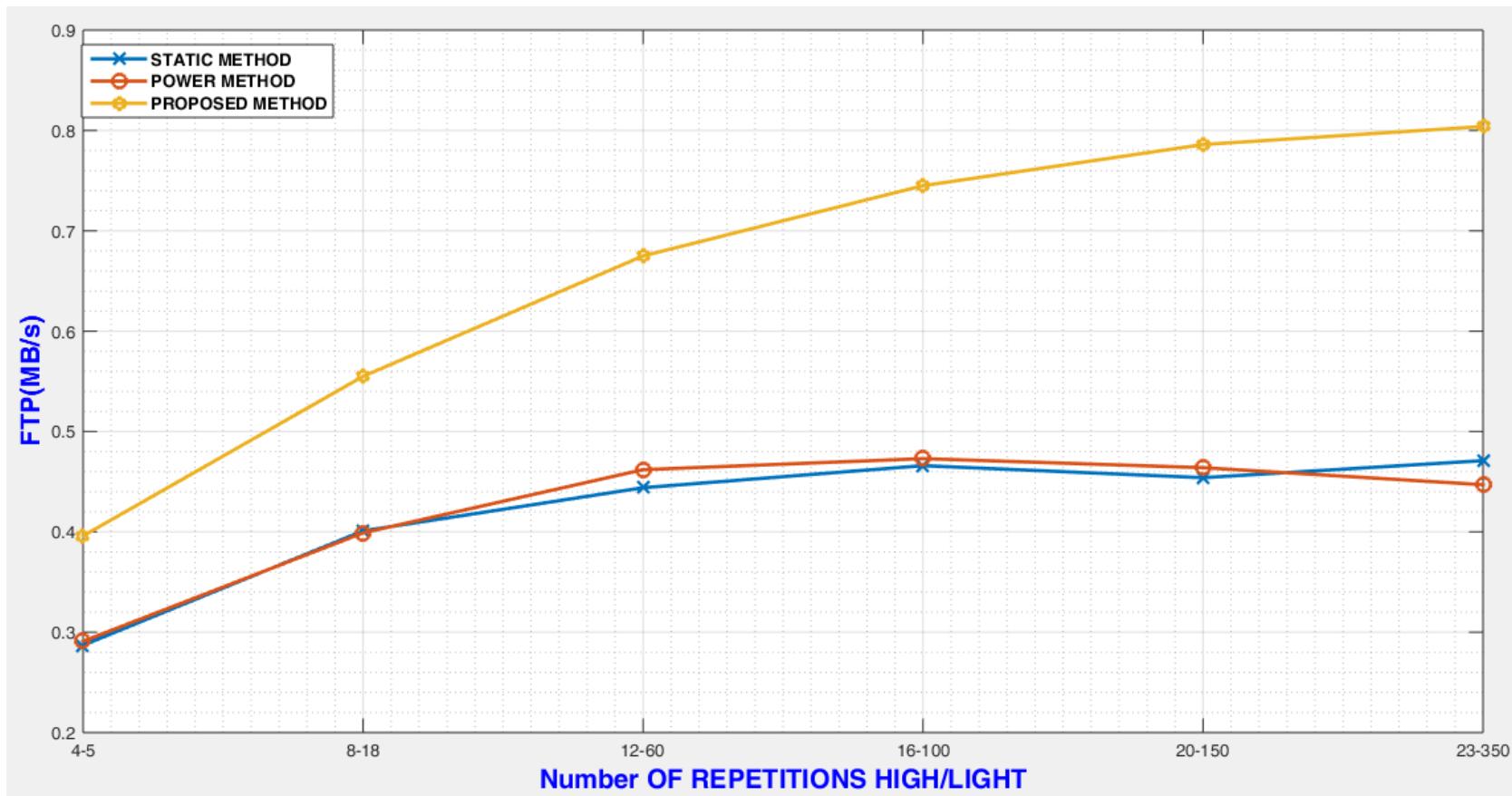


Figure 5-6: Throughput Results of Scenario Three - Increasing Loads of 65 Nodes

Chapter 6

Conclusions and Future work

6.1 Conclusions

As shown in chapter 5, the proposed method has achieved better performance compared to legacy methods. This is due to the fact that, it is measurement-based and considers the real network status in terms of:

- Users distribution.
- Number of Users.
- Users Traffic types.
- Interference relations among communication links.

Capturing the above parameters allows us better characterize the network conditions and hence allocate radio resources in a better method.

6.2 Future work

As a future work, we propose to extend the analysis and experiments to consider the features of the new emerging devices and technologies, like 802.11ac and 802.11n. In those technologies, channels bandwidth may be different as they support 40, 80, and 160MHz channels. This creates a new dimension in the study in terms of widths of allocated channels. Another aspect which may be considered as well, is the network of heterogeneous devices (802.11b, 802.11g, 802.11n, 802.11ac).

