

Copyright
by
Jeremy Kang-pen Chen
2007

The Dissertation Committee for Jeremy Kang-pen Chen
certifies that this is the approved version of the following dissertation:

**Frequency Allocation, Transmit Power Control, and Load
Balancing with Site Specific Knowledge for Optimizing Wireless
Network Performance**

Committee:

Theodore S. Rappaport, Supervisor

Gustavo de Veciana, Supervisor

Sanjay Shakkottai

Lili Qiu

Sriram Vishwanath

**Frequency Allocation, Transmit Power Control, and Load
Balancing with Site Specific Knowledge for Optimizing Wireless
Network Performance**

by

Jeremy Kang-pen Chen, B.S.E., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2007

Acknowledgments

This research is sponsored by NSF (National Science Foundation) Next Generation Software research program under the grant number ACI-0305644, NSF ITR (Information Technology Research for National Priorities) Grant CNS-0325788, and the industrial affiliate program at the Wireless Networking and Communications Group (WNCG) at the University of Texas at Austin.

Many thanks go to my advisor, Prof. Ted Rappaport. It has been my privilege working under him since fall 2002. He is an outstanding researcher, teacher, entrepreneur, inventor, and mentor. Prof. Rappaport's enthusiasm toward the engineering profession impressed me and educated me very much. I have learned so much from him.

Many thanks also go to my co-advisor, Prof. Gustavo de Veciana. He began co-advising me since December 2005. He is an expert in mathematical modeling. He is also very willing to help students. I am so privileged to have spent much time with him. From my two advisors, I learned how to define, solve, and gain insights into difficult engineering problems.

I also thank my committee members, Profs. Sanjay Shakkottai, Lili Qiu, and Sriram Vishwanath for their time and feedbacks.

Thanks to my colleagues in WNCG (including alumni); they have inspired me so much.

Thanks to WNCG staff, Cristine Alaya, Janet Preuss, Lori Lacy, Wanda Franklin, and Holly Brown.

I would like to thank my dear wife Esther; without her support, care, patience, and love, this dissertation could not have been done. Thanks to my parents Peter and Priscilla, who have always been supporting me. I thank my dear Lord Jesus Christ; He always supplies me with His life and riches. I also thank many Christians that met with me to enjoy the Word of God.

Frequency Allocation, Transmit Power Control, and Load Balancing with Site Specific Knowledge for Optimizing Wireless Network Performance

Publication No. _____

Jeremy Kang-pen Chen, Ph.D.
The University of Texas at Austin, 2007

Supervisors: Theodore S. Rappaport
Gustavo de Veciana

This dissertation is the first analytical and algorithmic work to exhibit the substantial gains that result from applying *site specific knowledge* to frequency allocation, transmit power control, and load balancing in wireless networks. Site specific knowledge refers to the use of knowledge of the surrounding propagation environment, building layouts, the locations of access points (APs) and clients, and the locations and electrical properties of physical objects. We assume a central network controller communicates with all APs, and has site specific knowledge which enables the controller to differentiate the sources of RF interference at every AP or user. By predicting the power from each interference source, the controller can allocate frequency channels, adjust transmit power levels, and balance loads among APs and clients in order to optimize throughput of the network. When site specific knowledge is not available, measurement-based algorithms may be used; we present three measurement-based frequency allocation algorithms that outperform the best published work by 18% for median user throughput. Then we present two site-specific knowledge-based frequency allocations that outperform the proposed measurement-based algorithms particularly for uplifting throughputs of the users who suffer low throughputs, e.g., we have gains of 3.75%, 11.8%, 10.2%, 18.2%, 33.3%, and 459% for 50, 25, 20, 15, 10, and 5 percentiles of user throughputs, respectively,

over the proposed measurement-based algorithms. Furthermore, we employ transmit power control to further improve clients' throughputs achieved by optimal site-specific knowledge-based frequency allocations; transmit power control can improve the 25, 10, 5, and 3 percentiles of users' throughputs by up to 4.2%, 9.9%, 38%, and 110%, and save power by 20%. Finally, a load balancing algorithm is proposed as an add-on that works seamlessly with frequency allocation and transmit power control algorithms. The load-balancing algorithm can improve median user throughput by about 26%. The work in this dissertation shows that site specific knowledge is an important means for optimizing performance of wireless networks.

Table of Contents

Acknowledgments	iv
Abstract	v
List of Tables	x
List of Figures	xi
Chapter 1. Introduction	1
1.1 Why Site Specific Knowledge Instead of Measurement-Based Techniques?	3
1.1.1 Site Specific Knowledge for Licensed Bands	6
1.1.2 Site Specific Knowledge for Unlicensed Bands	8
1.1.3 Summary Concerning Using Site Specific Knowledge	10
1.2 Frequency Allocation for Wireless Data Networks	11
1.3 Transmit Power Control in Wireless Networks	12
1.4 Load Balancing for Wireless Data Networks	12
Chapter 2. Literature Review	15
2.1 Prior Research on Site Specific Predictions of RF Channels and Network Throughput	15
2.1.1 RF Channel Prediction	15
2.1.2 Throughput Prediction	16
2.2 Prior Research on Frequency Allocations	17
2.3 Prior Research on Transmit Power Control	20
2.4 Prior Research on Load Balancing	21
Chapter 3. Improved Measurement-Based Frequency Allocation Algorithms for Wireless LAN or Cellular Networks	23
3.1 Overview, Main Contribution, and Organization of This Chapter	23
3.2 System Model and Notation	25
3.2.1 Metrics for Measurement-Based Frequency Channel Decisions	26
3.2.2 The Set of Interfering and Interfered Cells	30

3.3	Three Measurement-Based Algorithms	34
3.3.1	The No-Coord Algorithm	35
3.3.2	The Local-Coord Algorithm	35
3.3.3	The Global-Coord Algorithm	40
3.3.4	Implementation Concerns	40
3.3.5	Convergence	41
3.3.6	Characterization of Convergence Points	43
3.4	Simulation Results	44
3.4.1	Simulation Setup	45
3.4.2	Results and Discussion for Downlink-Only Scenarios	53
3.4.2.1	Results with Realistic Noise Floor	57
3.4.3	Results and Discussion for Downlink-and-Uplink Scenarios	60
3.5	Conclusions of This Chapter	61
3.6	Appendix	61
3.6.1	Modeling Measured Interference	61
3.6.2	Proofs of Lemmas 3.1 and 3.2, and Theorems 3.4 and 3.5	62
 Chapter 4. Site Specific Knowledge for Improving Frequency Allocations in Wireless LAN and Cellular Networks		69
4.1	Notation and Assumptions	72
4.2	Site-Specific Knowledge-Based Algorithms	72
4.2.1	The Site-Specific SINR (SS-S) Formulation	72
4.2.2	The Site-Specific Rate (SS-R) Formulation	74
4.2.3	A Local Optimization Algorithm for <i>SS-S</i> and <i>SS-R</i>	75
4.3	Simulation Setup and Results	76
4.3.1	Simulation with Realistic Noise Floor	82
4.4	Conclusions of This Chapter	86
 Chapter 5. Power Control with Site Specific Knowledge for Maximizing Throughput of the Network		87
5.1	Contrast with Prior Related Work	88
5.2	Notation and Assumptions	88
5.3	Transmit Power Control Problem	89
5.3.1	Background on Geometric Programming	91
5.3.2	Algorithms for Transmit Power Control	92

5.3.3	Implementation Concerns: Block Processing, Overhead, and Discrete Power Levels	94
5.4	Simulation Setup and Results	95
5.4.1	Simulation Setup	95
5.4.2	Simulation Results and Discussions	97
5.5	Conclusions of This Chapter	99
Chapter 6.	Load Balancing for Wireless Data Networks	100
6.1	Introduction	101
6.2	System Model and Notation	103
6.3	Maximum Sum Utility with Time Allocation	107
6.4	Simulation Results	112
6.5	Conclusions of This Chapter	115
Chapter 7.	Conclusions	116
Bibliography		118
Vita		125

