STUDY ON ACCESS POINT AND BANDWIDTH MANAGEMENT IN OVERCROWDED WLANS BY MATCHING ALGORITHM AND GAME THEORY

This thesis in partial fulfillment for the award of the degree of
Master of Science in Telecommunication Engineering

Submitted By

1. Sadia Sultana	**ID: 2016-3-98-005**
2. Anirban Sarkar	**ID: 2016-3-98-006**

Under the Supervision of

Dr. Mohammad Arifuzzaman
Assistant Professor
East West University, Bangladesh

Department of
Electronics and Communications Engineering
Faculty of Sciences and Engineering

Semester: Summer-2017

DECLARATION

We hereby declare that we carried out the work reported in this thesis in the Department of Electronics and Communications Engineering, East West University, Dhaka, Bangladesh under the supervision of Dr. Mohammad Arifuzzaman and also declare that this thesis is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All sources of knowledge used have been duly acknowledged.

Students' names & Signatures:

Sadia Sultana
ID: 2016-3-98-005

Anirban Sarkar
ID: 2016-3-98-006

APPROVAL

The Thesis titled **"Study on Access Point and Bandwidth Management in Overcrowded WLANs by Matching Algorithm and Game Theory"** has been submitted to the following respected members of the Board of Examiners of the Faculty of Sciences and Engineering in partial fulfillment of the requirements for the degree of Master of Science in Telecommunication Engineering on 2nd August, 2017 the following students and has been accepted as satisfactory.

1. Sadia Sultana	**ID: 2016-3-98-005**
2. Anirban Sarkar	**ID: 2016-3-98-006**

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Supervisor
Dr. Mohammad Arifuzzaman
Assistant Professor
Department of Electronics and Communications Engineering
East West University, Bangladesh

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Dr. M. Mofazzal Hossain
Professor & Chairperson
Department of Electronics and Communications Engineering
East West University, Bangladesh

ACKNOWLEDGEMENT

The completion of our thesis work brings with it a sense of satisfaction, but it is never complete without thanking everyone who made it possible and whose constant support has crowned our efforts with success.

We would like to sincerely thank Dr. Mohammad Arifuzzaman, Assistant Professor, Dept. of Electronics and Communications Engineering, East West University for allowing us to carry out the thesis and providing guidelines whenever necessary.

Our deepest regards to Dr. M. Mofazzal Hossain, Professor & Chairperson of Electronics and Communications Engineering Department for encouraging and inspiring us to carry out the thesis work.

We would also like to thank all the faculty members and staff of ECE Department for providing us with the required facilities and support towards the completion of the thesis.

We feel extremely happy to acknowledge and express our sincere gratitude to our parents for their constant support and encouragement and last but not the least, friends and well-wishers for their help, cooperation and solutions to problems during the thesis works.

Sadia Sultana
Anirban Sarkar

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Wireless Local Area Networks (WLANs) have been widely developed during this decade, due to their mobility and flexibility. As WLANs have become popular in many sectors, so we have to manage it efficiently. In a typical deployment of WLANs, the coverage areas of nearby access-points(APs) usually overlap with one another and a host may detect signals from multiple APs which degrade the network performance. The main purpose of our work to minimize the number of active APs in a overcrowded WLAS and to develop a bandwidth sharing techniques to reduce the wastage of bandwidth. This study represents the management of APs in overcrowded WLANs. In this thesis, firstly, we analyze the relationship of number of access point (AP) and achieved throughput for a given scenario. In the scenario we consider a number of parameters, i.e. given area, topology, number of users, expected rate and channel allocation vector. We find that in general with the increase of AP the performance of overall network improve. However after a certain number of deployed APs, further increase of AP does not bring substantial improvement in the network performance. More specially, performance deteriorates after a threshold value of number of APs due to increasing interference between adjacent channel (or co-channel) interference. We verify the throughput increasing rate with proper management of APs by simulation results using WIMNET (Wireless Internet-access Mesh Network) simulator.

Most of the cases we see that in an overcrowded WLAN, the host does not get their expected bandwidth. By using an effective bandwidth sharing techniques, this problem can be solved. In this research work, we’ve also proposed about the bandwidth sharing techniques for a given scenario where the different operators can share their bandwidth with each other by applying matching algorithm and game theory model.

Chapter 1

INTRODUCTION

Wireless communication has been widely deployed in the last two decades, changing the way people access the internet. It is an application of science and technology that has come to be vital for modern existence. Accessing the global network has become the most essential and indispensable part of our lifestyle. The increased demands for mobility and flexibility in our daily life demands lead the development from wired LANs to wireless LANs (WLANs). WLANs are a simple and cheap way to connect laptops, tablet PCs, smart phones, digital cameras, TVs and set-top boxes etc. to the internet in almost all places of daily life. WLANs can be found in private homes, universities, offices, restaurants and more recently even on airplanes, in cars or on trains [1]. They can be used for web surfing, video streaming, telephony and many other applications. This versatility along with the cheap hardware and the use of license-free radio spectrum has led to an ever increasing demand for bandwidth. Wireless system designers have been facing the continuously increasing demand for high data rates and spectrum sharing required by new wireless applications.

In this thesis, we propose an efficient game theoretic approach for host association which ensures user satisfaction.

1.1.Motivation

The term wireless communication was introduced in the 19th century with the invention of radio. Since then, the power of instant communication over long distance has been transformed society and made the world a smaller place. In this technology, the information can be transmitted through the air without requiring any cable or wires or other electronic conductors, by using electromagnetic waves like IR, RF, satellite, etc. Wireless technology provides us much profit like portability and flexibility, increased productivity, and lower installation costs. For this the wireless connection has become popular day by day over wired connection [2]. Demand for wireless access is creating hotspots in places like university campuses, offices, coffee shops, and airports etc. Major cities are planning to set up wireless local area networks (WLANs) for public use free of cost. Wireless communication is an ever-developing field, and the future holds many possibilities in this area. Researchers and developers are trying to make this technology more users friendly. We choose this field because there are more scopes to work in this field.

Wireless technology has become an essential part of life and has impacted the world in many important ways. Through the Best Wi-Fi Statistics of 2014 75% of people say a week without Wi-Fi would leave them grumpier than a week without coffee. 90% of all smart phones are equipped with Wi-Fi capabilities. Nowadays people use Wi-Fi rather than cellular data. This upward technology will sustain long time yet.

1.2.Related Works

Though from the application perspective the no. of APs, bandwidth utilization and interface have immense impact on the users, the relevant researches are still not in mature state. Nevertheless several relevant works have been reported in recent years.

In [3], authors studied the channel interference problem in WLANs where APs are located densely and they also proposed an AP cooperation system which detects the hidden terminal and the exposed terminal problems between stations by integrating the information obtained at different APs. But the paper doesn't concentrate on no. of AP management. In [4], authors introduce some techniques based on graph coloring algorithms for frequency allocation process in WLAN. Authors also suggest a preliminary message format for access points to exchange information. In [5], authors analyze the performance of three channel assignment strategies where each AP independently selects one of the three available channels respectively for random channel assignment. For local selection, APs are deployed sequentially and each AP selects the least congested channel. However authors use heuristic rather than any direct algorithm for channel assignment.

In [6], authors have proposed an analytical model to compute the saturation throughput performance in the presence of a finite number of terminals and in the assumption of ideal channel conditions. But the authors have not considered the concept of using different channels. In [7], authors have proposed a decentralized AP selection algorithm to achieve a minimum throughput with the reasonable computation and transmission overhead to be compatible with legacy APs without any modification. In [8], authors have proposed an AP selection algorithm to maximize the throughput while preserving newly arrived-user throughputs in a multi-rate WLAN. It supports limited user movements. Their goal is to maximize the throughput by optimizing the associated APs for hosts while the locations of the APs are fixed. On the other hand, we focus on reducing the number of active APs while maximizing the performance by optimizing the associated APs for hosts.

In [9], theoretical throughput upper limit and delay lower limit for IEEE 802.11 protocols are shown. Authors have found that increasing data rate is very challenging in case of ensuring consistent performance where other overhead have vital role. In [10], channel assignment for IEEE 802.11 and searching optimum access point for a host are considered jointly. Authors have shown that for a typical indoor environment patching algorithm can provide nearly optimum performance. The complexity of the optimum is also low. In [11], authors have developed and tested an open enterprise Wi-Fi solution based on virtual APs. A central controller is used to manage the handover scheme of the host. Authors also address QoS (Quality of Service) of real time services.

In [12], [13], a scheme for inter AP coordination in load balancing, frequency planning, power control have proposed. Authors also introduce the concept of adaptation of certain abstractions and concepts from Software Define Networks (SDNs) for its use in wireless networks. In [14], a distributed solution using Light Virtual Access Point (LVAP) was introduced for the direct exchange of information between APs. In [15], an analytical model based on a Markov chain to study the efficiency of the Linux network subsystem has been presented. Authors identify the softIRQ process as the main element in the Linux capturing stage and they have built a model that represents the different steps in the softIRQ and the computational cost.

In [16], authors have proposed an efficient and distributed algorithm named Client-Assisted Channel Assignment Optimization. A distributed channel assignment algorithm for uncoordinated WLANs to minimize interference with adjacent APs is considered. The algorithm leads to better channel assignment decisions at the APs. However, authors have not considered the mapping of the number of APs & throughput in a crowded network. In [17], authors studied the energy efficiency issue in dense WLANs and proposed an aggressive scheme for adaptation of the AP density to actual traffics. However authors have not addressed the impact of channel assignment in the overall throughput. Operators want to share their WLAN infrastructure in order to reduce cost and administration task.

In [18], author studied the history and current status of IEEE 802.11n and many PHY enhancements for higher data rates and they identified overhead as the fundamental problem of MAC inefficiency. Increasing transmission rate alone cannot help a lot. The overhead is very large when either the data rate is high or the frame size is small. There-fore, new efficient MAC strategies are especially needed. We propose several MAC enhancements to reduce overhead via frame aggregation from various aspects such as distributed vs. centrally controlled, ad hoc vs. infrastructure, uplink vs. downlink, single-destination vs. multi-destination, PHY-level vs. MAC-level, single-rate vs. multirate, immediate ACK vs. delayed ACK, and no spacing vs. SIFS spacing. A new throughput upper limit for the frame aggregation scheme is formulated.

In [19], authors have discussed the popularity of wireless local area networks (WLANs), which has resulted in their dense deployment in many cities around the world. The increased interference among different WLANs severely degrades the throughput achievable. This problem has been further exacerbated by the limited number of frequency channels available. An improved distributed and dynamic channel assignment scheme that is simple to implement and does not depend on the knowledge of the throughput function is proposed in this work. It also allows each access point (AP) to asynchronously switch to the new best channel. Simulation results show that our proposed scheme converges much faster than similar previously reported work, with a reduction in convergence time and channel switches as much as 77.3% and 52.3% respectively. When it is employed in dynamic environments, the throughput improves by up to 12.7%.

In [20], authors focused on bandwidth management in shared WLANs. They proposed an algorithm to allocate cell bandwidth to different operators in such a way that the traffic from one logical network can not affect throughput of the users from other ones. This algorithm can be implemented in large WLANs that use Wi-Fi. They evaluated the performance of the proposed method by simulating the transmission of FTP and HTTP traffics. In [21], authors have presented a novel AP association mechanism in multi-rate IEEE 802.11 WLANs. They have formulated the problem as a coalition matching game with complementarities and peer effects and we have provided a new practical control mechanism that provides nodes the incentive to form coalitions both solving the unemployment problem and reducing the impact of the anomaly in IEEE 802.11. In [22], authors have studied the resource allocation problem in EH-based D2D communications with downlink spectrum resource reusing. By employing SWIPT, UEs can harvest energy from the received signal power, and thus interference and noise can be introduced as beneficial outcomes. They've formulated a joint power control and partner selection problem to optimize EE performance of D2D pairs and the energy harvested by CUEs simultaneously with the consideration of UEs' preferences and satisfactions, which was then transformed to a two-dimensional matching between D2D pairs and CUEs (RBs). Finally, the performance of the proposed matching algorithm was compared with three heuristic algorithms.

In [23], authors have studied stable energy-efficient partner selection in cooperative wireless networks. Specifically, two distributed polynomial-time partner selection algorithms, namely ISM and MSM, have been designed based on the stable roommates matching problem. The proposed algorithms are compared with other matching algorithms, suggesting a tradeoff between stability and total energy consumption, and yielding comparable SER performance and network sum-rate. The authors investigated the case of an open spectrum shared area using game theory [24]. In particular, they proposed a new way of maximization of the network throughput and the provision of fairness. They calculated the Nash equilibrium and the Nash bargaining solution of a non-cooperative and of a cooperative power control game, respectively.

In [25], Authors presented a game theoretic approach for bandwidth allocation between infrastructure providers (InPs) and service providers (SPs). Here they introduced non cooperative game theory to analyze the interaction between InPs and SPs. At first they consider a problem which is lack of efficient bandwidth allocation between two entities. Resource allocation is an important issue in network virtualization due to the limited physical resources. To distribute physical resources to different SPs in a fair way is a critical issue for InPs. An efficient resource allocation is required to avoid congestion. The proposed approach is based on two-stage non cooperative games. The first stage of game is the bandwidth negotiation game where a SP requests bandwidth from multiple InPs. The second stage of the non-cooperative game is the bandwidth provisioning game, where different SPs compete for an amount of bandwidth in a shared physical link for a given InP. They showed that the Nash equilibrium state is reached in each stage of the game where neither the SP nor the InP has an incentive to change its strategy. Simulation results show that the proposed approach achieves higher bandwidth utilization in network virtualization, which improves network performance, and fairly distributes bandwidth between multiple service providers.

In [26], Authors proposed a game theoretic approach to encourage efficient behavior in solving the interaction between InPs and SPs. In the network virtualization environment a bandwidth allocation scheme is based on the non-cooperative game model and the concept of Nash Equilibrium. The proposed approach did not consider the situation when multiple SPs get physical resources from different InPs which results the competition among different InPs.

In [27], Authors investigated Nash strategies for the non-zero sum differential game with a focus on the pursuit-evasion problem. These Nash strategies are derived for single-pursuer single evader and n-pursuers single-evader game. The ideas of differential game theory and cooperative control are integrated to cope with the absence of information under the distributed information. Authors also present simulation results for those cases.