Bibliography

- [Akl07] R. Akl and A. Arepally, “Dynamic Channel Assignment in IEEE 802.11 Networks”, 2007 IEEE.
- [Aruba] Aruba 802.11n Networks, Version 8, 2011. The home page of Aruba is <http://www.arubanetworks.com>.
- [Bad13] A. Badgujar and S. Nikam, “A study of Channel Assignment Strategies used for Uncoordinated WLANs”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-2, Issue-5, June 2013.
- [Bok13] F. Bokhari and G. Zaruba, “i-POCA: Interference-aware Partially Overlapping Channel Assignment in 802.11-based Meshes”, 2013 IEEE.
- [Chi10] S. Chieochan, E. Hossain and J. Diamond, “Channel Assignment Schemes for Infrastructure-Based 802.11 WLANs: A Survey”, IEEE Communications Surveys & Tutorials, VOL. 12, No. 1 , first Quarter 2010.
- [Cui11] Y. Cui, W. Li and X. Cheng, “Partially Overlapping Channel Assignment Based on “Node Orthogonality” for 802.11 Wireless Networks”, 2011 IEEE.
- [Doc08] OPNET Modeler Documentation Set, Version: 14.5, 1986–2008 OPNET Technologies.

- [Elw12]** M. Elwekeil, M. Alghoniemy, M. El-Khamy, H. Furukawa and O. Muta, “Optimal Channel Assignment For IEEE 802.11 Multi-Cell WLANs”, 20th European Signal Processing Conference (EUSIPCO 2012).
- [Gas02]** M. Gast, “802.11® wireless Networks: The Definitive Guide”, Publisher: O’Reilly, April 2002.
- [Gon13]** D. Gong, M. Zhao and Y. Yang, “Channel Assignment in Multi-Rate 802.11n WLANs”, 2013 IEEE Wireless Communication and Networking Conference (WCNC): MAC.
- [Hai08]** M. Haidar, R. Ghimire, H. Al-Rizzo, R. Akl and Y. Chan, “Channel Assignment in an IEEE 802.11 WLAN Based on Signal to interference Ratio”, 2008 IEEE.
- [Lu12]** Z. Lu and H. Yang, Unlocking the Power of OPNET Modeler, PP 1-100, 2012 1st Edition.
- [Mish05]** A. Mishra, S. Banerjee and W. Arbaugh, “Weighted coloring based channel assignment for WLANs”, 2005.
- [Motorola12]** Motorola Solutions, WHAT YOU NEED TO KNOW ABOUT 802.11AC, July 2012.
- [Mur12]** A. Murad, “Heterogeneous uncoordinated deployment of WLANs: an evolving problem for current and future Wi-Fi access”, Int. J. Network Mgmt 2013; 23:66-79, published online 25 September 2012 in Wiley online Library(wileyonlinelibrary.com) DOI:10.1002/nem.1815.
- [Sri09]** S. Srivasak, K. Wongthavarawat, K. Limmongkol and S. Siwamogsatham, “Channel Assignment Strategies of IEEE 802.11 WLAN in a shared Public Hotspot Environment”, 2009 IEEE.

- [Tektronix] The homepage of Tektronix. : <http://www.tektronix.com/wifi>
- [Xu11] W. Xu, C. Hua and A. Haung, “Channel Assignment and User Association Game in Dense 802.11 Wireless Networks”, 2011 IEEE.
- [Yang13] Y. Yang, I. Chen and C. Yang, “Channel Assignment for infrastructure-Based 802.11 WLANs ”, International journal of future Computer and Communication, Vol. 2, No. 5, Oct. 2013.
- [Yue10] X. Yue, C. Wong and S. Chan, “A Distributed Channel Assignment Algorithm for Uncoordinated WLANs”, 2010.
- [Zha11] H. Zhang, H. Ji and W. Ge, “Channel Assignment with Fairness for multi-AP WLAN based on Distributed Coordination Function”, IEEE WCNC 2011- MAC.
- [Zho12] H. Zhou and Chao Liu, WLAN Channel Assignment Based on Channel Overlap Factor, 2012 second International Conference on Instrumentation and Measurement, Computer, Communication and Control.
- [Zho13] K. Zhou, X. Jia, Y. Chang and X. Tang, “Partially overlapping channel assignment for WLANs using SINR interference model”, Int. J. Commun. Syst. (2013), published online in Wiley online Library(wileyonlinelibrary.com) DOI:10.1002/dac.2527.

