On Large Throughputs in High Density Wireless Local Area Networks

Student Name: Mridula Singh

IIIT-D-MTech-CS-MUC-12-068

May 20, 2014

Indraprastha Institute of Inforation Technology
New Delhi

Thesis Committee
Dr. Sanjit Krishnan Kaul (Chair)
Dr. Pravesh Biyani
Dr. Vinay Ribeiro
Dr. Vivek Bohara

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Mobile and Ubiquitous Computing

Keywords: High Density Networks, Spatial Reuse, 802.11, WiFi, Optimization, Throughput Maximization, Enterprise Networks.

Certificate

This is to certify that the thesis titled **"On Large Throughputs in High Density Wireless Local Area Networks"** submitted by **Mridula Singh** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under my guidance and supervision in the Mobile and Ubiquitous Computing group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Sanjit Krishnan Kaul, Assistant Professor
Indraprastha Institute of Information Technology, New Delhi

Abstract

While the density of access points in enterprise settings has increased, the sharing of the spatial resource amongst links in 802.11 wireless local area networks remains inefficient. Conservative mechanisms based on a static carrier sense range (CSR) are used and are designed to avoid occurrence of interfering transmissions. Even when the CSR is adapted to allow interfering transmissions, it is with the goal of increasing spatial reuse, which may not translate to larger network throughput.

We formulate the network throughput optimization problem, which is to decide which links in a network must share in space (transmit data simultaneously) such that the network throughput is maximized. Links share in space by piggybacking on data transmission opportunities seized by another link using RTS/CTS as specified in the distributed coordination function (DCF) of 802.11. Sharing in space increases interference and hence reduces the PHY rate at which a link can send data. It also increases the opportunities a link gets to transmit data, however. We show that the optimization problem is NP hard. A relaxation of the problem gives an upper bound on network throughput. Computationally feasible algorithms that achieve a significant percentage of the upper bound are proposed. We will restrict our network modeling and evaluation to 802.11 networks in which all nodes always have a packet to send and are within carrier sense range of each other. Networks with a high density of clients and AP(s) are shown, via simulation, to achieve large throughput gains (up to 400% for a network of 25 clients and AP(s)), over standard 802.11.

Acknowledgments

I dedicate this thesis work to my parents, my two pillars of strength. They have always been there for me and have encouraged me in every step of the way to remain focused on my goal. I would like to express my deepest gratitude to my advisor, Dr. Sanjit Krishnan Kaul, for his guidance and support. I am grateful to him for teaching and discussing problems even if it had to be dealt with right from the basic fundamentals. The efforts he put in work always inspires me to work hard.

I sincerely thank Dr. Pravesh Biyani for discussing problems and providing input for dynamic approach. I would also thank Dr. Sandip Aine and Prof. Sumit Roy for their valuable inputs.

I wish to express my gratitude to Prof. Pankaj Jalote and other associated members for creating and managing such a wonderful institute.

I would like to thank Dr. Vinay Ribeiro (IIT-Delhi) for introducing me to the field of wireless networks.. It is my pleasure to have him in my thesis committee.

I will also like to thank all MUC lab members for detailed discussions on various research topics, especially Dheryta for discussing the concept of Computer Networks.

I was lucky to be in the company of such wonderful people at IIITD, those who made me laugh even during the rough patches. They have constantly been a source of positive energy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

802.11 wireless LAN(s) (WiFi) have made rapid improvements in the physical (PHY) layer bit rates supported over a wireless link between a client and its access point (AP). 802.11ac allows use of up to 160MHz of bandwidth, multiple antennas (MIMO), with supported rates as high as 1 Gbps when using multi-user MIMO [27]. The increasing PHY rates have been accompanied by a shift in the enterprise WiFi user's demand from just network coverage to high throughputs [28]. The enterprise network must now support large bits/sec/Hz/user. The need to support large user throughputs in high density 802.11 networks with a large number of access points and clients, say for example in operator hotspots, homes, campuses, stadiums, and enterprises, is in fact the push behind the recent High Efficiency WLAN (HEW) study group [1].

While available bandwidth and number of antennas at clients and AP(s) have increased manifold, sharing of the spatial resource between links connected to different access points remains inefficient. A client does not initiate a transmission to its AP, and vice versa, as long as a transmission with energy that exceeds the clear channel assessment (CCA) threshold is received, that is the transmission is within carrier sense range. The CCA is set as per the 802.11 standard specifications [3], so as to largely avoid simultaneous transmissions. To avoid too many links within carrier sense range of each other, often links are split across non-overlapping channel bandwidths, that is different AP(s) are configured to operate on non-overlapping bands [28]. The use of the wireless resource is nevertheless inefficient. In this work we show that links, in high AP and client density settings, could gain from transmitting simultaneously, that is sharing-in-space. While sharing-in-space increases interference, the links get to transmit for larger amounts of time and over larger bandwidths. While keeping access mechanisms as they are in 802.11, once access to transmit data has been gained by a link using RTS/CTS, we allow other links to piggyback on the data transmission opportunity created and transmit their data simultaneously with the link. The optimization problem is to choose links that can transmit simultaneously so that the throughput is maximized.

There are prior works [8] and [13] that allow spatial reuse to improve throughputs in mesh networks. Other works [24] have proposed mechanisms to allow simultaneous transmissions to improve throughputs by alleviating loss in throughput due to exposed nodes in 802.11 networks.

However, in the context of wireless LAN(s) with a high density of access points, leveraging gains from allowing links within carrier sense range to transmit data simultaneously is not well studied. In this work we propose suitable algorithms and show that very large gains in throughput are available.

Our specific contributions are as follows:

- We formulate the network throughput maximization problem for a network of links that may transmit simultaneously. The problem is shown to be NP-hard.
- We propose an upper bound on network throughput. Polynomial time algorithms are proposed.
- Simulations over networks of high to very high client and AP densities, and for low to high link SNR(s), show that the algorithms achieve network throughputs close to the upper bound.
- We demonstrate that large gains in throughput can be achieved.
- We illustrate, using simulations, the tradeoff between number of AP(s) and network throughput for different network densities and link SNR(s).

In Chapter 3 we describe our network and the modifications we propose to 802.11 DCF. The optimization problem is formulated in Chapter 4. An upper bound on the network throughput is described in Chapter 5. Polynomial time algorithms are described in Chapter 6. Chapter 7 describes our evaluation methodology and Chapter 8 show results from simulations. We conclude in Chapter 9.

Chapter 2

Related Works

The enterprise network must now support large bits/sec/Hz/user. We present related works that improve network throughput by exploiting PHY and MAC layer parameters, using distributed and centralized algorithms, in different types of networks including Mesh, Ad-hoc and enterprise networks. We also discuss recent work on decoding multiple transmissions that occur simultaneously.

There are many works that use space time division multiple access (STDMA) based scheduling and spatial reuse to improve network throughput. For example [8] and [13]. Links are allowed to transmit together as long as the resulting SINR exceeds a minimum threshold such that a minimum data rate is supported. [13] consider two methods of generating a schedule. One method use a graph representation of the network with limited information of interference and the other method uses full information of signal-to-interference ratio (SIR). In our work we consider 802.11 DCF based networks with large AP and client density. We use pairwise SIR information of each node to select which links must transmit together with the goal of maximizing network throughput.

If beacons are transmitted periodically, interference between all pair of links in a network of N nodes can be estimated in $O(N^2)$ time. Works such as [4] construct precise RF maps in the form of a conflict graph online by using the micro-probing technique, which operates over a millisecond time scale. [9] analyzes the effect of multiple interferers. Interferers that do not cause much interference individually, can cause significant interference when they all interfere together.

In [18], the authors consider tuning PHY parameters of transmit power and data rates and MAC backoff timer jointly, in order to optimize overall network throughput. The authors shows that system throughput is not a monotonically increasing/decreasing function of CS range but exhibits transitional point at which many choices of CS range are possible. Using the interference model the authors define a collision zone for each node. If all simultaneous transmissions belong to different collision zones, each transmission is assigned a data rate that is a function of its SINR.

[8] uses the threshold model in which decision of scheduling a link is binary. According to [19], deciding threshold for practical purpose is not easy. They propose a probabilistic model, in which scheduling is based on SINR and bit error rate (BER).

In [14], author shows that spatial reuse is characterized by the relation between the transmitters and transmitter-receiver distances. On increasing transmission power, spatial reuse can be improved in ambient noise dominated environments but not in interference limited scenarios. On the other hand, more simultaneous transmissions can be made physically possible by decreasing the average transmitter-receiver separation.

In [16], the authors use CSMA and include the possibility of packet errors due to collisions, which occur when multiple transmissions overlap in time. They use a Markov chain to model CSMA collisions and its overheads and propose throughput optimal scheduling algorithms. They also show relationship between CSMA parameters such as maximum packet length, short term fairness and achievable capacity regions.

Works like [28] allow dynamically changing the carrier sense (CS) threshold to improve spatial reuse, so that multiple flows can coexist in current 802.11 architectures. All AP(s) are controlled by an algorithm running on central controller, which allocates orthogonal channels to nearby AP(s) and adjusts transmit power values dynamically, thereby exploiting spatial heterogeneity in distribution of users at the hotspots.

In context of high density AP deployments, [22] lists the limits of conventional enterprise WLANs and proposes a software architecture DenseAP, which is a novel system for improving the performance of enterprise WLANs using a dense deployment of AP(s).To increase capacity, the APs must be assigned appropriate channels and the clients must make intelligent decisions about which AP to associate with. DenseAP provides significant improvements in performance over existing WLANs. They suggest that decisions about channel assignment, and associations must be based on a global view of the entire WLAN, rather than the local viewpoint of an individual client or AP.

In [11], authors study the effect of inter-cell interference in unplanned AP installations. The efficiency of an 802.11 network is determined by the number of active clients. They find that with a typical TCP-dominant workload,cumulative system throughput is characterized by the number of interfering access points rather than the number of clients. This motivates the possibility that a controller algorithm together with large number of AP(s) can increase network throughput.

Speculative scheduling is used in [5] for interference mitigation in enterprise WLANs where all AP(s) and clients operate in the same channel and a client can associate with multiple AP(s). They are design a centralized approach where the scheduler has the responsibility to schedule traffic. They suggest that a speculative component is necessary to handle delay incurred by the wireless medium in form of data rate adaptation algorithms, frame re-transmissions, and co-existence with other non-enterprise traffic. According to them, centralized design leads to better utilization of the channel and small mean delay-jitter on wired path between controller and AP(s). In our work too all algorithms are assumed to run on an enterprise WLAN controller.

In [23] authors propose MegaMIMO, which enables high density of independent access points (APs) to beamform their signals, and communicate with their clients on the same channel as if they were one large MIMO network, to scale network throughput with the number of transmitting devices. It requires that transmitters exchange decoded packets among themselves and thus, it works only for the downlink transmissions. On the other hand, [6] proposes Symphony, that improves the throughput for the uplink traffic. Here multiple AP(s) in an enterprise WLAN co-operate to decodes simultaneous transmissions using successive interference cancellation (SIC). These decoding schemes work without any central controller but need synchronization and low mutual-latency. In our work we do not propose any method to decode simultaneous transmissions. Our work is only concerned with the SINR to PHY rate mapping that is obtained as a result of the decoding scheme used.

Works [21], [17] suggest that simultaneous transmissions do not always end in collisions. In [17] it is shown by experiments shows that stronger signals can be decoded regardless of the timing relation with the weaker. In [21] authors use Message in Message (MIM), which ensures that a receiver correctly decodes the stronger signal by forcing the transmissions to follow an order.

Works like [7], [20], and [15] model the 802.11 DCF under different traffic and interference assumptions. [7] forms the basis of our model, which is described in Chapter 3.

Chapter 3

Network Model

The network, see Figure 3.1, consists of M access points. Access point (AP) a_k has clients $c_{k1}, \ldots, c_{kn_k}$ associated with it, where $1 \leq k \leq M$. Clients send and receive packets to and from the access point to which they are associated. AP a_k sends packets over n_k links. A packet for client c_{kj} is sent by a_k over link $a_k \rightarrow c_{kj}$. The client c_{kj} sends all its packets to a_k over the link $c_{kj} \rightarrow a_k$. All links in the network, including those that originate in the same AP, contend with each other for exclusive access to the medium. An AP can either transmit or receive on just one link at any given time. All AP(s) are connected to the enterprise WLAN controller via a high speed wired backbone. All algorithms we propose run on the controller. The AP(s) and clients send estimates of received power from other nodes to the controller, which uses this to calculate the signal-to-interference-and-noise ratios that will be used by the algorithms.

All nodes (access points and clients) in the network are within each other's carrier sense range and can sense each other's transmissions. Thus, all links have an accurate estimate of whether the channel is busy or idle. They access the medium using the distributed coordination function (DCF) with RTS/CTS, as specified in the 802.11 standard. The links are assumed to be saturated. That is, they always have a packet to send.