In [28], Authors investigated resource distribution problem such as a water distribution system (WDS). The flow of water should be controlled to achieve consumers demand. To keep WDS under control game theoretic approach is utilized. In [29], Authors focused on cooperative game theory based approach for job scheduling in cloud environment. Cloud computing provides computing resources to the consumers in the form of infrastructure, platform and software. When multiple users request for services, the cloud service provider has to schedule the request to satisfy users request and to meet Service Level Agreements (SLAs). For this authors have proposed a new job scheduling technique by using the concept of game theory and genetic algorithm to maximize resource utilization. Here the problem was formulated mathematically using the concepts of game theory and solution was proposed using Pareto optimality concept which uses Non-dominated Sorting Genetic Algorithm II (NSGA II). Experimental results show that this approach gives better when compared to non cooperative scheduling.

In [30], Authors proposed a game theoretic model of cooperation between over-the-top (OTT) service providers and mobile network operators (MNO). For this model, authors derived an analytical expression for the Nash bargaining solution (NBS) of the corresponding game using the Nash bargaining framework. The model results show that efficient cooperation between an OTT and an MNO leads to MNO deploying in the radio access infrastructure. Moreover, the cost of MNO’s infrastructure, if excessively high, then the cooperation between OTT and MNO is not sustainable.

In [31], authors investigated the problem of quality of experience (QoE) based multichannel allocation in 5G heterogeneous cellular networks (HCNs) with cross-tiered interference constraint. In this paper, small-cell users (SUs) QoE demands and the individual QoE losses of macro-cell users (MUs) are considered. Authors proposed a joint matching-coalitional game theoretical scheme to solve QoE-based complicated multichannel allocation problem. They divide this problem into two sub problem; one is intra-cell channel allocation for SUs and another is inter-cell channel allocation for SBSs. These problems can be solved by using two proposed games. They also proposed a joint channel allocation algorithm for the matching-coalitional game.

In [32], authors focused on making distributed resource allocation and orchestration a viable approach. Game theory is used to model interaction between users nad servers. Authors also presented two two-stage Stackelberg game where servers act as leaders of the game and users as followers. The proposed framework proves the existence and uniqueness of equilibrium. Numerical results show the effectiveness of the approach.

In [33], the authors considered the energy-efficient resource management issue in D2D cooperative relay communications. A pricing-based matching approach to jointly optimize relay selection, spectrum allocation and power control is also proposed by them. They formulated the joint optimization problem as an NP-hard four-dimensional matching problem and proposed a low-complexity two-stage optimization approach to provide a tractable solution one after another. Further again they proposed a pricing-based iterative matching algorithm to maximize the EE while guaranteeing the QoS requirements of both D2D TRs and CUEs for the matching problem in each stage and finally, the proposed algorithm was compared with some heuristic algorithms through simulations.

To match D2D pairs with cellular UEs the authors formulated the joint resource allocation problem as a one-to-one matching problem under two-sided preferences and employed the Gale–Shapley (GS) algorithm in [34].

In [35], authors provided the first comprehensive tutorial on using matching theory for developing innovative resource management mechanisms in wireless networks. At first, they provided the fundamental concepts of matching theory and discussed a variety of properties that allow the definition of several classes of matching scenarios. Finally, proposed three new engineering-oriented classes of matching theory, which can be adopted in wireless networking environments.

1.3.Organization of Thesis

This thesis is organized as follows. Chapter 2 introduces wireless network in details. Here we discussed about wireless local area network and its types, applications of Wi-Fi and IEEE802.11 protocols. Chapter 3 provides a brief description on our proposed work. The part I of chapter 3 describes access-point management in overcrowded WLANs. Network model and access-point management algorithm is presented in this chapter. In part II, we described host association techniques based on matching algorithm and game theoretic model. Chapter 4 presents all simulation results. Finally, chapter 5 presents the future implementation of this system and conclusions. Last but not the least, Appendix A contains all simulation codes.

Chapter 2

INTRODUCTION TO WIRELESS NETWORK

A wireless network is a flexible data communications system, which uses wireless media such as radio frequency technology to transmit and receive data over the air, minimizing the need for wired connections. The wireless communication revolution is bringing fundamental changes to data networking, telecommunication, and is making integrated networks a reality. Examples of wireless networks include cell phone networks, wireless local area networks (WLANs), wireless sensor networks, satellite communication networks, and terrestrial microwave networks.

2.1. Wireless Local Area Network (WLAN)

A Wireless Local Area Network (WLAN) links two or more devices using a wireless communication method. It usually provides a connection through an Access Point (AP) to the wider internet. This gives users the ability to move around within a local coverage area and still be connected to the network. Just as the cordless telephone frees people to make a phone call from anywhere in their home, a WLAN permits people to use their computers, laptops anywhere in the network area, such as an office building or corporate campus, universities restaurants and more recently even on airplanes, in cars or on trains. The increased demands for mobility and flexibility in our daily life are demands that lead the development from wired LANs to wireless LANs (WLANs).

Wireless Local Area Networks (WLANs) are ubiquitous today. WLANs are a simple and cheap way to connect laptops, tablet PCs, smart phones, digital cameras, TVs and set-top boxes etc. to the Internet in almost all places of daily life. They can be used for web surfing, video streaming, telephony and many other applications. This versatility along with the cheap hardware and the use of license-free radio spectrum has led to an ever increasing demand for bandwidth. Industry and academia have managed to satisfy this demand and tremendously increased the speed of WLANs over the last 15 years.

2.1.1. History of WLAN

Norman Abramson, a professor at the University of Hawaii, developed the world's first wireless computer communication network, ALOHAnet (operational in 1971), using low-cost ham-like radios. The system included seven computers deployed over four islands to communicate with the central computer on the Oahu Island without using phone lines. Wireless LAN hardware initially cost so much that it was only used as an alternative to cabled LAN in places where cabling was difficult or impossible. Early development included industry-specific solutions and proprietary protocols, but at the end of the 1990s these were replaced by the various versions of IEEE 802.11standards (in products using the Wi-Fi brand name).

Beginning in 1991, a European alternative known as HiperLAN/1 was pursued by the European Telecommunications Standards Institute (ETSI) with a first version approved in 1996. This was followed by a HiperLAN/2 functional specification with ATM influences accomplished February 2000. Neither European standard achieved the commercial success of 802.11, although much of the work on HiperLAN/2 has survived in the physical specification (PHY) for IEEE 802.11a, which is nearly identical to the PHY of HiperLAN/2. In 2009 802.11n was added to 802.11. It operates in both the 2.4 GHz and 5 GHz bands at a maximum data transfer rate of 600 Mbps. Most new routers are able to utilize both wireless bands, known as dual-band. This allows data communications to avoid the crowded 2.4 GHz band, which is also shared with Bluetooth devices and microwave ovens. The 5 GHz band is also wider than the 2.4 GHz band, with more channels, which permits a greater number of devices to share the space. Not all channels are available in all regions.

2.1.2. Types of WLANs

The basic building block of a WLAN network is the Basic Service Set (BSS), which is simply a group of stations that communicate with each other at PHY layer. Communication takes place within an area called the ‘basic service area’ which is defined by the propagation characteristics at a given rate in the medium. When a station is in the basic service area, it can communicate with the other members of the BSS. Generally, there are two types of BSSs: independent networks and infrastructure networks.

- **Independent BSS (IBSS):** Fig. 1 gives a representation of an Independent BSS (IBSS), which is also called an ad-hoc network. An independent BSS (IBSS) is an ad-hoc network that contains no access points, which means they cannot connect to any other basic service set. Stations in an IBSS can communicate directly with each other. As shown in Fig. 1, stations A, B and C can transmit packets directly to each other without requiring relaying. The smallest possible IEEE 802.11 network is an IBSS with two stations. Typically, IBSSs are composed of a small number of stations set up for a specific purpose and for a short period of time. One common use is to create a short-lived network to support a single meeting in a conference room. As the meeting begins, the participants create an IBSS to share data. When the meeting ends, the IBSS is dissolved.

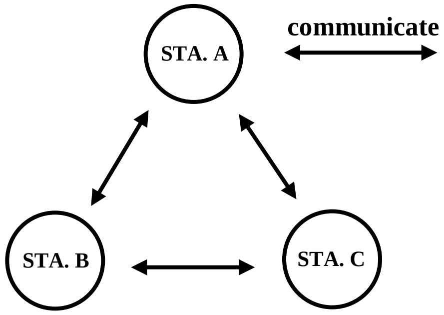


Figure 2.1: An example of Independent Basic Service

- **Infrastructure BSS:** Most Wi-Fi networks are deployed in infrastructure mode. In infrastructure mode, a base station acts as a wireless access point hub, and nodes communicate through the hub. Infrastructure networks are distinguished by the use of an AP. Infrastructure BSS is shown in Fig.2. From the Fig. 2 we can see that if station A needs to communicate with station B, the communication must take two hops: first, the station A transfers the packet to the AP; second, the AP relays the packet to station B. With all communications relayed through an AP, the basic service area corresponding to an infrastructure BSS is defined by the points in which transmissions from the AP can be received.

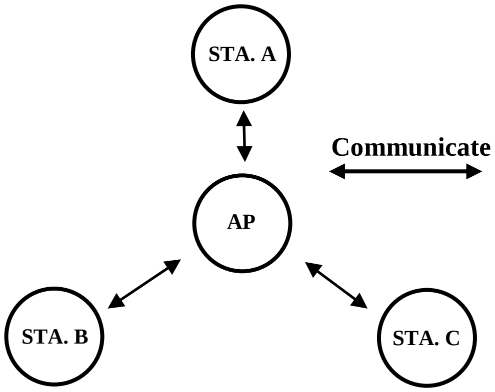


Figure 2.2: An example of Infrastructure Basic Service

2.1.3. Advantages of WLANs

The most obvious advantages of WLANs are mobility and flexibility. A WLAN gives the mobility to the users, so they can move around while still sharing the network resources as long as they are in the service area.

WLANs typically have a great deal of flexibility. A WLAN uses a number of APs to connect users to an existing network. The infrastructure side of a wireless network, however, is qualitatively the same whether it is connecting one user or a million users. To offer service in a given area, it is necessary to have APs in place. Once that infrastructure is built, however, adding a user to a wireless network is mostly a matter of authorization, which just takes a few seconds or less. Due to their mobility and flexibility, WLANs become popular over ‘wired’ networks.

2.2. Wi-Fi Technology

Wireless Fidelity (Wi-Fi) is one of the upcoming techniques in the internet world. Wi-Fi is a technology for wireless local area networking with devices based on the IEEE 802.11 standards. This Wi-Fi can be an alternate to Wired Technology. It creates a hidden path between the internet and the wired network. Wi-Fi network functioning can be done on the physical and the data link layer. Radio Frequency (RF) is used for transmitting data through air. Wi-Fi allows wireless connection for up to 20 meters, it is also known as Internet Routers.

Some of the Wi-Fi network topologies are given below:

- Access Point AP-based topology
- Star-based network topology
- Peer-to-peer topology
- Point-to-multipoint bridge topology

2.2.1. Applications of Wi-Fi Technology

- **In Educational Institutions**
    - There is Wi-Fi network connectivity in many colleges and universities that is easily accessible by all students.
    - The Wi-Fi helps students providing internet services for free and helps students in studies by letting them search and study on internet.

- Students also get the chance to do online courses and the global English programs in the university itself by the use of internet available through the Wi-Fi.
- Teachers can delivery lecture through internet by using Wi-Fi.

❖ **In Various Places**

- Wi-Fi provides services in private homes, businesses, as well as in public space at Wi-Fi hotspots set up either free of charge or commercially.
- Organizations and businesses, such as airports, hotels and restaurants often provide free use hotspot to attract customers.
- Authorities who wish to provide services or even to promote business in selected areas sometimes provide free Wi-Fi access.

❖ **Wi-Fi Gaming Console**

- Game consoles: PS3, XBOX 360, have available in Wi-Fi.
- Can connect to the internet wirelessly.
- Can perform as a media center.

❖ **Wireless Devices**

- A Wi-Fi phone is a wireless device that gives the dual benefits of connectivity and the cost savings of VoIP.
- Wi-Fi phone looks like the next big thing in the telecom revolution.
- Wi-Fi camera can connect the internet wirelessly and it offers cloud storage

2.2.2. Advantages and Disadvantages of Wi-Fi

❖ **Advantages**

- Mobility

- Ease of installation
- Flexibility
- Security
- Use unlicensed part of the radio spectrum
- Roaming
- Speed

❖ **Disadvantages**

- Slow speed in overcrowded area
- Less security
- Limit range
- High cost
- High power consumption

2.3.IEEE 802.11

Since its introduction in 1997, IEEE 802.11 has become the dominant WLAN standard. IEEE 802.11 is a member of the IEEE 802 family, which is a series of specifications for Local Area Network (LAN) technologies. IEEE 802 specifications are focused on the two lowest layers of the OSI 7-layer model1, because they incorporate both physical and data link components. IEEE 802.11 is a set of standards, which specifies a 2.4 and 5 GHz frequency bands with data rates 1 and 2 Mbps. With this standard, one could choose to use either frequency hopping or direct sequence.

In late 1999, the IEEE published two supplements to the initial 802.11 standard: 802.11a and 802.11b.

❖ **IEEE802.11a**

The IEEE 802.11a standard uses the same data link layer protocol and same format as the original IEEE 802.11 protocol, but it supports an Orthogonal Frequency-Division Multiplexing (OFDM) based air interface at the PHY layer. It operates in the 5 GHz band with data rates up to 54 Mbps. The advantages of this standard as compared to 802.11b include having much higher capacity and less RF interference with other types of devices like Bluetooth. However, this high carrier frequency also brings some disadvantages. Due to their smaller wavelength, the IEEE802.11a signals will be absorbed more heavily by walls and other solid objects in their path. As a result, the IEEE 802.11a signals cannot penetrate as far as those of the IEEE 802.11 b/g that operate in the frequency band of 2.4 GHz.

- **IEEE802.11b**

As with the initial standard, 802.11b operates in the 2.4 GHz band, but it includes 5.5 and 11 Mbps in addition to the initial 1 and 2 Mbps. The 802.11b standard only specifies direct sequence modulation. Devices operating in the 2.4 GHz range include microwave ovens, Bluetooth devices, baby monitors, cordless telephones, and some amateur radio equipment. In 802.11b, the signal range is good and not easily obstructed. Its disadvantage is home appliances may interfere on the unregulated frequency band.