Appendices

Appendix A

Appendix A-1: The first modified code is inserted in Procedure “static void wlan_mac_sv_init ()”.

```
static void
wlan_mac_sv_init ()
{
.
.
.

FIN (wlan_mac_sv_init ());

/*****************/
//You should change some variable according to numbers of
nodes (consist AP) and numbers of BSS AS:
//1. Line 49
//2. Line 51
//3. and some variables in State variable index of array
//4. in File
C:/Users/Ibrahim/op_models/number_nodes_remaining_in_bss.txt
number of nodes and BSS ---each line represents number of nodes
(nodes+AP) in each BSS--- ----number of lines represent number of
BSS---
//5. You should set BSS from 1,2,3, and so on

/*Initialize state variables*/
count_first_time_rcv_pkt=0;
count_read_AP_from_file=0;
count_change_channel_node=0;
sim_elapsed_time=0;
test_elapsed_time=0;
TimeBeginTest=100;
//Determine after what time do you want enter "TEST MODE"
"CHANGE"
duration_test=1;
//Determine duration Interval time in "TEST MODE" "CHANGE"
count_begintest=0;
count_endtest=0;
number_nodes_accum=0;
number_nodes=114;
//Determine Number of Nodes (nodes+AP) in your "Topology"
"CHANGE"
back_to_origin_ch=OPC_FALSE;
number_bss=9;
//Determine Number of BSS in your "Topology" "CHANGE"
bss_pre_change_channel=OPC_TRUE;
skip_pkt_analysis=OPC_FALSE;
```

```

for(i=0;i<3;i++)
{
    interference_ch[i]=0;
    accum_interferences[i]=0;
    counter_packets_outside_cell[i]=0;
    counter_packets_inside_cell[i]=0;
    cal_interference[i]=OPC_TRUE;
    channel_overlap[i]=OPC_FALSE;

    frequency_ch_array[i]=band_freq + 2401;
    band_freq+=25;
}

for(i=0;i<4;i++)
{
    count_test_ch[i]=0;
}

for(i=0;i<number_nodes;i++)
{
    for(j=0;j<4;j++)
    {
        interference_node_array[i][j]=0;
    }
}

for(i=0;i<number_bss;i++)
{
    number_nodes_exist_in_bss_array[i]=0;
    number_nodes_exist_accum_in_bss_array[i]=0;
    number_nodes_remaining_in_bss_array[i]=0;
    change_frequency_array[i]=0;
}

for(j=0;j<number_bss;j++)
{
    for(i=0;i<3;i++)
    {
        counter_packets_from_cells[j][i]=0;
    }
}

/*Open file "allow_test_node.txt" to initialize its value */
fp=fopen("C:/Users/ibrahim/op_models/allow_test_node.txt","w");
/*write value 1 on (allow_test_node.txt) file to allow first
node to begin test*/
fprintf(fp,"%d",1);

fclose(fp);

/*Open file "allow_test_bss.txt" to initialize its value */
fp=fopen("C:/Users/Ibrahim/op_models/allow_test_bss.txt","w");
/*write value 1 on (allow_test_bss.txt) file to allow nodes
in BSS (ID=1) to begin test*/

```

```

fprintf(fp,"%d",1);

fclose(fp);

/*Open file "number_nodes_remaining_in_bss.txt" to read
their values*/
fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_remaining_
in_bss.txt","r");

/*read values from (number_nodes_remaining_in_bss.txt) file
and fill them in Array (num_nodes_remaining_bss_array) to check
how many nodes remaining in BSS that wait Testing*/
j=0;
while(!feof(fp))
{
    fscanf(fp,"%d",&number_nodes_remaining_in_bss_array[j]);

    number_nodes_exist_in_bss_array[j]=number_nodes_remaining_in_
bss_array[j];

    number_nodes_change_channel_array[j]=number_nodes_remaining_
in_bss_array[j];

    number_nodes_exist_accum_in_bss_array[j]=number_nodes_remain
ing_in_bss_array[j]+number_nodes_accum;

    number_nodes_accum=number_nodes_exist_accum_in_bss_array[j];

    j++;
}

fclose(fp);

/*Open file "number_nodes_change_channel_bss.txt" to
initialize its value */
fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_change_cha
nnel_bss.txt","w");

/*write number_nodes_change_channel_array in
(number_nodes_change_channel_array.txt) file*/
for(i=0;i<number_bss;i++)
{
    fprintf(fp,"%d\n",number_nodes_change_channel_array[i]);
}

fclose(fp);

/*create new text file
"reslut_interferences_for_each_node.txt" to write the results of
interference on it*/
fp=fopen("C:/Users/Ibrahim/op_models/reslut_interferences_fo
r_each_node.txt", "w" );
fclose(fp);

/*create new text file "test_channel_file.txt" to write the
results of interference on it*/

```

```

        fp=fopen("C:/Users/Ibrahim/op_models/test_channel_file.txt",
"w" );
        fclose(fp);

        /*create new text file
"counter_packets_outside_cell_file.txt" to write number of
packets that received from outside Cell on it*/
        fp=fopen("C:/Users/Ibrahim/op_models/counter_packets_outside_
cell_file.txt", "w" );
        fclose(fp);

        /*create new text file
"counter_packets_inside_cell_file.txt" to write number of packets
that received from inside Cell on it*/
        fp=fopen("C:/Users/Ibrahim/op_models/counter_packets_inside_
cell_file.txt", "w" );
        fclose(fp);

        /*create new text file "time_begin_test_node_file.txt" to
write number of packets that received from inside Cell on it*/
        fp=fopen("C:/Users/Ibrahim/op_models/time_begin_test_node_fi
le.txt", "w" );
        fclose(fp);

        /*create new text file
"accumulative_interferences_for_each_AP.txt" to write the results
of accumulative interference on it*/
        fp=fopen("C:/Users/Ibrahim/op_models/accumulative_interferen
ces_for_each_AP.txt", "w" );
        fclose(fp);

        /*create new text file "list_nodes_finish_test.txt" to write
on it the name of the nodes according to thier order in Testing
Mode*/
        fp=fopen("C:/Users/Ibrahim/op_models/list_nodes_finish_test.
txt", "w" );
        fclose(fp);

        /*create new text file "number_nodes_finish_test.txt". Each
node finish its Test Mode add number "one" to this file, which AP
uses this value to ensure all nodes finish their Testing Mode*/
        fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_finish_tes
t.txt","w");
        fclose(fp);

        /*Open file "allow_node_change_channel.txt" to initialize
its value */
        fp=fopen("C:/Users/Ibrahim/op_models/allow_node_change_chann
el.txt","w");
        fclose(fp);

        /*write value 0 on (allow_node_change_channel.txt) file to
prevent node from changing its channel*/
        for(i=0;i<number_bss;i++)
        {
            fprintf(fp,"%d\n",change_frequency_array[i]);
        }