List of Tables

3.1	A variable ψ_m used in the protocol for <i>Local-Coord</i>	37
3.2	Three proposed measurement-based algorithms and two types of weighted interference functions yield six combinations that are shown between the first and the sixth row. The last two rows show two other published algorithms. The last column shows the abbreviation of each algorithm.	45
3.3	Scenarios considered in the simulations	50
3.4	Comparison of the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on) and the mean of <i>users' throughputs</i> in Mbps using site-specific prediction based algorithms, measurement-based algorithms, <i>LC</i> [1], or <i>CF</i> [2]. APs are on a 10-by-10 uniform AP layout with 400 users and 10 rogue RF interferers. The percentages indicate the throughput gains over <i>CF</i> ; the 5 and 3 percentiles are not compared to <i>CF</i> , since <i>CF</i> yields zero throughputs at these points. The 50 percentile of course corresponds to the median.	53
4.1	Comparison of the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on) and the mean of users' throughputs in Mbps using site-specific prediction based algorithms, measurement-based algorithms, <i>LC</i> [1], or <i>CF</i> [2]. APs are on a 10-by-10 uniform AP layout with 400 users and 10 rogue RF interferers. The percentages indicate the throughput gains over <i>CF</i> ; the 5 and 3 percentiles are not compared to <i>CF</i> , since <i>CF</i> yields zero throughputs at these points. The 50 percentile of course corresponds to the median.	80
5.1	<i>Power saving</i> for different network sizes, rogue interference, and network topologies.	97
5.2	Percentiles of throughput gains with continuous power levels, or 2, 2.5, 4, 5, or 10 dB of separation between discrete power levels.	98
6.1	Fairness index (cf. [3]) of user throughput allocation for two kinds of user distributions (cluster and uniform) in a network with 36 APs and 400 users. (Unit: %)	114

List of Figures

1.1	We consider downlink traffic only. The interference on c_2 is from a_2 , a_3 , and a_4 .	5
1.2	Desired signals and two different types of interferences.	9
1.3	An example of load balancing. Two mobile users that are pointed by arrows are <i>not</i> associated with the AP with the strongest signal; rather, these two users are associated with farther APs to obtain better data transmission throughput. ‘Ch1, Ch2, Ch3’ denote three orthogonal frequency channels.	13
3.1	An example of measured interference power at a_1, c_1, c_2 for Channels 1 - 4.	28
3.2	Potential interferers for transmissions between the AP a_1 and the user c_1 . The <i>set of interfering cells</i> of \mathbb{Z}_1 is $\mathbb{Q}_1 = \{2, 3, 4, 5\}$, as covered in the gray area.	31
3.3	The heights of <i>Solid</i> and <i>dotted</i> bars respectively signify the weighted interference power seen by cells <i>before</i> and <i>after</i> AP-1 switches from Channel 1 to Channel 2. The max weighted interference seen by Cells 1 – 4 decreases after AP-1 switches to Channel 2.	36
3.4	A protocol for the distributed implementation of <i>Local-Coord</i> .	38
3.5	The coverage area of each AP is modeled as a hexagon. The separation distance between two adjacent APs is denoted as d ; then the distance from an AP to its farthest user is $\frac{d}{\sqrt{3}}$.	47
3.6	Frequency allocation examples for 49 APs on a 7-by-7 <i>nonuniform</i> or <i>uniform</i> topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 49 APs; hollow objects denote 196 users; double-layered objects with inner part filled with black denote 20 rogues. The units of X and Y axes are meters.	51
3.7	<i>User throughput (in Mbps)</i> comparison in a setting with APs on a <i>uniform</i> 10-by-10 layout, 400 users, and 10 rogue RF interferers. Subfigure (b) is part of (a) with only 200 users with the lowest throughputs.	54
3.8	Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. <i>Nonuniform</i> and <i>uniform</i> AP layouts are denoted ‘nu’ and ‘u’, respectively.	55
3.9	<i>User throughput (in Mbps)</i> comparison in a setting with APs on a <i>uniform</i> 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with lower throughputs are shown. Noise floor is 10dB above the thermal noise.	58
3.10	Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. <i>Nonuniform</i> and <i>uniform</i> AP layouts are denoted ‘nu’ and ‘u’, respectively. Noise floor is 10dB above the thermal noise.	59

3.11	50 and 25 percentiles of users' throughputs (50P and 25P) respectively, including both downlink and uplink traffic, for 400 users on a 10-by-10 uniform AP layout with 10 rogues.	60
3.12	The activities of AP a_2 and user c_2 affect the interference on a_1 and c_1	62
4.1	Frequency allocation examples for 49 APs on a 7-by-7 <i>nonuniform</i> or <i>uniform</i> topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 49 APs; hollow objects denote 196 users; double-layered objects with inner part filled with black denote 20 rogues. The units of X and Y axes are meters.	77
4.2	<i>User throughput (in Mbps)</i> comparison in a setting with APs on a <i>uniform</i> 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with lower throughputs are shown.	78
4.3	Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. <i>Nonuniform</i> and <i>uniform</i> AP layouts are denoted 'nu' and 'u', respectively.	79
4.4	<i>User throughput (in Mbps)</i> comparison with different levels of noise floor in a setting with APs on a uniform 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with the lowest throughputs are shown.	83
4.5	Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. <i>Nonuniform</i> and <i>uniform</i> AP layouts are denoted 'nu' and 'u', respectively. Noise floor is 10dB above the thermal noise.	84
4.6	50P and 25P signify 50 and 25 percentiles of users' throughputs respectively, including both downlink and uplink traffic. Noise floor is 10dB above the thermal noise	85
5.1	Frequency allocation examples for 25 APs on a 5-by-5 <i>nonuniform</i> or <i>uniform</i> topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 25 APs; hollow objects denote 100 users; double-layered objects with inner part filled with black denote 10 rogues. The units of X and Y axes are meters.	95
5.2	Gains of median, and 25, 10, 5, and 3 percentiles (denoted 25P, 10P, 5P and 3P) of users' throughputs when transmit power control is employed, as compared with using constant transmit power of 100 mW at every AP. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. <i>Nonuniform</i> and <i>uniform</i> AP layouts are denoted 'nu' and 'u', respectively.	96
6.1	A simple network with 3 APs and 5 users. Three APs, denoted by $a_1, a_2,$ and a_3 , use disjoint channels. Five users are denoted by $c_1 - c_5$. $\mathbf{R}_{m,l}$ denotes the long-term transmission rate between a_m and user c_l . $\mathbf{T}_{m,l}$ denotes the time fraction allocated to c_l over the RF channel of a_m . The aggregate rate that c_4 receives from a_2 and a_3 are denoted b_4	103

6.2	An iterative algorithm to solve (6.2)	111
6.3	The median of user throughput.	114
6.4	The 25-percentile of user throughput.	115

Chapter 1

Introduction

Despite the growth and wide acceptance of wireless technology, fundamental differences between wireless and wired networks pose challenging issues for wireless network design and management. Unlike wired networks, where links have stable and fixed rates, the link bandwidths in wireless networks are somewhat *unpredictable*, since the bandwidths vary with many factors, such as transmit power levels, antenna directivity, multipath, and user movement. Moreover, in most wired networks, interference between different links are negligible unless the insulation around wires is problematic. By contrast, in wireless communications, physical environment is like an imperfect insulator which causes attenuation of radio signal powers; therefore, the signal reception at a receiver is affected by interference from undesired sources in the proximity of the receiver.

In order to resolve the *unpredictability* of wireless communication, techniques and algorithms for *site-specific channel prediction* for wireless communications have been developed in the past decade [4–19]. Propagation characteristics have been found to be highly site specific, since major propagation mechanisms (e.g. penetration, reflection, and diffraction) are highly related to obstacles in the surroundings. These algorithms can take a building CAD drawing or a satellite map and compute the radio propagation loss contours for indoor or outdoor environments. As of today, these site-specific prediction methods are mainly used in the *design* and *deployment* phase for engineers to pre-configure the network. What if they can be used at run-time on devices? As the processing power of devices becomes stronger, future devices (central network controllers, base stations (BS), access points (AP), or end users hand-held devices/laptops) can be equipped with site-specific prediction engines for instantaneous propagation modeling. Prediction engines may require that these devices have digital maps of

surrounding terrains, buildings, and other obstacles that may affect radio propagation.