- **IEEE802.11g**

In 2002 and 2003, WLAN products supporting a newer standard called *802.11g* emerged on the market. 802.11g attempts to combine the best of both 802.11a and 802.11b. 802.11g supports bandwidth up to 54 Mbps, and it uses the 2.4 GHz frequency for greater range. It is based on OFDM (orthogonal frequency division multiplexing). 802.11g access points will work with 802.11b wireless network adapters and vice versa. It provides fast maximum speed and its signal range is good and not easily obstructed. Its disadvantages are appliances may interfere on the unregulated signal frequency and its cost is more than 802.11b.

- **IEEE802.11n**

802.11n is an amendment that improves upon the previous 802.11 standards by adding multiple-input multiple-output antennas (MIMO). 802.11n operates on both the 2.4 GHz and the 5 GHz bands (optional). It operates at a maximum net data rate from 54 Mbps to 600 Mbps. Enterprises were already migrating to 802.11n networks based on the Wi-Fi Alliance's. It provides fastest maximum speed and best signal range. It is more resistant to signal interference from outside sources. Its cost is more than 802.11g and the use of multiple signals may greatly interfere with nearby 802.11b/g based networks.

- **IEEE802.11ac**

  IEEE 802.11ac is an amendment to IEEE 802.11, published in December 2013 that builds on 802.11n. 802.11ac utilizes dual-band wireless technology, supporting simultaneous connections on both the 2.4 GHz and 5 GHz Wi-Fi bands. 802.11ac offers backward compatibility to 802.11b/g/n and bandwidth rated up to 1300 Mbps on the 5 GHz band plus up to 450 Mbps on 2.4 GHz. As compared to 802.11n, it includes wider channels in the 5 GHz band, higher-order modulation.

- **IEEE802.11ah**

  IEEE 802.11ah is a wireless networking protocol published in 2017 to be called Wi-Fi HaLow as an amendment of the IEEE 802.11-2007 wireless networking standard. It uses 900 MHz license exempt bands to provide extended range Wi-Fi networks, compared to conventional Wi-Fi networks operating in the 2.4 GHz and 5 GHz bands. It also benefits from lower energy consumption and allowing the creation of large groups of stations or sensors that cooperate to share signals. It supports the concept of the Internet of Things (IoT). It can cover a one-kilometer radius. The protocol is intended to be competitive with Bluetooth 5 with its low power consumption, but with a wider coverage range. It aims at providing connectivity to thousands of devices under an access point. The protocol supports machine to machine (M2M) markets, like smart metering.

Chapter 3

PROPOSED WORK

This chapter provides the description on our proposed work in details. The part I describes about access-point management in overcrowded WLANs. The systems network model and access-point management algorithm is presented in this chapter. In part II, we described host association techniques based on matching algorithm and game theoretic model.

3.1.Access-Point Management in Overcrowded WLANs

In an overcrowded WLAN, the coverage areas of nearby access-points(APs) usually overlap with one another and a host may detect signals from multiple APs which degrade the network performance[36,37]. The main purpose of our work to minimize the number of active APs in overcrowded WLANs. This study represents the management of APs in overcrowded WLANs. We analyze the relationship of number of access point (AP) and achieved throughput for a given scenario. In the scenario we consider a number of parameters, i.e. given area, topology, number of users, expected rate and channel allocation vector. We find that in general with the increase of AP the performance of overall network improve. However after a certain number of deployed APs, further increase of AP does not bring substantial improvement in the network performance. More specially, performance deteriorates after a threshold value of number of APs. The main reason for this performance deterioration is the increase of interference between adjacent channel (or co-channel) interference. We verify the throughput increasing rate with proper management of APs by simulation results using WIMNET (Wireless Internet-access Mesh Network) simulator.

3.1.1. Minimum Average Throughput and Average Throughput Threshold

When the number of active physical APs (=C_1) is minimized, the number of hosts associated to one AP increases, which reduces the communication performance per host. We use the estimated average throughput per host that can be calculated from cost function defined in our previous work [26]. Then, we introduce the average throughput constraint that the estimated *minimum average* *throughput* C_2 per host must satisfy the given minimum throughput called the *average throughput threshold* G. The number of active APs may be increased when this *constraint* is not satisfied.

Let,

$AP_i,\ i = 1,2,3, \dots \dots \dots n\ (No. of\ AP)$

$Host_j,\ j = 1,2,3, \dots \dots \dots m\ (No. of\ Host)$

$sp_{ij},\ link\ speed$

$d_{ij},\ delay\ time$

Assume, AP_i is associated with a $Host_j$ with the link speed sp_{ij} and it always transmit one bit. In this case the expected delay time is $d_{ij} = {}^{1}/_{sp_{ij}}$.If multiple hosts are associated with AP_i, then the total delay time becomes $\sum_{i,j=1}^{n,m} d_{ij}$. Because each host can transmit one bit during this time if every host transmits the same amount of data and their transmission chances are fair. So the minimum average throughput per host with AP_i can be estimated by:

$$minT_{avg} = \frac{1}{\sum_{i,j=1}^{n,m} d_{ij}} \quad \dots \dots \dots \dots \dots \dots \dots . . (1)$$

Since, total delay time is equal to the maximum transmission time.

Hence,

$$\sum_{i,j=1}^{n,m} d_{ij} = Max(T_i) \quad \dots \dots\ \dots \dots \dots \dots \dots \dots (2)$$

From (1) & (2),

$$minT_{avg} = \frac{1}{Max\,(T_i)} = C_2$$

Where, T_i is the transmission time of AP_i among the active APs.

Now, the average throughput of bottleneck AP should not be smaller than minimum average throughput threshold.

$$C_2 \geq G \quad \dots . . \dots \dots \dots \dots \dots \dots . (3)$$

3.1.2. Access-Point Management Algorithm

In this subsection, we present the high-level view of the hosts to AP association algorithm. In our proposed algorithm, initially hosts are assigned to APs based on link speed constraints [3]. With the changes of the number of active APs, the procedure is executed several times until the algorithms converge. It can be noted that since we use the centralized local search, the algorithm converge in linear time. The process is briefly explained in Table I.

TABLE I
AP MANAGEMENT ALGORITHM

1st Step: Generates a list of associable hosts for each AP considering the link speed constraints.

2nd Step: Generates a list of active physical APs and their associated hosts.

3rd Step: Optimizes host association of AP to minimizing C_2 by local search method.

4th Step: Optimizes AP selection through minimizing both C_1 and C_2 by repeating local search method until G is satisfied.

5th Step: Swaps the association of hosts to improve and finalize host association.

3.1.3. Network Model

Here, we consider instance modeling a large-size cafeteria as a common WLAN environment [3]. Figure 3.1 represents the topology in this cafeteria instance. Here in the 35m×35m field, 3×3(=9) APs and 8×8(=64) hosts are located with the regular interval and use a personal computer or a smart phone for the internet access using WLANs.

An ideal case of a cafeteria is assumed here where each AP has 110m radius coverage area and link speed threshold is 40 Mbps and distance between each host is 5m.

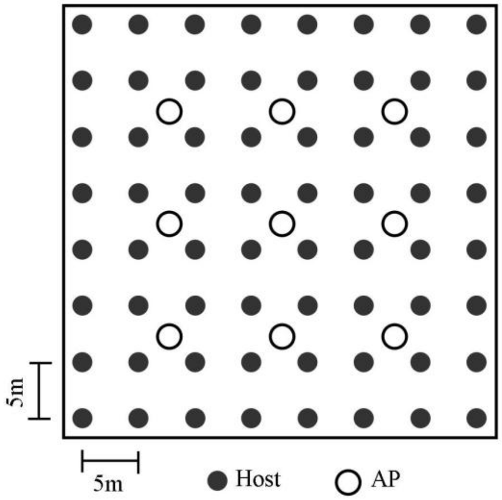


Figure 3.1: Cafeteria instance

3.1.4. Simulation Setup

The WIMNET simulator consists of multiple access-points (APs) as wireless routers that are connected through wireless links. It simulates least functions for wireless communications of hosts and APs that are required to calculate throughputs and delays. It has been developed to evaluate a large-scale WIMNET.

IEEE 802.11n protocol is used to enhance the performance and network speed of WLANs [21]. In this protocol, the number of non-interfered channels is two for 2.4 MHz bands and nine for 5 GHz bands. It introduces the MIMO (Multiple-Input-Multiple-Output) to provide the data rate up to 600 Mbps.

In our simulation we follow IEEE 802.11n standards as wireless interface. Accordingly we use the corresponding supportive version of simulator. To make sure the best environment for simulation, Intel core 2 Duo 3.33 GHz configured CPU is used which primary memory is 3 GB as well as Ubuntu 12.04 (Linux kernel 3.2.0) OS. Important simulation parameters are listed in Table II.

TABLE II
SIMULATION ENVIRONMENT

Simulator	WIMNET Simulator
Interface	IEEE 802.11n
No. of Access Point	9
No. of host	64
Average throughput	5MB – 10MB

3.2.Bandwidth Management by Matching Algorithm and Game Theory

We've proposed about the bandwidth sharing techniques for a given scenario where the different operators can share their bandwidth with each other. Matching algorithm is used for host association in WLAN and game theory considered as economic model.

3.2.1. Game Theory Application in Wireless Network

In recent years game theory has gotten more attention from the academic community as scientists have found areas where game theory can be used. The ability to model independent decision makers whose actions potentially affect other decision makers makes game theory attractive to analyze the performances of wireless communication systems. Recently, there has been growing interest in adopting game theoretic methods to wireless networks for power control, pricing, resource allocation, flow control and error control, load balancing and channel access schemes.

3.2.2. Game Theory Basic

Game theory is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. The analysis of a situation involving conflicting interests in terms of gains and losses among opposing players is the another definition of game theory . Its aim is to mathematically capture behavior in strategic situations, in which an individual's success in making choices depends on the choices of others. Historically, game theory developments were motivated by studies in economics, but in different fields as diverse as biology, computer science, social science and engineering has emerged by many interesting game theory applications. Power allocation in wireless communications, most notably congestion control in network traffic, problems of optimal routing, and optical networks are the basic example of game theory. In recent years there have even been some advances in to using game theory to analyze human decision making in a field called Neuro-economics. In communication technology Game theory is particularly appropriate for modeling and analyzing interactions involving communication, because the exchange of information between the two animals can be manipulated in the model and provides a very useful framework for analyzing communication. Game theory is also a tool for understanding expertise and increasing skill.

Let's start out by defining a few terms commonly used in the study of game theory:

- *Game:* Any set of circumstances that has a result dependent on the actions of two of more decision makers ("players")
- *Players:* A strategic decision maker within the context of the game
- *Strategy:* A complete plan of action a player will take given the set of circumstances that might arise within the game
- *Payoff:* The payout a player receives from arriving at a particular outcome. The payout can be in any quantifiable form, from dollars to utility.
- *Information Set:* The information available at a given point in the game. The term information set is most usually applied when the game has a sequential component.

- *Equilibrium:* The point in a game where both players have made their decisions and an outcome is reached.

The different types of games are formed on the basis of number of players involved in a game, symmetry of the game, and cooperation among players.

1. Cooperative and Non-Cooperative Games

Cooperative Games*:* In game theory, a cooperative game is a game with competition between groups of players due to the possibility of external enforcement of cooperative behavior. In cooperative games, players are convinced to adopt a particular strategy through negotiations and agreements between players. Cooperative game theory only describes the structure, strategies and payoffs of coalitions and provides analytical tools to study the behavior of rational players when they cooperate.

Non-Cooperative Games: On the other hand a non-cooperative game is a game with competition between individual players. This theory focuses on strategic choices resulting from interaction among competing players; each player chooses its own strategy independently for improving its own utility. It focuses on competitive scenarios and provides a low-level approach as it models all the procedural details of the game.

2. Normal Form and Extensive Form Games

Normal Form: In normal form games the description of game is represented by the form of matrix. When the payoff and strategies of a game are represented in a tabular form, it is also termed as normal form games. To identifying the dominated strategies and Nash equilibrium, the normal form game is use. It has a set of players and each player has a set of strategies. Each of them selects a strategy and plays their selections simultaneously. In this manner, no player is responding to another’s selection. Where, each player knows every player’s strategy set and utility function.

1.1.1.1 ***Extensive Form Games****:* In extensive form games the description of game is represented by the form of a decision tree. This type of game contains all the information about a game by defining who moves when, what each player knows when he moves, what moves are available to him, and where each move leads to etc. Whereas the normal form is more of a summary representation and Extensive form games help in the representation of events that can occur by chance and it consist of a tree-like structure in which the names of players are represented on different nodes.

3. Simultaneous Move Games and Sequential Move Games

Simultaneous Move Games: Simultaneous games are the one in which the move of two players is simultaneous without knowing the strategies that have been chosen by other players. In simultaneous move, players do not have knowledge about the move of other players. Simultaneous games are represented in normal form and denoted by payoff matrix.

Sequential Move Games: Sequential games are the one in which players are aware about the moves of players who have already adopted a strategy. This need not be perfect information about every action of earlier players. But the players do not have a deep knowledge about the strategies of other players. Sequential games are represented in extensive form and normally denoted by decision tree.

4. Constant Sum, Zero Sum, and Non-Zero Sum Games

Constant Sum Games: Constant sum game is the one in which the sum of outcome of all the players remains constant even if the outcomes are different. Here the sum of the payoffs of all players always adds up to the same constant figure for any particular outcomes. Zero sum game is a type of constant sum game.

Zero Sum Games: A game is called zero-sum if the sum of payoffs equals zero for any outcome. Zero-sum games are a special case of constant-sum games, in which choices by players can neither increase nor decrease the available resources. In zero sum game, the gain of one player is always equal to the loss of the other player. Examples of zero sum games are chess and gambling. In these games, the gain of one player results in the loss of the other player. The winnings of the winning players are paid by the losses of the losing players.

Non-Zero Sum Games: A non-zero sum game is the games in which sum of the outcomes of all the players is not zero. In non-zero sum games one decision maker’s gain or loss does not necessarily result in the other decision maker’s loss or gain. It can be transformed to zero sum game by adding one dummy player. The losses of dummy player are overridden by the net earnings of players. However, cooperative games are the example of non-zero games. This is because in cooperative games, either every player wins or loses.