```

```

fclose(fp);

/*Open file "number_nodes_finish_test.txt" to initialize its
value */
fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_finish_tes
t.txt","w");

/*write initial value 0 on (number_nodes_finish_test.txt)
file*/
fprintf(fp,"%d",0);

fclose(fp);

/*create new text file
"counter_packets_outside_cell_file.txt" to write number of
packets that received from Cells*/
fp=fopen("C:/Users/IBRAHIM/op_models/counter_packets_from_ce
lls_file.txt", "w" );
fclose(fp);

```

Appendix A-2: The second modified code is inserted in a new procedure which is called “rxgroup_recompute ”. to compute receiver group channel for every transmitter.

```

static int
rxgroup_recompute (Objid txch, Objid rxch)
{
    double frequency_tx,frequency_rx,bandwidth_tx,bandwidth_rx;
    /* Okay if not part of the same node */

    op_ima_obj_attr_get (txch, "min frequency", &frequency_tx);
    op_ima_obj_attr_get (txch, "bandwidth", &bandwidth_tx);
    op_ima_obj_attr_get (rxch, "min frequency", &frequency_rx);
    op_ima_obj_attr_get (rxch, "bandwidth", &bandwidth_rx);

    bandwidth_rx/=1000.0;
    bandwidth_tx/=1000.0;

    return ((frequency_tx + bandwidth_tx > frequency_rx) &&
(frequency_rx + bandwidth_rx > frequency_tx) &&
(op_topo_parent(op_topo_parent (op_topo_parent
(txch))) !=op_topo_parent(op_topo_parent (op_topo_parent
(rxch))))) ;
}

```

Appendix A-3: The third modified code is inserted in a new procedure which is called “test_channel_mode”. to apply proposed interference measurements.

```

void

test_channel_mode (int init_channel,int final_channel,int
bss_id_receiver_para,int bss_id_sender_para,int
test_elapsed_time_para,double Pi,float Li,float Ri)
{
    int                                     i,j,k;
    double                                frequency_tx,frequency_rx;
    Objid
    para_attr_objid_SND,wireless_lan_para_comp_attr_objid_SND;

    FIN(test_channel_mode (int init_channel,int
final_channel,int bss_id_receiver_para,int bss_id_sender_para));

    for(i=init_channel;i<final_channel;i++)
    {
        if (((i*duration_test) <= test_elapsed_time_para) &&
test_elapsed_time_para < ((i+1)*duration_test))
        {
            if (bss_id_receiver!=bss_id_sender)
            {
                ++counter_packets_outside_cell[i];
            }
            else if (bss_id_receiver==bss_id_sender)
            {
                ++counter_packets_inside_cell[i];
            }
            ++counter_packets_from_cells[bss_id_sender-
1][i];
            if (count_test_ch[i]==0)
            {
                j=0;
                do
                {
op_ima_obj_attr_set (txch_objid, "min frequency",
frequency_ch_array[i+j]);
op_ima_obj_attr_set (rxch_objid, "min frequency",
frequency_ch_array[i+j]);

op_ima_obj_attr_get (txch_objid, "min frequency", &frequency_tx);
op_ima_obj_attr_get (rxch_objid, "min frequency", &frequency_rx);

op_radio_txch_rxgroup_compute(OPC_OBJID_UNIVERSAL,rxgroup_recompute);

op_radio_txch_rxgroup_get (txch_objid, &num_receivers,
&receivers);

                if
(num_receivers==0)
                {

interference_ch[i+j]=0;
cal_interference[i+j]= OPC_FALSE;

```

```

channel_overlap[i+j]=OPC_FALSE;

++count_test_ch[i+j];

++j;

}

else
{



for(k=0;k<num_receivers;k++)

{

//Check if there is an Overlapping between channels in
RXGROUP
op_ima_obj_attr_get(op_topo_parent(op_topo_parent (op_topo_parent
(receivers[k])), "Wireless LAN
Parameters", &wireless_lan_para_comp_attr_objid_SND);

para_attr_objid_SND=op_topo_child(wireless_lan_para_comp_attr_obj
id_SND,OPC_OBJTYPE_GENERIC,0);

op_ima_obj_attr_get(para_attr_objid_SND,"BSS
Identifier",&bss_id_rxgroup);

if(bss_id_receiver!=bss_id_rxgroup)

{

channel_overlap[i+j]=OPC_TRUE;

}

}

op_prg_mem_free (receivers);

}

while ((num_receivers == 0) && (frequency_tx <
frequency_ch_array[2]));

if (num_receivers == 0)
{
op_ima_obj_attr_set (txch_objid, "min frequency",
origin_frequency);

op_ima_obj_attr_set (rxch_objid, "min frequency",
origin_frequency);

op_ima_obj_attr_get (txch_objid, "min frequency", &frequency_tx);
op_ima_obj_attr_get (rxch_objid, "min frequency", &frequency_rx);

op_radio_txch_rxgroup_compute
(OPC_OBJECT_UNIVERSAL,rxgroup_recompute);

