Throughput Maximization in Wireless Data Transmission

Prateek Pandey

Throughput Maximization in Wireless Data Transmission

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Technology in Computer Science

by

Prateek Pandey
(Roll No: CS-1401)

under the guidance of

Dr. Sasthi C. Ghosh
Assistant Professor
Advanced Computing and Microelectronics Unit

Indian Statistical Institute
Kolkata 700 108

July, 2016

To my family, friends and guide

CERTIFICATE

This is to certify that the dissertation entitled **'Throughput Maximization in Wireless Data Transmission"** submitted by **Prateek Pandey** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Sasthi C. Ghosh
Advanced Computing and Microelectronics Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisor, *Dr. Sasthi C. Ghosh*, Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support.

Prateek Pandey
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

The increasing popularity of wireless local area network (WLAN) have dramatically increased the density of access points (APs). If the node density is high, it results in strong interference and hence poor network performance. The channel assignment problem under this type of highly interfering environment is a challenging task. A lot of research has already been done to maximize the throughput of WLAN with limited channel resources under this environment. Because the problem of channel assignment is NP-hard, many researchers have given heuristics to improve the throughput with minimum network interference. Hoque et. al. [10] presented an algorithm which tries to activate as many links as possible at every slot to maximize the throughput. Every link can not be activated due to the limitation of number of non-overlapping channels (NOCs) in WLAN. If we always try to activate maximum number of links in every slot, some nodes may always suffer depending on their positions with respect to the positions of the APs. This report presents a greedy algorithm which improves the average throughput of the network and minimizes the network interference while taking care of user fairness. We propose an algorithm which uses the concept of priority scheduling in which each APs are assigning some priority value to each STA and we use these priority values for channel assignment and AP-STA association. We have shown that our algorithm produces better throughput and also schedules the user traffic earlier than the algorithm based on maximizing the number of links.

Keywords: *IEEE802.11b clients, Non-overlapping channels, Throughput Maximization, Priority Scheduling*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

There has been a lot of changes in physical layer technologies in the recent past. In spite of these modifications, today's wireless local area network (LAN) can not match its wired counterparts in providing sustained bandwidth [17]. The actual throughput obtained by the applications is reduced too much with the inclusion of overheads like 802.11 headers, errors and MAC contention. As the distance between nodes increases, the transmission rates decrease rapidly.

In most of the real world applications, everyone is using WLAN because of its popularity. The increasing popularity of WLANs have dramatically increased the density of APs. High node density may result strong interference and hence may produce poor network performance [5]. In wireless mesh networks, routers form the backbone of the network and have very less mobility. Each router has some transmission range in which only it can transmit the data. It also has an interference range within which its transmission interfere the others. Interference range is often more than the transmission range.

According to Cui et.al. [5], when two nearby wireless links communicate on the same frequency band (channel), they can't transmit the data simultaneously because of the interference. As a result the throughput of the system may be decreased dramatically because of interference from other links. Hence two interfering nodes can simultaneously transmit the data without interfering each other only if their assigned channels are non-overlapping. But the number of non-overlapping or orthogonal channels (NOC) is very limited, so the interference can not be completely eliminated in reality. Hence to minimize the interference, channel should be assigned very carefully.

In channel assignment problem, we assign channels to radio interfaces and our goal is always to achieve maximum channel utilization and to minimize the total interference in the network. According to Sridhar et.al. [17] channel assignment can be done in many ways. One of the approach is "Dynamic Channel Assignment Scheme" in which the channel is allocated on demand, i.e., on a per-packet basis. But this scheme requires frequent channel switching within each nodes which takes order of few milliseconds time and hence it cause delays of the order of few milliseconds. Because of switching of channels frequently this scheme needs high speed synchronization among the nodes while sending or receiving the data over a particular channel which is in reality very difficult to achieve without changing the 802.11 MAC.

Second approach is "Static Channel Assignment Scheme" which is done on the premise of ease of adaptability in commodity 802.11 hardware. These schemes can easily be extended to semi-dynamic by refreshing the channel assignment at regular fixed time interval.

Third type of approach is hybrid approach which uses semi dynamic and/or static channel assignment scheme for fixed interfaces and dynamic channel assignment scheme for switchable interfaces. Our approach is kind of dynamic approach in which channels are switched/changed.

1.2 Objective

In stadium like environment APs are usually placed very densely. To improve the network capacity and overall throughput in these type of environment, often dense deployment of APs is required. The channel assignment problem in this type of environment is a challenging task. Because the problem of channel assignment in wireless networks is proven to be NP-hard, one can not guaranteed give an optimal algorithm. Many researchers have already given heuristics to improve the throughput of the network. Many efforts have been devoted to maximize the throughput. Researchers have tried different approaches and presented some heuristics which improves the throughput, but in reality throughput of a system depends on many factors like interference of the network, placement of APs and STAs, load on the APs, number of slots required for the transmission, user's fairness etc. Different algorithms take care of different factors and gave their heuristic. They calculate throughput based on those selected factors.

Hoque et.al. [10] presented an algorithm in which they have taken number of links as a baseline to measure the throughput. They maximize the number of links and they showed that if we use partially overlapping channels (POCs), the number of links can be increased, hence the throughput of the system can be improved. In his work, if we consider that each node has some data which it wants to transmit to some other node then according to this algorithm, every link may not be activated due to limitation of number of non-overlapping channels (NOCs). So if we follow his algorithm, then some nodes may always suffer, they may never get the chance to transmit their data. Hence this algorithm is ignoring the fairness of user.

The main objective of this report is to present a greedy algorithm for scheduling the user traffic in such a way that it improves the throughput of the network and minimizes the network interference while taking care of user fairness. We propose an algorithm which is based on the concept of priority scheduling in which each AP

is assigning some priority value to each STA and we use the priority values for the channel assignment and association of STAs with APs. Through simulations we have shown that our proposed algorithm produces better average throughput than the algorithm based on maximizing the number of links.

Chapter 2

Previous Literature and Benchmarks

Cui et.al. [5] proposed a novel Interference Factor I_c that captures the degree of interference between two channels at different positions. This interference factor is employed to formulate an interference minimization problem for partially overlapping channel assignment to maximize the aggregated network throughput. Also, an approximate algorithms, MICA, to tackle the optimization problem via relaxation and rounding, was proposed by them. MICA which stands for Minimum Interference for Channel Allocation, minimizes the sum of the weighted interference.

Mishra et.al. [14] presented the first attempt to model partial overlap between channels in a very systematic manner. Through the model, they illustrated that the use of partially overlapped channels is not always harmful. In fact, a careful use of some partially overlapped channels can often lead to significant improvements in spectrum utilization and application performance. They demonstrated this through analysis as well as through detailed application-level and MAC-level measurements. Additionally, they illustrated the benefits of their developed model by using it to directly enhance the performance of two previously proposed channel assignment algorithms — one in the context of wireless LANs and the other in the context of multi-hop wireless mesh networks.

Mishra et.al. [13] defined specific mechanisms that can transform partially overlapped channels into an advantage, instead of a peril. They constructed simple analytical and empirical models of the interference occurring in IEEE 802.11 networks, and illustrated two scenarios where the interference can be exploited. First, they applied partially overlapping channels to improve spatial channel re-use in WLANs. Second, they leveraged such channels to enable nodes with a single radio interface to communicate more efficiently with their peers in 802.11 ad-hoc mode potentially using multi-hop paths. They evaluated both capabilities through test bed measurements.

Feng et.al. [8] presented mathematical models to compute the capacity improvement ratio comparing POC-based designs with traditional designs and address those issues in the existing work. They proposed two separate optimization models for one hop and multi-hop networks for POC-based design. They introduced the orthogonality constraint in their mathematical formulation.

Ding et.al. [7] proposed an extension to the traditional conflict graph model, weighted conflict graph, to model the interference between wireless links more accurately. Based on this model, they first presented a greedy algorithm for partially overlapping channel assignment, and then proposed a novel genetic algorithm, which had the potential to obtain better solutions.

Cui et.al. [12] proposed novel channel allocation and link scheduling algorithms in the MAC layer to enhance network performance. Due to different traffic characteristics in multi-hop WMNs compared to those in one-hop 802.11 networks, they performed their optimization based on end-to-end flow requirement, instead of the sum of link capacity. In addition, they discussed other factors affecting the performance of POC, including topology, node density, and distribution.