3.0.1 Standard 802.11 DCF Operation

Details of standard 802.11 DCF operation can be found in [7]. We only present details relevant to our work. The *saturation* throughput[1] T_{sat} of such a network is given by equation (13) in [7]. We have

$$T_{\text{sat}} = \frac{P_s P_{\text{tr}} E[\mathcal{D}]}{(1 - P_{\text{tr}})\sigma + P_{\text{tr}} P_s T_s + P_{\text{tr}}(1 - P_s) T_c}. \tag{3.1}$$

Here, as defined by Bianchi in [7], P_{tr} is the probability that the channel is busy, that is at least one packet transmission is taking place. P_s is the conditional probability of a successful transmission, one that is decoded correctly, conditioned on the channel being busy. Note that,

[1] In this work by throughput we mean saturation throughput.

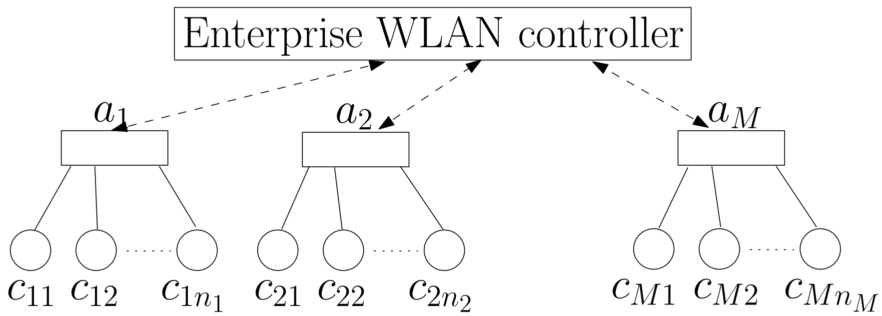


Figure 3.1: Nodes a_k, $k = 1, \ldots, M$, are the access points. Access point a_k has clients $c_{k1}, \ldots, c_{kn_k}$ associated with it. The AP(s) are connected to the controller via a high speed wired backbone.

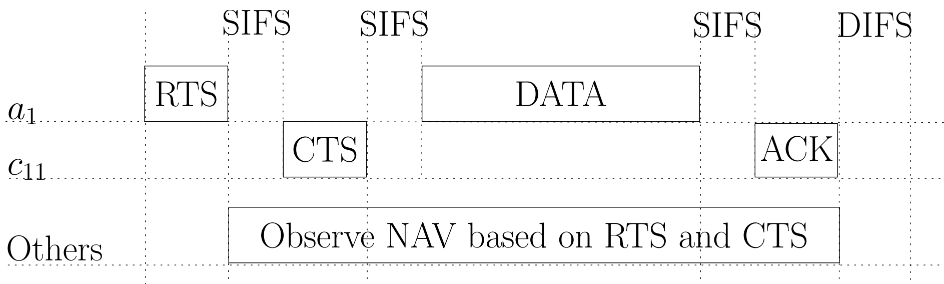


Figure 3.2: RTS/CTS mechanism in 802.11 DCF.

as per the model for DCF in [7], transmission over a link is successful only if no other link starts transmitting simultaneously. When RTS/CTS is enabled, successful transmission of data requires a link to reserve the medium. This process is started by sending a RTS. The medium is reserved successfully by the link only if no other links start transmitting when the RTS is being sent. Bianchi defines three kinds of slots — the idle slot, a slot during which successful transmission takes place, and a slot during which a collision (more than one simultaneous RTS transmissions) takes place. T_s is the average length in seconds of a slot during a successful transmission (payload, headers, and other protocol overheads like RTS/CTS and ACK are included). T_c is the average length in seconds of a slot during a collision. Note that a collision can only take place between RTS packets. The length of an empty slot is σ. The length $\overline{\sigma}$ of an *average* slot is thus given by $\overline{\sigma} = (1 - P_{\text{tr}})\sigma + P_{\text{tr}} P_s T_s + P_{\text{tr}}(1 - P_s)T_c$, which is the denominator of (3.1). $E[\mathcal{D}]$ is the average length in seconds of data payload at any link in the network. The saturation throughput T_{sat}, given by (3.1), is the fraction of an *average* slot that is used for successful payload transmission.

Let τ be the probability that a link starts transmitting. Let N be the total number of links in the network. We have $P_{\text{tr}} = 1 - (1 - \tau)^N$ and $P_s = N\tau(1 - \tau)^{N-1}/P_{\text{tr}}$. An example timeline of a successful data packet transmission is shown in Figure 3.2, where AP a_1 sends a packet to client c_{11}. All other nodes set their network allocation vectors (NAV) based on information in the RTS and the CTS, so as to not attempt transmission while a_1 completes data transmission and c_{11} sends an ACK in response.

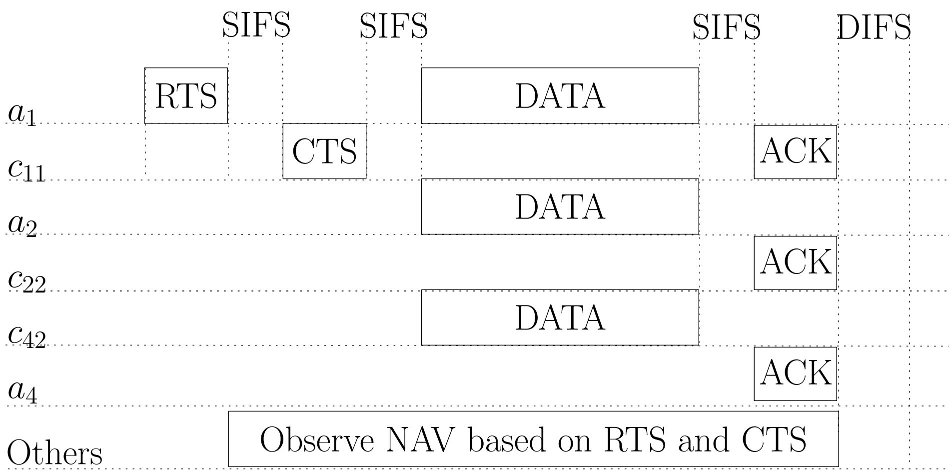


Figure 3.3: Modified RTS/CTS mechanism.

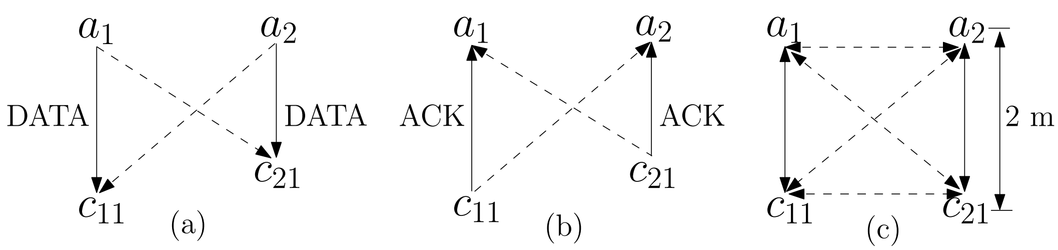


Figure 3.4: Figures (a) and (b) show the interfering flows when DATA and ACK transmissions take place. Figure (c) shows example AP and client placements that we use in simulations.

Let link i have a signal-to-noise ratio of SNR_i when i transmits exclusively over its data transmission opportunity. Its physical rate[2] of transmission R_i bits/sec/Hz is given by $R_i = \log_2(1 + \mathrm{SNR}_i)$. Let $\mathcal{T}_i$ be the time interval over which i sends data. Without loss of generality, we will assume the bandwidth to be 1 Hz. Let $\tau_s = \tau(1-\tau)^{N-1}$. The data payload $\mathcal{D}_i$ bits sent by i in an average slot and the corresponding throughput T_i bits/sec are given by

$$\mathcal{D}_i = \tau_s \mathcal{T}_i \log_2(1 + \mathrm{SNR}_i);\; T_i = \mathcal{D}_i / \overline{\sigma}. \tag{3.2}$$

3.0.2 Proposed Modification to DCF: Links Piggyback On Data Transmission Opportunities Seized By Others

Consider a subset $S = \{i_1, \ldots, i_{|S|}\}$ of links in the network. Note that $|S|$ is the size of the set S. The links in the subset contend for access to the medium amongst themselves and with links in complementary set S^c. We will call a set S of links a share-in-space (SS) set if the links in it *piggyback* on data transmission opportunities seized by any one of the links in the set, that is all links in the set transmit data during the opportunity. Note that in standard DCF SS sets are singleton.

[2]The contributions in this work can be extended to any chosen mapping between SNR/SINR and PHY rate in a straightforward manner. The choice of Shannon's capacity formula helps us demonstrate with clarity why our proposals lead to throughput gains in 802.11 networks.

A link in S seizes a data transmission opportunity by successfully transmitting a RTS message. Let's say that link $i_k \in S$ sends a RTS successfully. This event occurs with probability $\tau_s = \tau(1-\tau)^{N-1}$. Data transmission over the link will start an interval SIFS after completion of the RTS/CTS message exchange. The other links in S hear the RTS/CTS exchange and know that it belongs to link i_k that is a part of their SS set S. So they piggyback on the data transmission opportunity obtained by i_k and start transmissions of their data simultaneously with that of i_k. Note that while these links in the set $\{S \setminus i_k\}$ transmit data simultaneously with i_k, the DCF at each of the links maintains state as under standard 802.11 operation. For example, these links pause their backoff counters at the start of the RTS transmission by i_k and resume the counters only after the medium is sensed idle for DIFS. Figure 3.3 shows links $a_2 \to c_{22}$ and $c_{42} \to a_4$ piggybacking on the data transmission opportunity seized by the link $a_1 \to c_{11}$.

As a result of allowing piggybacking on data transmit opportunities seized by links in the set, every link in set S gets a data transmission opportunity with a probability $|S|\tau_s$. However, the increase in opportunity may or may not translate into larger throughput. This is because the simultaneous data transmissions of the links in the SS set S interfere with each other. Thus, they must occur at physical layer rates smaller than rates that would be possible in the absence of interference. Next we will formalize this fact.

As part of the SS set S, link i will on an average get $|S|$ opportunities to transmit data for every opportunity it got under standard DCF operation. However, link i will see interference from data transmissions of the $|S|-1$ other links in the SS set. Let the signal-to-interference-and-noise ratio of link i be $\mathrm{SINR}_i^{(S)}$. Note that $\mathrm{SINR}_i^{(S)} \leq \mathrm{SNR}_i$. The average payload (in bits) transmitted by i over an average slot of the modified DCF operation, therefore, is

$$\mathcal{D}_i^{(S)} = \tau_s |S| \mathcal{T}_i \log_2(1 + \mathrm{SINR}_i^{(S)}). \tag{3.3}$$

We will now argue that the average slot length of the modified DCF is the same as that of the standard DCF.

Recall that every data transmission over a link (see Figures 3.2 and 3.3) is followed by an ACK by the receiver. The ACK is transmitted in the reverse direction. Figure 3.4(a) shows two links $a_1 \to c_{11}$ and $a_2 \to c_{21}$ transmitting data simultaneously. The transmission from a_2 interferes with reception of data from a_1 at c_{11} and that from a_1 creates interference at c_{21}. The ACK(s) corresponding to the data transmissions flow in the opposite direction as shown in figure 3.4(b). The SINR of the link $a_1 \to c_{11}$ when transmitting data may not be the same as the SINR of the link $c_{11} \to a_1$ that transmits the ACK. When links are part of a SS set, their data transmissions will occur in parallel and so will the transmissions of ACK(s) that correspond to the data transmissions. The rates of data and ACK transmission will in general be different.

We assume that all data transmissions occur over a fixed time interval[3], that is $\mathcal{T}_i = \mathcal{T}_j$ for all i, j. ACK packets are of a fixed format and size, however. Under standard DCF operation,

[3] This assumption leads to time fairness across links, which is desirable in 802.11 networks with links that operate at different physical layer rates [25].

their transmission takes a fixed amount of time. Under the modified operation, given that different links will operate at different rates, ACK packets on different links will take different lengths of time. However, since the size of an ACK packet is much smaller (or can be made smaller given the saturated node assumption) than that of a data packet, in this study, we ignore differences in ACK transmission times on different links. From the above observations, one can conclude that the proposed modifications to DCF do not change T_s and T_c. Neither do they change σ, P_{tr} and P_s. That is the length of an average slot $\overline{\sigma}$ is unchanged. However, the payload transmitted by link i during the average slot is now (3.3) instead of (3.2). The total payload $\mathcal{D}_S = \sum_{i\in S} \mathcal{D}_i^{(S)}$ bits transmitted by all links in SS set S over an average slot and the corresponding sum throughput T_S bits/sec of the set S, are given by

$$\mathcal{D}_S = \sum_{i\in S} \tau_s |S| \mathcal{T} \log_2(1 + \mathrm{SINR}_i^{(S)});\ T_S = D_S/\overline{\sigma}. \tag{3.4}$$

Let $\widetilde{\mathcal{D}}_S$ be the total payload transmitted by links in the set S under standard 802.11 DCF operation. Let $\widetilde{T}_S$ be the corresponding sum throughput. Using (3.2), we have

$$\widetilde{\mathcal{D}}_S = \sum_{i\in S} \tau_s \mathcal{T} \log_2(1 + \mathrm{SNR}_i);\ \widetilde{T}_S = \widetilde{\mathcal{D}}_S/\overline{\sigma}. \tag{3.5}$$

Consider the equations (3.4) and (3.5). Note that, for all i, $\mathrm{SINR}_i^{(S)} < \mathrm{SNR}_i$. As a result, when piggybacking, each link sends at a smaller PHY rate, and hence transmits fewer bits in a single transmission opportunity, than when using the standard DCF model. However, piggybacking gives the links an average of $|S|$ more data transmission opportunities. The throughput of a link benefits from piggybacking when this *spatial multiplexing* gain of $|S|$ outdoes the PHY rate penalty incurred due to increased interference.