```

```

back_to_origin_ch=OPC_TRUE;
}

++count_test_ch[i+j];

}

/*we use (Else) / Except first packet from interference
calculation because it may be comes from prev channel so that we
use else if not if*/

else if (bss_id_receiver!=bss_id_sender && cal_interference[i]==
OPC_TRUE)
{
/* node k summarize the measured power and activity from all
neighboring cells which operate on the same channel I= summation
(Pi*Li)/Ri for all captured packets*/

interference_ch[i] = interference_ch[i] +
((Pi*Li)/Ri)*pow(10,10);

op_ima_obj_attr_get (txch_objid, "min frequency", &frequency_tx);
op_ima_obj_attr_get (rxch_objid, "min frequency", &frequency_rx);
}

} //End of if (((i*duration_test) <= test_elapsed_time) &&
test_elapsed_time < ((i+1)*duration_test))

} //End of for(i=0;i<3;i++)

FOUT;
}

```

Appendix A-4: final modified code is inserted in a procedure which is called “wlan_physical_layer_data_arrival”.

```

static void
wlan_physical_layer_data_arrival (void)
.
.
.
wlan_rcvd_frame_ptr = op_pk_get (i_strm);
/*********************************************
//To speed Simulation spead analysis pkt until Node change its
channel
if(skip_pkt_analysis==OPC_FALSE)
{
    /*Initialize variable*/
    Pi=0;
    Li=0;

```

```

Ri=0;

/* Define object ID for node which received packet*/
objid_RCVD = op_topo_parent(op_id_self());

/*Obtain the values assigned to different attributes
(Receiver)*/
op_ima_obj_attr_get(objid_RCVD,"Wireless LAN
Parameters",&wireless_lan_para_comp_attr_objid_RCVD);

para_attr_objid_RCVD=op_topo_child(wireless_lan_para_comp_attr_objid_RCVD,OPC_OBJTYPE_GENERIC,0);

op_ima_obj_attr_get(objid_RCVD,"name",&name_node);

/*Get the BSS ID for (Receiver)*/
op_ima_obj_attr_get(para_attr_objid_RCVD,"BSS
Identifier",&bss_id_receiver);

/*Define object ID for node which create packet*/
objid_SND=op_topo_parent(op_pk_creation_mod_get
(wlan_rcvd_frame_ptr));

/*obtain the values assigned to different attributes (Sender)*/
op_ima_obj_attr_get(objid_SND,"Wireless LAN
Parameters",&wireless_lan_para_comp_attr_objid_SND);

para_attr_objid_SND=op_topo_child(wireless_lan_para_comp_attr_objid_SND,OPC_OBJTYPE_GENERIC,0);

/*Get the BSS ID for (Sender)*/
op_ima_obj_attr_get(para_attr_objid_SND,"BSS
Identifier",&bss_id_sender);
}

sim_elapsed_time = op_sim_time();

/*Determine time on which Node X Begin calculate interference*/
if(TimeBeginTest < sim_elapsed_time)
{
    /*Node X will read file (allow_test_node.txt) to check if
allow to begin test*/
    if (count_begintest==0)
    {

        fp=fopen("C:/Users/Ibrahim/op_models/allow_test_node.txt","r
");
        //read value from (allow_test_node.txt) file, if its 1 Node X
        will begin test, else it can not

        fscanf(fp,"%d",&check_allow_option_node);

        fclose(fp);

        /*Open file "allow_test_bss.txt" to Check the order of BSS for
Testing */
}
}

```

```

fp=fopen("C:/Users/Ibrahim/op_models/allow_test_bss.txt","r");

//read value from (allow_test_bss.txt) file, if its the same BSS
that Node X belong it, then Node X will begin test, else it can
not
fscanf(fp,"%d",&bss_id_allow_test);

fclose(fp);

/*Open file "number_nodes_remaining_in_bss.txt" to read their
values*/

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_remaining_in_bs
s.txt","r");

/*read values from (number_nodes_remaining_in_bss.txt) file and
fill them in Array (num_nodes_remaining_bss_array) to check how
many nodes remaining in BSS that wait Testing*/
j=0;

while(!feof(fp) && j<number_bss)
{
    fscanf(fp,"%d",&number_nodes_remaining_in_bss_array[j]);
    j++;
}

fclose(fp);

/*Open file "number_nodes_change_channel_bss.txt" to read values
of number_nodes_change_channel_bss*/

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_change_channel_
bss.txt","r");

j=0;
while(!feof(fp) && j<number_bss)
{
    fscanf(fp,"%d",&number_nodes_change_channel_array[j]);
    j++;
}

fclose(fp);

//to except first BSS from checking previous
number_nodes_change_channel_array condition

if(bss_id_receiver>1)
{
    if(number_nodes_change_channel_array[bss_id_receiver-2]!=0)
    {
        bss_pre_change_channel=OPC_FALSE;
    }
    else if(number_nodes_change_channel_array[bss_id_receiver-
2]==0)