In this dissertation, we exploit the site-specific predictions for optimizing wireless network performances; particularly, by doing frequency allocation, transmit power control, and load balancing, throughput of network can be improved. Chapter 2 reviews and summarizes prior research on site specific prediction techniques, frequency allocations, transmit power control, and load balancing. In Chapter 3, we present improved frequency allocation algorithms that are measurement-based; these algorithms are shown to outperform all published work on frequency allocations. We outperform the best published work by 15% and 18% for mean and median user throughputs respectively, and 81%, 168%, and 1011% for 25, 20, and 15 percentiles of user throughputs, respectively. The work in Chapter 3, however, does not use site specific knowledge. Chapter 4 presents frequency allocation algorithms that exploit site specific knowledge and show these algorithms even outperform those in Chapter 3 up to 3.75%, 11.8%, 10.2%, 18.2%, 33.3%, and 459% for 50, 25, 20, 15, 10, and 5 percentiles of user throughputs, respectively. Note that in Chapters 3 and 4 and other published work on frequency allocations [1, 2, 20–22], the transmit powers of APs are fixed. After the optimal frequency allocation has been found using the fix transmit power, transmit power control can further improve network throughput. In Chapter 5, we optimize transmit power and find that we can improve the 25, 10, 5, and 3 percentiles of users' throughputs by up to 4.2%, 9.9%, 38%, and 110%, and save power by 20%. Furthermore, when the density of APs increases, the number of APs to which a client can connect increases. It is important to associate clients to APs so that the loads on APs are balanced and users are associated with less-congested APs to increase their throughputs. In Chapter 6, we present a load-balancing algorithm that allocates bandwidth among APs and users subject to heterogeneous fairness and application requirements; the proposed load-balancing algorithm achieves at least 26% gain of median user throughput over other algorithms in the literature. Frequency allocation algorithms in Chapters 3 and 4, transmit power control in Chapter 5, and the load-balancing algorithm in Chapter 6 can work seamlessly, since transmit power control improves user throughput achieved by frequency allocations, and load balancing can be viewed as an add-on to frequency

allocations and transmit power control to further improve user throughputs. Frequency allocations and transmit power control are performed on a longer time scale (say 5 minutes) to optimize average throughputs of users, whereas load balancing is performed when any user joins or leaves, whose time scale is often shorter than that for frequency allocations (say 5, 30, or 60 seconds). Finally, Chapter 7 concludes this dissertation.

In this introductory chapter, we will briefly describe the benefit of using site specific knowledge in Section 1.1 for practical problems of wireless system design. We will use site specific knowledge to solve the problems of frequency allocations, transmit power control, and load balancing in wireless local area networks, as described in Sections 1.2, 1.3, and 1.4, respectively.

1.1 Why Site Specific Knowledge Instead of Measurement-Based Techniques?

In brief, the advantage of using site specific knowledge is to predict *a priori* path loss¹ between any pair of radio frequency (RF) transmitter and receiver, when the locations of the transmitter and receiver are obtained via GPS (Global Positioning System) or other position location technologies. More precisely, site specific prediction can predict *channel impulse response* between a transmitter and a receiver; a channel impulse response usually consists of multipaths, where each path has two key parameters: amplitude and the angle of arrival (or, phase). Suppose multipaths can be resolved at the receiver; then, the total received power at the receiver is simply the sum of the power of each resolved ray. Then, the path loss can be computed by dividing the transmit power to the received power. The resolution of multipaths can be done by using MIMO or other signal processing techniques. In this dissertation, we assume that the signal processing on multipaths has been done and consider only path losses, computed with the resolved and combined received power. It is

¹*Path loss* is defined as the ratio of the transmit power of the RF transmitter to the received power at the RF receiver, when no interference and noise exists in the environment. *Path gain* is the inverse of the path loss.

known that the environment affects the path losses. Nevertheless, empirical results show that by modeling large fixed partitions and items in the environment (such as walls, book shelves, and cubicles), the predicted path losses and empirical results are within 4 dB accuracy [4–19]. With the high accuracy, site specific predictions can be a useful means for wireless network optimization.

Several indoor position location approaches, based on signal strength sensing, are widely known today and used in some WLANs [23, 24]. Other triangulation methods can also be used to locate a client. Modern cellular handsets are equipped with GPS chips or other position location technologies. State-of-the-art GPS can work not only outdoors but also indoors; various vendors, e.g. Metris and SnapTrack, provide indoor GPS solutions. The indoor GPS technology by Metris can be compared to the matrix of satellites that create the Global Positioning System; instead of satellites, Metris’ indoor GPS uses small infrared laser transmitters that emit laser pulses to create a measurement universe. Then, photo detectors pick up the signals and compute angle and positions based on the timing of the arriving light pulses. Later when we mention site specific knowledge in this dissertation, we may implicitly include the knowledge of the locations of APs and users, if there is no ambiguity.

The strength of centralized site specific predictions is that the central controller can *predict the strength of each individual constituent of the interference on RF devices (APs or mobile users)* in a very short time. Knowing individual constituents of the interference on every device can help formulate a global optimization problem, thereby maximizing throughputs and saving power, etc. Distributed measurement-based algorithms (with the knowledge of APs’ transmit powers via message exchanges) can learn individual constituent of interference over time; nevertheless, the time needed to learn individual components may be too long, when the number of interfering APs is large, as explained by the following example. We use Fig. 1.1 to illustrate a simple measurement-based algorithm for learning every individual constituent of the interference that c_1 sees. In this dissertation, we assume that interference from difference sources are additive. Let S_{c_l, a_m} denote the interference power from a_m to c_l . We can compute the individual values of S_{c_1, a_2} , S_{c_1, a_3} , and S_{c_1, a_4} by doing three or more measurements. When

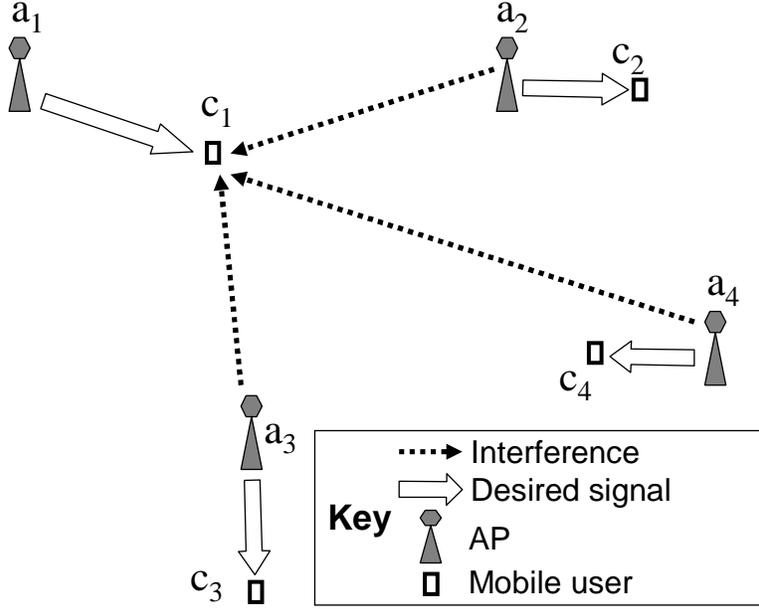


Figure 1.1: We consider downlink traffic only. The interference on c_2 is from a_2 , a_3 , and a_4 .

a_4 is off, c_1 can measure the sum of S_{c_1,a_2} and S_{c_1,a_3} ; say the sum is α . Similarly, c_1 can measure the sum of S_{c_1,a_2} and S_{c_1,a_4} (say the sum is β), as well as the sum of S_{c_1,a_3} and S_{c_1,a_4} (say the sum is γ). Then, we have the following three linear equations.

$$\begin{aligned}
 S_{c_1,a_2} + S_{c_1,a_3} &= \alpha \\
 S_{c_1,a_2} + S_{c_1,a_4} &= \beta \\
 S_{c_1,a_3} + S_{c_1,a_4} &= \gamma
 \end{aligned} \tag{1.1}$$

Since the three equations above are linearly independent, the individual values of S_{c_1,a_2} , S_{c_1,a_3} , and S_{c_1,a_4} can be solved. Suppose the number of base stations or APs is M , a client c_l can learn the individual interference power from M APs after c_l performs enough times (at least M times) of interference measurements to form a linearly independent system of equations that have M variables. The learning time of this algorithm could become too long as M becomes large; hence, this algorithm may not scale well. The algorithm can be simplified if each client learns the interference power from only the APs that are in the range of causing non-negligible interference at the client. In order to know which base stations are in the interference range, site specific knowledge (such as the environments and the locations of APs

and clients) is needed. Saving the learning time for measurement-based algorithms is a topic for ongoing and future work. In this dissertation, we choose to use site specific knowledge to predict *a priori* the interference power between any transmitter and any receiver.