The obtained throughput depends on how user traffic is being scheduled. If we always try to activate number of links using both NOCs and POCs, some users may always suffer from degraded throughput because of their disadvantageous position. If the APs which are getting higher data rate are being scheduled first, the throughput in initial slots will be very high but in later slots it will be very less and hence the aggregate throughput of the network will be degraded. In this report, we have scheduled the user traffic in such a way that the throughput in a slot can be less than the throughput of a slot in future but the overall aggregate throughput will be high. In other words, we are not maximizing the throughput of individual slots, instead we are maximizing the throughput over a span of slots. Hence each user gets fair access to the channel resource. Accordingly we propose an algorithm based on priority scheduling to schedule the user traffic. Through simulations we have shown that our proposed algorithm produces better aggregate throughput.

Chapter 3

Preliminaries

In this chapter we explain few terminologies, which is being and will be used further in this report.

3.1 Channel

In telecommunications and computer networking, a communication channel or channel, refers either to a physical transmission medium such as a wire, or to a logical connection over a multiplexed medium such as a radio channel. A channel is used to convey an information signal, for example a digital bit stream, from one or several senders (or transmitters) to one or several receivers. A channel has a certain capacity for transmitting information, often measured by its bandwidth in "Hz" or its data rate in "bits per second" (bps).

3.2 Interference

In telecommunications, interference is anything which modifies, or disrupts a signal as it travels along a channel between a source (sender) and a destination (receiver). The term typically refers to the addition of unwanted signals to a useful signal.

The region in which a transmitter can send its signal is called transmission range. The region within which two or more transmissions interfere each other is called interference range. Interference range is often more than transmission range.

In figure 3.1, T_1 and T_2 are the transceivers. In theoretical study, the transmission range is considered to be circular for the ease of calculations. Both transceivers have a circular transmission range and the dark area which is overlapping is the interference region.
According to Hoque et.al. [10], interference is categorized in three types - co-channel interference, self channel interference, adjacent channel interference (ACI).

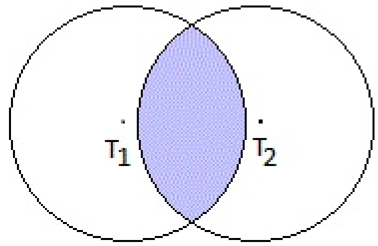

Figure 3.1: Interferencce region

3.2.1 Co-channel Interference

Interference generated due to transmission on the same channel concurrently. Hence if two nodes interfere each other, they can not use same channel at the same time.

3.2.2 Self Channel Interference

The biggest challenge is to remove the self interference. If there is one sender and two receivers then transmission on both the links is not possible at the same time with one channel due to self interference. Hence two channels are required to make the transmission possible simultaneously.

3.2.3 Adjacent Channel Interference

This is the type of interference perceived by a node (say 'x') when a different node (say 'y') in its interference range is transmitting some data and using the channel which is partially overlapping to the channel assigned to 'x'.

3.3 Interference Factor

Its a measure of effective spectral overlapping between two channels. Interference factor accounts for the amount of channel overlapping between two interfering APs. According to Chieochan et.al. in [3], the overlapping channel interference factor indicates how much two interfering channels are overlapped in frequency.

3.4 Interference Range

The interference range of a transmission depends on the transmission power of the transmitter. Therefore, the amount of spatial re-use of the same channel depends on the choice of transmission power. In channel allocation schemes, that only use orthogonal channels, it is often unavoidable to assign the same channel. The co-channel interference restricts the nodes from parallel communications [8]. POCs also interfere each other but according to Feng et.al. in [8], the received signal power from a sending node is lower if the receiving node uses a POC compared to using the same channel as the sender. Hence, the interference range of POCs is often much smaller than the typical co-channel interference range. Such reduced interference range of POCs enables more parallel transmissions, essentially increasing the capacity of the network as discovered in [12] [14] [16].

For calculating interference ranges for different channel separations, Zhenhua Feng and Yaling Yang [8] had performed experiment several times and gave the result as shown below in table 3.1. We are directly using their results in our work.

δ	0	1	2	3	4	5
IR(δ)	13.26	9.21	7.59	4.69	3.84	0

Table 3.1: Interference Range for the different channel separations

Here IR(δ) refers to the interference range for a channel separation of δ, where $\delta = |i - j|$.

3.5 Priority Scheduling

In terms of operating system, each process in the system is given a priority value by the operating system, and the scheduling is done according to the priority of each process. Priority is calculated based on memory requirements, time requirements or any other resource requirements.

Process with highest priority is to be executed first and so on i.e. among all the ready-to-run processes, a higher priority process gets CPU first whereas lower priority process waits. Priority scheduling is a form of preemptive scheduling where priority is the basic of preemption.

The preemptive scheduler has a clock interrupt task that can provide the scheduler with options to switch after the task has had a given period to execute the time slice. This scheduling system has the advantage of making sure that no task hogs the processor for any time longer than the time slice. However, this scheduling scheme is vulnerable to process or thread lockout: since priority is always given to higher-priority tasks, the lower-priority tasks could wait an indefinite amount of time. One common method of arbitrating this situation is aging which gradually increments the priority of waiting processes at each time slice, ensuring that they will all eventually execute. Most Real-time operating systems (RTOSs) have preemptive schedulers.

3.6 Priority Queue

A priority queue is an abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according to their order in the queue.

While priority queues are often implemented with heaps, they are conceptually distinct from heaps. A priority queue is an abstract concept like a "list" or a "map", just as a list can be implemented with a linked list or an array, a priority queue can be implemented with a heap or a variety of other methods such as an unordered array.

Element can be integer, float, string, structure etc. Each element is associated with a priority value. In data structure, queue works on the concept of "First In First Out", i.e., the element which was inserted (enqueue) first, will come out first. But in priority queue the element with highest priority will come out no matter at which place it was inserted.

3.7 Graph Coloring Problem

In graph theory, graph coloring is a special case of graph labeling. it is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color. This is called a vertex coloring.

Graph coloring is computationally hard. It is NP-complete to decide if a given graph admits a k-coloring for a given k except for the cases $k = 1$ and $k = 2$. In particular, it is NP-hard to compute the chromatic number [9]. The 3-coloring problem remains NP-complete even on planar graphs of degree 4 [6]. However, k-coloring of a planar graph is in P, for every $k > 3$, since every planar graph has a 4-coloring (and thus, also a k-coloring, for every $k \geq 4$).

Chapter 4

Proposed Model

In the era of Internet, WLANs are becoming more popular day by day. Most of the institutes, organizations are deploying the WLANs in its premises. With the increased popularity and deployment of WLANs, managing the wireless spectrum efficiently is becoming increasingly important and necessary. In this report we are focusing on channel assignment which is a specific resource sharing problem in the context of 802.11-based WLANs.

IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing WLAN computer communication in the 900 MHz and 2.4, 3.6, 5, and 60 GHz frequency bands. They are created and maintained by the Institute of Electrical and Electronics Engineers (IEEE) LAN/MAN Standards Committee (IEEE 802) [18].

If we consider an inside of a building environment, there are multiple APs which are functional. Each AP operates on an administrator- specified channel. In 802.11 WLANs, the wireless card (WI-FI card) of a user (STA) scans the wireless medium to identify the access points and associates with the AP which have strongest signal [13]. For minimizing the interference between different nearly placed APs, administrators conduct detailed Radio Frequency (RF) site surveys, often using spectrum analyzers, before setting up APs within the building and assigning specific channels to them [11].

For using WLAN, there is no need of purchasing the license of the frequency spectrum. Generally some regulatory body, e.g., the Federal Communications Commission in the USA allots the license of the frequency spectrum. Each WLAN standard (802.11/a/b/g) defines a fixed number of channels for use by APs and STAs (mobile users) [13]. In our work, we are considering IEEE 802.11b clients. The 802.11b standard has a maximum raw data rate of 11 Mbit/s, and uses the same media access method defined in the original standard.