```

```

    {

        bss_pre_change_channel=OPC_TRUE;
    }
}

//if Node X read value check_allow_option_node==1 and exist in the
//same BSS which exist in TEST Mode and never Test before then node
//will proceed in TEST Mode

if (check_allow_option_node==1 &&
bss_id_receiver==bss_id_allow_test &&
number_nodes_remaining_in_bss_array[bss_id_receiver-1]!=0 &&
bss_pre_change_channel)
{
}

//if Node X read value check_allow_option_node==1 from
//(allow_test_node.txt) then node will not read from the file
//(allow_test_node.txt) another time

++count_begintest;

fp=fopen("C:/Users/Ibrahim/op_models/allow_test_node.txt","w");

//write value 0 on (allow_test_node.txt) file to prevent onther
node to begin test

fprintf(fp,"%d",0);

fclose(fp);

TimeBeginTest_node = op_sim_time();

//Record Time on which node begin test

fp=fopen("C:/Users/Ibrahim/op_models/time_begin_test_node_file.tx
t","a");

fprintf(fp, "%s\n", name_node);

fprintf(fp,"%f\n",TimeBeginTest_node);

fclose(fp);

}

}

test_elapsed_time=op_sim_time() - (TimeBeginTest_node);

if(check_allow_option_node==1 &&
bss_id_receiver==bss_id_allow_test &&
number_nodes_remaining_in_bss_array[bss_id_receiver-1]!=0 &&
bss_pre_change_channel)
{
    init_channel=0;
}

```

```

final_channel=3;

Pi=op_td_get_dbl(wlan_rcvd_frame_ptr,OPC_TDA_RA_RCVD_POWER);

Li=op_pk_total_size_get(wlan_rcvd_frame_ptr);

Ri=op_td_get_dbl(wlan_rcvd_frame_ptr,OPC_TDA_RA_RX_DRATE);

test_channel_mode
(init_channel,final_channel,bss_id_receiver,bss_id_sender,test_el
apsed_time,Pi,Li,Ri);

/*if the Node X ends test it will permit another node to begin
test*/
if (count_endtest==0 && test_elapsed_time > (3*duration_test))
{
    ++count_endtest;

/*to end test for this node make (check_allow_option_node=0) */
check_allow_option_node=0;

//After finish its Test Mode Back to its Tx and Rx Original
channel if it is not change to its original channel yet

if(back_to_origin_ch==OPC_FALSE)
{
    op_ima_obj_attr_set (txch_objid, "min frequency",
origin_frequency);

    op_ima_obj_attr_set (rxch_objid, "min frequency",
origin_frequency);

    op_radio_txch_rxgroup_compute
(OPC_OBJID_UNIVERSAL,rxgroup_recompute);

    op_ima_obj_attr_get (txch_objid, "min frequency",
&frequency_tx);

    op_ima_obj_attr_get (rxch_objid, "min frequency",
&frequency_rx);

    back_to_origin_ch=OPC_TRUE;
}

//Allow other Node to Begin Test

fp=fopen("C:/Users/Ibrahim/op_models/allow_test_node.txt","w");

                                //write value 1 on
(allow_test_node.txt) file to permit onther node to begin test

```

```

fprintf(fp,"%d",1);

fclose(fp);

--number_nodes_remaining_in_bss_array[bss_id_receiver-1];

/*Open file "number_nodes_remaining_in_bss.txt" to write modified
values of num_nodes_remaining_bss_array*/

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_remaining_
in_bss.txt","w");

/*Write modified values of num_nodes_remaining_bss_array to
(number_nodes_remaining_in_bss.txt) file to point to how many
nodes remaining in BSS that wait Testing*/

for(i=0;i<number_bss;i++)
{
    fprintf(fp,"%d\n",number_nodes_remaining_in_bss_array[i]);

}
fclose(fp);

/* when value of num_nodes_remaining_bss_array[i]=0, then every
node in this BSS fininsh its TEST Mode*/
if(number_nodes_remaining_in_bss_array[bss_id_receiver-1]==0)
{
/*Open file "allow_test_bss.txt" to write on it Next BSS that
will proceed in Test Mode*/

fp=fopen("C:/Users/Ibrahim/op_models/allow_test_bss.txt","w"
);

++bss_id_allow_test;

//Write Next BSS ID that will Proceed In Test Mode on file
(allow_test_bss.txt)

fprintf(fp,"%d",bss_id_allow_test);

fclose(fp);

}

//Open file to write name of nodes on it according to their order
in Testing Mode

fp =
fopen("C:/Users/Ibrahim/op_models/list_nodes_finish_test.txt",
"a");

fprintf(fp, "%s\n", name_node);

fclose(fp);

```

```

//Open file to write number of packets that received from outside
cell

fp =
fopen("C:/Users/Ibrahim/op_models/counter_packets_outside_cell_file.txt", "a");

fprintf(fp, "%s\n", name_node);
for(i=0;i<3;i++)
{
    fprintf(fp, "%d\n", counter_packets_outside_cell[i]);
}

fclose(fp);

//Open file to write number of packets that received from inside
cell

fp =
fopen("C:/Users/Ibrahim/op_models/counter_packets_inside_cell_file.txt", "a");

fprintf(fp, "%s\n", name_node);
for(i=0;i<3;i++)
{
    fprintf(fp, "%d\n", counter_packets_inside_cell[i]);
}

fclose(fp);

//Open file to write number of packets that received from
Different cells

fp =
fopen("C:/Users/IBRAHIM/op_models/counter_packets_from_cells_file.txt", "a");

fprintf(fp, "%s\n", name_node);

for(j=0;j<number_bss;j++)
{
    fprintf(fp, "PKTs FROM BSS = %d are ....\n",j+1);

    for(i=0;i<3;i++)
    {

        fprintf(fp, "during channel %d it recevied
%d\n", (5*i)+1,counter_packets_from_cells[j][i]);
    }
}

fclose(fp);

//Open file to write which channel is tested for each node
fp =
fopen("C:/Users/Ibrahim/op_models/test_channel_file.txt", "a");