The path loss predictions for all possible transmitter-receiver pairs are useful for interference management and on-line resource allocation algorithms (e.g. for frequency channels, bandwidth, power). In the following subsections, we present the ideas of using site specific knowledge on licensed or unlicensed bands (without or with rogue interferers).

1.1.1 Site Specific Knowledge for Licensed Bands

Let us use the frequency allocation problem as an example. Suppose there are only two orthogonal frequency channels available, and each AP is assigned with either one of these two channels. Let us consider the wireless data network in Fig. 1.1, and assume for now there is much more downlink (from APs to users) than uplink traffic. Hence, we focus on interference management for downlink traffic. We assume there are no other RF signal emitters other than the APs and the mobile users shown on Fig. 1.1, since we consider licensed bands in this subsection. Suppose that a central network controller communicates with all APs via wireline network and knows the locations of all users and can predict the path losses between every pair of AP and user. The central controller would like to maximize the average signal-to-interference-plus-noise ratio (SINR) or throughputs seen across all users.

Site specific knowledge can predict the interference from every single interference, given that the interfering source is active. The APs and the clients may be on or off and sending either uplink or downlink traffic. The central network controller could gather the activity states (on or off) of every AP and client and dynamically change the channels of all APs and clients whenever the activity states of APs and clients change (e.g. turning on or off) in order to minimize interference seen by APs and clients and maximize throughputs; however, such a scheme has a high complexity. In this dissertation, we consider a simpler scheme where APs and clients use a fixed set of channels during a longer period of time (say 5 minutes). During this period, APs may turn on and off, but the channels remain the same; by doing

so, the complexity is much lower, since channel switching occurs less frequently. Then a question arises: how do we determine the channels that would work well irrespective of APs' activities (on or off)? Our hypothesis is that we can optimize the channels for a specific case where all APs are sending downlink traffic; then, the optimal channels for this case can also work well in other cases. Of course, the channel gains between APs and clients vary from time to time; hence, in each time period (say 5 minutes), the channel gains are sampled, and the optimization is performed with respect to the sampled channel gains. Our hypothesis is based on the empirical results that downlink traffic dominates in WLANs [11, 12, 25]. In this dissertation, we show by simulation that our hypothesis is correct. So for now, we assume that the central network controller does not care about the activities of APs, but simply optimizes the channels assuming all APs are sending downlink traffic.

A central network controller with site specific knowledge can predict the signal power at each user from the desired AP and the interference power at each user from any interfering AP, once the locations of users are reported back to APs via short control packets. Then, the network controller can perform a centralized optimization algorithm to easily maximize the average SINR at users. In other words, the network controller has a bird's-eye view on the entire wireless network and can solve on-line resource allocation problems in a better way. As the network becomes larger, having site specific knowledge will improve resource allocation, as will be shown in later chapters of this dissertation.

The benefit of using site specific knowledge is obvious for *licensed bands* where all RF signal transmitters are under the control of a single vendor (usually in a bounded geographical area). There are hardly any rogue RF interference emitters because of the FCC regulation (or equivalent RF spectrum regulations in other countries). On licensed bands, the central network controller has complete control on the interference that is generated and can even 'design' interference.

1.1.2 Site Specific Knowledge for Unlicensed Bands

Additional issues arise in *unlicensed bands* where there are uncontrolled RF interference emitters; these emitters may be RF transmitter in independent networks. These uncontrolled RF emitters may be APs from independent networks, microwave ovens, or other RF devices. From the perspective of our controlled networks, we will call the interference from uncontrolled RF emitters *background interference* or *rogue interference*, and call the uncontrolled RF emitters *rogue RF interferers*. Some measurements need to be performed at APs and users to determine the background interference. For example, the central network controller may periodically require the APs to stop transmitting for a short duration of time (say, one second). In this duration, APs take turns in requiring all users associated with them to perform measurements of background interference; note that each user needs to measure the background interference for all available frequency channels. The users then feedback to APs these measured background interference. Site specific knowledge along with measurements of background interference make the estimations of SINR at users or APs more accurate. Note that the measured background interference may consist of interference from a number of rogue interferers. In this dissertation, we do not separate the constituents of the rogue interference. Simulation results show that the aggregate background interference is useful enough to improve the throughput of the network. Nevertheless, it is possible to decompose the constituents of the aggregate background by using triangulation methods, for example [23, 24].

We will use the example in Fig. 1.2 to illustrate such measurements of background interference. We consider the case where multiple APs communicate with multiple users in an indoor wireless local-area network (WLAN). The framework in Fig. 1.2 can also apply to wireless metropolitan-area network (WMAN) if the wooden walls and cubicle walls are replaced with trees, buildings, mountains, and other common buildings and obstacles in outdoor environments. First, a_1 informs c_1 and c_2 by broadcasting a short control message; then, c_1 and c_2 feedback the measured background interference. The background interference at c_1 should be the sum of the noise floor of the radio RF environment and the interference from

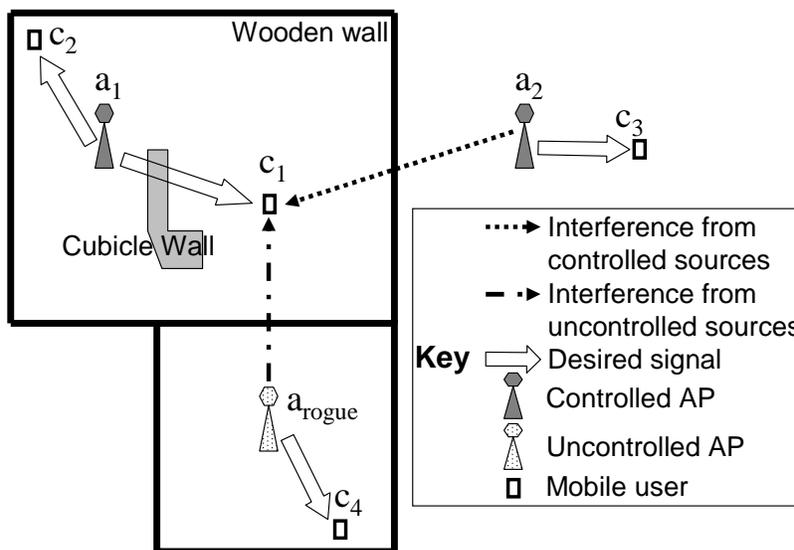


Figure 1.2: Desired signals and two different types of interferences.

the uncontrolled RF source a_{rogue} , and similarly for c_2^2 .

Suppose only two orthogonal frequency channels are available, and a_{rogue} operates on the first channel. What is the best frequency allocation for a_1 and a_2 ? Again, we would like to determine the *best* frequency allocation for the case where all APs are sending downlink traffic. Suppose a central network controller³ that controls APs a_1 and a_2 has a propagation prediction engine with the site-specific knowledge for the particular bounded area shown on Fig. 1.2. With the site specific knowledge and measured background interference, this network controller can determine that a_1 on the second channel and a_2 on the first channel are the solution that induces the least interference on users, particularly on c_1 , for the downlink-only case. In this small example, the optimum frequency allocation may also be obtained by using distributed measurement-based algorithms. For large networks with many APs and users, however, site specific knowledge enables the central network controller to know the signal

²In Fig. 1.2, we assume the interference from a_{rogue} affects c_1 greatly but does not affect c_2 or c_3 .

³The central network controller is not shown on the figure, as this controller communicates with a_1 and a_2 via wireline networks (e.g. backbone Internet). For illustration purposes, we assume the site specific predictions are performed at the central network controller; however, APs that have adequate computing capabilities may also be equipped with site specific knowledge and perform propagation predictions.

power or the interference power between any pair of AP and user, thereby producing a better solution for frequency or other resource allocation problems.

1.1.3 Summary Concerning Using Site Specific Knowledge

We are interested in an enterprise-like network that consists of many APs and clients. We assume a central network controller can communicate with all controlled APs and know the locations of all controlled APs and clients. We assume that the central network controller can predict signal strength between any AP and client, given the AP is active and the central network controller knows the transmit power of the AP. Note that the central controller must know the active transmitters at any point in time in order to predict correct interference at all times; this information may be too costly to obtain, but time sampling may be done. Since downlink volume presently dominates WLAN traffic, this paper considers a case where all APs are actively transmitting downlink traffic. It is reasonable to assume that frequency allocation is optimized with respect to this most active case, since in this case, frequency allocation is crucial for interference mitigation at users.