There are 14 frequency channels which comes under 802.11b standard of which only 1 through 11 are generally used. Channels 12 and 13 are not normally used in order to avoid any potential interference in the adjacent restricted frequency band, 2,483.5–2,500.0 MHz [19]. Here an important point to note regarding these channels is

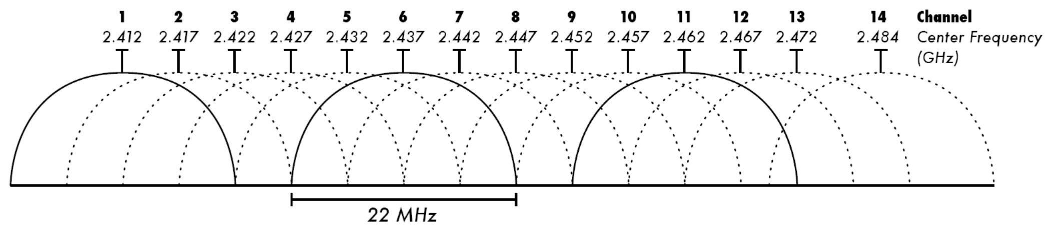

Figure 4.1: Graphical representation of Wi-Fi channels in the 2.4 GHz band

that the channel actually represents the center frequency that the transceiver within the radio and AP uses (e.g., channel 1 is used for 2.412 GHz and channel 2 is for 2.417 GHz). The separation between two consecutive center frequencies is only 5 MHz, and an 802.11b signal occupies approximately 30 MHz of the frequency spectrum. The signal falls within about 15 MHz of each side of the center frequency. As a result, an 802.11b signal on any channel overlaps with several adjacent channel frequencies causing interference (also known as adjacent channel interference). This leaves only three channels as shown in figure 4.1 - channel 1, 6, and 11 [1] that can be used simultaneously without causing interference [13] and these channels are commonly known as orthogonal or non-overlapping channels (NOCs).

4.1 Setup

In our model, APs and STAs are placed uniformly at random as shown in figure 4.2 where “circles are APs” and “stars” are STAs. For the purpose of time complexity we have assumed k APs and n STAs.

We have uniformly assigned the data to each of the STA. Each station has some data, i.e., station s_i has d_i kilo bytes of data as shown in table 4.1.

STA	s_1	s_2	s_3	..	..	..	..	s_n
Data (KB)	d_1	d_2	d_3	..	..	..	..	d_n

Table 4.1: STA s_i has d_i KBytes of data

We have assumed standard frame size (F) which is 1 KB (1024 Bytes) as taken by [4]. We calculate the number of frames an STA need to send , i.e., $\lfloor d_i/F \rfloor$ as shown in table 4.2.

STA	s_1	s_2	s_3	..	..	..	..	s_n
Frames	f_1	f_2	f_3	..	..	..	..	f_n

Table 4.2: STA s_i has f_i frames

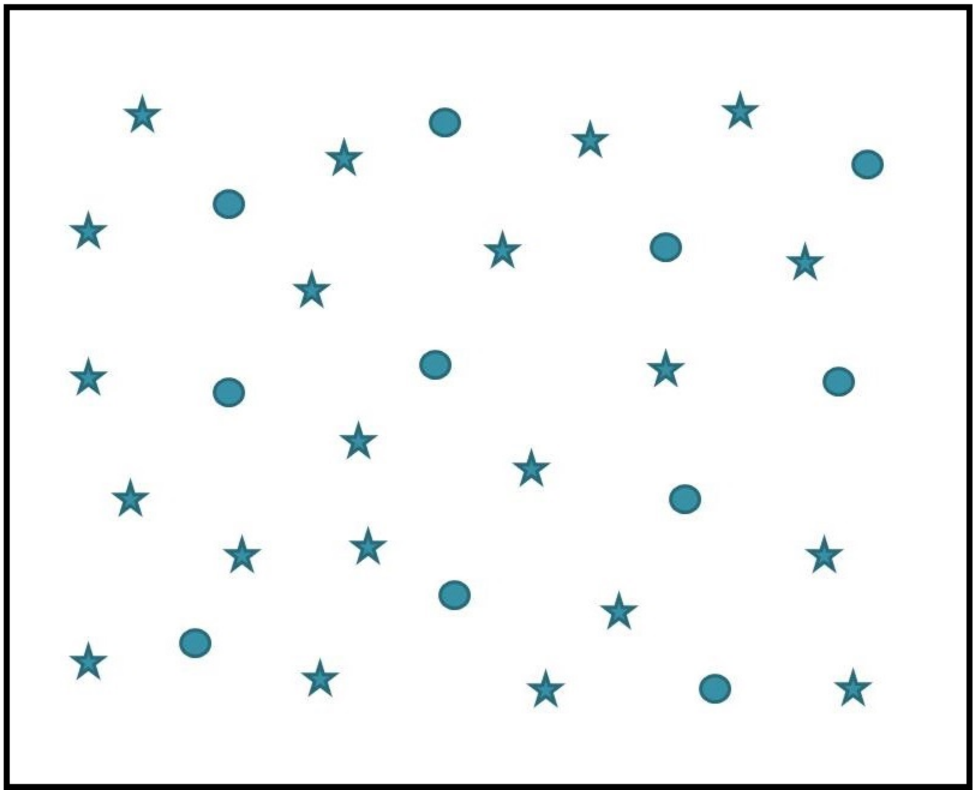

Figure 4.2: Setup

As shown in figure 4.2, we know the co-ordinates of STAs and APs, so we calculate the distance of an STA from an AP. We have used a simple wireless channel model in which, the data rate obtained by the stations depends only on the distance to the AP. Adopting the values which are commonly advertised by 802.11b vendors, we assume that the data rate (bit rate) of stations within 50 meters from AP is 11 Mbps, within 80 meters is 5.5 Mbps, within 120 meters is 2 Mbps, and 1 Mbps within 150 meters, respectively [2]. The maximum transmission range of an AP is 150 meters, outside this range we are assuming the data rate is zero.

Now for each STA, we have a vector/array of data rates of size k as shown in table 4.3. It is quite obvious that the STA will get higher data rate from nearer AP and lesser data rate from AP which is far from the STA.

STA/AP	AP_1	AP_2	AP_3	..	..	..	..	AP_k
s_1	r_{11}	r_{12}	r_{13}	..	..	..	..	r_{1k}
s_2	r_{21}	r_{22}	r_{23}	..	..	..	..	r_{2k}
s_3	r_{31}	r_{32}	r_{33}	..	..	..	..	r_{3k}
..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..
s_n	r_{n1}	r_{n2}	r_{n3}	..	..	..	..	r_{nk}

Table 4.3: STA s_i is getting r_{ij} Mbps data rate from AP_j

We have assumed a fixed slot length (T) of 10 milliseconds. Now if station s_i is associated with AP_j, it is getting r_{ij} data rate. So it can send $r_{ij} * T$ KB data, i.e., $\lfloor(r_{ij} * T)/F\rfloor$ number of frames in each slot. Hence station s_i needs $\lceil f_i/((r_{ij} * T)/F)\rceil$ number of slots to transmit its data while associated with AP_j. For each STA, we calculated this and also the minimum number of slots it needs. So for each STA, now we have an array of total number of slots the STA needs to transmit its data as shown in table 4.4.

STA/AP	AP_1	AP_2	AP_3	..	..	..	..	AP_k
s_1	t_{11}	t_{12}	t_{13}	..	..	..	..	t_{1k}
s_2	t_{21}	t_{22}	t_{23}	..	..	..	..	t_{2k}
s_3	t_{31}	t_{32}	t_{33}	..	..	..	..	r_{3k}
..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..
s_n	t_{n1}	t_{n2}	t_{n3}	..	..	..	..	t_{nk}

Table 4.4: STA s_i needs t_{ij} slots when associated with AP_j

Our problem is to maximize the aggregate throughput by scheduling the user traffic while respecting the user's fairness.

4.2 Interference Model

The throughput of a wireless network heavily depends on the interference. In a dense environment, if the number of APs are very high, and if we activate all the APs at the same time, interference may increase drastically. Hence the throughput of the network will be degraded. Hence, we can not activated all the APs at the same time due to interference. To model the interference in our work, we have used the concept of interference graph and interference matrix.

4.2.1 Interference Graph

If we consider AP as nodes/vertices, and two nodes have an edge between them if they are in the interference range of each other, we call this graph as interference graph.