Note on calculation of SINR$_i$: Let SIR_{ij} be the signal-to-interference ratio at link i due to interference from j. All links are assumed to have a fixed energy budget that is independent of the size of SS set they are a part of. Also, links distribute the available energy equally over all transmit opportunities that they get as a result of being in a SS set. If a link transmits with power $P_t$when it is part of a singleton SS set containing only the link, then on being a part of a SS set of size $|S|$, every data transmission opportunity it gets, it will transmit with power $P_t/|S|$. For a link i that is part of a SS set S containing $|S|$ links, SINR of link i can be written as $\mathrm{SINR}_i = (|S|\mathrm{SNR}_i^{-1} + \sum_{j\in S\setminus\{i\}} \mathrm{SIR}_{ij}^{-1})^{-1}$. Here SNR_i is the SNR of the link i, as in (3.2), when it transmits data alone.

General 802.11 *network settings:* It is worth noting here that in more general network settings, for example, networks with unsaturated nodes [20], or networks with hidden nodes [15], the possibility of gains by simultaneous transmissions exist as motivated above using Bianchi's model. While in Bianchi's model all links have the same probability τ_s of getting exclusive access successfully, in more general 802.11 based networks this probability will be different for different links. The analysis above can be thus extended to more general scenarios. The optimization problem and algorithms presented later in this paper are applicable in general. All evaluation in this paper is for networks modeled by Bianchi, however. Also, catering to more general scenarios will need additional modifications to the DCF protocol as not all links may be in each other's carrier sense range.

Decoding of simultaneous data transmissions: In this work we do not consider the practical aspects of decoding simultaneous transmissions in 802.11 WLAN settings. However, there are recent works for example [6], [23] that use multiple AP(s) in an enterprise WLAN to decode simultaneous transmissions. Next we begin to answer which transmissions must be be simultaneous so as to maximize network throughput.

Chapter 4

Optimization Problem

Let $\mathcal{N} = \{1, 2, \ldots, N\}$ be the set of all links in the network. A *partition* of $\mathcal{N}$ is a set of mutually exclusive and collectively exhaustive (MECE) share-in-space (SS) sets. Consider a partition $P = \{S_1, \ldots, S_{|P|}\}$, where the S_i, $1 \leq i \leq |P|$, are SS sets. By the definition of P, $S_i \cap S_j = \emptyset$, for $i \neq j$. Links within a given SS set piggyback on data transmission opportunities seized by any link within the set. Their data transmissions *share-in-space*. Let the throughput of the network when it is partitioned as P be T_P. We have

$$T_P = \sum_{p \in P} T_p, \tag{4.1}$$

where T_p is the throughput of SS set p and is given by (3.4).

Define $\mathcal{P}$ to be the set of all partitions of the network. We want to find the partition P that maximizes the network throughput. The optimization problem is given by

$$\text{Maximize:} \quad T_P \tag{4.2}$$

$$\text{subject to:} \quad P \in \mathcal{P}. \tag{4.3}$$

This optimization problem is combinatorial in nature and can be proven to be equivalent to a Generalized Bin packing problem, which is NP hard and therefore has an exponential computational complexity. This brings us to the next lemma.

Lemma 4.0.1 *Problem (4.2)-(4.3) is equivalent to a generalized cost variable sized bin packing problem [10].*

Proof of Lemma 4.0.1:

In the bin packing problem, objects of different volumes must be packed into a finite number of bins or containers each of volume V in a way that minimizes the number of bins used. The size of object vary and size of bins is constant. Cost associated with each bin is its size. For example, if size of our bin is V, the sum of the size of objects it contain should be less than or equal to V.

We reduce our problem into a variant of the bin packing problem, known as generalized cost variable sized bin packing (GCVS) problem. This is a general version of bin packing, where an available bin size is associated with a cost, which may be smaller or larger than its size. The costs of different bin sizes are unrelated.

According to GCVS we get bins of type r, where type is defined by its size. We are given a supply of N items that need to pe partitioned into subsets. Each subset has some cost associated with it. If a subset i is assigned to a bin j, let's say that we get a packaging cost of $C_{i,j}$. The aim is to find a feasible solution and minimizes the sum of packaging cost.

In our problem we are given N links and we can get all possible partitions of the set of links with a throughput (benefit) associated with each partition. If we assign a partition of size J to a bin, its cost will be equal to the throughput of the partition. Our aim is to pack all the links such that sum throughput is maximized.

Chapter 5

Upper Bound on Network Throughput

Recall that only links that do not share an AP can share in space. If there are $M \leq N$ AP(s) in the network, then a maximum of M links can be in a share-in-space set. The throughput of a SS set is calculated using (3.4). Calculate the throughputs of all SS sets that contain $i \leq M$ links. Let $\text{BEST}(i)$ be the maximum of the calculated throughputs. If $M < N$, then for $M < i \leq N$, let $\text{BEST}(i) = 0$.

Consider a partition $P = \{S_1, \ldots, S_{|P|}\}$ of the network $\mathcal{N}$. Let $I_k = |S_k|, 1 \leq k \leq |P|$. Note that the set $\{I_1, I_2, \ldots, I_{|P|}\}$ is an integer partition of N. Also, it is a fact that the throughput of the network partitioned as P, $T_P \leq \sum_{i \in \{I_1, \ldots, I_{|P|}\}} \text{BEST}(i)$. Define $\text{IP}(N)$ to be the set of all integer partitions of N. We want to find an upper bound on the throughput of all partitions of network $\mathcal{N}$. The bound is provided by the solution to the following optimization problem. The optimization problem, call it Optimal-Integer-Partition (OIP for short) is

$$\text{Maximize:} \quad \sum_{i \in I} \text{BEST}(i) \tag{5.1}$$

$$\text{subject to:} \quad I \in \text{IP}(N). \tag{5.2}$$

Note that the problem (5.1)-(5.2) is a relaxation of the problem (4.2)-(4.3). Specifically, we are relaxing the constraint (4.3) in the latter. That is the selected SS sets do not need to form a partition of the network $\mathcal{N}$. We want to maximize the throughput. The optimizer of (5.1)-(5.2) is just the set of sizes of SS sets, however chosen, that maximize throughput.

Optimal-Integer-Partition can be solved using the dynamic program (has a time complexity of $O(N^2)$ [26]):

$$\text{SolveOIP}(N) = \max\big[\text{BEST}(N), \max_{k \in [1, \lfloor N/2 \rfloor]} (\text{SolveOIP}(N-k) + \text{SolveOIP}(k))\big]. \tag{5.3}$$

It has a time complexity of $O(N^2)$ [26]. For a network to achieve the upper bound returned

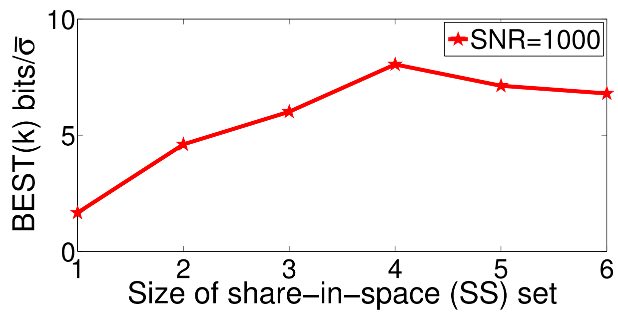


Figure 5.1: BEST(i) for an example 6 link network.

by OP, it must be possible to select mutually exclusive and collectively exhaustive SS sets such that the sizes of the sets are as suggested by the optimizer I^* and each selected SS set has a throughput (calculated using (3.4)) that equals the maximum of throughputs of SS sets of its size. For example, if $N = 4$, then $\text{IP}(4) = \{(1,1,1,1),(1,1,2),(1,3),(2,2)\}$. If $I^* = (2,2)$, then OP suggests that the 4 links be divided in two SS sets of 2 links each. To achieve maximum throughput, there must exist a way to select two mutually exclusive SS sets such that both the sets have a throughput of BEST(2). However, the above is not always possible.

To exemplify, assume that sizes of I_1 and I_2 belong to K^*. Note that OP does not worry about the actual links that will constitute the SS sets. Therefore, it is likely that the size I_1 SS set that has a throughput $\text{BEST}(l_1)$ has one or more links in common with the size I_2 SS set that has a throughput of $\text{BEST}(I_2)$.

One scenario in which a network will achieve the upper bound returned by OIP is when the optimizer I^* suggests that all links must share in space (one SS set that contains all links).

Consider an example network consisting of $N = 6$ links, with no two links sharing an AP. The pairwise SIR (dB) matrix $\mathcal{SIR}$ for the network is

$$\mathcal{SIR} = \begin{bmatrix} - & \text{-6.02} & 7.96 & 9.54 & 19.55 & 21.94 \\ & - & 6.02 & 7.96 & 19.08 & 21.58 \\ & & - & -6.02 & 16.9 & 20.0 \\ & & & - & 16.26 & 19.55 \\ & & & & - & 9.54 \\ & & & & & - \end{bmatrix}. \tag{5.4}$$

The matrix is symmetric and we show only the upper triangle. The diagonal corresponds to links' self-interference and is not applicable as the links are half-duplex. Figure 5.1 plots the maximum throughput BEST(i), for a network of six links with no two links sharing an AP, that is achieved when i of the 6 links transmit simultaneously, for $1 \le i \le 6$. The total number of integer partitions, that is the size of the set IP(6), is 11. The optimizer $I^* = (2,2,2)$. Therefore, Optimal-Integer-Partition suggests that the throughput of the network can be maximized by selecting three SS sets (may or may not be mutually exclusive and collectively exhaustive), each of size 2. The upper bound on throughput is $3 \times \text{BEST}(2) = 13.82$ bits every average slot length

$\overline{\sigma}$. This is a throughput gain of 38% over links operating as per standard DCF under which each link gets a throughput of BEST(1).

5.0.3 Heuristic Greedy Approach to find Upper Bound:

To find an upper bound in polynomial time, we propose a heuristic based algorithm. The algorithm reduces the search space used to find the upper bound. This algorithm does not guarantee to return same result as of OIP but for our experiments this algorithm performs quite well. Error with respect to OIP is tabulated in Section 8.3.1.

We can represent this algorithm by a tree data structure. Consider each subset creation part as generating nodes of a tree, and number of element in a set represent its level. We start with calculating value of BEST(N), which is the value of the root node and then proceed toward its leaves. To move a level down in tree we find possible nodes (Possible Subset $\widehat{PS}$), that is subset of size equal to level. We then calculate throughput associated with these nodes and keep only K nodes (forming $\widehat{P}$) having maximum throughput at this level. We discard all other nodes of this level, which means that subtrees are discarded. One level down, we search for BEST only on children of selected parents of current level. In this way our search space is reduced to K subset per level, where $K \leq \binom{N}{level}$.

If at each level we select N subsets (nodes), the Complexity of this algorithm is $O(N^5)$, If we select N^2 subsets at each level the complexity of this algorithm becomes $O(N^9)$.

Algorithm 1: Heuristic Greedy Approach to find Upper Bound

Input: $\mathcal{N}$ `// Set of links in the network`
Output: BEST `// Maximum of the calculated throughputs for [1,2...N] SS set size`
$\widehat{P} = \{1, 2..., N\}$; `// Initialize`
$\overline{\mathcal{N}} = \mathcal{N}$;
for $i = 0$ *to* $|\overline{\mathcal{N}}|$ **do**
  $\widehat{PS} = \{S | S \subset \widehat{P}(K) \wedge |S| = |N| - i, \forall K\}$
  $[\widehat{PS}, T]$ = Calculate throughput associated with each subset of $\widehat{PS}$
  $[\widehat{PS}, T] = Sort[\widehat{PS}, T]$ in decreasing order of throughput(T)
  $\widehat{P}$ = Keep K subsets from $\widehat{PS}$, with maximum throughput and discard others.
  BEST(N-i)= Max(T) `// maximum Throughput for SS set of size N-i`
end

Chapter 6

Approaching the Upper Bound

Optimal-Integer-Partition (5.1)-(5.2) gives an upper bound on throughput that is in general not achievable. We propose two algorithms: (1) Reevaluate Optimal-Integer-Partition and (2) Begin Sharing in Space. Later, in Chapter 8, we will illustrate via simulations that both algorithms achieve a large fraction of the gains promised by the upper bound.