5. Symmetric and Asymmetric Games

Symmetric Games: In symmetric games, strategies adopted by all players are same and it can exist in short-term games because in long-term games the number of options with a player increases. The decisions in a symmetric game depend on the strategies used, not on the players of the game. If the two players interchange their moves, the payoffs are also interchanged. The chicken, the prisoner's dilemma and the stag hunt all are the symmetric game example.

Asymmetric Games: Which strategies adopted by players are different is known as asymmetric game. However, decision making in asymmetric games depends on the different types of strategies and decision of players. For an example this game is entry of new organization in a market because different organizations adopt different strategies to enter in the same market. The ultimatum game and similarly the dictator game have different strategies for each player which is the most common example of asymmetric games.

6. Perfect Information and Imperfect Information

An important subset of sequential games consists of games of perfect information. A game is one of perfect information if, in extensive form, all players know the moves previously made by all other players. Simultaneous games cannot be games of perfect information, because the conversion to extensive form converts simultaneous moves into a sequence of moves with earlier moves being unknown. Most games studied in game theory are imperfect-information games. Interesting examples of perfect-information games include the ultimatum game and centipede game. Recreational games of perfect information games include chess and checkers. Many card games are games of imperfect information, such as poker or contract bridge.

Perfect information is often confused with complete information, which is a similar concept. Complete information requires that every player know the strategies and payoffs available to the other players but not necessarily the actions taken. Games of incomplete information can be reduced, however, to games of imperfect information by introducing moves by nature.

7. Discrete and Continuous Games

Discrete games have a finite number of players, moves, events, outcomes, etc. and many concepts can be extended. On the other hand continuous games allow players to choose a strategy from a continuous strategy set.

1.1.1.2 Nash Equilibrium

The Nash equilibrium is a game theoretic solution concept that is normally applied in economics. It was introduced by John Nash in 1950 and has emerged as one of the fundamental concepts of game theory. It is a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players and no player has anything to gain by changing only his own strategy.

3.2.3. Matching Algorithm Basic

The basic wireless resource management problem can be presented as a matching problem between resources and users. The resources can be represented as base stations, time-frequency chunks, power, or others. Users can be devices, stations, or smart phone applications. Each user and resource has a limit that defines the maximum number of players with which it can be matched. The main goal of matching is to optimally match resources and users. Each user (resource) builds a ranking of the resources (users) using a preference relation. The concept of a preference represents the individual view that each resource or user has on the other set, based on local information. In its basic form, a preference can simply be defined in terms of an objective utility function that quantifies the QoS achieved by a certain resource-user matching. However, a preference is more generic than a utility function in that it can incorporate additional qualitative measures extracted from the information available to users and resources.

A matching is essentially an allocation between resources and users. The basic solution concept for a matching problem is the so-called two-sided stable matching. A matching is said to be two-sided stable, if and only if there is no blocking pair (BP). A BP for a stable marriage case is defined as a pair of user and resource (u, r), where u prefers r to its currently matched user j, and r prefers u to its currently matched resource k. Thus, u will leave i to be matched to r and r would prefer being matched to user u than user k. Recently, matching theory has emerged as a promising technique for wireless resource allocation which can overcome some limitations of game theory and optimization.

There are mainly two types of matching one is two sided matching and another one is one sided matching. In briefly these are described below.

1. Two-sided Matching:

Two-sided matching model can be applied in many practical settings, such as corporate hiring, marriage, and university admission. The phrase ‘two-sided’ refers to the fact that in such markets, participants are partitioned into two disjoint sets e.g., men and women in a marriage market, students and colleges, firms or workers. Each agent on one side has preferences over the set of agents on the opposite side. Each participant on one side of the market wishes to be matched to a candidate from the other side of the market. Gale and Shapley (1962) introduced an algorithm to obtain a stable matching for any such two-sided market. They focused on college admissions and marriage. The traditional model of two-sided matching assumes that all agents fully know their own preferences. The first desired requirement for a matching is stability: a matching is stable if no pair of agents prefers to break their respective partnerships in order to be matched together.

- **One-to-one matching**

  Each player can be matched to at most one member of the opposite set. The most prominent example is the stable marriage problem in which men and women need to be matched for marriage.

- **Many-to-one matching:**

  Here, in one of the sets, at least one player can be matched to multiple players of the opposing set, while in the other set; every player has exactly one match. One example is the college admissions problem in which one student can be matched to one university while a university can recruit multiple students.

- **Many-to-many matching:**

  At least one player within each of the two sets could be matched to more than one member in the other set. Many-to-many matching are the most general type of problems and it admits many examples such as creating partnerships in peer-to-peer networks. There exist other classifications for matching problems, such as based on the partitioning of players, and the preference requirement for players.

2. One-sided Matching:

While most of the research on matching market focused on two-sided matching prior to late 1990s, this trend has changed since then. Two early models of one-sided matching played an important role in this change. House allocation and housing markets is one of the most common examples of one-sided matching.

Recently, matching theory has emerged as a promising technique for wireless resource allocation which can overcome some limitations of game theory and optimization [38]. The advantages of matching theory for wireless resource management include:

- Suitable models for characterizing interactions between heterogeneous nodes, each of which has its own type, objective, and information.
- The ability to define general “preferences” that can handle heterogeneous and complex considerations related to wireless quality of service (QoS).
- Suitable solutions, in terms of stability and optimality that accurately reflect different system objectives.

- Efficient algorithmic implementations that is inherently self-organizing and amenable to fast implementation.

3.2.3.1. Proposed Algorithm

At first we consider a set of host H, where H = $\{h_1, h_2, h_3 \ldots h_n\}$ and A is the set of APs, where A = $\{a_1, a_2, a_3 \ldots a_n\}$. We want to establish many to one connection between APs and host. That means the arrived hosts are matched with active AP. This is the result of this matching algorithm. Every AP and every host build their preference list which is based on game theory.

In the matching process, all arrived host propose their nearest active APs to be connected with that AP. However, it is possible that more than one host will propose the same AP. For this, we consider a blocking pair set. The APs that received request more than its capacity, in that case, that AP will be considered as blocking pair. When any host proposes its nearest AP to make a connection, the AP checks its capacity. If its capacity is not full the APs are directly matched with the requested host directly. At this situation blocking pair set is null.

At times, active APs receive request more than its capacity that means the blocking pair set is not null. At this time every active AP will check its preference list to make a connection. If the new arrived host is in the higher ranked in its preference list than any currently connected host. At that time the APs will make a connection with the new host and handover the lower ranked host to another host. But if the APs do not get any extra benefit from the new host afterward it will deny that host. The algorithm ends if there is no new request from hosts.

We summarize the two dimensional matching algorithm in Algorithm 1. Figure 3.6 shows the initial proposal establishment to form host-AP pair and Figure 3.7 shows the many to one matching result. The flow chart is given in Figure 3.8.

Algorithm 1: The Two-Dimensional Matching Algorithm

```
Input: 'A' set of AP's and 'H' set of Host's.
Output: All users are matched with APs denoted by a set X.
Initialization:
Every AP, $a \in A$ builds their preference list.
Set $X(a) = \emptyset, \forall a, \Omega = \emptyset$ (blocking pair).
while $\exists X(a) = \emptyset$ do
    for h $\in H$ do
        Every host proposes to AP to be connected.
    end for
    Find the APs that have received request more than its capacity and put it into blocking pair set $\Omega$.
    if $\Omega = \emptyset$ then
        APs are matched with the requested hosts directly.
    else
        for $\Omega \neq \emptyset$ then
            For each $a_k \in A$
            if preference $(h_i)$ > preference $(h_j)$
                accept $h_i$
                reject $h_j$
            else
                accept $h_j$
                reject $h_i$
            Update: Update $\Omega$ of $a_k$.
            end for
    end if
end white
```

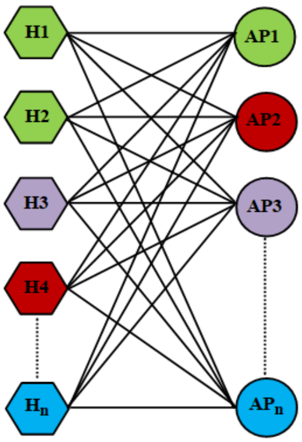


Figure 3.2: Initial proposal to form Host-AP pair

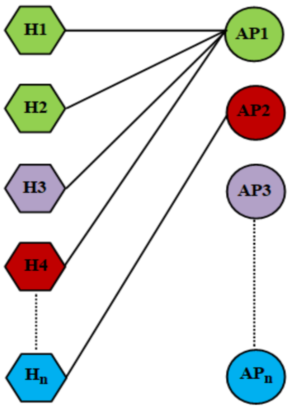
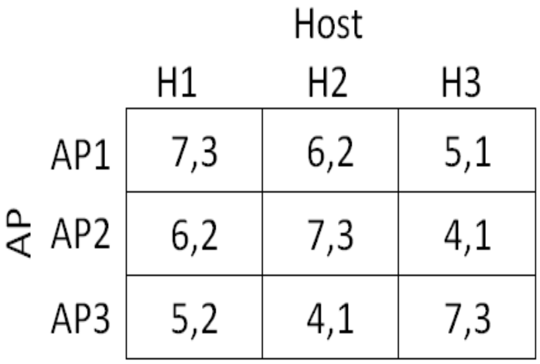


Figure 3.3: Many to one matching

In our work, game theory is used as an economic engineering model. By using payoff matrix each AP will build their preference list. The payoff matrix is given below:



		Host		
		H1	H2	H3
AP	AP1	7,3	6,2	5,1
	AP2	6,2	7,3	4,1
	AP3	5,2	4,1	7,3

Figure 3.4: Payoff matrix

This matrix represents the utility function for APs and hosts. Each AP always gives higher priority to its own host. So, if AP1 is connected with its own host that means H1, then its utility is high 7. But if AP1 is connected with H2 and H3, its utilities are 6 and 5 respectively. By this way AP1 will build its preference list where H1 is the first, H2 is second and H3 is the third choice. Similarly, others AP will build their preference list.

3.2.4. Flow Chart for Bandwidth Management

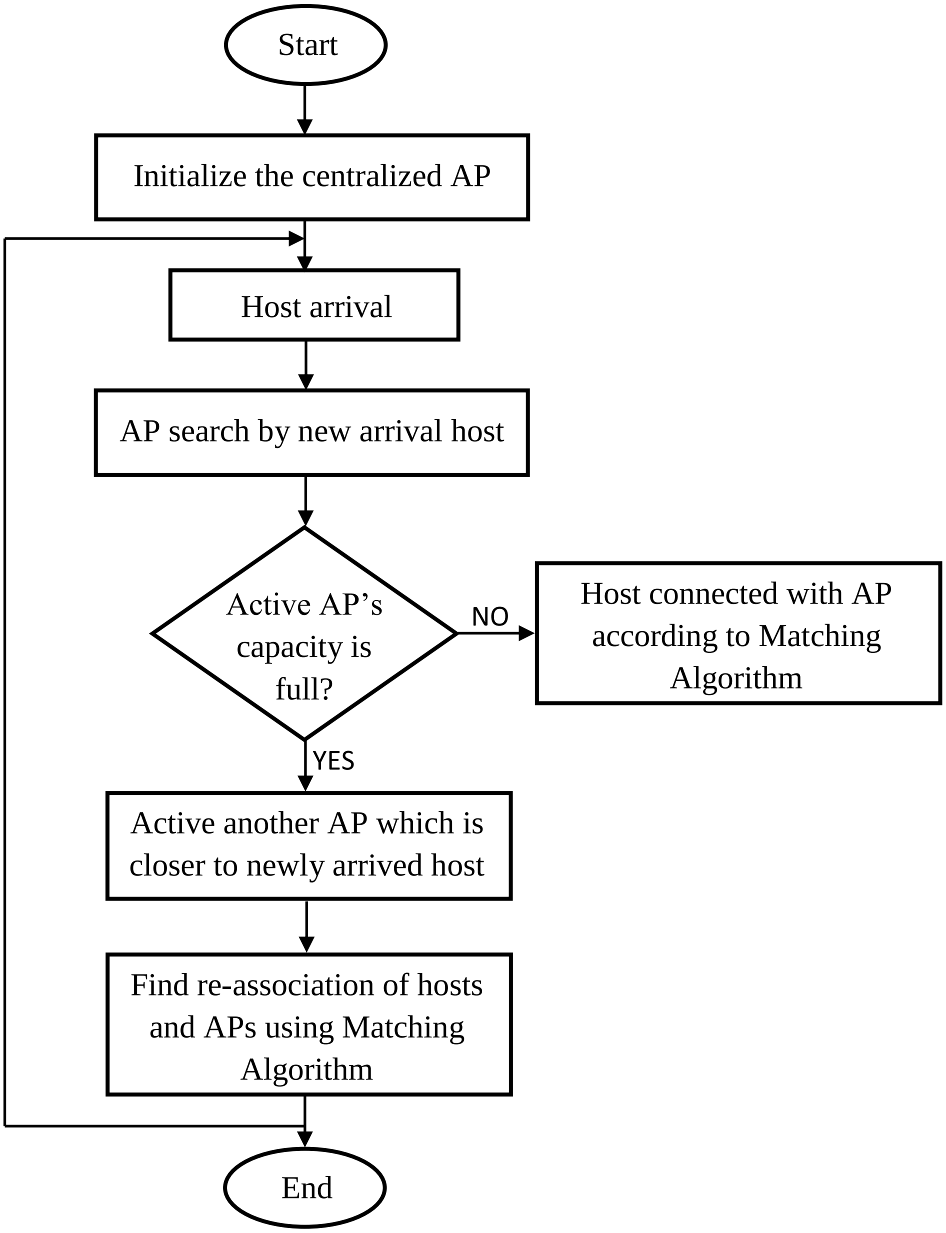


Figure 3.5: Bandwidth management flow chart

Chapter 4

RESULTS AND DISCUSSIONS

For cafeteria instance, we obtain the overall throughput of the scenario using the AP management algorithm where we change the average throughput threshold from 0 Mbps to 12.5 Mbps and link speed threshold is 40 Mbps. Here we consider two cases; one is same channel for all APs and another one is three different channels for different APs. We assume that every time users move, the active APs are stopped and new APs are turned on. However, in real situation, some APs cannot be turned off if they are currently associated with active hosts in service area. Those two different cases are described below.