4.2.2 Interference Matrix

The adjacency matrix of this graph is called interference matrix (IM). We create the Interference Matrix of size $k \times k$ as shown in table 4.5, where k is the number of APs. The ij^{th} element of the matrix is 1, if AP_i and AP_j interfere, otherwise 0 as shown by equation 4.2.2. Dense deployment of APs is required to improve the network capacity and throughput. In this dense environment, due to co-channel interference, we may not activate all the APs at the same time. Hence careful selection of AP is required for better throughput. We will use interference matrix while selecting APs for activation.

$$IM_{ij} = \begin{cases} 1 & \text{if } d(AP_i, AP_j) \leq IR \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where d(AP_i, AP_j) is the Euclidean distance between AP_i and AP_j and IR is the interference range of APs.

	AP_1	AP_2	AP_3	..	..	AP_k
AP_1	1	..	..	..	..	..
AP_2	..	1	..	..	..	..
AP_3	..	..	1	..	..	..
..	..	..	..	..	..	..
..	..	..	..	..	..	..
AP_k	..	..	..	..	..	1

Table 4.5: Interference Matrix

We have used the concept of priority queue (explained in chapter 3), i.e., each AP has a priority queue of size n. Each AP has assigned some priority to each STA. If an AP gets activated, it will select the STA which is having the highest priority and after completion of a slot, the AP will modify the priority of the STA which it has served in this slot. This way we are ensuring that no user will suffer from being acquiring required slots.

4.3 Throughput Calculation

We are maximizing the throughput of the network. The throughput of a particular slot is basically how much traffic is being scheduled in that slot. Our approach schedules the traffic earlier than the traditional approach, i.e., the number of slots for both the approaches are different. Hence, for comparison, we calculate throughput per slot and and take the logarithm of this value. We execute this algorithm n times on the same number of APs and STAs and take the average throughput to mitigate the effect of randomization. Let, in i_{th} iteration x_i KB be the user traffic scheduled in n_i number of slots. Hence, throughput per slot is

$$Throughputperslot = x_i/n_i \tag{4.2}$$

The average throughput is represented as equation 4.2.

$$AverageThroughput = (1/n)\sum \log(x_i/n_i) \quad \forall i \in 1, 2 \ldots n \tag{4.3}$$

where n is the total number of slots. In the chapter 7, we have shown that our algorithm produces better average throughput and per slot throughput.

Chapter 5

The Proposed Algorithm

In the dense deployment of APs, if we consider this as a graph problem where nodes are the APs and if two APs interfere each other, we connect those nodes by an edge. The problem of channel assignment to APs is same as "graph coloring" problem in the graph (explained in chapter 3). As already discussed, the problem of graph coloring is NP complete. So we we have tried to present a greedy approach for the activation of APs and association of AP-STA.

In our algorithm, there are 4 sections - priority assignment, selection of the APs, i.e., channel assignment, selection of the STA for each activated AP and updation of priority for $(k+1)^{th}$ slot of the STA which was associated with some AP in k^{th} slot. Each AP has assigned a priority to each STA which is being calculated based on data rate STA is getting from the AP. We are using this priority in the selection of APs and STAs.

5.1 Priority Assignment

In our work, we have incorporated the concept of priority scheduling. As discussed in chapter 3, in priority scheduling the central processing unit (CPU) assigns some priority based on some factors like memory requirement, resource requirement etc. to each process appears in ready queue.

In our work, AP can be considered as CPU because CPU is responsible for managing processes and in our work AP is responsible for managing stations. Process needs CPU to perform its task and STA needs AP to transmit its data. in this way, we thought of using this policy in our work. Hence AP assigns some priority to each STA which wants to transmit the data.

To assign the priority to each station, we calculate the data rate a station is getting from each access point. For the calculation of data rate, we use Euclidean distance as the distance measure and based on the distance of STA from AP, we calculate the data rate for each STA. Then for each STA, we calculate the number of slots it needs for transmitting the data if it is associated with an AP and we call the priority as the number of slots it needs for transmission as shown by equation 5.1. Hence the priority of a station s_i for AP_j is given as equation 5.1

$$pr_{ij} = t_{ij} \tag{5.1}$$

where t_{ij} is the number of slots s_i needs to transmit its complete data while associated with AP_j.

Now to increase our accuracy of the above method, we went deeper and analyzed the two cases, one was if there is one STA which is getting some data rate from an AP which is the highest (in our case 11 Mbps) and one STA which is getting moderate data rate from two or more than two APs. So in this situation which STA should have more priority, one which have one best option (let say STA which is getting 11 Mbps data rate from an AP) or other which have so many good options but none of the option is the best (let say STA which is not getting 11Mbps data rate from any AP but its getting 5.5Mbps or 2 Mbps rate from many APs). So we modified the priority to penalize the STA which is having only one best option because there was a big difference between the priorities of both STAs. Hence we reduced the difference between both the priority values. For this, we calculated the minimum and maximum number of slots for each STA and we change the priority value as given in equation 5.2.

$$pr_{ij} = t_{ij} + max(t_{ij}) - min(t_{ij}) \quad \forall j \in AP \tag{5.2}$$

where t_{ij} is the number of slots s_i needs to transmit its complete data while associated with AP_j, $max(t_{ij})$ and $min(t_{ij})$ is the maximum and minimum number of slots it needs while associated with any AP from AP_0 to AP_k respectively. But later we found that initially the algorithm is giving good throughput but in later slots this modification has reverse effect. So we changed our method back to the previous one, i.e., method 5.1.
The algorithm for priority assignment is given as algorithm 1.

Algorithm 1: Priority Assignment

```
for ∀s_i ∈ {s_0, s_1, ..., s_{k-1}} do
    for ∀AP_i ∈ {AP_0, AP_1, ..., AP_{k-1}} do
        pr_ij = t_ij
```

5.2 Selection of APs

To select which set of APs is more important, some criteria is must. So we are using priority as a criteria for AP selection. For each AP, we have calculated the priority of each STA by algorithm 1 and put these values in a priority queue of that particular AP with the STA index. Now each AP has a priority queue of size $1 \times n$ where n

is the number of stations. The queue shown in table 5.1 contains $(STA, priority)$ tuples, i.e., each STA with its priority according to this AP.

$(s_1,\, Pr_{11})$	$(s_2,\, Pr_{12})$	$(s_3,\, Pr_{13})$	..	..	..	$(s_n,\, Pr_{1n})$

Table 5.1: STA s_i has priority Pr_{ji} according to AP_j

We put the priority queues of each AP in a list and we generate a "priority matrix" of size $k \times n$, in which each row is basically a priority queue of an AP as shown in table 5.2.

$(s_1,\, Pr_{11})$	$(s_2,\, Pr_{12})$	$(s_3,\, Pr_{13})$	..	..	..	$(s_n,\, Pr_{1n})$
$(s_1,\, Pr_{21})$	$(s_2,\, Pr_{22})$	$(s_3,\, Pr_{23})$	..	..	..	$(s_n,\, Pr_{2n})$
$(s_1,\, Pr_{31})$	$(s_2,\, Pr_{32})$	$(s_3,\, Pr_{33})$	..	..	..	$(s_n,\, Pr_{3n})$
..	..	..	..	..	..	..
..	..	..	..	..	..	..
..	..	..	..	..	..	..
$(s_1,\, Pr_{k1})$	$(s_2,\, Pr_{k2})$	$(s_3,\, Pr_{k3})$	..	..	..	$(s_n,\, Pr_{kn})$

Table 5.2: Priority Matrix (PM)

In the algorithm for the selection of APs for activation (algorithm 2), we use priority matrix and interference matrix. The algorithm is given as algorithm 2.

The algorithm returns two arrays "isActivated[ ]" and "allocatedChannel[ ]" both of size k and have values as shown in equation 5.3 and 5.4 respectively.

$$isActivated_i = \begin{cases} 1 & \text{if } AP_i \text{ is activated} \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

$$allocatedChannel_i = \begin{cases} \text{channel among 1, 6 or 11} & \text{if } AP_i \text{ is activated} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

Once we are done with the activation of APs, we select STAs to associate with the activated APs. In the dense environment of APs, there are many STAs in surroundings of an AP. Hence, there can be many STAs which can be associated eith an AP. Different STAs get different data rate from different APs based on their distance from the APs. Hence, we need to carefully select STAs also in each slot.