6.0.4 Reevaluate Optimal Integer Partition (ROIP)

Our aim is to solve (4.2)-(4.3) to find the optimal partition P^* of the network $\mathcal{N}$. However, the problem has an exponential computational complexity. ROIP finds a partition $\widehat{P}$ of $\mathcal{N}$ by reevaluating OIP on links in $\mathcal{N}$ that do not belong to a SS set that has already been assigned to $\widehat{P}$. By doing so it leverages our ability to solve OIP, without violating the requirement (4.3) that the chosen SS sets form a partition of the network $\mathcal{N}$.

Algorithm 2: Reeval-Optimal-Integer-Partition (ROIP)

Input: $\mathcal{N}$ `// Set of links in the network`
Output: $\widehat{P}$ `// Partition of the network` $\mathcal{N}$
$\widehat{P} = \emptyset$; `// Initialize`
$\overline{\mathcal{N}} = \mathcal{N}$;
while $\overline{\mathcal{N}} \neq \emptyset$ **do**
  $[I^*, \mathrm{UB}] =$ Optimal-Integer-Partition($|\overline{\mathcal{N}}|, \overline{\mathcal{N}}$);`// Solve Optimal-Integer-Partition for the network` $\overline{\mathcal{N}}$
  `//` $I^* \in$ `IP(`$|\overline{\mathcal{N}}|$`) is the optimizer and UB is the optimum value, which is the upper bound on the achievable throughput for the set of links` $\overline{\mathcal{N}}$
  $I_{\max} = \max(I^*)$;
  `//` $I_{\max}$ `is the largest integer in the partition` I^*
  $S =$ SS set that has throughput BEST($I_{\max}$);
  $\overline{\mathcal{N}} = \overline{\mathcal{N}} \setminus S$;
  $\widehat{P} = \widehat{P} \cup S$;
end

Algorithm 2 summarizes the approach. Let $\overline{\mathcal{N}}$ be the set of links in the network that have not been added to $\widehat{P}$. We start with finding the optimal integer partition I^* of $|\overline{\mathcal{N}}|$, by solving (5.1)-

(5.2). Note that solving the problem requires calculation of BEST(i), $1 \le i \le |\overline{\mathcal{N}}|$, for the set of links $\overline{\mathcal{N}}$. Let $I^* = \{I_1, I_2, \ldots, I_{|I^*|}\}$. Let $I_{\max}$ be the maximum of $I_1, I_2, \ldots, I_{|I^*|}$. Select a set of $I_{\max}$ links from the network $\overline{\mathcal{N}}$, such that throughput of the selected set is BEST($I_{\max}$). At least one such set will exist. The selected set is an SS set and is added to the partition $\widehat{P}$. The above process is now repeated. When the algorithm terminates, it returns $\widehat{P}$, which contains mutually exclusive and collectively exhaustive SS sets. Finally, note that the choice of picking the largest size $I_{\max}$ set is motivated by our simulations based observations that it is better than other choices like picking the smallest.

For our six link example in Chapter 5, ROIP chooses a partition of $(2, 2, 1, 1)$. Recall that OIP had chosen $(2, 2, 2)$. The SS sets chosen by ROIP are $\{1, 6\}$, $\{2, 5\}$, $\{3\}$, and $\{4\}$. Note that $\{1, 6\}$ has a throughput of BEST(2). However, all other pairs of links have a smaller throughput. Given that mutually exclusive SS sets must be chosen, ROIP ends up choosing an integer partition of $N = 6$ that is different from the one chosen by OIP. While OIP had suggested an upper bound of 38.6% gain over standard DCF, ROIP achieves a gain of 20.6%.

The time complexity of ROIP is dependent on that of OIP. Recall that OIP requires calculation of BEST(i) over $1 \le i \le N$. The size of the problem OIP can become very large even for moderate N. We have considered two approximations of OIP: *ApproxN*, which limits the search space to $O(N)$ and has a time complexity of $O(N^5)$, and *ApproxN*^{2}, which has corresponding values of $O(N^2)$ and $O(N^9)$.

ROIP takes a maximum of $N/2$ iterations. The time complexity of ROIP is $O(NC)$ when the approximation of OIP is $O(C)$. Therefore, ROIP has a time complexity of $O(N^6)$ when *ApproxN* is used, and that of $O(N^{10})$ when *ApproxN*^{2} is used. Next we will describe *Begin Sharing in Space*, which has a much better worst case time complexity of $O(N^3)$.

6.0.5 Begin Sharing in Space (BSS)

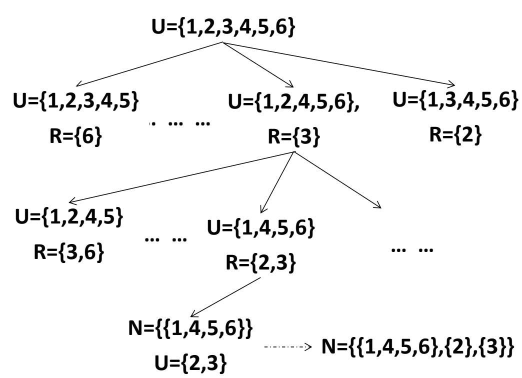


Figure 6.1: Demonstration of Begin Sharing in Space using a 6 link topology with link SNR = 1000. The topology was introduced in Chapter 6. BEST(i) for this network was shown in Figure 5.1.

Algorithm 3 summarizes the approach. Unlike ROIP, BSS does not solve Optimal-Integer-Partition. It starts with the set $\overline{\mathcal{N}}$, which is initialized to the network $\mathcal{N}$. The throughputs of

Algorithm 3: Begin Sharing in Space (BSS)

```
Input: 𝒩: // Set of links in the network
Output: P̂ // Partition of the network 𝒩
P̂ = ∅; // Initialize
N̄ = 𝒩;
R = ∅;
while N̄ ≠ ∅ do
    for k = 1 to |N̄| do
        P1 = N̄(k);// Set P1 to the k^th element (link) in set N̄.
        P2 = N̄ \ P1;
        T(k) = T_P1 + T_P2;// Throughputs of sets P1 and P2 calculated using
            Equation (3.4)
    end
    k* = arg max_{k∈{1,...,|N̄|}} T(k);
    // T_N̄ = 0 if N̄ contains links that share AP(s)
    if T_N̄ < T(k*) then
        R = R ∪ N̄(k*);
        N̄ = N̄ \ N̄(k*);
    else
        P̂ = P̂ ∪ N̄;
        P̂ = P̂ ∪ BSS(R);
        break;
    end
end
```

all possible partitions of $\overline{\mathcal{N}}$ into mutually exclusive sets of size 1 and $|\overline{\mathcal{N}}|-1$ are calculated. The maximum of these throughputs is compared with the throughput $T_{\overline{\mathcal{N}}}$ of $\overline{\mathcal{N}}$.

If the maximum is larger than $T_{\overline{\mathcal{N}}}$, the set $\overline{\mathcal{N}}$ is set to the size $|\overline{\mathcal{N}}|-1$ set from the partition that gave the maximum. The single link from the partition is added to R. The process is then repeated. If the maximum is smaller, $\overline{\mathcal{N}}$ is added to partition $\widehat{P}$, $\overline{\mathcal{N}}$ is set to R, and the process is repeated. If $\overline{\mathcal{N}}$ contains links that cannot share in space, BSS sets its throughput to 0.

Figure 6.1 demonstrates the working of BSS for the six link topology described in Chapter 6. The algorithm begins at level 1 (top most) with all 6 links sharing-in-space. The nodes at the next level in the tree consist of 2 SS sets of sizes 5 and 1. There are a total of 6 nodes of which 3 are shown. Partitioning the links into the SS sets $\{1,2,4,5,6\}$ and 3 yields the largest throughput. Further, the split of $\{1,4,5,6\}$ and 2 is the best split for $\{1,2,4,5,6\}$ and yields larger throughput. The algorithm moves to level 3. We have $U=\{1,4,5,6\}$ and $R=\{2,3\}$. The throughput of U cannot be improved by splitting. It becomes the first SS set returned by the algorithm. The above procedure (not shown in figure) is now carried out on $R=\{2,3\}$ and gives us the two SS sets $\{2\}$ and $\{3\}$. A throughput gain of 14% is obtained over standard DCF.

The time complexity of BSS is $O(N^3)$. The tree traversal takes $O(N^2)$ time. Also, when the partition $\widehat{P}$ ends up with N SS sets, the tree is traversed N times.

Chapter 7

Simulation Setup

Our network consists of 25 clients that are distributed uniformly and independently over a two dimensional region. We choose regions of different areas. Specifically, we choose different values of average area/client. Network topologies are simulated where on an average one client is placed every 2, 4, 8, 16, 32, and 50 m^2. To simulate very high density of access points over small regions, we place an AP at a height of 2 m above each client. A client is associated with the AP placed above it. Example placement of 2 clients and their respective AP(s) is shown in Figure 3.4(c). The bi-directional link between each client and AP has a given SNR. We choose SNR values of $1, 10, 100$ and 1000. All results are averages over 50 network instances generated for every SNR and area/client selection.

To keep the simulation time of OIP manageable, we do not distinguish between the forward direction (link from an AP to its client) and the reverse direction (link from the client to its AP). Specifically, while links in both directions can seize data transmission opportunities as per 802.11 DCF, they are treated as one entity by the optimization problem when scheduling them as part of a SS set[1]. Recall that links that share an AP cannot share-in-space. Thus clubbing the forward and reverse directions does not reduce the available spatial multiplexing gains. We do need a redefinition of combined link's SINR, however. Recall from Chapter 3 that calculation of SINR requires pairwise SIR values. For a combined link, the pairwise SIR is assumed to be the smallest SIR obtained under the assumption that a pair of combined links are transmitting simultaneously and each of the combined links can be transmitting in either the forward or the reverse direction. Combining link directions is thus likely to lead to a larger penalty due to interference when simultaneous transmissions take place, leading to smaller throughput gains.

Let $T_{\mathcal{N}}$ be the throughput of the network when using the proposed modified DCF that allows sharing in space and let $\widetilde{T}_{\mathcal{N}}$ be the throughput when using standard DCF. Note that $T_{\mathcal{N}}$ is calculated using (4.1). Equation (4.1) requires knowledge of the network partition, which is calculated using OIP, ROIP or BSS. $\widetilde{T}_{\mathcal{N}}$ is calculated by substituting $S = \mathcal{N}$ in (3.5). Throughput

[1]When a link from a SS set seizes a data transmission opportunity, the direction of transmission of the other links may be chosen as desired and does not affect our gains. For example, both directions may be given equal opportunity to piggyback on data transmission opportunities by alternating between the forward and the reverse.

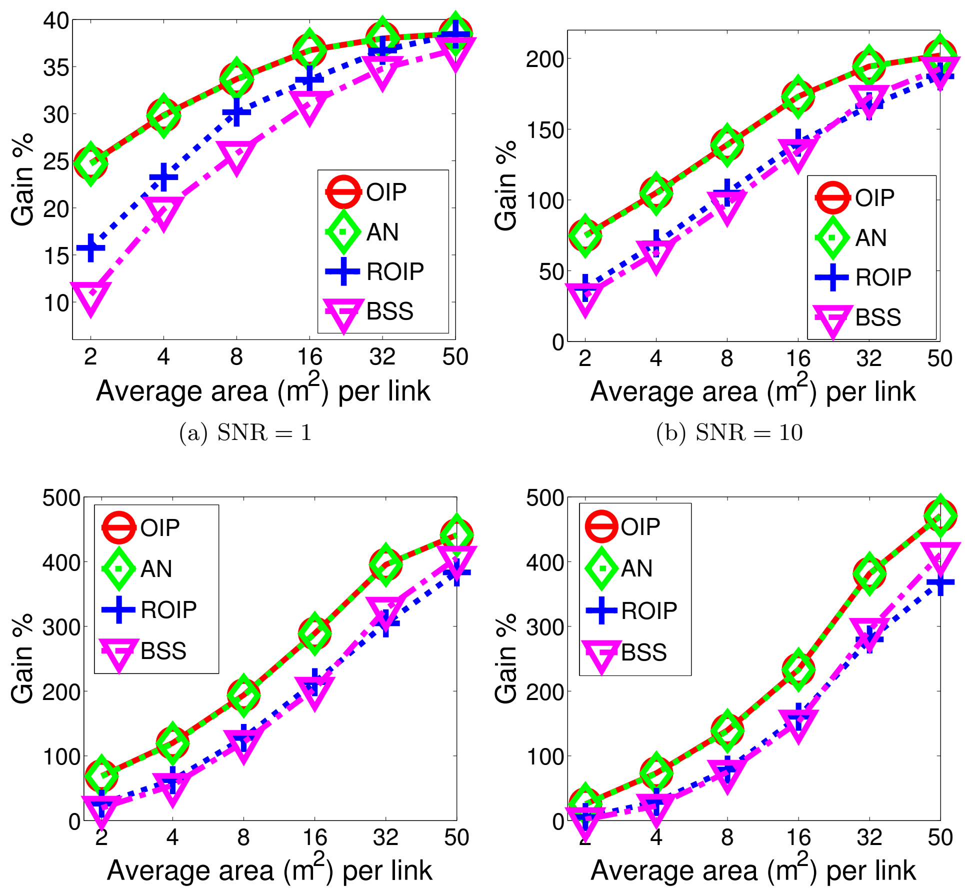
gains (%) are given by $(T_{\mathcal{N}} - \widetilde{T}_{\mathcal{N}}) \times 100/\widetilde{T}_{\mathcal{N}}$.