4.1. Same Channel (CH-1) to All APs

TABLE III shows the simulation result of using same channel for all APs. The value of minimum average throughput and overall throughput are obtained by simulations using the WIMNET simulator. At first we evaluate the number of active APs by changing the average throughput threshold, where link speed threshold is fixed for all cases. To ensure 2.5 Mbps average throughput threshold, it needs 3 active APs. Thus more APs will be activated for ensuring higher average throughput. This table also shows that the position of bottleneck APs will change with the no. of active AP, but always the no. of bottleneck AP is one and the no. of associated hosts in bottleneck AP decreases with the increase of active APs.

As our assumption, the expected minimum average throughput will increase with the increase of active AP. But through the simulation results, we can see that the minimum average throughput increasing rate is not very high because here we’ve used same channel for all active APs.

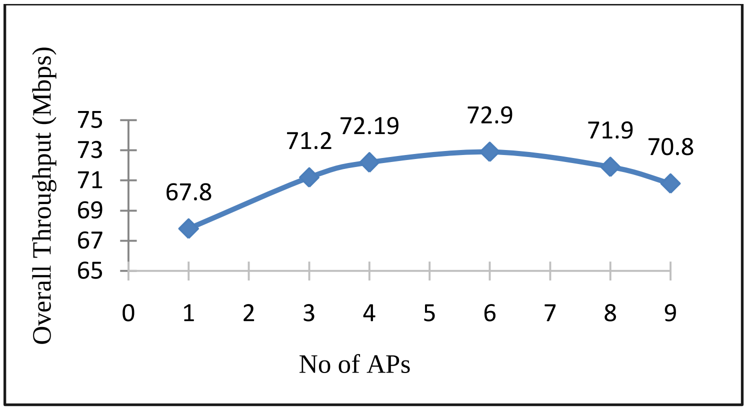


Figure 4.1: APs vs Overall Throughput (for same channel)

From Figure 4.1, we see that overall throughput increases with the expansion of APs. After certain point the overall throughput decreases instead of increasing. From this figure we can also see that after 6 APs, the overall throughput decreases with increase of AP. This point is the saturation point. After this point further increase of APs does not bring satisfactory outcome. Here we are using same channel for all APs. So, the main reason for this performance deterioration is the increase of interference between adjacent channels.

TABLE III
SIMULATION RESULTS FOR SAME CHANNEL (CH-1) TO ALL AP

Simulation Information							Results	
Average throughput threshold (Mbps)	Link speed threshold (Mbps)	Total Active AP	Bottleneck AP's position	No. of host in bottleneck AP	Expected max. avg. delay	Expected min. avg. throughput (Mbps)	Overall throughput (Mbps)	Min. avg. throughput (Mbps)
0	40	1	5	64	0.87	1.14	67.8	1.06
2.5	40	3	3	22	0.27	3.76	71.2	1.11
5	40	4	1	16	0.19	5.2	72.19	1.13
7.5	40	6	1	11	0.12	8.11	72.9	1.14
10	40	8	4	8	0.09	11.45	71.9	1.15
12.5	40	9	9	8	0.09	11.58	70.8	1.11

4.1.1. Network Association for Same Channel

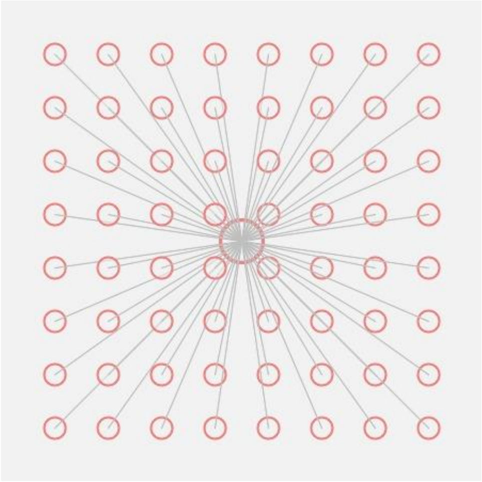


Figure 4.2: Nearest network association for avg. throughput threshold 0.0 Mbps and link speed 40Mbps

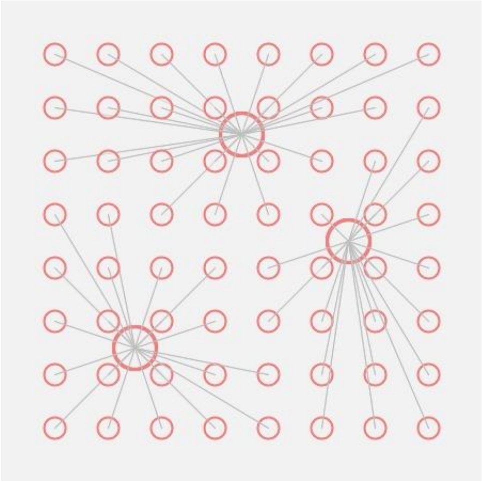


Figure 4.3: Nearest network association for avg. throughput threshold 2.5 Mbps and link speed 40Mbps

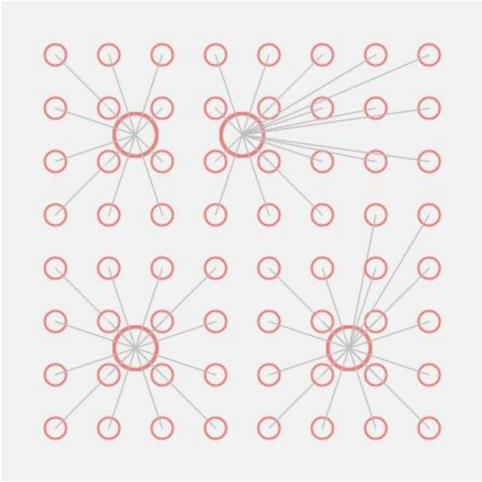


Figure 4.4: Nearest network association for avg. throughput threshold 5.0 Mbps and link speed 40Mbps

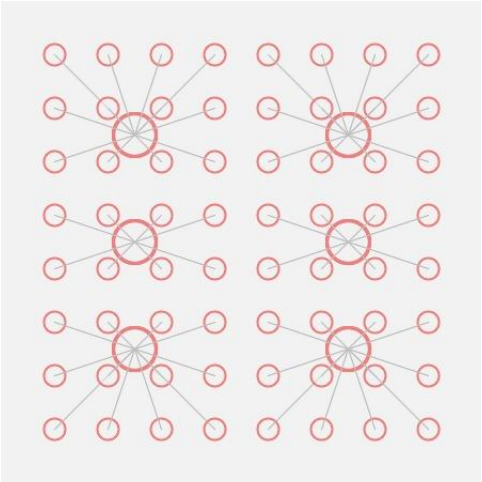


Figure 4.5: Nearest network association for avg. throughput threshold 7.5 Mbps and link speed 40Mbps

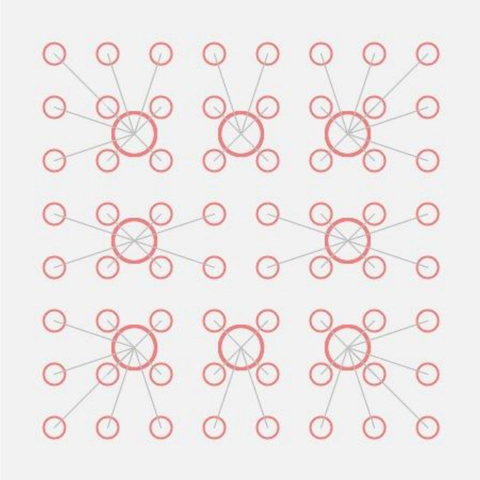


Figure 4.6: Nearest network association for avg. throughput threshold 10.0 Mbps and link speed 40Mbps

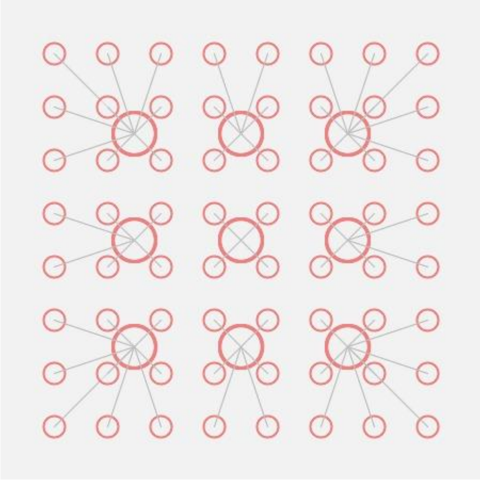


Figure 4.7: Nearest network association for avg. throughput threshold 12.5 Mbps and link speed 40Mbps

4.2. Different Channels to All APs

In this part, we use all different channels for different APs as shown in Figure 4.8. At a time we use all different channels into all different active APs. TABLE IV shows the simulation result of using all different channels for all APs. The value of minimum average throughput and overall throughput are obtained by simulations using the WIMNET simulator. At first we evaluate the number of active APs by changing the average throughput threshold, where link speed threshold is fixed for all cases. Thus more APs will be activated for ensuring higher average throughput. This table also shows that the position of bottleneck APs will change with the no. of active AP, but always the no. of bottleneck AP is one and the no. of associated hosts in bottleneck AP decreases with the increase of active APs.

Through simulation results, we can see that the minimum average throughput increasing rate is not very high because of same channel assignment in all active APs in TABLE III. But from TABLE IV it is shown that the overall throughput increasing rate is too high and minimum average throughput increasing rate is also high but not more than previous. The graphically representation of these condition is shown in Figure 4.8. So, the main reason for this high throughput increasing is minimal adjacent channel interferences for using all different channels. But this result is not matched with practical situation. That’s why we will not consider this simulation.

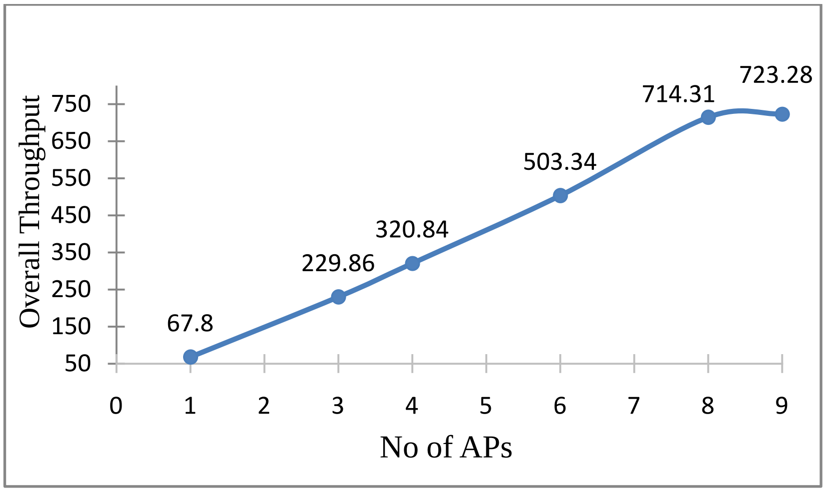


Figure 4.8: APs vs Overall Throughput (for different channels)

TABLE IV
SIMULATION RESULTS FOR DIFFERENT CHANNELS TO ALL AP

Simulation Information								Results	
Average throughput threshold (Mbps)	Link speed threshold (Mbps)	Activated AP and channel assign		Bottle-neck AP's position	No. of host in bottleneck AP	Expected max. avg. delay	Expected min. avg. throughput (Mbps)	Overall throughput (Mbps)	Min. avg. throughput (Mbps)
		Total active AP	Selected AP (Channel)						
0	40	1	5(1)	5	64	0.87	1.14	67.8	1.06
2.5	40	3	3(1),4(2),8(3)	3	22	0.27	3.76	229.86	3.59
5	40	4	1(1),3(2),4(3),9(4)	1	16	0.19	5.2	320.84	5.01
7.5	40	6	1(1),2(2),3(3),7(4), 8(5),9(6)	1	11	0.12	8.11	503.34	7.87
10	40	8	1(1),2(2),3(3),4(4), 6(5),7(6),8(7),9(8)	4	8	0.09	11.45	714.31	11.17
12.5	40	9	1(1),2(2),3(3),4(4), 5(5),6(6),7(7),8(8), 9(9)	9	8	0.09	11.58	723.28	11.3

4.2.1. Network Association for Different Channels

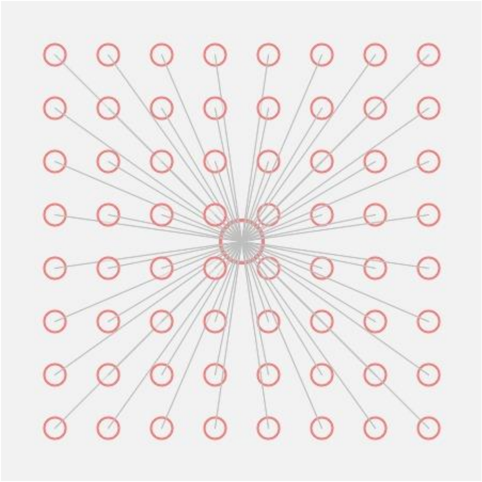


Figure 4.9: Nearest network association for avg. throughput threshold 0.0 Mbps and link speed 40Mbps

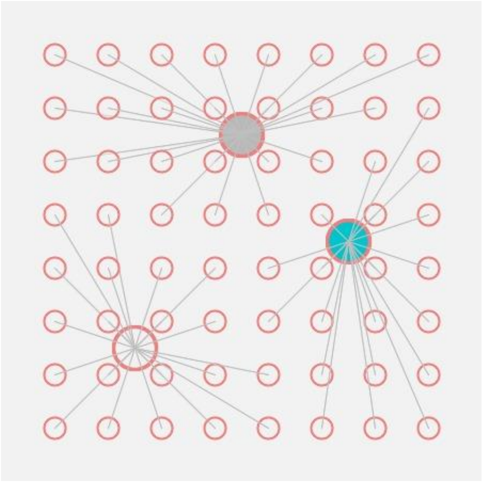


Figure 4.10: Nearest network association for avg. throughput threshold 2.5 Mbps and link speed 40Mbps