Algorithm 2: Selection of APs

Input: $isActivated = [\ ]_k$, Priority Matrix, Interference Matrix
Output: $isActivated$

for $\forall AP_i \in \{AP_0, AP_1, \ldots, AP_{k-1}\}$ **do**
    For each AP, find the highest priority in its priority queue and put it in an array (let say $importanceOfAP$);
Find which AP has the highest importance i.e. AP with the highest priority among the highest priorities of each APs (let say AP_i);
Activate AP_i and set $isActivated_i = 1$ and assign channel 1 to AP_i by setting $allocatedChannel_i = 1$;
for $\forall unactivatedAPs$ **do**
    Find next AP which has the highest importance (let say AP_j);
    Find the APs which interfere AP_j i.e. adjacent APs of AP_j in interference graph (use interference matrix);
    for $\forall AP_k \in adj(AP_j)$ **do**
        if AP_k *is activated* **then**
            check which channel is allocated to it.;
            if *a channel among 1, 6 and 11 is free* **then**
                set $isActivated_j = 1$, activate AP_j by allocating a channel (if more than one channels are free allocate the minimum among them) i.e. $allocatedChannel_i = min(freechannelamong1, 6or11)$;
            else
                set $isActivated_j = 0$;
Return the array $isActivated$;

5.3 Selection of STAs

At this step, we have known the APs which are activated among all the k APs. If we see in the Figure 4.2, an AP has a lot of STAs in its surroundings, so if it is activated, it can select any of the STA and same is for STAs. An STA has so many APs near to it so it can also be associated with any of the APs. Hence selection of STA is also a factor which affects the overall throughput, hence it should also be taken into account for improving the overall throughout. Each STA has a priority value which is being assigned by APs. If some STA is nearer to an AP, it will get comparatively higher data rate, so it needs less number of slots to transmit its data, hence intuitively this STA should get higher priority for that particular AP.

For the selection of STAs also, we are using “priority value” as the selection criteria. For all the activated APs, we are choosing those STAs which is in need for a slot for transmitting its data. In practical many scenarios are possible like one STA has same priority for more than one AP or one AP is giving same priority to more than one STA. In our work we have taken care of every case very carefully. The algorithm for the selection of STAs for association with activated APs is using two arrays. one is "APselectsSTA" of size $2 \times k$ as shown in table 5.3 and it takes value as shown in equation 5.5.

AP	AP_1	AP_2	AP_3	..	..	AP_k
STA	s_{i_1}	s_{i_2}	s_{i_3}	..	..	s_{i_k}
Priority	$Pr_{i_1 1}$	$Pr_{i_2 2}$	$Pr_{i_3 3}$	..	..	$Pr_{i_k k}$

Table 5.3: AP_j is serving s_{i_j} having priority $Pr_{i_j j}$

$$APselectsSTA_j = \begin{cases} \begin{cases} s_i \\ Pr_{ij} \end{cases} & \text{if } AP_j \text{ selects } s_i \\ \begin{cases} -1 \\ -1 \end{cases} & \text{otherwise} \end{cases} \tag{5.5}$$

And the second array is "availabilityOfSTA" of size $2 \times n$ as shown in table 5.4 and it takes value as shown in equation 5.6.. This array is used to check whether an STA is available for selection or it has already been selected by some other AP and if it is selected by some AP, 0th index tells the serving AP and 1st index, priority of the STA with which it is being served.

STA	s_1	s_2	s_3	..	..	s_n
AP	AP_{i_1}	AP_{i_2}	AP_{i_3}	..	..	i_k
Priority	Pr_{1i_1}	Pr_{2i_2}	Pr_{3i_3}	..	..	Pr_{ki_k}

Table 5.4: STA s_i is being served by AP_j

$$availabilityOfSTA_i = \begin{cases} \begin{cases} AP_j \\ Pr_{ij} \end{cases} & \text{if } s_i \text{ is served by } AP_j \\ \begin{cases} -1 \\ -1 \end{cases} & \text{otherwise} \end{cases} \tag{5.6}$$

The algorithm for the selection of STAs for association with activated APs is given as algorithm 3.

Algorithm 3: Selection of STAs

Input: $isActivated = [\ ]_k$, Priority Matrix

Output: $APselectsSTA$

1 **for** $\forall AP_j \in \{AP_0, AP_1, \ldots, AP_{k-1}\}$ **do**

2   **if** $AP_j is activated i.e. isActivated_j = 1$ **then**

        find the highest priority station from its priority queue, make a tuple $(priority, s_i, AP_j)$ and put it in a list (let say $importanceOfSTA$);

3   **else**

        put an invalid tuple $(maxNumber, -1, AP_j)$ in the list;

4 sort the list in ascending order. By sorting, the STAs will be arranged in decreasing order of its priority;

5 take the first tuple of this list and check

6 **if** *its valid tuple i.e.* $importanceOfSTA(0)(1) \neq -1$ **then**

    if the tuple is (Pr_{ij}, s_i, AP_j), associate s_i with AP_j and set station s_i is not available by setting,

    $availabilityOfSTA(0)_i = AP_j$,

    $availabilityOfSTA(1)_i = Pr_{ij}$,

    $APselectsSTA(0)_j = s_i$,

    $APselectsSTA(1)_j = Pr_{ij}$;

7 **for** $\forall validtuple \in importanceOfSTA$ **do**

8   **if** *the tuple is* (Pr_{ij}, s_i, AP_j)*, and if* s_i *is unallocated,* **then**

        associate s_i with AP_j and set station s_i is not available.;

9   **else**

        check which AP is serving s_i, let say AP_k with priority Pr_{ik}, then check;

10   **if** AP_j *has lower priority* **then**

            remove s_i from the current position of $importanceOfSTA$, find next highest priority station from the priority queue of AP_j, place this STA at its correct location in $importanceOfSTA$ so that this list remains sorted and continue;

11   **else**

            if AP_j has same priority then take stations with next highest priority from the priority queue of both of the APs i.e. AP_i and AP_j

            compare the priorities of both stations and associate s_i with the AP which has lower next highest priority and if needed replace the other one with next highest priority station and continue;

5.4 Priority Updation

The third step gives an association of STAs with the activated APs. After completion of each slot, the priority of each associated STA is being updated by priority updation algorithm which is given as algorithm 4.

Algorithm 4: Priority Updation

1 calculate the data transmitted in current slot and the residual demand of each station;
2 **for** $\forall AP_j \in \{AP_0, AP_1, \ldots, AP_{k-1}\}$ **do**
    if AP_j is activated, find the STA associated by this AP (let say s_i);
    calculate the number of frames s_i can send while being associated with AP_j;
    calculate the residual demand of s_i;
3 If the residual demand of any STA becomes zero, remove the station from the priority queue of all the APs;
4 update the priority of STA which have transmitted data in the current slot by adding min slots that STA needs, i.e., $pr(s_i) = pr(s_i) + minslots$;

5.5 Overall Algorithm

The 4 steps are already explained above. The overall algorithm is given as algorithm 5.

Algorithm 5: Overall Algorithm

1 for each AP, calculate the initial priority of each STA using algorithm 1;
2 Repeat step 3 to 4 until the residual demand of all the STAs becomes zero;
3 select which set of APs can be activated using algorithm 2;
4 select which set of STAs should be associated with the activated APs using algorithm 3;
5 after completion of the current slot, update the priority of each associated STAs using algorithm 4;

Chapter 6

Complexity of the Algorithm

In our algorithm, for time complexity purpose, we have taken n STAs and k APs. The algorithm is devided in 4 sections in which algorithm 1 to algorithm 4 are executing continuously until all the STAs have transmitted its data. Time complexity of each algorithm is explained below.

6.1 Time Complexity of Priority Assignment

For each AP, we calculate priority of each STA. For calculating the priority value, we calculate the number of slots required from each AP. There are k APs, hence for one STA it takes $O(k)$ time. There are n STAs, hence for n STAs, it takes $O(kn)$ time. Hence the time complexity of the algorithm 1 is $O(kn)$.

6.2 Time Complexity of AP Selection

In the algorithm 2 (Selection of APs), we have created a priority queue for each AP. For creating priority queue of one AP with n stations takes $O(n)$ time. For k APs, it takes $O(kn)$ time. Now we are creating a priority queue (importanceOfAP) and putting the highest priorities of each AP. This takes $O(k \log k)$ time (instead we can simply put the highest priorities of all the APs and sort, but both takes the same time). Now we scan this sorted list linearly and for each AP, we check if it is interfering with other activated AP or not. If its interfering then We look for the available channel and allocate the channel if available. It takes $O(k^2)$ time. Hence the algorithm 2 takes $O(k^2)$ time.