Wireless propagation is assumed to follow the pathloss model. For every simulated network topology, we calculate the gains corresponding to the exact upperbound on throughput given by OIP, approximation of the same given by *ApproxN* (AN), see Chapter 6, and throughput gains achieved when using ROIP and BSS. We ran simulations for pathloss exponents of $\alpha = 2, 3$.

For understanding how throughput changes with number of access points, we need to fix the positions of the AP(s), and select the clients that will associate with them. Given N clients distributed as above uniformly and independently over a two dimensional region.we group the clients into M clusters using k-mean clustering. One AP is assigned to each cluster. The assigned AP is placed 2 m above the midpoint of its clients' cluster. SNR of the each client is now function of the distance between client and the AP location. For finding links in SS set, the restriction is that, SS set can have nodes from each cluster but only one node of a cluster can become part of one SS set.

Chapter 8

Results



(a) SNR = 1 (b) SNR = 10

(c) SNR = 100 (d) SNR = 1000

Figure 8.1: The plots show the upper bound on throughput gains calculated by OIP, its approximation *ApproxN* (AN), gains from ROIP and BSS. Pathloss exponent is 3. A m^2 per link implies that on an average we have a link every A m^2. The network consists of 25 AP(s) and each AP has one client.

Figure 8.1 shows throughput gains obtained for $\alpha = 3$. Gains for $\alpha = 2$ are of a similar order of magnitude. They are a bit smaller than gains obtained for $\alpha = 3$, which can be explained by smaller SINR(s) due to less attenuation of interference between links in the network when $\alpha = 2$. From the figure it is clear that ROIP and BSS achieve a large fraction of the upperbound on gains. AN is seen to approximate OP very well. For the results shown, we used AN to solve Optimal-Integer-Partition for ROIP. Even for modest link SNR(s) of 10 and high link densities (say one link every 8 m^2) gains of about 100% are achieved by BSS and ROIP. But for its much lower time complexity, BSS compares well to ROIP.

Gains increase with decreasing density (links are farther apart) of links because interference between links reduces. Low SNR (say SNR = 1) leads to low SINR(s) and allowing simultaneous transmissions lead to smaller improvements in throughput. This is because when SINR(s) are small link throughputs do not benefit from multiplexing gains from simultaneous transmissions and most links transmit data alone. The chosen partition consists of many singleton SS sets.

8.1 Effect of Increasing SNR

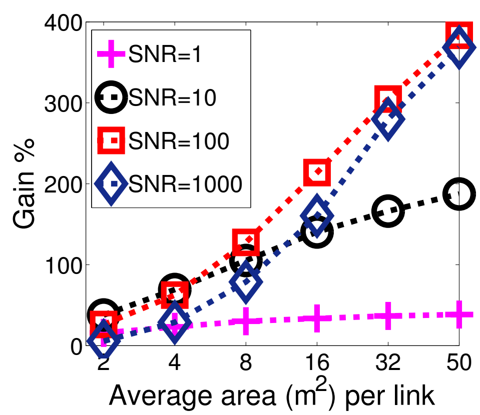


Figure 8.2: The plot showing together the throughput gains (shown separately in Figures 8.1a-8.1d) given by ROIP for different link SNR(s).

As seen in Figure 8.2, increasing SNR (and hence interference) does not translate into throughput gains. At SNR=100, throughput gains are more than SNR=1000 and the clear reason is that increasing SNR will also increase interference on the links sharing-in-space. If we want to increase spatial reuse by piggybacking, we have to decide on SNR value also. This hints at the need for power control.

8.2 SINR to PHY rate mapping for a 802.11a receiver

Figure 8.3 shows the corresponding results when using the SINR to PHY rate mapping for a 802.11a receiver, shown in Table 8.1, instead of the Shannon's capacity formula we used in (3.4) and (3.5). The gains obtained are significant.In the figure, SNR= 1000 outperforms SNR= 100 for less dense networks, but for larger densities SNR= 100 is a better choice.

Table 8.1: SINR-Data rates mapping for IEEE 802.11a standard as specified in Table II of [18].

Rates (Mbps)	$SINR$ Threshold β (dB)
54	24.56
48	24.05
36	18.80
24	17.04
18	10.79
12	9.03
9	7.78
6	6.02

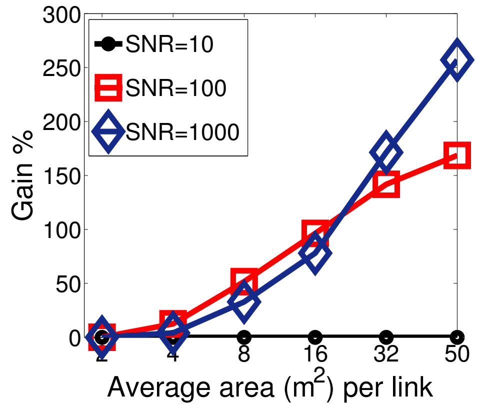


Figure 8.3: The plot to shows throughput gains obtained, for different link SNR settings, on using the 802.11a SINR to PHY rate mapping. Note that we skip SNR = 1 since 802.11a supports a minimum rate of 6.5 Mbps at a minimum SINR of about 6dB.

8.3 Zooming in to the Results

We will demonstrate the impact of the optimization on a per link basis. We will evaluate fairness, look at the distribution of the size of SS sets, and last but not the least, present results on the efficacy of the approximation algorithms.

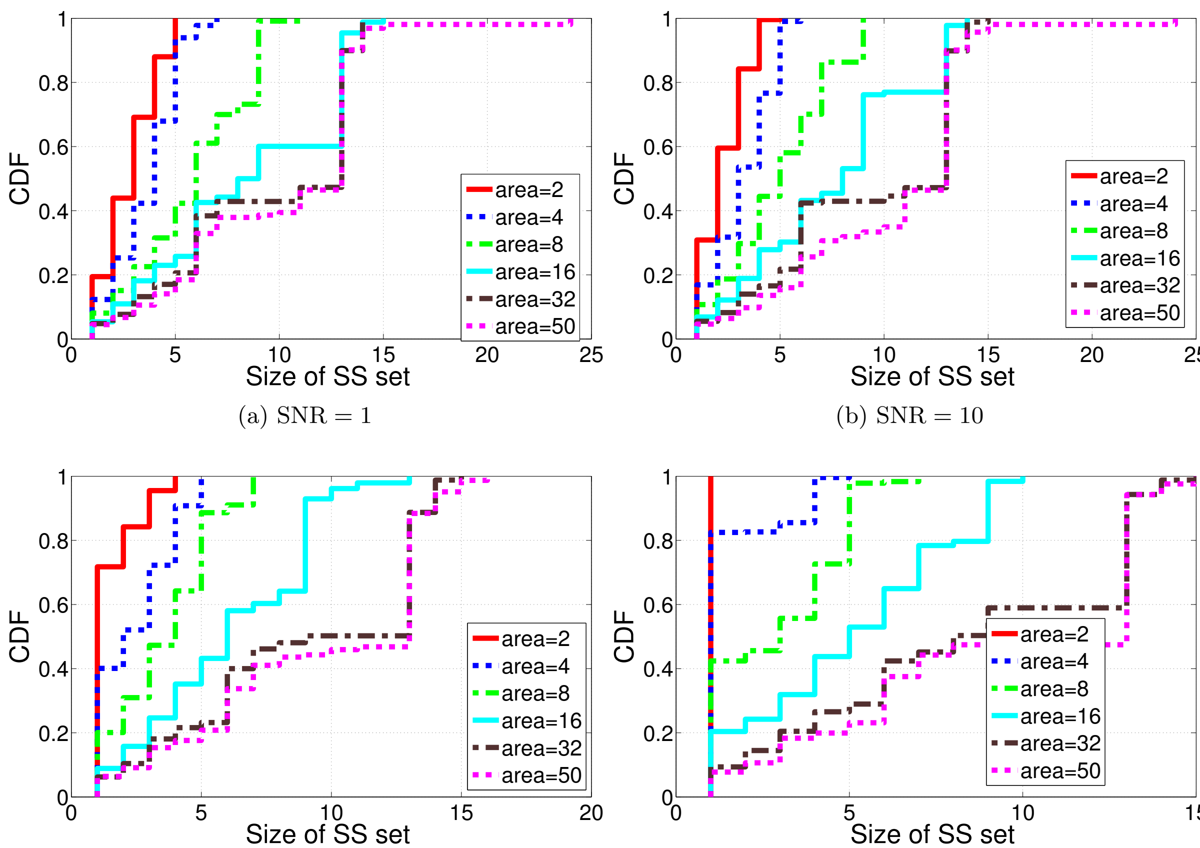
8.3.1 Error in Calculating Upper Bound with Heuristic Greedy Approach

Table 8.2: Percentage error incurred in calculation of the upper bound (5.1)-(5.2) when using *ApproxN* (AN) and *ApproxN*^{2} (AN^2).

	σ=0		σ=4		σ=8	
	AN	AN^2	AN	AN^2	AN	AN^2
Mean	0.32	0.27	2.10	0.39	2.4	0.36
90^{th} pct	0.80	0.00	6.38	0.02	7.94	0.02
95^{th} pct	1.53	0.00	12.09	1.07	13.90	0.98
98^{th} pct	3.50	0.009	20.57	6.81	19.42	7.64
Max	28.34	12.27	51.41	34.33	35.49	23.12

Finally, in Figure 8.1a - 8.1d, note that the approximation *ApproxN* of OP does not lead to significant differences in gains with respect to OP and may be used in lieu of OP. Table 8.2 tabulates the percentage errors, with respect to the actual solution of OP, when using $ApproxN$ and $ApproxN^2$. The approximations are evaluated for the pathloss model of propagation ($\sigma^2 = 0$) and also assuming log normal shadowing ($\sigma^2 = 4, 8$ dB) [12]. We are showing results averaged over 50 topologies for each SNR and density pair. The mean errors are found to be small. However, we do find outliers that lead to large errors.

8.3.2 Size of SS Set using ROIP



(a) SNR = 1 (b) SNR = 10

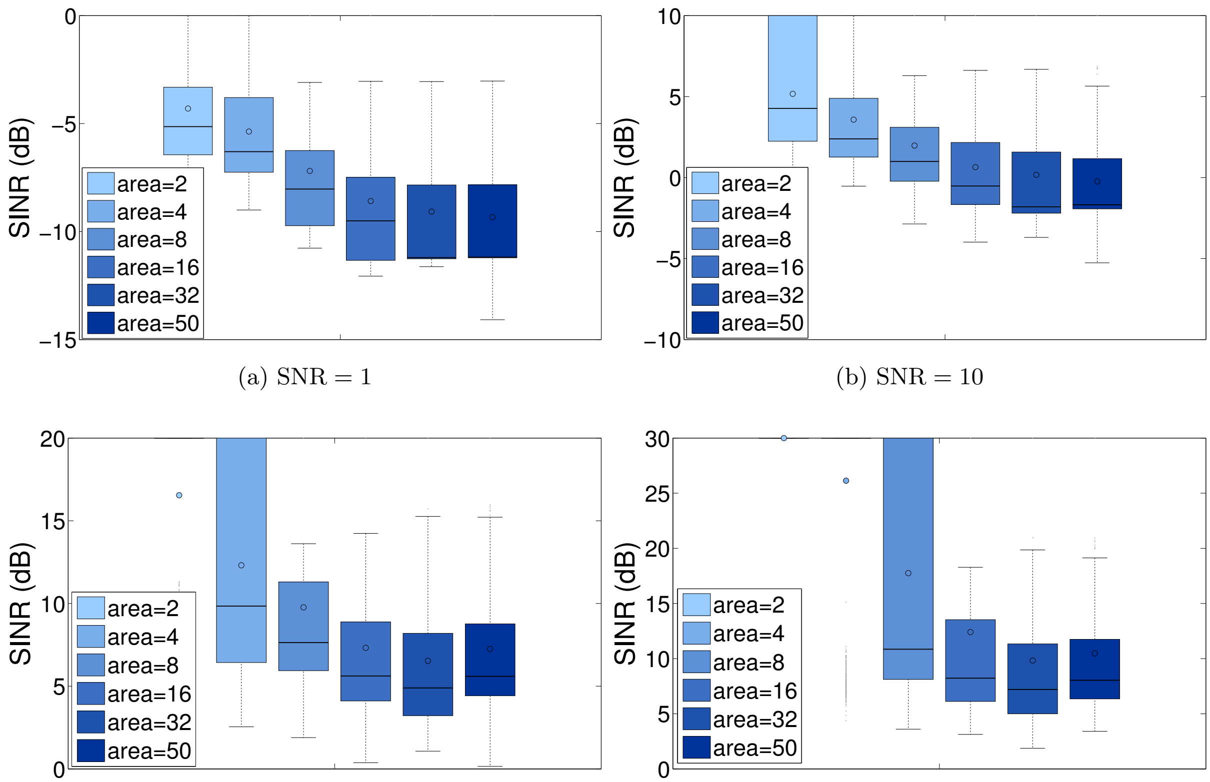
(c) SNR = 100 (d) SNR = 1000

Figure 8.4: Size of SS sets using ROIP

Figure 8.4a - 8.4d shows the distribution (CDF) of the size of SS sets in partitions returned by ROIP for each of the six network densities. For very dense networks, where AP(s) are located nearby to each other, the SS set size is small. For less dense networks we see more links in SS set, going all the way upto 25.