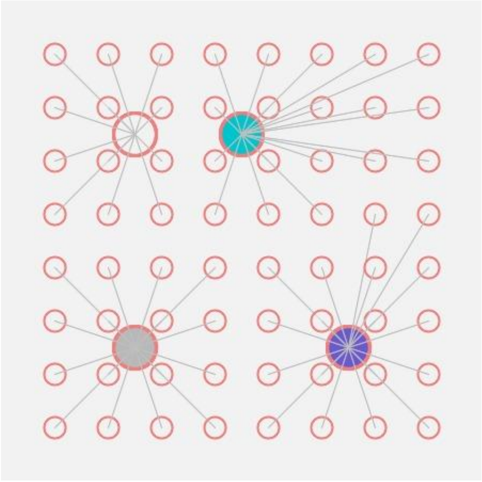


Figure 4.11: Nearest network association for avg. throughput threshold 5.0 Mbps and link speed 40Mbps

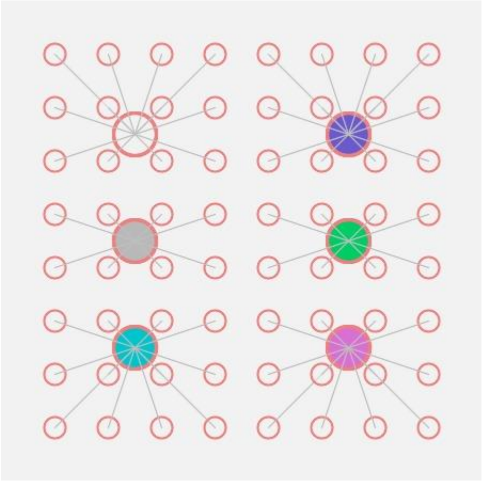


Figure 4.12: Nearest network association for avg. throughput threshold 7.5 Mbps and link speed 40Mbps

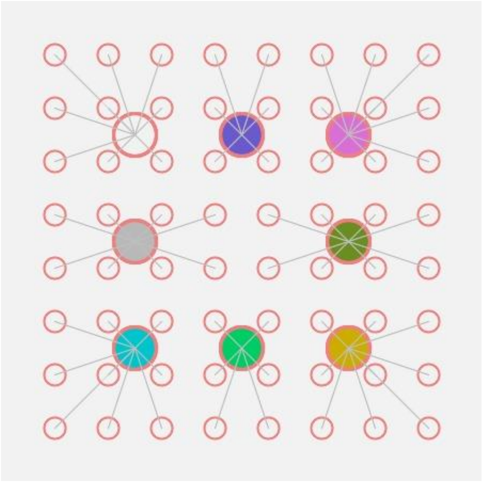


Figure 4.13: Nearest network association for avg. throughput threshold 10.0 Mbps and link speed 40Mbps

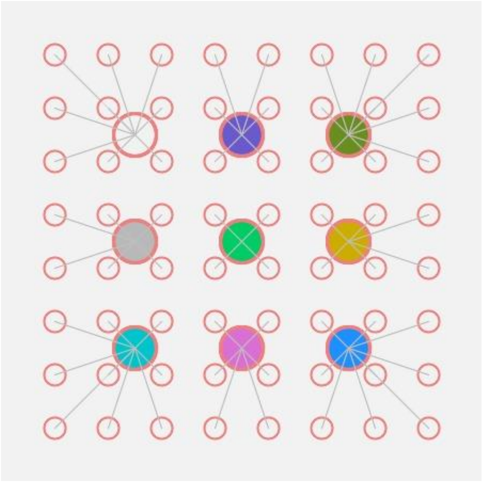


Figure 4.14: Nearest network association for avg. throughput threshold 12.5 Mbps and link speed 40Mbps

4.3. Three Different Channels (1, 6 & 11) to Different APs

In this part, we use three different channels for different APs as shown in Figure 4.15. We assigned 9 APs with 3 different channels. For effective performance gain, it is possible to consider the channel assignment problem with several established algorithms (e.g. Graph Coloring Algorithm). However for simplicity purpose we arbitrarily assign the channel to each AP.

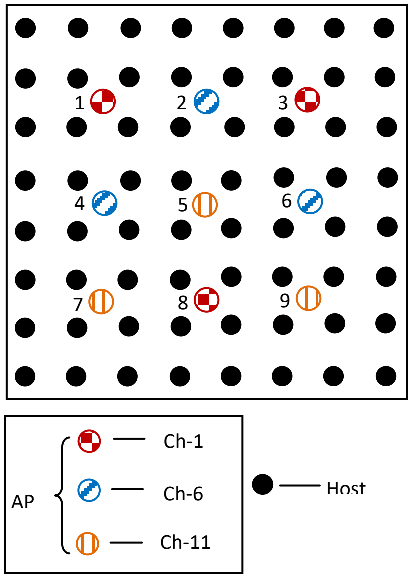


Figure 4.15: Channel assignment for all APs in Cafeteria instance

TABLE V
SIMULATION RESULTS FOR THREE DIFFERENT CHANNELS (1, 6 & 11) TO DIFFERENT APS

Simulation Information								Results	
Average throughput threshold (Mbps)	Link speed threshold (Mbps)	Activated AP and channel assign		Bottle-neck AP's position	No. of host in bottleneck AP	Expected max. avg. delay	Expected min. avg. throughput (Mbps)	Overall throughput (Mbps)	Min. avg. throughput (Mbps)
		Total active AP	Selected AP (Channel)						
0	40	1	5(11)	5	64	0.87	1.14	67.8	1.06
2.5	40	3	3(1),4(6),8(1)	3	22	0.27	3.76	109.86	1.72
5	40	4	1(1),3(1),4(6),9(11)	1	16	0.19	5.2	154.06	2.41
7.5	40	6	1(1),2(6),3(1),7(11), 8(1),9(11)	1	11	0.12	8.11	158.27	2.47
10	40	8	1(1),2(6),3(1),4(6),6 (6),7(11),8(1),9(11)	4	8	0.09	11.45	216.98	3.43
12.5	40	9	1(1),2(6),3(1),4(6),5 (11),6(6),7(11),8(1), 9(11)	9	8	0.09	11.58	232.83	3.64

Simulation result of using three different channels for different active APs is listed in TABLE V. Here we are using three channels: Ch-1, Ch-6 and Ch-11. When three APs like 3rd, 4th and 8th are activated, then Ch-1 is assigned to 3rd AP, Ch-6 is assigned to 4th AP and again Ch-1 is assigned to 8th AP. This table also shows that the overall throughput will increase with the increasing of no. of active APs. As we are using 3 different channels for all active APs, the minimum average throughput will increase compared to using same channel for all active APs. But still it is not equal as expected minimum average throughput.

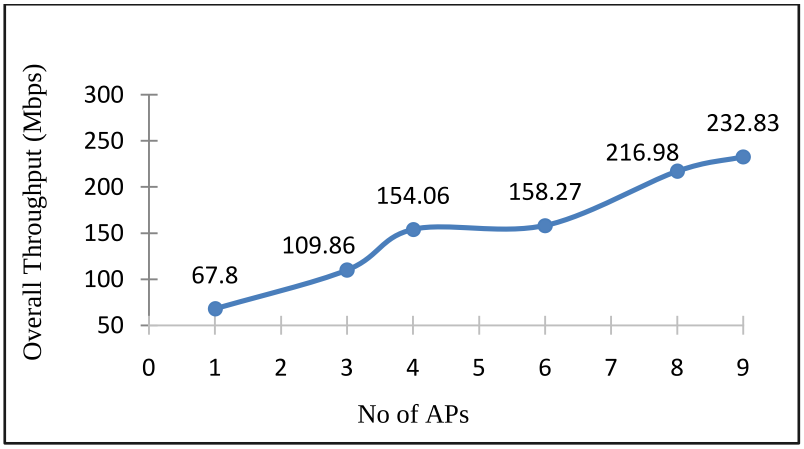


Figure 4.16: APs vs Overall Throughput (for three different channels)

From Figure 4.16, we see that with the increase of AP the corresponding overall throughput also increases. This is because the host uses different channels. However, we see that with the increase of the no. of AP the gain in throughput is not very consistent. As shown in Figure 4.16, the throughput increasing rate is not significant for 4 to 6 active APs. But for 6 to 8 active AP the overall throughput increasing rate is high. It is because of the orientation of the channel and AP.

4.3.1. Network Association for Three Different Channels

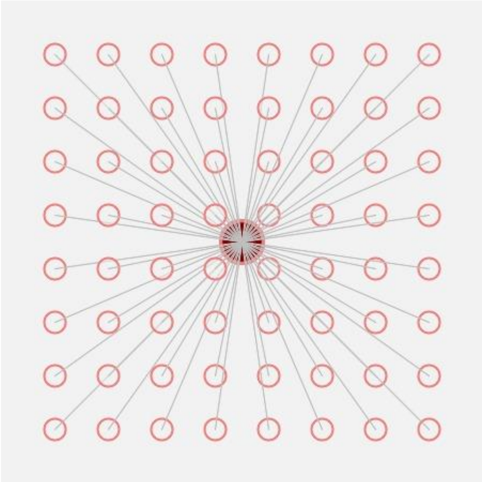


Figure 4.17: Nearest network association for avg. throughput threshold 0.0 Mbps and link speed 40Mbps

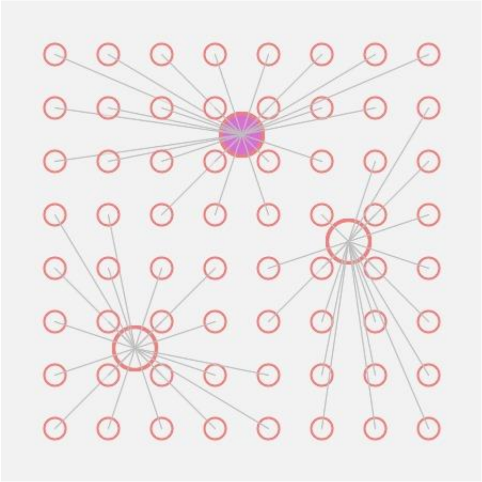


Figure 4.18: Nearest network association for avg. throughput threshold 2.5 Mbps and link speed 40Mbps

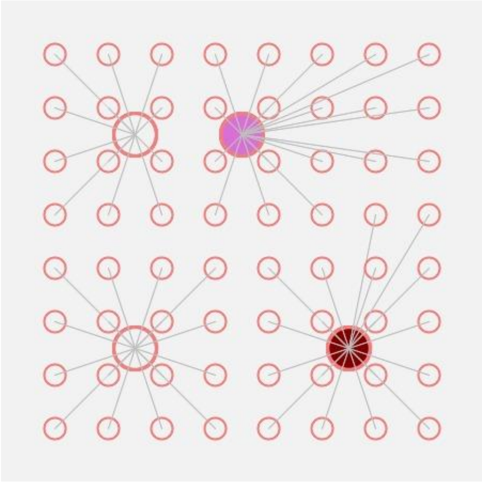


Figure 4.19: Nearest network association for avg. throughput threshold 5.0 Mbps and link speed 40Mbps

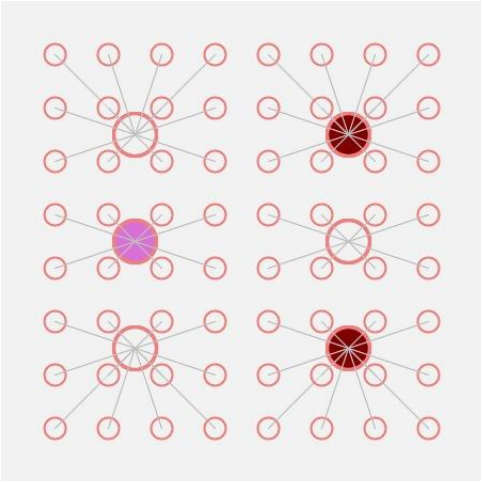


Figure 4.20: Nearest network association for avg. throughput threshold 7.5 Mbps and link speed 40Mbps

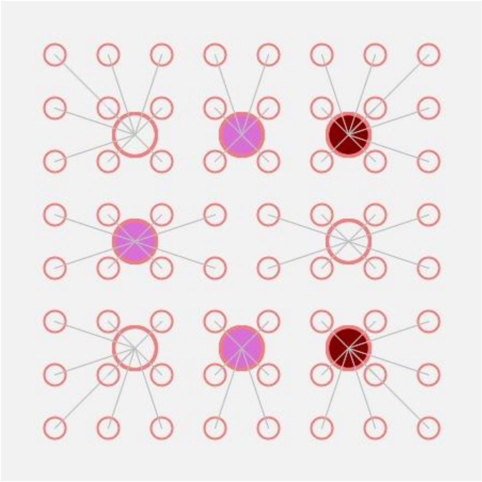


Figure 4.21: Nearest network association for avg. throughput threshold 10.0 Mbps and link speed 40Mbps

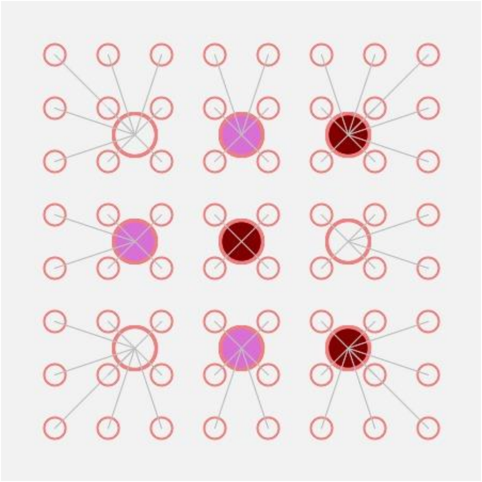


Figure 4.22: Nearest network association for avg. throughput threshold 12.5 Mbps and link speed 40Mbps

Chapter 5

CONCLUSIONS AND SCOPE OF FUTURE WORKS

5.1. Conclusions

In this research, we have studied on access point (AP) management in overcrowded WLANs. We analyze the relationship of APs and achieved throughput for a given scenario. Here we have presented two different simulation results for AP management. By adding active APs we've ensured the minimum average throughput for any host. We verified the effectiveness through simulations using WIMNET simulator. We have also proposed a model for a given scenario where the different operators can share their Bandwidth with each other by applying Matching Algorithm and Game Theory model.

5.2. Future Works

In future studies, we have a plan to evaluate our proposed work through experimental setup and implementation with the consideration of host arrival rate by Poisson distribution techniques.

REFERENCES

[1] M. Gast. "*802.11 Wireless Networks: The Definitive Guide*". O'Reilly Media, Inc., 2nd edition, April 2005.

[2] A. S Tanenbaum. "Computer Networks". Prentice Hall, 5th edition, October 2010.