```

```

fprintf(fp, "%s\n", name_node);
for(i=0;i<3;i++)
{
    fprintf(fp, "%d\n", count_test_ch[i]);
}

fclose(fp);

//Open file to write results of interferences in Testing Mode
fp =
fopen("C:/Users/Ibrahim/op_models/reslut_interferences_for_each_node.txt", "a");

fprintf(fp, "%d\n", bss_id_receiver);

for (m=0; m<3;m++)
{
    fprintf(fp, "%f\n", interference_ch[m]);
}
fclose(fp);

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_finish_test.txt",
", "r");

//read value from (number_nodes_finish_test.txt) file, and add
one to this value
fscanf(fp,"%d",&increment_node);

++increment_node;

fclose(fp);
fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_finish_test.txt",
", "w");

//Increment value "increment_node " one in
(number_nodes_finish_test.txt) file, to point to number of nodes
that complete Testing Mode

fprintf(fp,"%d",increment_node);

fclose(fp);

} //End of if (count_endtest==0 && sim_elapsed_time >
(TimeBeginTest_node+(3*duration_test)))

} //End of if(check_allow_option_node==1)

} //End of if(TimeBeginTest < sim_elapsed_time)

//only For AP
if((TimeBeginTest+(3*duration_test)*number_nodes_exist_accum_in_b
ss_array[bss_id_receiver-1]) < sim_elapsed_time)
{
    if(ap_flag==OPC_BOOLINT_ENABLED &&
count_read_AP_from_file==0)
    {
/*Open file "number_nodes_remaining_in_bss.txt" to read their
values*/

```

```

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_remaining_in_bs
s.txt","r");

/*read values from (number_nodes_remaining_in_bss.txt) file and
fill them in Array (num_nodes_remaining_bss_array) to check how
many nodes remaining in BSS that wait Testing*/

j=0;

while(!feof(fp) && j<number_bss)
{
    fscanf(fp,"%d",&number_nodes_remaining_in_bss_array[j]);
    j++;
}

fclose(fp);

if(number_nodes_remaining_in_bss_array[bss_id_receiver-
1]==0)
{
    ++count_read_AP_from_file;

//Open file to read resluts of interferences in Testing Mode
fp =
fopen("C:/Users/Ibrahim/op_models/reslut_interferences_for_each_n
ode.txt", "r");

i=0; j=0;

while (!feof(fp) &&
i<number_nodes_exist_accum_in_bss_array[bss_id_receiver-1])
{
    if(j<4)
    {
        fscanf(fp,"%lf",&interference_node_array[i][j]);
    }

    j++;
}

else
{
    j=0;
    i++;
}

}
}

fclose(fp);

for (j=0;j<number_nodes_exist_accum_in_bss_array[bss_id_receiver-
1];j++)
{
}

```

```

if(bss_id_receiver==interference_node_array[j][0])
{
    for(i=0;i<3;i++)
    {
        accum_interferences[i]+=interference_node_array[j][i+1];
    }
}
}

/*Open file
"accumulative_interferences_for_each_AP.txt.txt" to write
accumulative_interferences on it*/
fp=fopen("C:/Users/Ibrahim/op_models/accumulative_interferences_for_each_AP.txt","a");

/*write
accumulative_interferences on file
"accumulative_interferences_for_each_AP.txt.txt"*/
for(i=0;i<3;i++)
{
    fprintf(fp,"%f\n",accum_interferences[i]);
}

fclose(fp);

minimum_interference = accum_interferences[0];
channel_number = 1;

for(i=0;i<3;i++)
{
    if(accum_interferences[i] < minimum_interference)
    {
        minimum_interference = accum_interferences[i];
        channel_number = i*5+1;
    }
}

freq_after_cal = ((channel_number-1)*5)+frequency_ch_array[0];

/*Open file "allow_node_change_channel.txt" to read its
frequencies to modify them*/
fp=fopen("C:/Users/Ibrahim/op_models/allow_node_change_channel.txt","r");

j=0;
while(!feof(fp) && j<number_bss)
{

```

```

fscanf(fp,"%lf",&change_frequency_array[j]);
j++;
}

fclose(fp);

change_frequency_array[bss_id_receiver-1]=freq_after_cal;

/*Open file "allow_node_change_channel.txt" to write modified
frequencies on it*/
fp=fopen("C:/Users/Ibrahim/op_models/allow_node_change_channel.txt",
"wt");

/*write value 0 on (allow_node_change_channel.txt) file to
prevent node from changing its channel*/

for(i=0;i<number_bss;i++)
{
fprintf(fp,"%f\n",change_frequency_array[i]);
}
fclose(fp);
}// End if(number_nodes_remaining_in_bss_array[bss_id_receiver-
1]==0)
}//End if(ap_flag==OPC_BOOLINT_ENABLED &&
count_read_AP_from_file==0)
}//End

if((TimeBeginTest+(3*duration_test)*number_nodes_exist_accum_in_b
ss_array[bss_id_receiver-1]) < sim_elapsed_time)

// For each node in BSS

if((TimeBeginTest+(3*duration_test)*number_nodes_exist_accum_in_b
ss_array[bss_id_receiver-1]) < sim_elapsed_time &&
count_change_channel_node==0)
{
/*Open file "allow_node_change_channel.txt" to read its frequency
to change its channel to it*/

fp=fopen("C:/Users/Ibrahim/op_models/allow_node_change_channel.txt",
"r");

j=0;
while(!feof(fp) && j<number_bss)
{
fscanf(fp,"%lf",&change_frequency_array[j]);
j++;
}
fclose(fp);
if(change_frequency_array[bss_id_receiver-1]!=
{
op_ima_obj_attr_set(txch_objid, "min frequency",
change_frequency_array[bss_id_receiver-1]);

op_ima_obj_attr_set(rxch_objid, "min frequency",
change_frequency_array[bss_id_receiver-1]);
}