SS set size is dependent on SNR and density of the network. On increasing SNR, SS set size of less dense network increases, as more transmit opportunities due to piggybacking overcomes the drawback of increased interference. For highly dense networks, size of SS set reduces when SNR increases because the effects of increased interference dominate.

8.3.3 SINR of links on using ROIP to calculate the SS sets



(a) SNR = 1 (b) SNR = 10

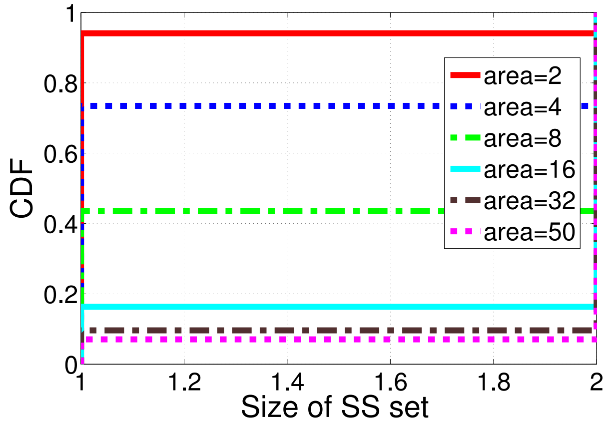
(c) SNR = 100 (d) SNR = 1000

Figure 8.5: SINR of links while SS sets are created using ROIP.

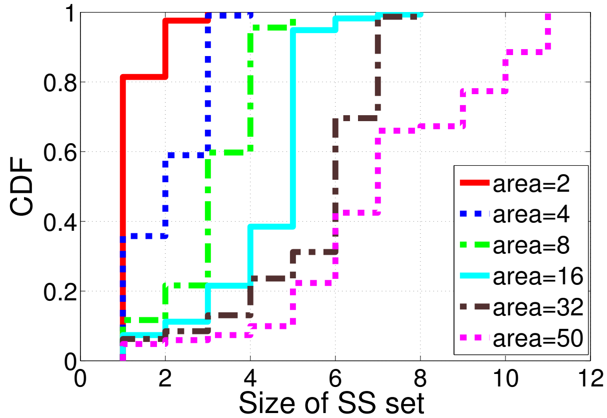
Figure 8.5a - 8.5d shows the boxplots of SINR (dB) of the links when scheduled by ROIP. For the highly dense networks, links do not share-in-space. Hence, SINR is close to SNR. As the number of links in SS set increases the SINR drops, but throughput gains increase due to piggybacking.

If we consider area= $50m^2$, at SNR=100 (see Figure 8.5c) the SINR value is reducing but overall gain is increasing due to increased size of SS sets. On SNR=1000 (see Figure 8.5c) the SINR value is increased but SS set size is reduced and gains are nearby equal to SNR=100.

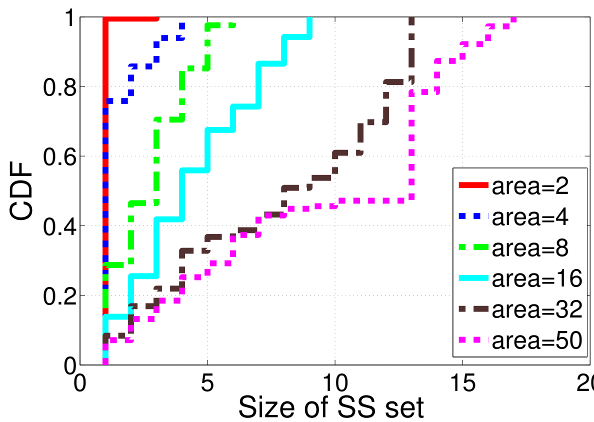
8.3.4 Size of SS Sets when using Discrete Mapping as specified in Table 8.1



(a) SNR = 10



(b) SNR = 100



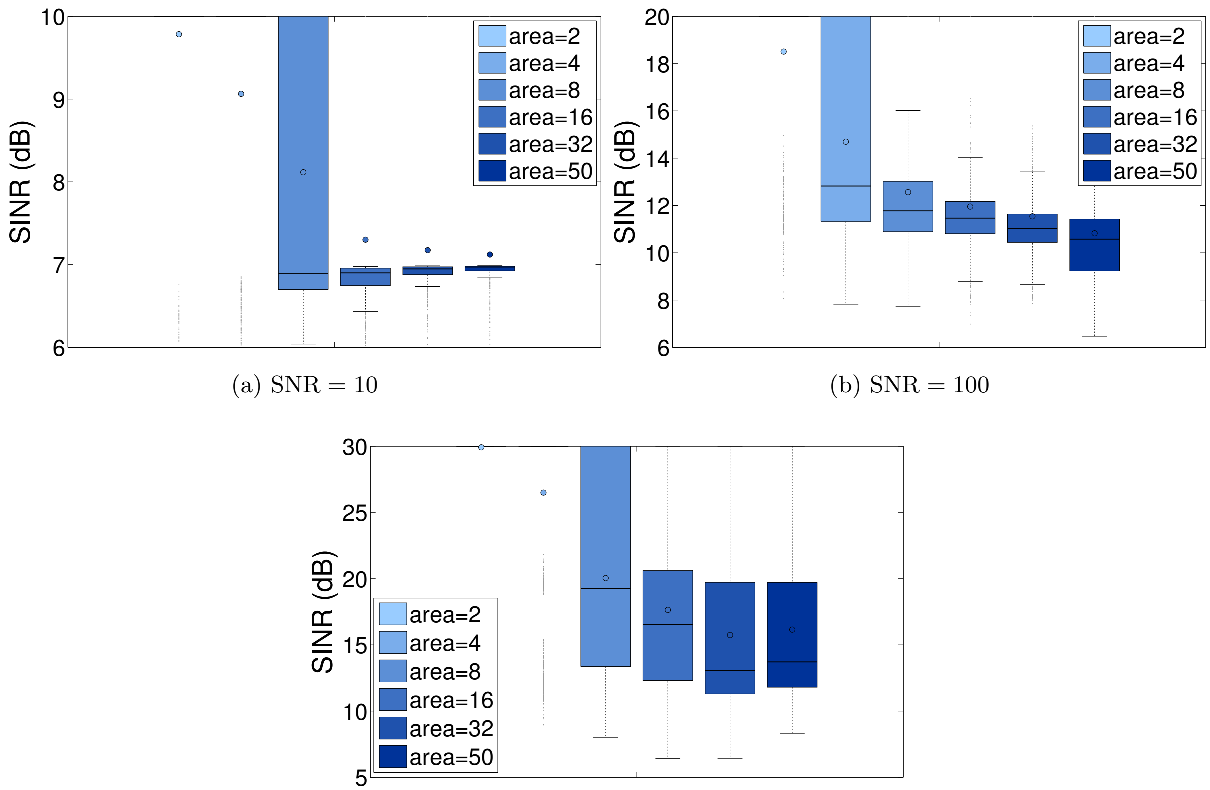
(c) SNR = 1000

Figure 8.6: Size of SS Sets created using Discrete Mapping as shown in Table 8.1.

There is not any rate mapping for SNR=1 (0dB) for 802.11a as seen in Table 8.1. Figure 8.6a - 8.6c shows the size of SS sets for SNR(s) of 10,100 and 1000 respectively.

8.3.5 SINR of links when using Discrete Mapping as specified in Table 8.1

Figure 8.7a- 8.7c show the SINR corresponding to Discrete rate mapping for SNR(s) 10,100 and 1000 respectively. We see a trend in the figures. As the network becomes less dense or highly dense, choice of SINR reduces. In highly dense networks SINR is quite close to SNR, as most of the links do not want to share. In less dense network SINR is quite less than SNR, as most of the links benefit from piggybacking on transmission opportunities by others.



(a) SNR = 10

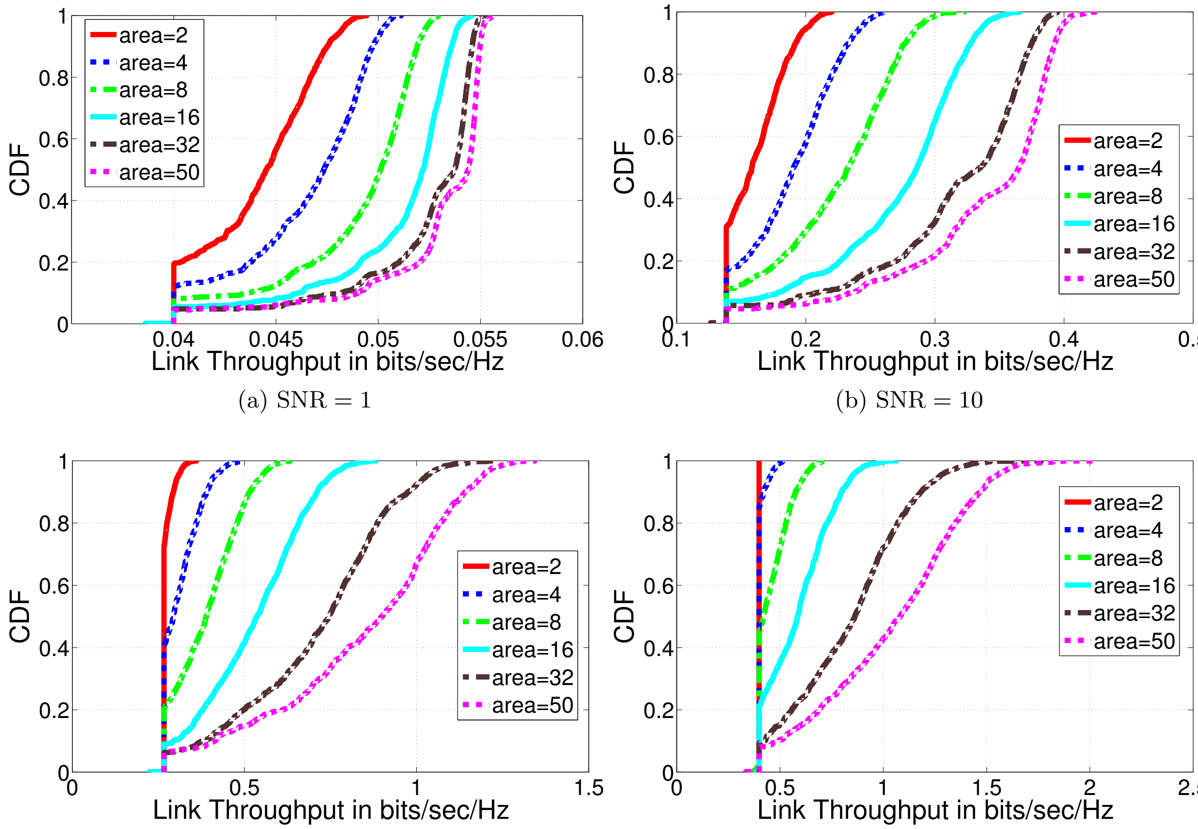
(b) SNR = 100

(c) SNR = 1000

Figure 8.7: SINR of links when SS sets are created using Discrete Mapping as shown in Table 8.1.

8.4 Fairness Index

In this section we shows the result of throughput achieved by link when topology links are partitioned by using ROIP and the discrete mapping as specified in Table 8.1. We are also showing result of Jain's fairness index. Fairness value is calculated over each topology and average of values is shown in tables.