[3] K. Nishide, H. Kubo, R. Shinkuma and T. Takahashi, "Detecting hidden and exposed terminal problems in densely deployed wireless networks," in *IEEE Trans. Wireless Commun*., Vol.11, Issue 11, November 2012.

[4] J. Riihijarvi, M. Petrova and P. Mahonen, "Frequency allocation for WLANs using graph colouring Techniques", *Proc. 2nd Annual Conf. Wireless On-demand Network Systems and Services (WONS)*, Washington DC, WA, pp. 216-22, January 2005.

[5] A. Baid and D. Raychaudhuri, "Understanding channel selection dynamics in dense Wi-Fi networks", in *IEEE Commun. Mag. ,* Volume: 53, Issue: 1, January 2015.

[6] G. Bianchi, "IEEE 802.11-saturation throughput analysis", *IEEE Commun. Lett.* , Vol.2, Issue 12, December 1998.

[7] F. Xu, X. Zhu, C. C Tan, Q. Li, G. Yan, and J. Wu, "Smart Assoc: Decentralized access point selection algorithm to improve throughput". in *IEEE Trans. Parallel Distrib. Syst.* , 24(12):2482{2491, December 2013.

[8] S. Miyata, T. Murase, and K. Yamaoka. "Novel access-point selection for user QoS and system optimization based on user cooperative moving", in *IEICE Trans. Commun.* , 95(6):1953{1964, June 2012.

[9] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11", in *IEEE Commun. Lett.* , vol. 6, no. 8, August. 2002.

[10] X. Ling and K. L. Yeung, "Joint access point placement and channel assignment for 802.11 wireless LANs", in IEEE Trans. Wireless Commun. , Volume: 5, Issue: 10, October. 2006.

[11] L. Sequeria, J. L. de la Cruz, J. R-Mas, J. Saldana, J. F. Navajas and J. Almodovar, "Building an SDN enterprise WLAN based on virtual APs", in IEEE Commun. Lett. , Volume: 21, Issue: 2, February. 2017.

[12] R. Riggio, T. Rasheed and M. K. Marina, "Poster: Programming software-defined wireless networks", in Proc. MobiCom, pp. 413-416, 2014.

[13] R. Riggio, T. M. Rasheed and R. Narayanan, "Virtual network functions orchestration in enterprise WLANs", in Proc. IM, pp. 1220-1225, May 2015.

[14] M. E. Berezin, F. Rousseau and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks", Proc. VTC Spring, pp. 1-5, May 2011.

[15] L. Zabala, A. Ferro and A. Pineda, "Modelling packet capturing in a traffic monitoring system based on Linux", in Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS), pp. 1-6, July. 2012.

[16] X. Yue, C.F. Wong and S.H.G. Chan, "CACAO: Distributed client-assisted channel assignment optimization for uncoordinated WLANs", in IEEE Trans. Parallel Distrib. Syst., Volume: 22, Issue: 9, September. 2011.

[17] F. Ganji, L. Budzisz, and A. Wolisz, "Assessment of the power saving potential in dense enterprise WLANs," in Proc. IEEE 24th Int. Symp. Pers. Indo. Mob. Radio Commun. . (PIMRC2013), pp. 2835-2840, September. 2013.

[18] Y. Xiao, "IEEE 802.11n: enhancements for higher throughput in wireless LANs," in *IEEE Wireless Communications*, Volume 12, Issue 6, December, 2005.

[19] M. Drieberg, F. C. Zheng, R. Ahmad and S. Olafsson, "An improved distributed dynamic channel assignment scheme for dense WLANs," in *Information, Communications & Signal Processing*, 6th International Conference on, 2007.

[20] A. Nazari, N. Yazdani and N. Rezaee, "Bandwidth Management in Shared WLAN," in *Communications and Networking in China*, ChinaCom '06. First International Conference on, 2006.

[21] M. Touati, R. El-Azouzi, M. Coupechoux, E. Altman and J. M. Kelif, "A Controlled Matching Game for WLANs," in *IEEE Journal on Selected Areas in Communications,* 35*,* pp. 707-720*,* 2017.

[22] Z. Zhou, C. Gao, C. Xu, T. Chen, D. Zhang and S. Mumtaz, "Energy-Efficient Stable Matching for Resource Allocation in Energy Harvesting-Based Device-to-Device Communications," in *IEEE Access* , Volume: PP, Issue: 99, 06 March 2017.

[23] M. W. Baidas and M. M. Afghah, "A matching-theoretic approach to energy-efficient partner selection in wireless networks," in *Proc. International Wireless Communications and Mobile Computing Conference (IWCMC)*, Dubrovnik, Croatia, pp. 1260–1265, August 2015.

[24] E. A. Panaousis, C. Politis and G. C. Polyzos, "Power Control Using Game Theory in a Shared Open Spectrum," Unpublished.

[25] M. S. Seddiki and M. Frikha, "A non-cooperative game theory model for bandwidth allocation in Network Virtualization", in *Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2012 XVth International,* 1 February2013.

[26] Y. Zhou, Y. Li, G. Sun, D. Jin, L. Su and L. Zeng, "Game Theory Based Bandwidth Allocation Scheme for Network Virtualization", in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE,* 10 January 2011.

[27] W. Lin, Z. Qu and M. A. Simaan, "A Design of Distributed Nonzero-Sum Nash Strategies", in *Decision and Control (CDC), 2010 49th IEEE Conference on,* 22 February 2011.

[28] A. Pashaie, L. Pavel and C. J. Damaren, "Population Dynamics Approach for Resource Allocation Problems", in *American Control Conference (ACC), 2015,* 30 July 2015.

[29] A. Ananth and K. Chandrasekaran, "Cooperative Game Theoretic Approach for Job Scheduling in Cloud Computing", in *Computing and Network Communications (CoCoNet), 2015 International Conference on,* 18 February 2016.

[30] J. Kibilda, F. Malandrino and L. A. DaSilva, "Incentives for Infrastructure Deployment by Over-the-Top Service Providers in a Mobile Network: A Cooperative Game Theory Model", in *Communications (ICC), 2016 IEEE International Conference on,* 14 July 2016.

[31] D. Wu, Q. Wu, Y. Xu J. Jing and Z. Qin, "QoE-Based Distributed Multichannel Allocation in 5G Heterogeneous Cellular Networks: A Matching-Coalitional Game Solution", in *IEEE Access* (Volume*:* 5), 20 September 2016.

[32] S. D'Oro, L. Galluccio, S. Palazzo and G. Schembra, "A Game Theoretic Approach for Distributed Resource Allocation and Orchestration of Softwarized Networks", in *IEEE Journal on Selected Areas in Communications* (Volume*: 35, Issue: 3, March 2017*), 20 February 2017.

[33] C. Xu , J. Feng , B. Huang , Z. Zhou , S. Mumtaz and J. Rodriguez, “Joint Relay Selection and Resource Allocation for Energy-Efficient D2D Cooperative Communications Using Matching Theory”, in *Appl. Sci.* 2017, 7, 491; doi:10.3390/app7050491.

[34] Zhou, Z.; Ota, K.; Dong, M.; Xu, C. “Energy-efficient matching for resource allocation in D2D enabled cellular networks”, *IEEE Trans. Veh. Technol*. 2016, PP, 1–13.

[35] Y. Gu , W. Saad, M. Bennis , M. Debbah and Z. Han, “Matching theory for future wireless networks: fundamentals and applications,” in *IEEE Communications Magazine*, Volume: 53, Issue: 5, May 2015.

[36] M. E. Islam, N. Funabiki, and T. Nakanishi, “Extensions of access-point aggregation algorithm for large-scale wireless local area networks,” in *International Journal of Networking and Computing*, Volume 5,Number 1, Pages 200-222, January. 2015.

[37] M. E. Islam, N. Funabiki, T. Nakanishi, and K. Watanabe, “An access point aggregation approach for energy-saving wireless local area networks,” in *IECE Trans. on Commun.* , E96-B (12):2986-2997, December 2013.

[38] E. A. Jorswieck, “Stable matchings for resource allocation in wireless networks,” in *Proc. of the 17th International Conference on Digital Signal Processing (DSP)*, Corfu, Greece, Jul. 2011.

Appendix A

All Codes are listed here for AP Management Algorithm:

```
// Variable Declaration for AP Info.
struct APInfo
    {
        int APID;
        float PositionX,PositionY;
        int GroupID;
        int NoConHost;
        int ConHost[DEFAULT_MAX_HOST];
        int ifActive;
        int HostMargin;
        int ifSatisfied;
        int nodeID;
        int ChannelID;
    }
// Variable Declaration for Host Info.
struct HostInfo
    {
        int HostID;
        float PositionX,PositionY;
        int GroupID;
        float HostActiveRate;
        int NoConAP;
        int ConAP[DEFAULT_MAX_AP];
        int AssocAP;
        float AssocAP_HostLinkSpeed
        int NoSat;
        int SatAP[DEFAULT_MAX_AP];
        int nodeID;
        int ChannelID;
    }
int main(int argc, char *argv[])
{
    int noOfActiveAp, sumLinkCandidate;
    float tempSumAP_HostLinkSpeed;
    srand ( atoi(argv[4]) );
        s1=(const char *) argv[1];
    ipFileNamePart = substring(s1, 5, strlen(s1)-5-4);

     //***************** Input Section *********************

```

```
    inputNetworkInfo(argv[1]);

// **********  Initialize important loop parameters /*
    LOOP_COUNT_LB_TX_TIME = 40 * NoAP * NoHost;
    LOOP_COUNT_HILL_CLIMB_LB_TX_TIME = 5*NoAP;//5 * NoAP;
    LOOP_COUNT_AP_SELECTION_OPTIMIZATION = NoAP;
    HILL_CLIMB_MUTATION_FACTOR = 10;
    AP_SELECTION_HILL_CLIMBING_RATIO = 50;
    LOOP_COUNT_LB_TX_TIME = atoi(argv[5]);
    LOOP_COUNT_HILL_CLIMB_LB_TX_TIME = atoi(argv[6]);
    LOOP_COUNT_AP_SELECTION_OPTIMIZATION = atoi(argv[7]);
    HILL_CLIMB_MUTATION_FACTOR = atoi(argv[8]);
    AP_SELECTION_HILL_CLIMBING_RATIO = atoi(argv[9]);
    THROUGHPUT_IMPROVE_MARGIN = strtod(argv[10], NULL);


    linkSpeedThreshold = strtod(argv[2], NULL);
    AverageHostThroughputThreshold = strtod(argv[3], NULL);

    // ****************** Preprocessing ******************

    calculateNoOfWallsBtwEachHost_AP_Pair();
    determineAssociabilityOfAP_HostUsingOnlyCoverageRadius();
    // ****************** Algorithm: 'Minimizing # of AP' Section ******************

    //   *****    Step 1: Preparation    *****         //
    PHASE_INITIALIZATION();

    sumLinkCandidate = 0;
    for(int i = 0; i < NoAP; i++)
    {
        tempSumAP_HostLinkSpeed=0.0;
        sumLinkCandidate = sumLinkCandidate + AINew[i].NoConHost;
AINew[i].PositionY,AINew[i].NoConHost);
        for(int j = 0; j < AINew[i].NoConHost; j++)
        {
            tempSumAP_HostLinkSpeed = tempSumAP_HostLinkSpeed +
AP_HostLinkSpeed[i][AINew[i].ConHost[j]-1];
            Host[j]-1].PositionX,HINew[AINew[i].ConHost[j]-1].PositionY));
        }
    }


    //   ***** Step 2: Selection of active APs and their associated hosts   *****  //
    PHASE_AP_SELECTION();
```

```

    //   ***** Step 3: Consider Maximum Permitted Host   *****  //
 //   PHASE_ASSOCIATION_MODIFICATION();

    //   **** Post processing *************

    noOfActiveAp=0;
    for(int i=0; i<NoAP; i++)
        {
             if(AIActive[i].ifActive == 1)
             {
                 AIActive[i].nodeID = noOfActiveAp;            // *** Assign Node ID for Active Aps here

                 noOfActiveAp++;
             }
            }

        for (int i=0; i<NoHost; i++)
        {
            HIActive[i].HostID = HINew[i].HostID;
            HIActive[i].PositionX = HINew[i].PositionX;
            HIActive[i].PositionY = HINew[i].PositionY;
            HIActive[i].GroupID = HINew[i].GroupID;
            HIActive[i].HostActiveRate = HINew[i].HostActiveRate;
            HIActive[i].AssocAP = HINew[i].AssocAP;
            HIActive[i].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[HIActive[i].AssocAP-1][i];

            HIActive[i].nodeID = noOfActiveAp+i;
            HIActive[i].NoConAP = 0;
            for (int j=0; j<HINew[i].NoConAP; j++)
            {
                if(AIActive[HINew[i].ConAP[j]-1].ifActive==1)  // consider only Active APs as associable APs
                {
                    HIActive[i].ConAP[HIActive[i].NoConAP] = HINew[i].ConAP[j];
                    HIActive[i].AP_HostLinkSpeed[HIActive[i].NoConAP] = AP_HostLinkSpeed[HINew[i].ConAP[j]-1][i];
                    HIActive[i].NoConAP++;
                }
            }

        }
        inputChannelIDforActiveAPs();