Therefore, the central network controller assumes all APs are always sending downlink traffic, and tries to optimize the frequency allocations and transmit powers for this specific scenario (downlink-only scenario). Simulation results in this dissertation show that the optimal channels for the downlink-only scenario works well also in a network mixed with downlink and uplink traffic, as long as downlink traffic dominates (which is true in most WLANs according to [11, 12, 25]). Based on the downlink-only assumption, the central controller can use the knowledge of channel gain between every AP and client to minimize interference seen at clients and maximize throughputs. In addition, on unlicensed bands, interference from rogue interference needs to be measured. Periodically APs stop sending traffic and require clients to measure background interference coming from rogue interferers. Each client measures the in-situ *aggregate* background interference. By using the aggregate background interference and knowledge of channel gain between every AP and client, the central network controller can estimate accurate SINR at each client with the assumption that all APs are actively

sending downlink traffic, thereby minimizing interference seen by clients and maximizing the throughputs of clients. In this dissertation, we consider perfect site specific knowledge; in other words, we assume that the actual path loss between any transmitter and receiver can be correctly predicted by the site specific knowledge. The effect of imperfect predictions of channel gains is an ongoing and future work.

1.2 Frequency Allocation for Wireless Data Networks

For the two frequency allocation examples in Fig. 1.1 and Fig. 1.2, we assume that downlink traffic volume dominates the total traffic volume, and thus uplink traffic volume is ignored (according to [11,12,25], the ratio of downlink volume to uplink volume is 5:1 in typical wireless LAN). The same assumptions will also be used as we describe the frequency allocation algorithms in Chapters 3 and 4. Under such assumptions, we optimize the frequency allocation for a most active case where all APs are actively transmitting downlink traffic. It is reasonable that frequency allocation is optimized with respect to this most active case, since in this case, proper frequency allocations are crucial for interference mitigation at users. On the other hand, we consider both downlink and uplink traffic (with ratio 5:1) in a modest case where APs are sometimes idle; then, interferences are naturally mitigated due to APs' inactivity. In such a modest case, frequency allocation plays a less important role in interference mitigation; hence, the optimality of frequency allocation may not be critical in terms of users' average SINR or average throughput. Simulation results in Chapters 3 and 4 show that the frequency allocation optimized for the most active case still performs very well for the modest scenario with both downlink and uplink traffic, and also show that the frequency allocation algorithms based on site specific knowledge outperform the measurement-based approach. When site specific knowledge is unavailable, our proposed measurement-based algorithms can be used, since they still outperform other published frequency-allocation algorithms.

1.3 Transmit Power Control in Wireless Networks

In Chapters 3 and 4, we optimize frequency allocations to minimize co-channel interference and maximize throughput of network, assuming the transmit power of APs and users are fixed. Increasing the transmit power of an AP can potentially increase the downlink throughput from this AP to its clients, but also induces larger interference on nearby APs or clients that are on the same channel, thereby lowering their throughputs. Like Chapters 3 and 4, we focus on optimizing downlink transmissions, i.e., we control the transmit powers of APs. In a network with multiple APs and clients, optimizing transmit powers requires the knowledge of path gains between APs and clients, which can be estimated by using site specific knowledge. A central network controller that communicates with all controlled APs and has site specific knowledge can optimize transmit powers at APs to maximize and balance the throughputs of all clients in the network. Our transmit power control works seamlessly with the best frequency allocation algorithm to date (i.e., the one presented in Chapter 4) to further improve users' throughputs, e.g., we improve the 25, 10, 5, and 3 percentiles of users' throughputs by up to 4.2%, 9.9%, 38%, and 110%, and save power by 20%.

1.4 Load Balancing for Wireless Data Networks

Since base stations or APs can operate on non-overlapping channels to avoid interfering with one another, people consider increasing the capacity of WLAN by increasing AP density and configuring channels appropriately. The scheme where neighboring APs provide overlapping coverage in a region also ensures continuity of network access when users roam. This is made possible by the increase of the number of non-overlapping channels in WLAN standards; for example, IEEE 802.11a provides up to twelve non-overlapping channels, whereas 802.11b/g provides three. Today's wireless LANs have the characteristics of user congestions at certain locations. Due to inappropriate association of users to APs, the bandwidth allocation may be inefficient and unfair among users and across APs. As the number of APs to which a user can connect increases, an algorithm that efficiently associates users to APs

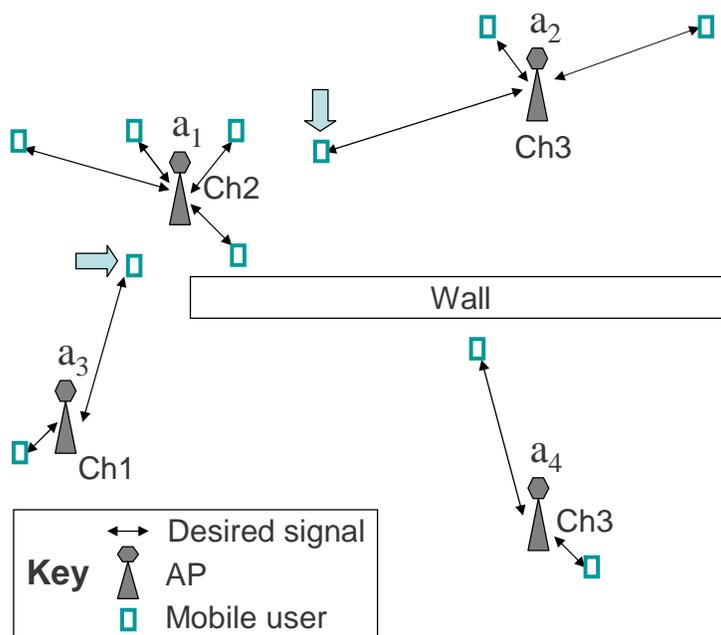


Figure 1.3: An example of load balancing. Two mobile users that are pointed by arrows are *not* associated with the AP with the strongest signal; rather, these two users are associated with farther APs to obtain better data transmission throughput. ‘Ch1, Ch2, Ch3’ denote three orthogonal frequency channels.

becomes critical for power, bandwidth, and quality of service (QoS) management. In most commercial 802.11 products, however, the default Strongest-Signal-First (SSF) approach, in which each user chooses an AP with the strongest signal, results in unevenly distributed loads among APs and poor performance [26]. Therefore, load balancing techniques are needed to balance the load at APs and to provide fair throughput across all users. In Fig. 1.3, we show the idea of load balancing. Suppose each AP has been allocated one of the three orthogonal frequency channels (denoted by Ch1, Ch2, or Ch3) to reduce inter-cell interference. Note that there are two mobile users pointed by arrows. These two users are *not* associated with the AP with the strongest signal (i.e. a_1); rather, they are associated with farther APs (a_2, a_3) to obtain better data transmission throughput. By associating these two users to farther APs, the load of a_1 is reduced. Load balancing is crucial when the spatial distribution of users are non-uniform, which is often the case according to field measurements in [27]. Load-balancing

can be done in either a distributed or a centralized way; we will present algorithms that can work in both ways.

Chapter 2

Literature Review

Past research has shown that RF signal parameters and throughput of wireless links can be predicted quickly and accurately by using site specific knowledge and position location capability. We exploit the site-specific predictions for optimizing wireless network performances; particularly, by doing frequency allocation, transmit power control, and load balancing, throughput of network can be improved. We first review prior research on site specific predictions, and then review the work on frequency allocation, transmit power control, and load balancing, respectively.

2.1 Prior Research on Site Specific Predictions of RF Channels and Network Throughput

First, Section 2.1.1 presents prior work on site-specific channel prediction techniques. Then, Section 2.1.2 presents work on throughput prediction.