(a) SNR = 1
(b) SNR = 10
(c) SNR = 100
(d) SNR = 1000

Figure 8.8: Throughput of the links according to ROIP.

8.4.1 Link Throughput Fairness using ROIP

Figure 8.8a - 8.8d show the throughput of links, using ROIP we get upto 4x gains in throughput of a link. There are a few outliers that suffer a reduction in comparison to their throughput when using standard 802.11 DCF.

There are few outliers suffering loss in throughput, but maximum loss is 16% and only 0.3% links are suffering this loss. The links that suffer a loss in throughput are seen to be a part of large SS sets.

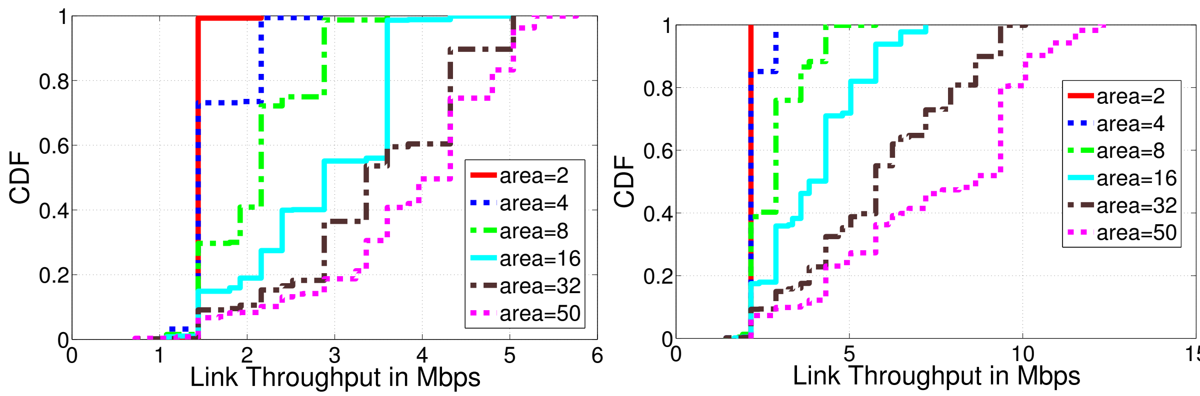
From Table 8.3 showing Jain's fairness index of the all SNR and node densities for ROIP, we conclude that partitioning by using ROIP ensures fairness.

Table 8.3: Link Throughput Fairness Index Using ROIP.

SNR	area=2	area=4	area=8	area=16	area=32	area=50
1	0.9971	0.9971	0.9972	0.9968	0.9977	0.9976
10	0.9838	0.9763	0.9723	0.9689	0.9674	0.9723
100	0.9949	0.9761	0.9555	0.9273	0.9071	0.9159
1000	0.9998	0.9966	0.9720	0.9373	0.8982	0.8801

We have also evaluated results for BSS added as Appendix A. In BSS SS set size increases. At SNR=1 and 10, we do not get any outliers. At SNR=100, we get 1.4% outliers with the maximum loss of 8%. At SNR=1000, we get 4.09% outliers with maximum loss of 17%. Maximum number of outliers belongs to intermediate density networks.

8.4.2 Link Throughput Fairness using Discrete Mapping as shown in Table 8.1



(a) SNR = 100

(b) SNR = 1000

Figure 8.9: Throughput of the links according to 802.11a mapping Table 8.1

Table 8.4: Fairness Index using Discrete Mapping as shown in Table 8.1.

SNR	area=2	area=4	area=8	area=16	area=32	area=50
10	1	1	1	1	1	1
100	0.9770	0.9268	0.9078	0.8820	0.8836	0.8726
1000	1.0000	0.9714	0.9030	0.8580	0.8233	0.8228

For SNR=10 (10dB), all links get equal throughput. SINR we get is in range of (6-10 dB) and maximum SS set size is equal to 2. If SS set size is 1, SINR of the link will be equal to SNR and link can transmit with the rate of 12 Mbps. When links share-in-space, they can choose only one link to share, on sharing they both can transmit at the rate of 6 Mbps and due to piggybacking they again get overall throughput of .48 Mbps. We did not get any outliers at SNR=10.

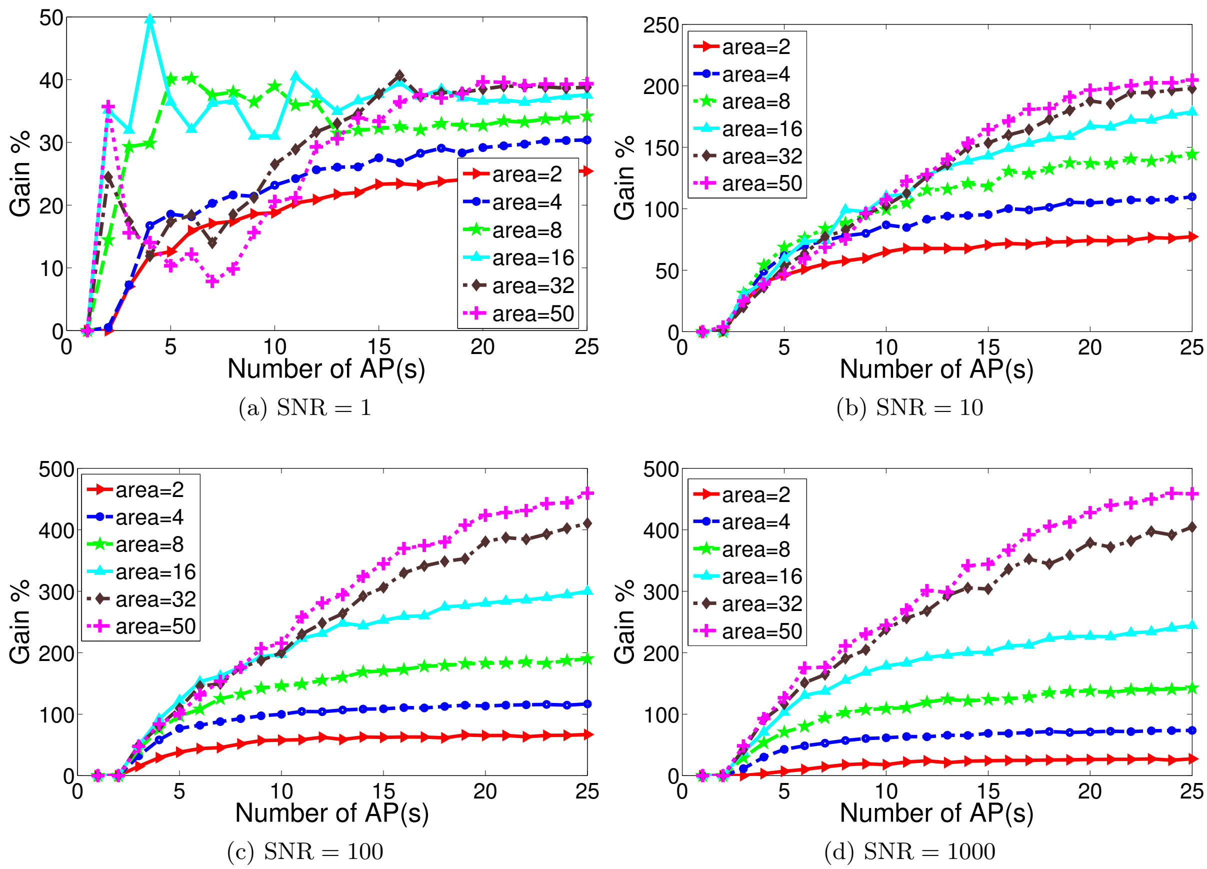
Figure 8.9a - 8.9b show throughput corresponding to SNR=100 and 1000 respectively. At SNR=100, if links share in time, they get throughput of (36/25)=1.44 Mbps. On Discrete mapping minimum link throughput for SNR=100 is 0.72 Mbps. It is about 50% loss in actual throughput of link. But only 1.01% links suffer this loss in throughput. We are getting upto 4x improvement in a link throughput.

At SNR=1000, we are getting gain of upto 5x gain in throughput of a link. We are getting few outliers also. On time sharing each link should get (54/25)=2.7 Mbps. We are getting minimum throughput of 1.44 Mbps. which is loss of about 46%, but only 0.5% links are suffering this loss in throughput.

In Table 8.4, we show jain's fairness index value of each SNR and density pair. All values are averaged over 50 topologies. The result shows that creating SS set using discrete mapping leads to throughput fairness.

8.5 How many Access Points?

Figures 8.10a - 8.10d plot gains in throughput as a function of number of access points refer to Chapter 7 for details on AP placement and client association. Each figure corresponds to a different nearby-SNR. Note that for large densities of $2m^2$ and $4m^2$ the gains flatten very quickly as a function of the number of access points. In fact, about 3-5 access points for a given network suffice and adding more AP(s) does not increase the gains much. However, for a low density of 0.1, gains accrue as more AP(s) are added.



(a) SNR = 1

(b) SNR = 10

(c) SNR = 100

(d) SNR = 1000

Figure 8.10: The plots show the tradeoff between the number of access points and throughput gains, for different nearby-SNR(s) and client densities.

Chapter 9

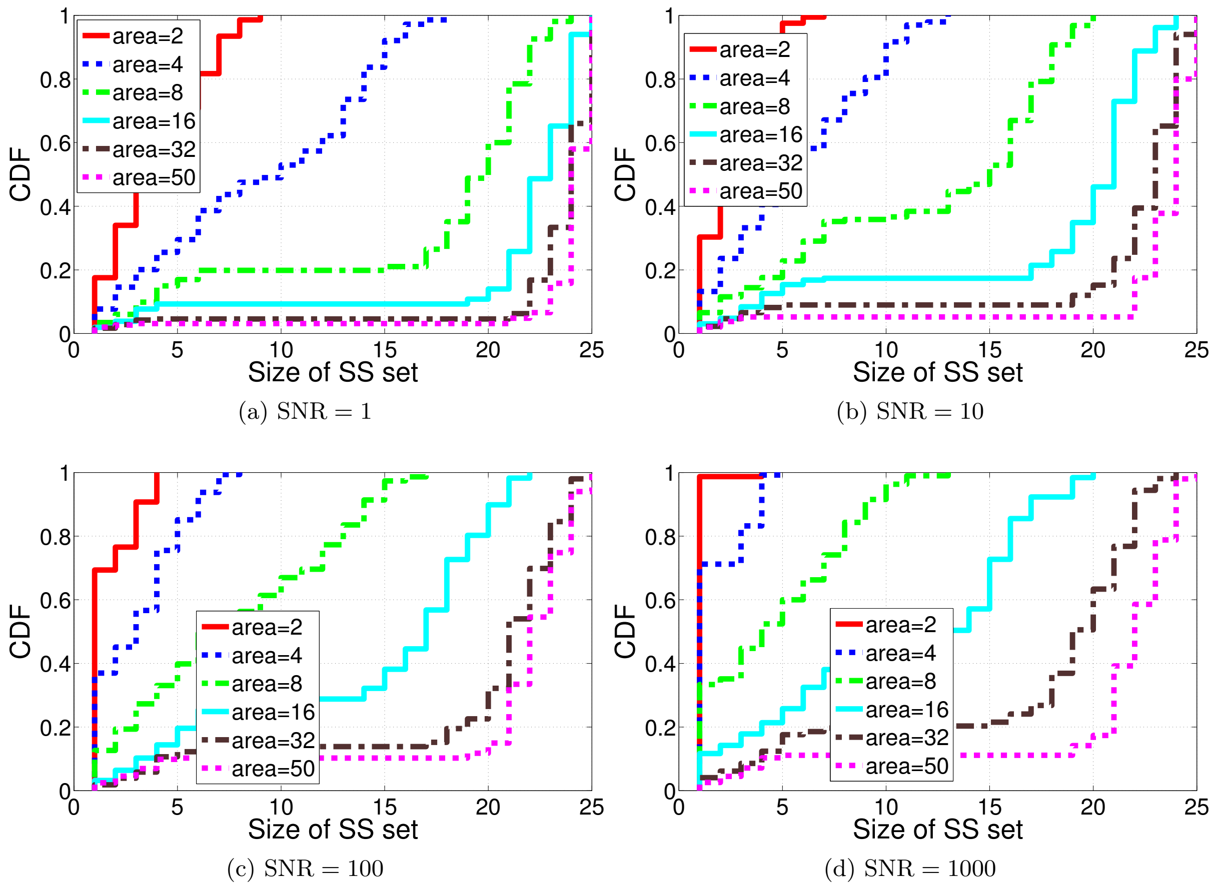
Limitations, Conclusion and Future Work

We extended the standard 802.11 DCF model to allow for simultaneous data transmissions. We formulated a network throughput maximization problem with the goal of finding links that must transmit simultaneously so that the network throughput is maximized. Relaxation of this NP-hard problem allowed us to calculate an upper bound on network throughput. We proposed two computationally feasible algorithms and demonstrated their efficacy via simulations. Very large gains of up to 400% were achieved for networks with 25 clients and AP(s) over standard 802.11 DCF.