6.3 Time Complexity of STA Selection

While associating STAs with APs, we have a pointer on each priority queue head and we scan all the queue linearly. For making or maintaining the priority queue of one AP takes $O(n \log n)$ time. Instead of this priority queue, if we use sorted list, it also takes $O(n \log n)$ time. All the APs may not be activated at a time, only a set of APs is possible to be activated. In case if all the APs are activated then for k APs, it takes $O(kn \log n)$ time. Linear scan will also take the same time, i.e., $O(kn \log n)$. Hence the algorithm 3 takes $O(kn \log n)$ time.

6.4 Time Complexity of Priority Updation

In priority updation, we update the priority of the STAs which are being associated with the activated APs. In case if all the APs are activated, we change the priority of associated STAs and enqueue these STAs back in their respective queues. One enqueue operation takes $\log n$ time, hence k enqueue operation will take $k \log n$ time. Hence the algorithm 4 takes $O(k \log n)$ time.

6.5 Time Complexity of Overall Algorithm

In the overall algorithm, algorithm 1 to algorithm 4 are executing continuously until all the STAs have transmitted its data. The no. of slots can be assumed to be constant, i.e., c. Hence the total time will be

$$O(kn) + c(O(k^2) + O(kn \log n) + O(k \log k))$$

Hence the time complexity of our algorithm is $O(kn \log n)$.

Chapter 7

Simulation Results

We conduct extensive experiments to evaluate the performance of our algorithm in large dense wireless network. In our simulation setup, we have considered the same WLAN environment as in [15] where, a number of APs and a number of STAs (users) are uniformly placed in a $100 \times 100 meters^2$ area. We have fixed a minimum distance of 8 meters between every pair of APs. We have not assumed any minimum distance between STAs, i.e., minimum distance between any pair of STAs is zero because in practical situation the users can be co-located.

An example placement of 50 STAs and 200 APs is shown in figure 7.1 where the red circles are APs and green stars are STAs.

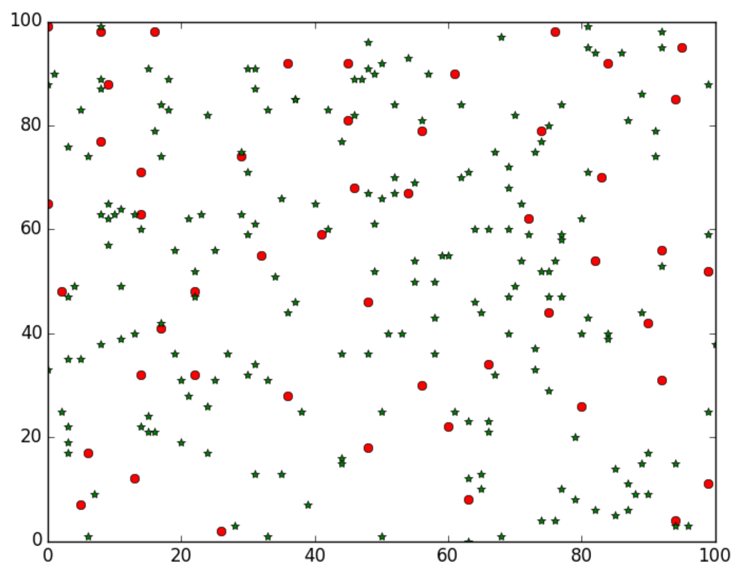

Figure 7.1: Uniform placement of 50 APs and 200 STAs

The frame size is taken to be 1024 Bytes, i.e., 1 KB as in [4] and the slot length is taken to be 10 milliseconds. We have considered IEEE 802.11b clients only.

With this simulation environment, we evaluate the performance of the proposed algorithm in terms of average throughput obtained under two different scenarios. In the first scenario number of APs is taken as 20 and number of STAs is varying from 80 to 120 with a step of 5 to represent different STA densities. In the second scenario number of APs is 40 and number of STAs is varying from 160 to 200 with a step of 5.

In each scenario we have taken three minimum and maximum data limits for each STA as from 100KB to 1000KB, 100KB to 500KB and 500KB to 1000KB respectively. We have conducted different experiments to show the effect of number of APs and number of STAs on average throughput for all three cases. In all cases we have considered 100 different runs and reported the average results to mitigate the effect of randomization.

In figure 7.2, 7.3 and 7.4 the relationship of variation in number of slots with number of STAs is shown where the number of APs is 20 and STAs is varying from 80 to 120 with a step of 5. The data range for each STA is from 100KB to 1000KB, 100KB to 500KB and 500KB to 1000KB respectively. Its very intuitive that if number of STAs increases, the traffic increases. Hence, the number of slots increases.