2.1.1 RF Channel Prediction

Techniques and algorithms for *site-specific channel prediction* for wireless communications have been developed in the past decade [4–19]. The work in [4] presents one of the first models that can predict RF path loss. The prediction is based on the number of walls, office partitions, and floors between the transmitter and the receiver, as well as the separation distance between the transmitter and receiver, i.e., T-R separation. Each kind of partition was found to induce a certain decibel (dB) of signal attenuation. Suppose two kinds of partitions (e.g., soft partitions and concrete walls) have attenuation factors of AF_s and AF_c in dB,

respectively. Suppose the path loss at a reference distance d_0 is denoted $PL(d_0)$, which can be found by calibration or by the formula given below

$$PL(d_0) = 20 \log_{10} \left(\frac{4\pi d_0}{\lambda} \right), \quad (2.1)$$

where λ denotes the wavelength of the radio wave carrier. Then the average path loss in dB for a T-R separation of d can be modeled as

$$\overline{PL}(d)[\text{dB}] = PL(d_0)[\text{dB}] + 10 \cdot n \cdot \log_{10} \left(\frac{d}{d_0} \right) + p \cdot AF_s[\text{dB}] + q \cdot AF_c[\text{dB}], \quad (2.2)$$

where p and q denote the number of soft partitions and concrete walls between the transmitter and the receiver, respectively. Other variant forms of (2.2) were introduced in [4]. Path loss exponents (n) were found to be environment dependent, and statistical data were presented in [4] for n . Path losses predicted by the models in [4] have about 5.8 dB of standard deviation from measured data, and can be as small as 4 dB for some data sets. In general, the path loss prediction models are accurate within 6 dB.

The SMT PlusTM tool is an easy-to-use coverage prediction tool that assists in the design of indoor wireless systems. SMT PlusTM uses the path loss models introduced in (2.2) and [4] to predict coverage.

The Site-Specific System Simulator for Wireless communications (S^4W) [7,13] provides an accurate computation of the path losses as well as an estimate of the performance of a targeted wireless system (e.g. WCDMA data communication systems). These propagation modeling techniques require intensive and time-consuming computation. Nevertheless, quick turn-around models that perform first-order approximations of the site-specific models have been developed. A commercial product called LANPlanner[®] [13] uses clever engineering assumptions and measurement integration to rapidly model indoor and outdoor environments.

2.1.2 Throughput Prediction

The work in [11, 12, 14–17, 23] presents models that can predict user data throughput in a site specific manner. The work in [11, 12] has conducted the first systematic measurement study for network traffic and users' throughputs in public wireless local area networks

(WLAN). Accurate throughput models have been developed based on extensive measured data of the public WLAN in Schlotzsky's Inc., a national restaurant chain, and can be useful for optimization of throughput and energy in wireless networks. It has been found that mobile clients' throughputs with different applications in WLAN can be well modeled by the application type and the signal-to-interference-and-noise ratio (SINR) seen by users. The empirical throughput models were applied to blindly predict throughputs in different environments, and were shown to have good accuracy. It has also been found that the measured public WLAN traffic was highly asymmetric with higher traffic from access points (AP) to mobile clients. In addition, inbound and outbound packet sizes distributed very differently. Although file downloading and peer-to-peer applications sometimes generated high network demands, the majority of public WLAN users used HTTP protocol. Knowing the asymmetry helps the design and optimization of frequency allocations, transmit power control, and other aspects of wireless networks. In summary, the measurement results and models in [11,12] show that a key to future WLAN deployment may be to use accurate site-specific propagation algorithms for design, as well as real-time control of networks.

2.2 Prior Research on Frequency Allocations

Several algorithms in the literature [1,2,20–22] have tried to solve the WLAN frequency allocation problem in many different ways, but have not found the optimum solution.

The work in [21] assumes each AP has a different fixed traffic load, and defines the effective channel utilization of an AP as the fraction of time the channel is used for data transmission or is sensed busy due to interference from other APs; then, the maximum effective channel utilization among all APs is minimized. AP placement and frequency allocation are jointly optimized in [22] with the same objective of minimizing the max channel utilization as in [21].

The frequency allocation problem is modeled as a minimum-sum-weight vertex-coloring problem in [20] where vertices are APs, and the weight of each edge between two APs denotes

the number of clients that are associated with either one of these two APs and are interfered by the other AP. In brief, the objective of the algorithm in [20] is to find a frequency allocation that minimizes the conditional sum weight (the weight between any two APs is counted toward the conditional sum weight only if these two APs use the same channel). The authors in [2] mention that the shortcoming of [20] is the *over-estimation* of the interference seen by users. Consider a simple example where the interference regions of four APs overlap and have a single user that lies in the overlapping region. Then, the algorithm in [20] may assign four channels to these APs respectively so that the conditional sum weight is zero and thus minimized; otherwise, the conditional sum weight will be greater than zero. However, the algorithm in [2] can find the optimum solution for this illustrative example, which is that only two channels (say channel-1 and channel-2) are needed for zero interference at the only user, when this user is assigned to an AP on Channel-1 and the other three APs use Channel-2. The vertex-coloring method is inefficient in that edges between APs over-count the interference at the single user.

The work in [2] focuses on minimizing the channel conflicts as seen by users and has been shown to outperform [20]. Two key notions enable conflict minimization: the *range set* and *interference set* of each client. The *range set* of a client c consists of all APs that c can associate with. In other words, c can associate with any AP in the range set and form a client-AP link, and the other APs in the range set may potentially induce interference at c , if they are on the same channel as c . An AP a is in the *interference set* of a client c if a is not in the range set of c , and a or any client within range of a are within communication range of the client c or an AP within range of c . The interference set captures all possible APs whose downlink or uplink traffic may interfere with the traffic from or to c . The centralized algorithm in [2] takes the range set and interference set of all clients as input and maximizes the number of clients that are *conflict free*. A client c is *conflict free* if c is associated to an AP on channel- j , and no other AP in the range set or interference set of c is on channel- j . The work in [2] also presents a modified algorithm that incorporates load balancing. Yet [2] lacks a mechanism to handle *rogue RF interferers*, i.e., intentional or unintentional RF interferers,

microwave ovens, or other RF devices that also operate on the same unlicensed bands as WLAN. Hence the performance of [2] degrades in the presence of rogue interferers. Even with a few or no rogue interferer, the performance of [2] is not as good as our proposed algorithms in Chapters 3 and 4 due to the *binary interference model* used in [2].

The frequency allocation algorithm in [1] is distributed and runs at every AP. Each AP selects a channel according to a certain probability distribution, and measures the in-situ interference of the selected channel. If the measured interference at the AP is within a predefined acceptable level, the AP will stay on this channel; otherwise, this AP will set the probability of the current channel as zero, increase the probabilities of the other channels, and then re-select a channel. Suppose there exists a channel configuration for all APs so that the interference at every AP is within acceptable levels; then, the distributed channel selection algorithm in [1] is proven to converge to one such proper channel allocation. The weakness of the convergence result in [1], however, is the assumption that a feasible channel allocation must exist for all APs to be within acceptable interference. In networks with high density of APs or high interference from other RF devices, there may be no feasible channel allocation; in such a case, the algorithm in [1] does not converge. One could in principle set a higher acceptable level for the algorithm in [1] to work in high-interference regimes, but [1] does not mention methods to adapt the acceptable level. It is not trivial to adapt this level, since setting a high level will degrade network performance, but setting a low level will yield no feasible solutions. The non-convergence result of [1] in the high interference regime is due to the *binary interference model*. Our work in Chapters 3 and 4 use a *physical* rather than *binary* model for interference; that is, we assume that interference power is a continuous quantity, which properly represents the real world.

2.3 Prior Research on Transmit Power Control

Chiang and Bell [28] present algorithms to solve nonlinear utility¹ maximization over powers and rates for three scenarios of wireless cellular networks: (1) single-cell downlink case without interference; (2) multi-cell uplink/downlink case with interference; and (3) end-to-end connections in a hybrid (wireless and wired backbone) network. The second category in [28] is similar to the multiple-cell system this dissertation is interested. The work in [28] assumes that a central network controller knows which APs and clients are actively sending data (downlink or uplink, respectively), and optimizes transmit powers and transmission rates for these active APs and clients. Whenever the set of active APs and clients changes, the central network controller has to know the new set and perform the optimization of power and rates again. Obviously, the overhead induced by [28] is considerable.

Foschini and Miljanic [29] consider that base stations are always sending downlink traffic, and presents a distributed power control algorithm to minimize transmit powers so that each user's SINR meets the minimum SINR requirement. Uplink traffic is not considered in [29].

Xiao [30] *et al* focus on a case when no feasible transmit power solution exists to satisfy the SINR constraints for all clients in the entire wireless network; in such case, [29] does not converge. The work in [30] may turn off several nodes to reduce interference levels, in order that a solution that satisfies the SINR constraints can be found.

Hanly [31] and Yates and Huang [32] extend the work in [29] by considering jointly optimal base-station selection and power control.