```

```

op_radio_txch_rxgroup_compute
(OPC_OBJID_UNIVERSAL,rxgroup_recompute);

op_ima_obj_attr_get (txch_objid, "min frequency", &frequency_tx);
op_ima_obj_attr_get (rxch_objid, "min frequency", &frequency_rx);

/*skip analysis pkts if the node chnage its channel*/
skip_pkt_analysis=OPC_TRUE;

++count_change_channel_node;

/*Open file "number_nodes_change_channel_bss.txt" to read values
of number_nodes_change_channel_bss*/

fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_change_channel-
bss.txt","r");

j=0;
while(!feof(fp) && j<number_bss)
{
fscanf(fp,"%d",&number_nodes_change_channel_array[j]);
                j++;
}

fclose(fp);

--  

number_nodes_change_channel_array[bss_id_receiver-1];

/*Open file
"number_nodes_change_channel_bss.txt" to write values of
number_nodes_change_channel_bss*/
\
fp=fopen("C:/Users/Ibrahim/op_models/number_nodes_change_channel-
bss.txt","w");

for(i=0;i<number_bss;i++)
{
fprintf(fp,"%d\n",number_nodes_change_channel_array[i]);

}

fclose(fp);
}

}

/*********************************************
/* Get the type of the received WLAN frame and check */
/* whether it is marked as a bad packet in the pipeline */
/* stages. Consider the possibility that we may receive a*/
/* noise packet from a powerful jammer, which we need */
/* simply discard.
.
.
.
```

Appendix A-4: We declare some of state variables as shown below

Type	Name
int	bss_id_sender
int	bss_id_receiver
char	msg[128]
double	interference_ch[3]
double	count_first_time_rcv_pkt
double	sim_elapsed_time
double	TimeBeginTest
double	test_elapsed_time
int	duration_test
int	num_receivers
Objid*	receivers
int	count_test_ch[4]
double	frequency_change
double	origin_frequency
Boolean	cal_interference[3]
double	frequency_ch_array[3]
int	count_begintest
int	count_endtest
FILE*	fp
int	check_allow_option_node
double	TimeBeginTest_node
int	increment_node
Boolean	channel_overlap[3]
int	bss_id_nxgroup
int	count_read_AP_from_file
int	number_nodes_testing
double	interference_node_array[114][4]
double	accum_interferences[3]
Boolean	back_to_origin_ch
int	number_nodes_remaining_in_bss_array[9]
int	number_bss
int	bss_id_allow_test
int	number_nodes_accum

اختيار القناة الأفضل في شبكات الاتصال اللاسلكية المحلية (WLAN) المبنية على المعيار IEEE 802.11

إعداد: ابراهيم أحمد عبدالله الواوي.
اشراف: الدكتور مراد صبيح.

الملخص

تعتبر شبكة الاتصال اللاسلكية المحلية (WLAN) المبنية ضمن مواصفات المعيار (IEEE 802.11) من أشهر التقنيات المنتشرة والمهيمنة في الشركات، البيوت، الجامعات والمcafes. وللحصول على تغطية أوسع لشبكة الانترنت في هذه المناطق فإن العديد من الأجهزة المرجعية (Access Points) توضع وتنشر بكثافة. لكن هذا العدد الكثيف من الأجهزة المرجعية (Access Points) يُنتج ما يعرف بالتدخل (interference) ما بين هذه الأجهزة في حال تم ضبط القنوات (channels) التي تعمل عليها هذه الأجهزة بطريقة عشوائية، وهذا بدوره يؤدي إلى التقليل من الكفاءة والفعالية لهذه الأجهزة. وبما أن عدد القنوات المتعامدة (orthogonal channels) التي يدعمها المعيار (IEEE 802.11) محدود فإن هذا يؤدي إلى ضبط أكثر من جهاز مرجعي (Access Point) في نفس المنطقة على نفس القناة (channel) وبالتالي يؤدي إلى حدوث تداخل (interference) ما بين هذه الأجهزة (Access Points). وبظهور تقنيات جديدة في المعيار (802.11n) و المعيار (802.11ac) مثل ربط القنوات (channel bonding) فإن مشكلة التداخل (interference) تضاعفت. هنالك العديد من الأبحاث التي اجريت للحد والتخفيف من هذه المشكلة (التدخل) (interference) كان أبرزها التركيز على موضوع اختيار القناة الأنسب للتخفيف من التداخل ما بين الأجهزة المرجعية (Access Points).

في هذه الرسالة قمنا بوضع واقتراح مقياس ومعيار لقياس التداخل (interference)، واقترحنا خوارزمية تقوم على هذا المقياس في اختيارها القناة الأنسب. بحيث تقلل من التداخل ما بين الأجهزة المرجعية. اردفنا هذا العمل بمحاكاة (Simulation). أظهرت فيه النتائج تفوق طريقتنا على غيره من الطرق المقترحة.