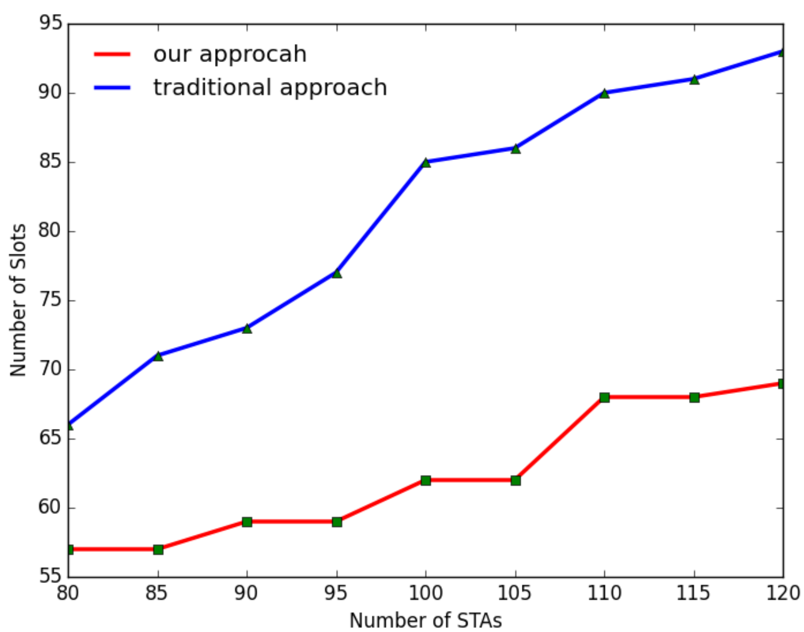

Figure 7.2: No. of slots with 20APs and the data range is 100KB to 1000KB

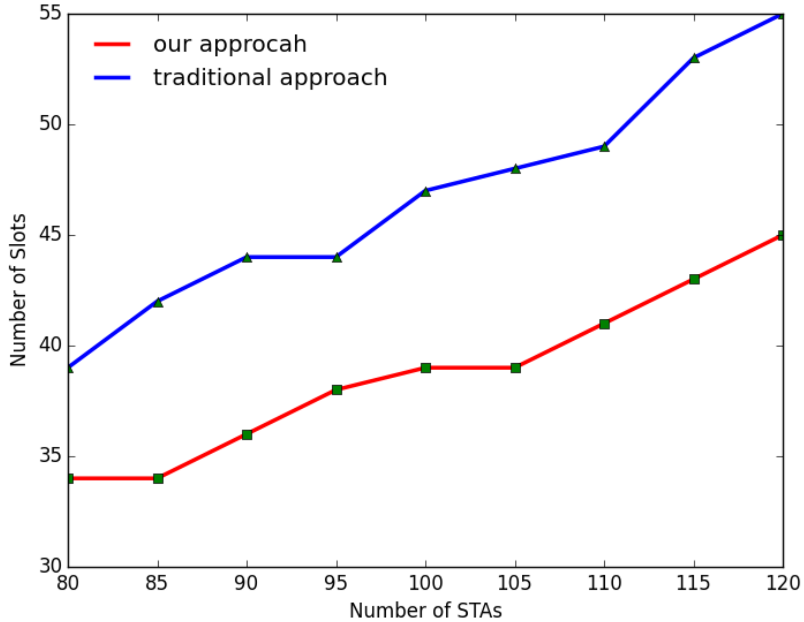

Figure 7.3: No. of slots with 20APs and the data range is 100KB to 500KB

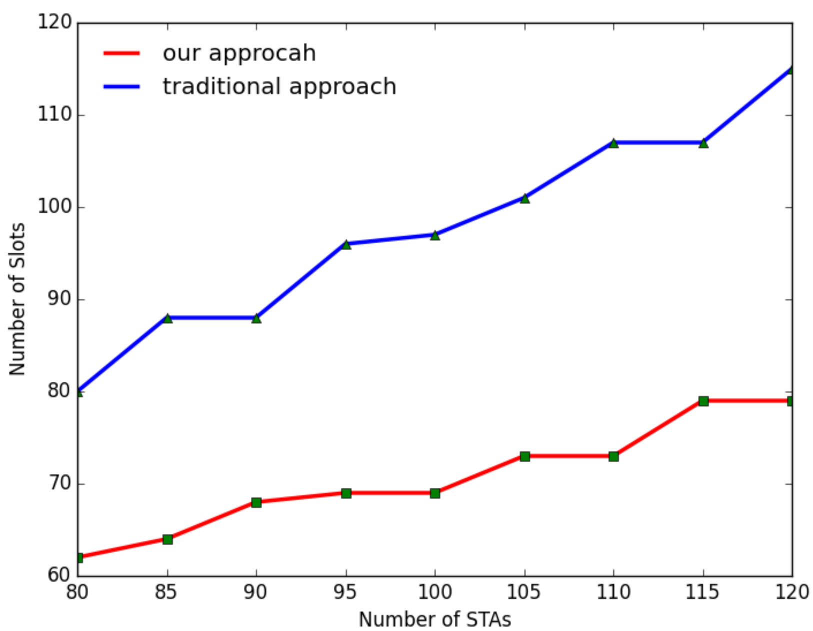

Figure 7.4: No. of slots with 20APs and the data Range is 500KB to 1000KB

In figure 7.5, 7.6 and 7.7 the relationship of variation in number of slots with number of STAs is shown while the number of APs has been increased to 40 and STAs is varying from 160 to 200 with a step of 5. The data range for each STA is from 100KB to 1000KB, 100KB to 500KB and 500KB to 1000KB respectively.

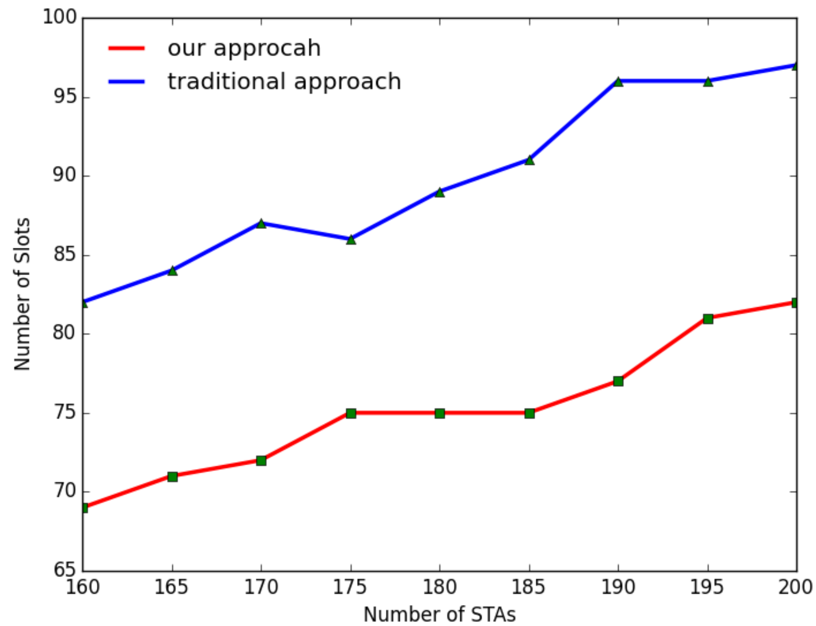

Figure 7.5: No. of slots with 40APs and the data range is 100KB to 1000KB

It is clear from the simulation results that our algorithm schedules the traffic earlier than the traditional approach in every scenario.

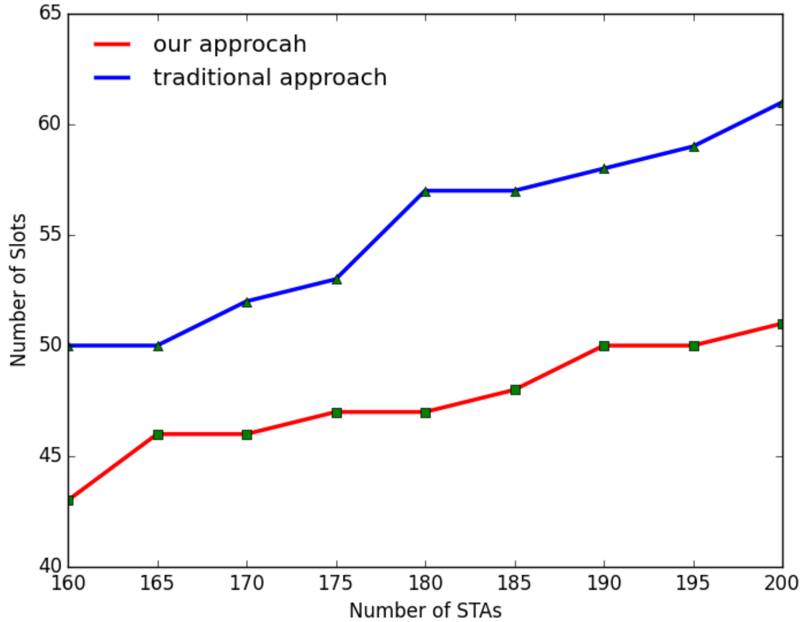

Figure 7.6: No. of slots with 40APs and the data range is 100KB to 500KB

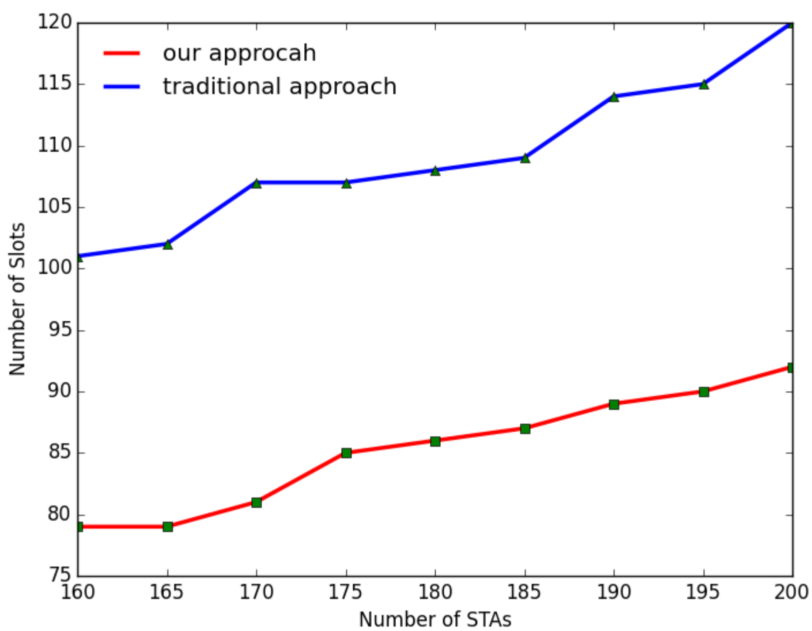

Figure 7.7: No. of slots with 40APs and the data range is 500KB to 1000KB

In figure 7.8 the relationship of variation in average throughput (per slot) with number of STAs is shown where number of APs is 20 and STAs is varying from 80 to 120 with a step of 5.
In figure 7.9 the relationship of variation in average throughput (per slot) with number of STAs is shown where number of APs has been increased to 40 and STAs is varying from 160 to 200 with a step of 5.