¹Communication system performances are usually measured by some nonlinear network utility function of rates. In [28], the sum of all users' individual utility functions is the network utility. In general, a *network utility function* need not be a summation of all user's utility functions, but rather can be a nonlinear combination of them. For example, a network utility can be defined as the minimum of all users' utility function. Then, maximizing the network utility maximizes the bottleneck link in the entire network.

2.4 Prior Research on Load Balancing

Several heuristic load-balancing schemes for wireless LAN have been presented [26, 33–35]. First, several vendors (such as Cisco, Symbol, Trapeze, Aruba, and Meru) make APs with load-balancing functionalities; however, there is a paucity of literature on how these functionalities are designed, and we suspect ad-hoc methods or techniques based on wired-network literature are used.

Balachandran *et al* [26] observed that APs with such functionalities periodically send beacons with current load. Based on the data sheet, the load is denoted by the number of users, bit error rates, and signal strengths. However, several measurement studies have shown that the number of users is not a good metric to determine the load [12, 27]. Balachandran *et al* proposes that each arriving user explicitly asks for a minimum and a maximum bound on bandwidth/throughput [26]. Then, APs perform admission controls to associate the arriving user to an AP that is within the user’s radio range and has the most available capacity. The decision of admissions is made by a centralized admission control server that keeps the load information of all APs. The protocol design and system architecture for QoS negotiations and admission controls have been presented in detail in [26] and are shown to improve the degree of load balance by over 30% and user bandwidth allocation by up to 52% in comparison with schemes with little load balancing. The work in [33, 34] presents a decentralized load balancing algorithm that can be applied to IEEE 802.11a/b/g without modifying the standards while being transparent to end users. It was shown by example that the throughput of a station increases from 1.5 to 2 Mbps, and packet delays can be reduced from 450 to 8 ms. Fukuda *et al* provides decentralized AP selection algorithms [35], wherein a user either selects the AP that would possibly provide her the largest throughput or avoids the AP that has users with low throughput. Simulation results show that the algorithms in [35] allow approximately five times more users to achieve a certain throughput (e.g., 300 Kb/s). While work in [26, 33–35] outperform schemes with little or no load balancing, they are not shown to be optimal. To the best knowledge of the authors, the only work that achieves some form of optimality in load balancing is [36], which achieves max-min fairness of user bandwidth.

We note again that none of the load balancing work takes advantage of site specific knowledge. We will investigate the benefits and effects of using site specific knowledge on optimization algorithm designs.

Chapter 3

Improved Measurement-Based Frequency Allocation Algorithms for Wireless LAN or Cellular Networks

This chapter¹ presents three algorithms that outperform all other published work for allocating a limited number of orthogonal frequency channels to access points (APs) in wireless networks. Unlike other work, we minimize interference seen by both *users* and *APs*, we use a *physical* rather than *binary* model for interference, and we mitigate the impact of rogue RF interference. Our three algorithms have different mechanisms of switching the channels of APs based on the in-situ interference measured at clients and/or APs. The convergence of the algorithms is proven and characterized. Our algorithms consistently yield high throughput gains irrespective of network topology, the level of AP activity, and the number of controlled APs, rogue interferers, and available channels. We outperform the best published work by 15% and 18% for mean and median user throughputs respectively, and 81%, 168%, and 1011% for 25, 20, and 15 percentiles of user throughputs, respectively.

3.1 Overview, Main Contribution, and Organization of This Chapter

We consider wireless packet-switched networks formed by access points (or base stations) and their clients (or users). Each access point (AP) transmits downlink traffic or receives uplink traffic to/from its associated clients. APs can operate on orthogonal frequency channels to avoid interfering with one another. However, when the number of frequency channels is limited relative to the number of APs, some APs inevitably use the same channel and

¹Part of the work in this chapter has been submitted to *IEEE Globecom 2007* [37].

induce co-channel interference. In order to reduce such interference, judicious channel reuse mechanisms are necessary. The same problem exists in cellular networks.

Recall that the work in [2] minimizes the number of clients whose transmissions suffer channel conflicts; a client associated with an AP suffers conflicts if other clients or other APs interfere with the client or the AP under consideration. The definition of channel conflict in [2] is more comprehensive than those in [20–22]; the work in [2] has been shown to outperform [20–22].

None of [2, 20–22] considers that interference from independent, i.e., noncooperative networks or other RF interferers needs to be detected by sensors and avoided by proper frequency allocation. (We refer to such RF interferers as *rogue interferers* or *rogues* for the sake of brevity.) Rogue interferers are not uncommon [38], since the frequency bands for WLANs are unlicensed (2.4 and 5 GHz), and APs deployed by individuals in a spontaneous manner may induce interference on other noncooperative networks. The only published algorithm that handles rogue interferers is introduced in [1]; however, it does not converge in high-interference regime due to the *binary model for interference*, which is also used in [2, 20–22]. Our work considers a *physical* rather than binary model for interference; that is, we assume that interference power is a continuous quantity, which properly represents the real world.

Most traffic in WLANs is downlink [11]; hence, maximizing downlink throughput and signal-to-interference-and-noise ratio (SINR) seen by *users* are key to proper network design. The work in [1, 21, 22] minimizes the interference at *APs* rather than minimizes that at *users*, as is done in [2, 20], and thus often perform poorer than [2, 20].

The *main contribution* of this work is our three new algorithms that outperform all other published work, i.e., those in [1, 2, 20–22]. The proposed algorithms perform well mainly because they: (1) minimize interference seen by *users* rather than that seen by *APs*; (2) use a *physical* model rather than a *binary* model for interference; and (3) have the ability to deal with *rogue interferers*. We propose that all or a subset of clients measure the *in-situ* interference power on all frequency channels periodically when their associated APs are idle,

and report the average measured power to their associated APs. This technique is used in mobile-assisted hand-off (MAHO) in the cellular field [10], and results in this chapter may also be applied to cellular networks. APs also measure *in-situ* interference power. Since the measurements at APs or clients are performed during their idle time, the overhead is negligible. Each AP then computes a metric called *weighted interference* which captures the overall interference as seen by itself and its clients, by placing different weights on its and the clients' in-situ measurements according to the clients' traffic loads, signal strengths, and uplink and downlink traffic volume.

Organization: Section 3.2 introduces the system model and notation, and describes the *weighted interference* in detail. The three proposed algorithms, denoted *No-Coord*, *Local-Coord*, and *Global-Coord*, have different mechanisms for iteratively switching frequency channels in order to reduce the weighted interference seen in a single cell, a group of nearby cells, or all cells, respectively, where a *cell* means an AP (or base station) and its associated users. Section 3.3 presents the mechanisms used by the three algorithms and their convergence. Then Section 3.4 shows by simulation that our algorithms substantially outperform those in [1, 2, 20–22]. Section 3.5 concludes this chapter, followed by the appendix in Section 3.6, which describes the idea concerning modeling the interference as well as proofs to the theorems presented in this chapter.

3.2 System Model and Notation

We first describe basic notation; then Section 3.2.1 describes *weighted interference*, a metric used in the three proposed algorithms to capture the overall interference of each cell. The second subsection, Section 3.2.2 defines notation used exclusively for the proposed *Local-Coord* algorithm.

Basic Notation: Throughout this dissertation, we interchange the use of the terms ‘frequency channel’ or ‘channel’, whenever there is no ambiguity. Suppose M APs, indexed by $\mathbb{M} = \{1, 2, \dots, M\}$, operate on K orthogonal frequency channels, indexed by

$\mathbb{K} = \{1, 2, \dots, K\}$. By *orthogonality* we mean that any two APs operating on different frequency channels induce no, or negligible, interference on each other. We index users (or clients) by $\mathbb{L} = \{1, 2, \dots, L\}$. We denote the set of identities of APs by $\mathbb{X}^a = \{a_m : m \in \mathbb{M}\}$ and that of clients by $\mathbb{X}^c = \{c_l : l \in \mathbb{L}\}$, respectively. We assume for this work that the locations of the APs and the clients do not vary with time. Let $\mathbb{X} = \mathbb{X}^a \cup \mathbb{X}^c$, and assume that no APs or users are at the same location. Let \mathbb{L}_m ($\mathbb{L}_m \subseteq \mathbb{L}$) denote the set of users that are associated with the AP a_m . We assume every user is associated with a single AP; hence $\bigcup_{m \in \mathbb{M}} \mathbb{L}_m = \mathbb{L}$, and $\mathbb{L}_{m_1} \cap \mathbb{L}_{m_2} = \phi, \forall m_1, m_2 \in \mathbb{M}$, such that $m_1 \neq m_2$, where ϕ denotes the empty set. We define a *cell* \mathbb{Z}_m as the set which consists of a_m and all the users associated with a_m , that is, $\mathbb{Z}_m = \{a_m\} \cup \mathbb{L}_m$.