Future work includes considering general 802.11 networks, real flows (TCP), mobility, and measurements of SINR(s) in high density settings. Specifically, in our work we consider only mutually exclusive and collectively exhaustive (MECE) sets. We are currently evaluating the impact of relaxing this constraint. Also, in our evaluation, we assume that all links are saturated. The work can be extended for unsaturated flows in a straightforward manner. We are conducting more detailed evaluations using TCP and UDP traffic. In our study we have assumed the pathloss model for wireless propagation. We plan to conduct real world experiments to measure SINR values in environments with large number of access points and clients. The values obtained will then be used to evaluate gains in throughput when using the proposed algorithms.

Appendix A

BSS Results



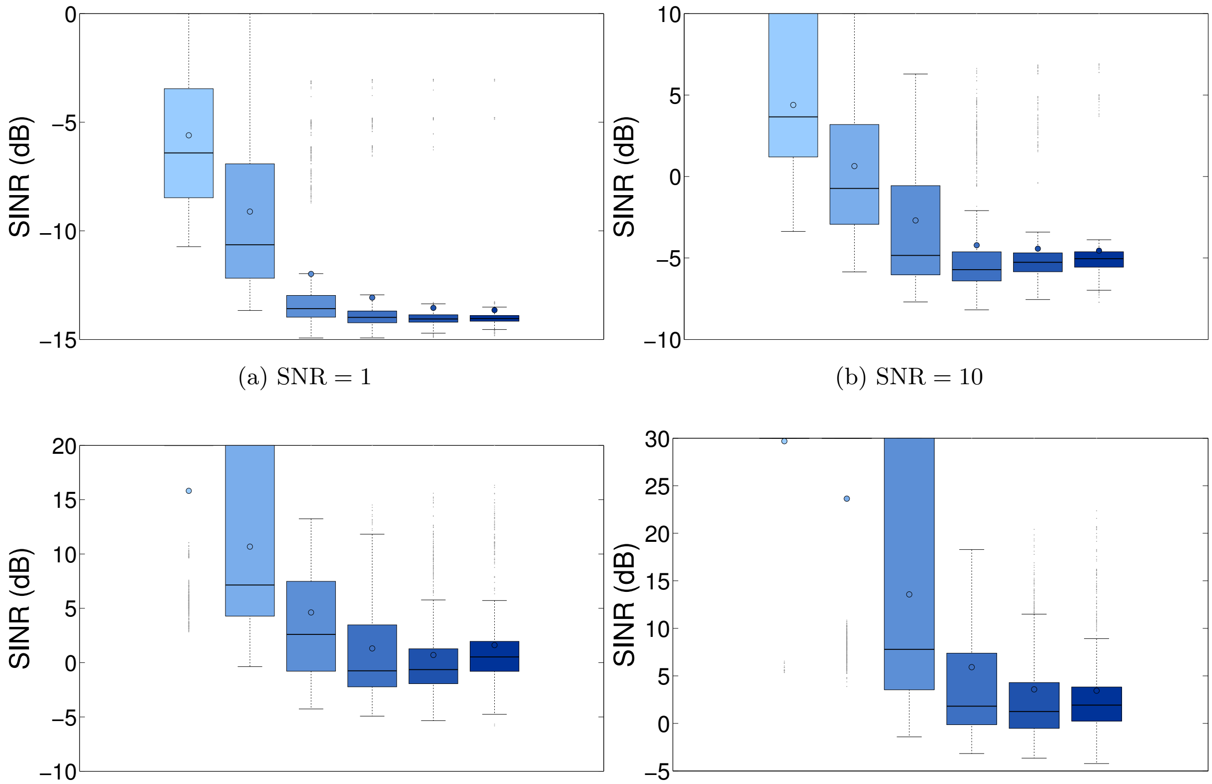
(a) SNR = 1

(b) SNR = 10

(c) SNR = 100

(d) SNR = 1000

Figure A.1: Size of SS Sets created using BSS



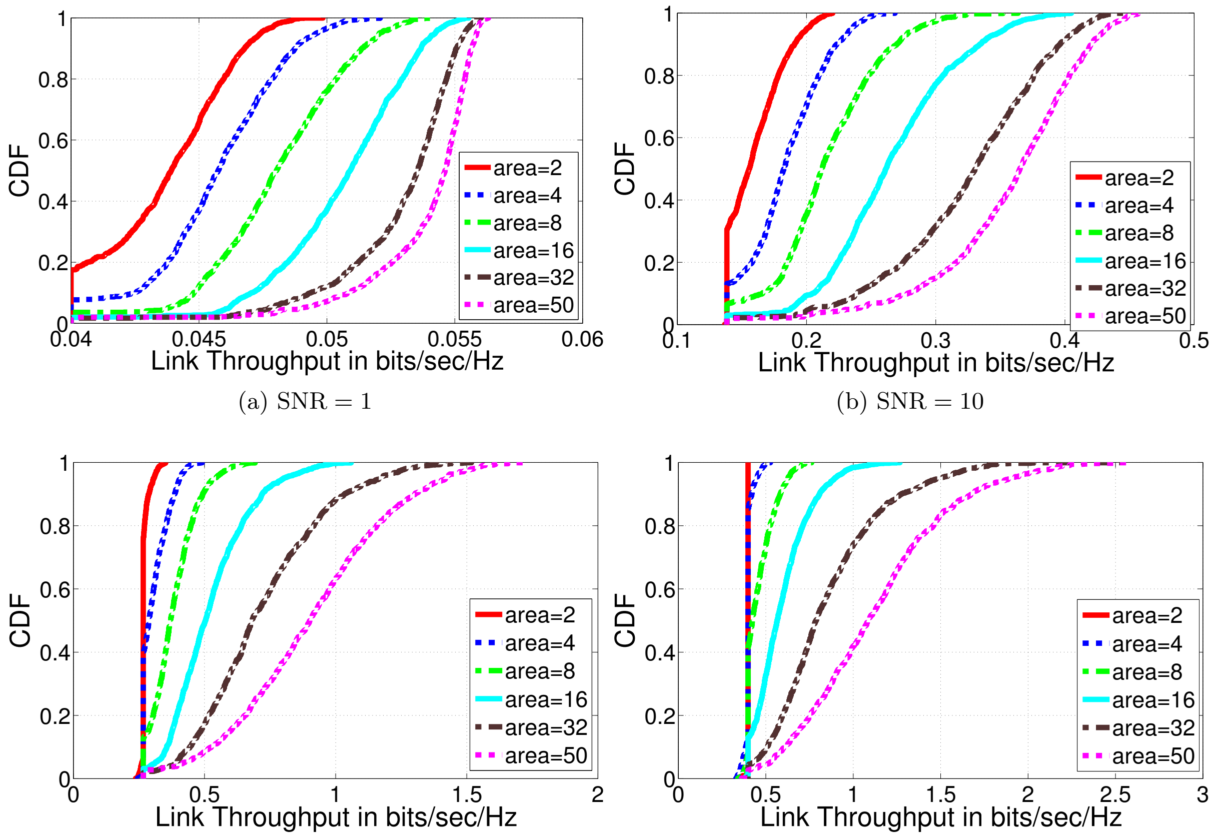
(a) SNR = 1 (b) SNR = 10

(c) SNR = 100 (d) SNR = 1000

Figure A.2: SINR of links when SS sets are created using BSS

Table A.1: Fairness Index using BSS.

SNR	area=2	area=4	area=8	area=16	area=32	area=50
1	0.9970	0.9970	0.9970	0.9971	0.9975	0.9975
10	0.9838	0.9762	0.9684	0.9659	0.9660	0.9726
100	0.9973	0.9764	0.9563	0.9290	0.9076	0.9089
1000	0.9999	0.9971	0.9722	0.9378	0.8888	0.8853



(a) SNR = 1

(b) SNR = 10

(c) SNR = 100

(d) SNR = 1000

Figure A.3: Throughput of the links using BSS

Bibliography

[1] High Efficiency WLAN. `http://standards.ieee.org/news/2013/IEEE_802.11_HEW.html`.

[2] Ieee standard for information technology–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - amendment 8: Medium access control (mac) quality of service enhancements. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)* (2005), 1–212.

[3] Specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)* (Mar. 2012).

[4] Ahmed, N., Ismail, U., Keshav, S., and Papagiannaki, K. Online estimation of rf interference. In *Proceedings of the 2008 ACM CoNEXT Conference* (New York, NY, USA, 2008), CoNEXT '08, ACM, pp. 4:1–4:12.

[5] Ahmed, N., Shrivastava, V., Mishra, A., Banerjee, S., Keshav, S., and Papagiannaki, K. Interference mitigation in enterprise WLANs through speculative scheduling. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking* (New York, NY, USA, 2007), ACM, pp. 342–345.

[6] Bansal, T., Chen, B., Sinha, P., and Srinivasan, K. Symphony: Cooperative packet recovery over the wired backbone in enterprise WLANs. MobiCom '13, ACM.

[7] Bianchi, G. Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on 18*, 3 (2000), 535–547.

[8] Brar, G. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *in Wireless Mesh Networks, in Proc. ACM MobiCom* (2006), ACM Press, pp. 2–13.

[9] Das, S. M., Koutsonikolas, D., Hu, Y. C., and Peroulis, D. Characterizing multi-way interference in wireless mesh networks. In *In Proc. of ACM WiNTECH* (2006), ACM Press, pp. 57–64.

[10] Epstein, L., and Levin, A. An aptas for generalized cost variable-sized bin packing. *SIAM Journal on Computing 38*, 1 (2008).

[11] Ergin, M. A., Ramachandran, K., and Gruteser, M. Understanding the effect of access point density on wireless lan performance. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking* (New York, NY, USA, 2007), MobiCom '07, ACM, pp. 350–353.

[12] Goldsmith, A. *Wireless communications.* Cambridge university press, 2005.

[13] Grönkvist, J., and Hansson, A. Comparison between graph-based and interference-based stdma scheduling. MobiHoc '01, ACM, pp. 255–258.

[14] Guo, X., Roy, S., and Conner, W. Spatial reuse in wireless ad-hoc networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th* (Oct 2003), vol. 3, pp. 1437–1442 Vol.3.

[15] Jang, B., and Sichitiu, M. IEEE 802.11 saturation throughput analysis in the presence of hidden terminals. *IEEE/ACM Transactions on Networking 20*, 2 (Apr. 2012), 557–570.

[16] Jiang, L., and Walrand, J. Approaching throughput-optimality in distributed csma scheduling algorithms with collisions. *Networking, IEEE/ACM Transactions on 19*, 3 (2011), 816–829.

[17] Lee, J., Kim, W., and et al. An experimental study on the capture effect in 802.11 a networks. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization* (2007), ACM, pp. 19–26.

[18] Lin, T.-Y., and Hou, J. Interplay of spatial reuse and sinr-determined data rates in csma/ca-based, multi-hop, multi-rate wireless networks. In *INFOCOM 2007. IEEE* (2007), pp. 803–811.

[19] Maheshwari, R., Cao, J., and Das, S. Physical interference modeling for transmission scheduling on commodity wifi hardware. In *INFOCOM 2009, IEEE* (April 2009), pp. 2661–2665.

[20] Malone, D., Duffy, K., and Leith, D. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Trans. Netw. 15*, 1 (2007), 159–172.

[21] Manweiler, J., and et al. Order matters: Transmission reordering in wireless networks. *IEEE/ACM Transactions on Networking 20*, 2 (2012), 353–366.

[22] Murty, R., Padhye, J., and et al. Designing high performance enterprise wi-fi networks. In *NSDI* (2008), vol. 8, pp. 73–88.

[23] Rahul, H. S., Kumar, S., and Katabi, D. Jmb: Scaling wireless capacity with user demands. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (New York, NY, USA, 2012), SIGCOMM '12, ACM, pp. 235–246.

[24] Shrivastava, V., Ahmed, N., and et. al. CENTAUR: realizing the full potential of centralized wlans through a hybrid data path. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2009), MobiCom '09, ACM.

[25] Shrivastava, V., Rayanchu, S., Yoonj, J., and Banerjee, S. 802.11 n under the microscope. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement* (2008), ACM, pp. 105–110.

[26] Stanton, D., and White, D. *Constructive combinatorics.* Springer-Verlag New York, Inc., 1986.

[27] Van Nee, R. Breaking the gigabit-per-second barrier with 802.11ac. *Wireless Communications, IEEE 18*, 2 (2011), 4–4.

[28] Vasan, A., Ramjee, R., and Woo, T. Echos - enhanced capacity 802.11 hotspots. In *INFOCOM 2005. IEEE* (2005), vol. 3.