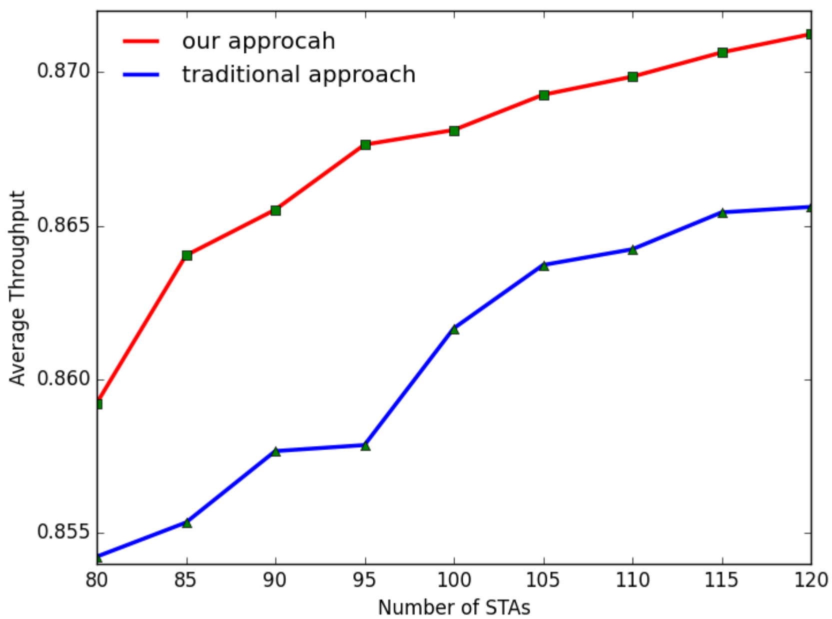

Figure 7.8: Average throughput with 20APs and the data range is 500KB to 1000KB

Figure 7.8 and 7.9 shows that our algorithm produces better average throughput per slot than the traditional approach.

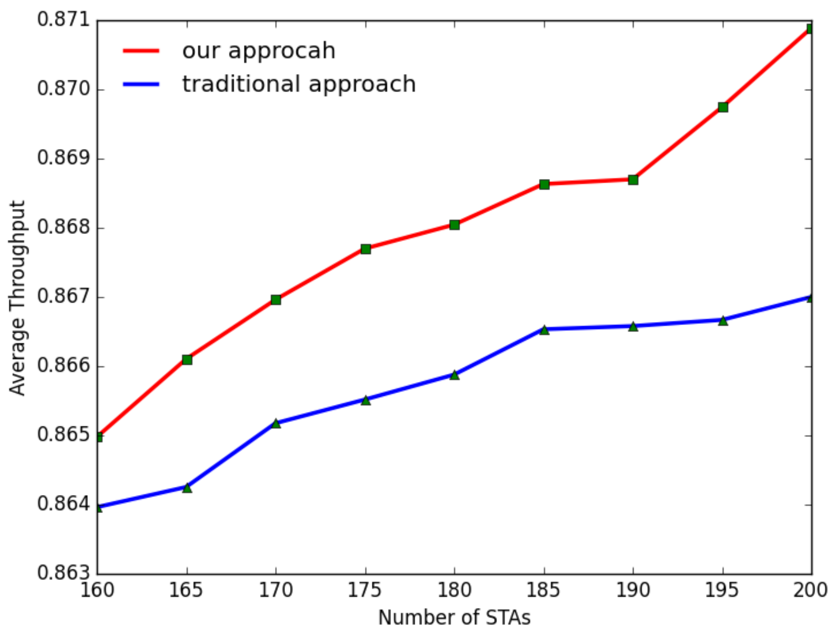

Figure 7.9: Averagee throughput with 40APs and the data range is 500KB to 1000KB

Chapter 8

Conclusion and Future Work

Throughput of WLAN depends on many factors. We have presented a greedy algorithm for channel assignment and association in WLAN. In our work we have considered interference, user's demand, number of slots required and the user's fairness. There are few factors which we have not considered yet but We are working on these factors and trying to find those factors which affects throughput per slot. For example traffic on an AP in AP selection. In STA selection while associating STAs with APs, few factors may be effective like - type of user's demand (e.g text, video etc.), delay of an STA etc. and can further improve the aggregate throughput.

Bibliography

[1] Aichele, C., Flickenger, R., Fonda, C., Forster, J., Howard, I., Krag, T., Zennaro, M., et al.: Wireless networking in the developing world (2007)

[2] Bejerano, Y., Han, S., Li, E.L.: Fairness and load balancing in wireless lans using association control. IEEE/ACM Trans. Netw. 15(3), 560–573 (2007), http://doi.acm.org/10.1145/1295237.1295244

[3] Chieochan, S., Hossain, E., Diamond, J.E.: Channel assignment schemes for infrastructure-based 802.11 wlans: A survey. IEEE Communications Surveys and Tutorials 12(1), 124–136 (2010), http://dx.doi.org/10.1109/SURV.2010.020110.00047

[4] Choi, E.C., Huh, J.D., Kim, K.S., Cho, M.H.: Frame-size adaptive mac protocol in high-rate wireless personal area networks. ETRI journal 28(5), 660–663 (2006)

[5] Cui, Y., Li, W., Cheng, X.: Partially overlapping channel assignment based on ”node orthogonality” for 802.11 wireless networks. In: INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10-15 April 2011, Shanghai, China. pp. 361–365 (2011), http://dx.doi.org/10.1109/INFCOM.2011.5935182

[6] Dailey, D.P.: Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete. Discrete Mathematics 30(3), 289–293 (1980)

[7] Ding, Y., Huang, Y., Zeng, G., Xiao, L.: Channel assignment with partially overlapping channels in wireless mesh networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet, WICON 2008, Maui, Hawaii, USA, November 17-19, 2008. p. 38 (2008), http://dx.doi.org/10.4108/ICST.WICON2008.4907

[8] Feng, Z., Yang, Y.: How much improvement can we get from partially overlapped channels? In: WCNC 2008, IEEE Wireless Communications & Networking Conference, March 31 2008 - April 3 2008, Las Vegas, Nevada, USA, Conference Proceedings. pp. 2957–2962 (2008), http://dx.doi.org/10.1109/WCNC.2008.517

[9] Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified np-complete problems. In: Proceedings of the sixth annual ACM symposium on Theory of computing. pp. 47–63. ACM (1974)

[10] Hoque, M.A., Hong, X., Afroz, F.: Multiple radio channel assignement utilizing partially overlapped channels. In: Proceedings of the Global Communications Conference, 2009. GLOBECOM 2009, Honolulu, Hawaii, USA, 30 November - 4 December 2009. pp. 1–7 (2009), http://dx.doi.org/10.1109/GLOCOM.2009.5425288

[11] Jim, G.: Assigning 802.11 b access point channels. online], http://www.80211planet. com/tutorials/article/0 4000 (2002)

[12] Liu, H., Yu, H., Liu, X., Chuah, C., Mohapatra, P.: Scheduling multiple partially overlapped channels in wireless mesh networks. In: Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, 24-28 June 2007. pp. 3817–3822 (2007), http://dx.doi.org/10.1109/ICC.2007.629

[13] Mishra, A., Banerjee, S., Arbaugh, W.A.: Weighted coloring based channel assignment for wlans. Mobile Computing and Communications Review 9(3), 19–31 (2005), http://doi.acm.org/10.1145/1094549.1094554

[14] Mishra, A., Shrivastava, V., Banerjee, S., Arbaugh, W.A.: Partially overlapped channels not considered harmful. In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/Performance 2006, Saint Malo, France, June 26-30, 2006. pp. 63–74 (2006), http://doi.acm.org/10.1145/1140277.1140286

[15] Mukherjee, S., Ghosh, S.C.: Throughput improvement using partially overlapping channels in WLAN with heterogeneous clients. In: Wired/Wireless Internet Communications - 14th IFIP WG 6.2 International Conference, WWIC 2016, Thessaloniki, Greece, May 25-27, 2016, Proceedings. pp. 335–347 (2016), http://dx.doi.org/10.1007/978-3-319-33936-8_26

[16] Rad, A.H.M., Wong, V.W.S.: Partially overlapped channel assignment for multichannel wireless mesh networks. In: Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, 24-28 June 2007. pp. 3770–3775 (2007), http://dx.doi.org/10.1109/ICC.2007.621

[17] Sridhar, S., Guo, J., Jha, S.: Channel assignment in multi-radio wireless mesh networks: a graph-theoretic approach. In: 2009 First International Communication Systems and Networks and Workshops. pp. 1–10. IEEE (2009)

[18] wiki: Ieee 802.11, https://en.wikipedia.org/wiki/IEEE_802.11

[19] wiki: Ieee 802.11, https://en.wikipedia.org/wiki/List_of_WLAN_channels