```

```
    // ***************** Throughput Update Section *********************

    // *** Step 1: Preprocessing

    TEST_NEAREST_AP_HOST_ASSOCIATION();

    PHASE_RANDOM_MOVE_TO_REDUCE_TX_TIME_FOR_BOTTLENECK_AP();

    PHASE_AP_SELECTION_OPTIMIZATION();

   PostAssociationOptimization();
    // ***************** Post Algorithm Section *********************
    inputChannelIDforActiveAPs();
    noOfActiveAp=0;
    for(int i=0; i<NoAP; i++)   if(AIActive[i].ifActive == 1)
    {
        AIActive[i].nodeID = noOfActiveAp;
        noOfActiveAp++;
    }
    for (int i=0; i<NoHost; i++)
HIActive[i].nodeID = noOfActiveAp+i;
    generateOutputAndSimulationInputFiles("AfterOptimization_", AIActive, HIActive);
    return 0;
}

void PHASE_INITIALIZATION()
{
    int curHost,curAP;
    float maxLinkSpeedForHost[NoHost];
    int maxLinkSpeedAPforHost[NoHost];
    float tempLinkSpeed;
    calculateInitialAP_HostLinkSpeed();
        for (int i=0; i<NoHost; i++)
    {
        HINew[i].HostID = HI[i].HostID;
        HINew[i].PositionX = HI[i].PositionX;
        HINew[i].PositionY = HI[i].PositionY;
        HINew[i].GroupID = HI[i].GroupID;
        HINew[i].HostActiveRate = HI[i].HostActiveRate;
        HINew[i].NoConAP = 0;

        maxLinkSpeedForHost[i] = 0.0;
        maxLinkSpeedAPforHost[i] = -1;
    }
    for (int i=0;i<NoAP;i++)
```

```
        {
            AINew[i].APID = AI[i].APID;
            AINew[i].PositionX = AI[i].PositionX;
            AINew[i].PositionY = AI[i].PositionY;
            AINew[i].GroupID = AI[i].GroupID;
            AINew[i].NoConHost = 0;

            for (int j=0;j<AI[i].NoConHost;j++)
            {
                if (Aggr[AI[i].GroupID-1][HI[AI[i].ConHost[j]-1].GroupID-1]==1)
                {
                     HI[AI[i].ConHost[j]-1].PositionX, HI[AI[i].ConHost[j]-1].PositionY );
                    tempLinkSpeed = AP_HostLinkSpeed[i][AI[i].ConHost[j]-1];
                    if(tempLinkSpeed > maxLinkSpeedForHost[AI[i].ConHost[j]-1])
                    {
                        maxLinkSpeedForHost[AI[i].ConHost[j]-1] = tempLinkSpeed;
                        maxLinkSpeedAPforHost[AI[i].ConHost[j]-1] = i;
                    }
                    if(tempLinkSpeed >= linkSpeedThreshold)                     {
                        curAP=i+1;
                        curHost= AI[i].ConHost[j];
                        AINew[i].ConHost[AINew[i].NoConHost]= curHost;
                        AINew[i].NoConHost++;
                        HINew[curHost-1].ConAP[HINew[curHost-1].NoConAP]=curAP;
                        HINew[curHost-1].AP_HostLinkSpeed[HINew[curHost-1].NoConAP] = tempLinkSpeed;
                        HINew[curHost-1].NoConAP
                    }
                }
            }
        }
        for (int i=0; i<NoHost; i++)
        {
            if(HINew[i].NoConAP == 0)
            {
              printf("BEST FIT!! ");

              curAP = maxLinkSpeedAPforHost[i];
              AINew[curAP].ConHost[AINew[curAP].NoConHost]= i+1
              AINew[curAP].NoConHost++;
              HINew[i].ConAP[HINew[i].NoConAP]=curAP+1
              HINew[i].AP_HostLinkSpeed[HINew[i].NoConAP]=maxLinkSpeedForHost[i];
              HINew[i].NoConAP++;
            }
        }
```

```
        for(int i=0; i<NoAP; i++)
        {
            AINew[i].ifActive = 0;
            AIActive[i].ifActive = 0;
            AIActive[i].APID = AINew[i].APID;
            AIActive[i].PositionX = AINew[i].PositionX;
            AIActive[i].PositionY = AINew[i].PositionY;
            AIActive[i].GroupID = AINew[i].GroupID;
            AIActive[i].NoConHost = 0;
        }
    }

    void PHASE_AP_SELECTION()
    {
        int max,maxAP;
        float maxSumAP_HostLinkSpeed,tempSumAP_HostLinkSpeed;
        for(int i=0;i<NoHost;i++)   HINew[i].AssocAP=-1;
        do
        {
            maxSumAP_HostLinkSpeed=0.0;
            maxAP=-1;
            for(int i=0;i<NoAP;i++)
            {
                if(AINew[i].ifActive == 0)            {
                    tempSumAP_HostLinkSpeed=0.0;
                    for(int j=0;j<AINew[i].NoConHost;j++)
                    {
                        if(HINew[AINew[i].ConHost[j]-1].AssocAP==-1)  tempSumAP_HostLinkSpeed =
tempSumAP_HostLinkSpeed + AP_HostLinkSpeed[i][AINew[i].ConHost[j]-1];
                        Host[j]-1].PositionX,HINew[AINew[i].ConHost[j]-1].PositionY));
}
                    if(tempSumAP_HostLinkSpeed>maxSumAP_HostLinkSpeed)
                    {
                        maxSumAP_HostLinkSpeed=tempSumAP_HostLinkSpeed;
                        maxAP=i;
                    }
                }
            }

            if(maxAP!=-1)
            {
                AINew[maxAP].ifActive = 1;
                AIActive[maxAP].ifActive = 1;
                AIActive[maxAP].NoConHost = 0;
                for(int i=0;i<AINew[maxAP].NoConHost;i++)
```

```
                {
                    if(HINew[AINew[maxAP].ConHost[i]-1].AssocAP==-1)
                    {
                        HINew[AINew[maxAP].ConHost[i]-1].AssocAP = maxAP+1;
                        HINew[AINew[maxAP].ConHost[i]-1].AssocAP_HostLinkSpeed =
AP_HostLinkSpeed[maxAP][AINew[maxAP].ConHost[i]-1];
AIActive[maxAP].ConHost[AIActive[maxAP].NoConHost]=AINew[maxAP].ConHost[i];
    }
                }
            }
            flag=1;
            for(int i=0;i<NoHost;i++)
                if(HINew[i].AssocAP==-1)
                {
                    flag=0;
                    break;
                }
        }
    while(flag==0
    }

    void PHASE_RANDOM_MOVE_TO_REDUCE_TX_TIME_FOR_BOTTLENECK_AP()
    {
        int curHost,curAP;
        float tempMaxTxTime, globalMaxTxTime = INITIAL_MAX_TX_TIME;
        int tempFlag, noOfMovableLowLSpHost, movableLowLSpHosts[NoHost];
        int noOfMovableHost, movableHost[NoHost], bestLSpAPforHost[NoHost];
        float distanceBtwAP_Host, bestLSpforHost[NoHost];
        noOfMovableHost = 0;
        noOfMovableLowLSpHost = 0;
        for (int i=0; i<NoHost; i++)
        {
            bestLSpforHost[i] = 0.0;
            bestLSpAPforHost[i] = -1;
            for(int j=0; j<HIActive[i].NoConAP; j++)
                    {
                        if(AP_HostLinkSpeed[HIActive[i].ConAP[j]-1][i] > bestLSpforHost[i])
                        {
                            bestLSpforHost[i] = AP_HostLinkSpeed[HIActive[i].ConAP[j]-1][i];
                            bestLSpAPforHost[i] = HIActive[i].ConAP[j]-1;
                        }
                    }
            if(HIActive[i].NoConAP > 1)
                {
                    movableHost[noOfMovableHost]=i;
```

```
                noOfMovableHost++;
            }
    }

    if(noOfMovableHost > 0)
    {

        for(int k=0; (k<LOOP_COUNT_HILL_CLIMB_LB_TX_TIME)  ; k++)
        {
            tempMaxTxTime = Local_Sesrch_RANDOM_MOVE_TO_REDUCE_TX_TIME_FOR_BOTTLENECK_AP();
            if(tempMaxTxTime < globalMaxTxTime)
            {
                globalMaxTxTime = tempMaxTxTime;
                SaveSolution(AIActive, AITempSol, HIActive, HITempSol);
            }
            noOfMovableLowLSpHost = 0;
            for (int i=0; i<NoHost; i++)
            {
                if( (HIActive[i].NoConAP > 1) && (HIActive[i].AssocAP_HostLinkSpeed != bestLSpforHost[i]) )
                {
                    movableLowLSpHosts[noOfMovableLowLSpHost] = i;
                    noOfMovableLowLSpHost++;
                }
            }

            for(int i = 0; i < noOfMovableLowLSpHost; i++)
            {
                curHost = movableLowLSpHosts[i];

                curAP = bestLSpAPforHost[curHost];

                AIActive[curAP].NoConHost++;
                AIActive[curAP].ConHost[AIActive[curAP].NoConHost-1]=curHost+1;
                tempFlag=0;
                for(int j=0; j< AIActive[HIActive[curHost].AssocAP-1].NoConHost; j++)
                {
                    if(tempFlag == 1) AIActive[HIActive[curHost].AssocAP-1].ConHost[j-1] = AIActive[HIActive[curHost].AssocAP-1].ConHost[j];
                    if(AIActive[HIActive[curHost].AssocAP-1].ConHost[j] == curHost+1) tempFlag = 1;
                }
                AIActive[HIActive[curHost].AssocAP-1].NoConHost--;
            }
                do
```

```
                    {
                        curAP = HIActive[curHost].ConAP[rand() % HIActive[curHost].NoConAP]-1;
                    }
while(curAP == (HIActive[curHost].AssocAP-1));

                    AIActive[curAP].NoConHost++;
                    AIActive[curAP].ConHost[AIActive[curAP].NoConHost-1]=curHost+1;
                    tempFlag=0;
                    for(int j=0; j< AIActive[HIActive[curHost].AssocAP-1].NoConHost; j++)
                    {
                        if(tempFlag == 1) AIActive[HIActive[curHost].AssocAP-1].ConHost[j-1] = AIActive[HIActive[curHost].AssocAP-1].ConHost[j];
                        if(AIActive[HIActive[curHost].AssocAP-1].ConHost[j] == curHost+1) tempFlag = 1;
                    }
                    AIActive[HIActive[curHost].AssocAP-1].NoConHost--;
                    HIActive[curHost].AssocAP = curAP+1;
                    HIActive[curHost].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[curAP][curHost];
                }
            }
            SaveSolution(AITempSol, AIActive, HITempSol, HIActive);
        }
}

void PostAssociationOptimization()
{

     int temp_i, temp_j, swapAP, ifSwapped, k, l;
     float currentMaxTxTime, newMaxTxTime;

     float oldSum, newSum;

    for(int i=0; i<NoAP; i++)
        {
        for(int j=0; j<NoHost; j++)
           printf("AP%d-H%d: %f¥t",i+1, j+1, HIActive[j].HostActiveRate/AP_HostLinkSpeed[i][j]);
        printf("¥n");
        }

      currentMaxTxTime = calculateMaxTxTimeAmongAPs();

      printf("¥n");
      for(int i=0; i<NoAP; i++) if(AIActive[i].ifActive == 1)
      printf("¥nAP%d->TxTime:%f¥t",i+1, calculateTxTimeForAP(i));
```

```
        ifSwapped = 1;
        do
        {
          ifSwapped = 0;
          for( int i=0; i<NoHost; i++)
              {
                  for( int j=0; j<NoHost; j++)
                  {
                      if( (i==j) || (HIActive[i].AssocAP == HIActive[j].AssocAP) ) continue;
                      oldSum = 0.0;
                      newSum = 0.0;

                      oldSum = ( HIActive[i].HostActiveRate / HIActive[i].AssocAP_HostLinkSpeed ) + ( HIActive[j].HostActiveRate / HIActive[j].AssocAP_HostLinkSpeed);
                      newSum = ( HIActive[i].HostActiveRate / AP_HostLinkSpeed[HIActive[j].AssocAP-1][i] ) + ( HIActive[j].HostActiveRate / AP_HostLinkSpeed[HIActive[i].AssocAP-1][j] );

                      if(newSum < oldSum)
                      {
                          printf("¥nPostAssociationOptimization here¥n");
                          temp_i = 0;
                          for(k=0; k<HIActive[i].NoConAP; k++)
                              if(HIActive[i].ConAP[k] == HIActive[j].AssocAP)
                              {
                                  temp_i = 1;
                                  break;
                              }
                          temp_j = 0;
                          for(k=0; k<HIActive[j].NoConAP; k++)
                              if(HIActive[j].ConAP[k] == HIActive[i].AssocAP)
                              {
                                  temp_j = 1;
                                  break;
                              }
                          if( (temp_i == 1) && (temp_j == 1))
                          {
                              printf("¥nSwapped here (H%d-AP%d , H%d-AP%d)¥n",i+1, HIActive[i].AssocAP, j+1, HIActive[j].AssocAP);
                              for(k=0; k<AIActive[HIActive[i].AssocAP-1].NoConHost; k++)
                                  if( (AIActive[HIActive[i].AssocAP-1].ConHost[k]) == (i+1))
                                  {
                                      AIActive[HIActive[i].AssocAP-1].ConHost[k] = j+1;
                                      break;
                                  }
```

```
                                for(l=0; l<AIActive[HIActive[j].AssocAP-1].NoConHost; l++)
                                    if( (AIActive[HIActive[j].AssocAP-1].ConHost[l]) == (j+1))
                                    {
                                        AIActive[HIActive[j].AssocAP-1].ConHost[l] = i+1;
                                        break;
                                    }
                                swapAP = HIActive[i].AssocAP;
                                HIActive[i].AssocAP = HIActive[j].AssocAP;
                                HIActive[i].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[HIActive[i].AssocAP-1][i];

                                HIActive[j].AssocAP = swapAP;
                                HIActive[j].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[HIActive[j].AssocAP-1][j];
                                newMaxTxTime = calculateMaxTxTimeAmongAPs();
                                printf("¥n");
                                printf("currentMaxTxTime=%f¥n",currentMaxTxTime);
                                for(int m=0; m<NoAP; m++) if(AIActive[m].ifActive == 1) printf("AP%d->TxTime:%f¥t",m+1, calculateTxTimeForAP(m));
                                printf("newMaxTxTime=%f¥n",newMaxTxTime);

                                if(newMaxTxTime <= currentMaxTxTime)
                                {
                                    currentMaxTxTime = newMaxTxTime;
                                    ifSwapped = 1;
                                }
                                else if(newMaxTxTime > currentMaxTxTime
                                {
                                    swapAP = HIActive[i].AssocAP;
                                    HIActive[i].AssocAP = HIActive[j].AssocAP;
                                    HIActive[i].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[HIActive[i].AssocAP-1][i];
                                    HIActive[j].AssocAP = swapAP;
                                    HIActive[j].AssocAP_HostLinkSpeed = AP_HostLinkSpeed[HIActive[j].AssocAP-1][j];
                                    AIActive[HIActive[i].AssocAP-1].ConHost[k] = i+1;
                                    AIActive[HIActive[j].AssocAP-1].ConHost[l] = j+1;
                                    printf("¥nRolled Back¥n");
                                }
                            }
                        }
                    }
                }
        }while(ifSwapped == 1);
}
```