Let f_m ($f_m \in \mathbb{K}$) denote the channel that a_m operates on, and let $\vec{f} = (f_1, f_2, \dots, f_M)$ denote the channels of all M APs. Let $\mathbb{A}_k(\vec{f}) = \{m \in \mathbb{M} : f_m = k\}$ denote the set of APs that use channel k ; note that we explicitly write the dependence of \mathbb{A}_k on \vec{f} .

3.2.1 Metrics for Measurement-Based Frequency Channel Decisions

In brief, the weighted interference of each cell (say \mathbb{Z}_m) is intended to capture the overall interference in the cell, and is therefore defined as a weighted sum of the average in-situ measurements at a_m and at all clients associated with a_m , i.e., at every $u \in \mathbb{Z}_m$. We propose that a_m or the clients associated with a_m measure their in-situ interference power when there is no traffic within Cell \mathbb{Z}_m , i.e., a_m is neither transmitting or receiving data. The average in-situ measured interference power at u (for every $u \in \mathbb{Z}_m$) on channel k is denoted $I_k^u(\vec{f})$. The averaging period is a design choice and could be the same as the period that an AP switches its channel, say 1, 2, or 5 minutes. $I_k^u(\vec{f})$ is lower-bounded by noise floor.

An AP can choose a frequency channel in order to reduce the co-channel interference at the AP or at the users associated with this AP. We assume that each AP, say a_m , bases its choices on a particular weighted interference function $W_k^m(\vec{f})$ for different channels $k \in \mathbb{K}$. Note $W_k^m(\cdot)$ is a function of the channel allocation vector \vec{f} . Intuitively, a_m has a tendency to choose a channel with a lower $W_k^m(\vec{f})$. In the following, we will describe how $W_k^m(\vec{f})$ captures

the interference detected at both a_m and the users associated with a_m .

Suppose $u \in \mathbb{Z}_m$ denotes either an AP or a user in the cell \mathbb{Z}_m . When the devices within \mathbb{Z}_m are all *idle* (neither transmitting nor receiving), the device u measures the interference power at each channel k . Then u computes the average of the measured interference power on each channel k over a predefined period of time, and define this average as $I_k^u(\vec{f})$, $k \in \mathbb{K}$. In brief, $I_k^u(\vec{f})$ denotes the long-term average interference power that u measures at channel k . The averaging period T_A is a design choice and could be the same as the period that an AP switches its frequency channel, say 1, 2, or 5 minutes. $I_k^u(\vec{f})$ is lower-bounded by noise floor. Note that both the AP a_m and the users $\{c_l : l \in \mathbb{L}_m\}$ measure interference from all cells other than \mathbb{Z}_m ; hence, $I_k^u(\vec{f})$ does not include signals generated by devices in \mathbb{Z}_m . However, $I_k^u(\cdot)$ depends on f_m , i.e., the channel on which the devices in \mathbb{Z}_m operate. For example, if a_m operates on channel k , the devices in \mathbb{Z}_m induce interference on some devices in other cells that are also on channel k , which in turn degrade the data transmission rate and prolong the transmission time. Since the devices in other cells are active for a longer time interval, the devices in \mathbb{Z}_m see a larger interference from these other cells. The work in [39] first studied such mutual coupling phenomenon among multiple cells. Note that the *interference power* refers to not only the noise floor but also the interference from co-channel APs, users, or other rogue RF interferers that happen to be active when the device u is measuring the interference. Thus, interference, as used here, is actually interference plus noise. If u is an AP, it is reasonable to assume that u measures the interference power when u itself is *idle*, since such measurements would produce negligible overhead. On the other hand, if u is a user, u knows when a_m is idle by overhearing the control packets sent by a_m ; it is reasonable for u to measure the interference only when a_m is idle, since the transmit power of a_m or any other user in \mathbb{Z}_m should not be counted as the interference on u .

We define a weighted interference function as seen by AP a_m and its clients on channel k by

$$W_k^m(\vec{f}) = \sum_{u \in \mathbb{Z}_m} B_k^u(I_k^u(\vec{f})), \quad k \in \mathbb{K}, \quad (3.1)$$

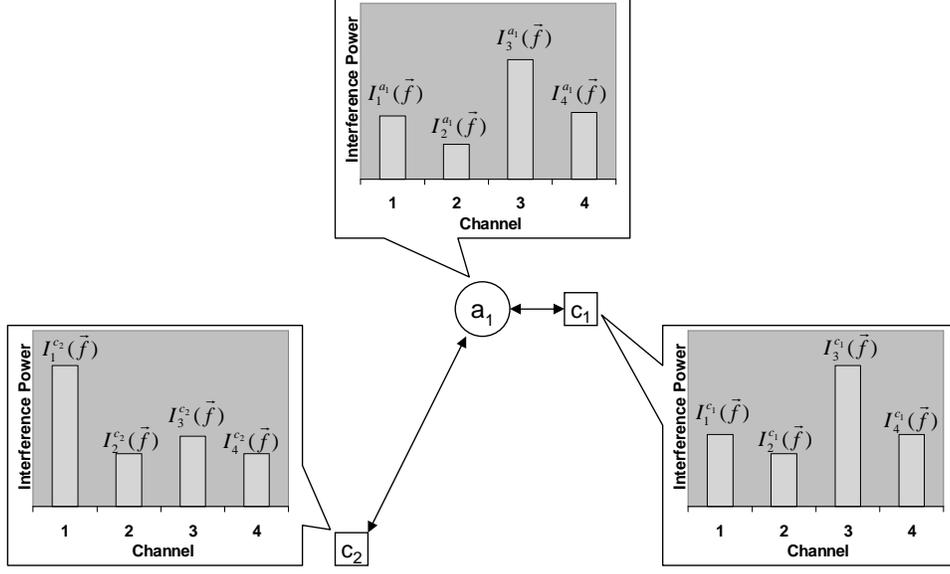


Figure 3.1: An example of measured interference power at a_1, c_1, c_2 for Channels 1 - 4.

where $B_k^u(\cdot)$ is a nonnegative and non-decreasing function that captures the weight of the in-situ measurement at u . We require that $W_k^m(\vec{f}) > 0$ to capture the noise floor in the real world. $B_k^u(\cdot)$ should be designed to reflect the difference of clients' traffic demands, signal strengths, and uplink and downlink traffic volume. The function $B_k^u(\cdot)$ captures the importance or weight of the *uplink and downlink traffic load* and *signal strength* between an AP and its users. Note that the weighted interference function defined here depends on \vec{f} , as mentioned earlier, is the case for the measured interference function $I_k^u(\vec{f})$. We will use an example to explain the significance of $W_k^m(\vec{f})$ and the design of $B_k^u(\cdot)$. Consider $K = 4$ and $\mathbb{Z}_1 = \{a_1, c_1, c_2\}$ in Fig. 3.1, and they receive interference $\{I_k^u(\vec{f}), u \in \mathbb{Z}_1, k \in \mathbb{K}\}$ from uncontrolled APs or RF devices that are not shown in the figure. We have $W_k^1(\vec{f}) = B_k^{c_1}(I_k^{c_1}(\vec{f})) + B_k^{c_2}(I_k^{c_2}(\vec{f})) + B_k^{a_1}(I_k^{a_1}(\vec{f}))$, $\forall k \in \mathbb{K} = \{1, 2, 3, 4\}$. Suppose c_1 receives a stronger signal from a_1 than c_2 , and thus c_1 has a larger tolerance to interference. Therefore, $B_k^{c_1}(\cdot)$ and $B_k^{c_2}(\cdot)$ should be designed to reflect that $I_k^{c_2}(\vec{f})$ contributes more to $W_k^1(\vec{f})$ than $I_k^{c_1}(\vec{f})$ does. Also note the interference seen by the AP affects uplink capacity, whereas that seen by the clients affect downlink capacity. Suppose, for example, the downlink traffic to c_1 is greater

than that to c_2 , which is also greater than the uplink traffic to a_1 . Then $B_k^{c_1}(\cdot)$, $B_k^{c_2}(\cdot)$, and $B_k^{a_1}(\cdot)$ should be designed to reflect that $I_k^{c_1}(\vec{f})$ contributes the most to $W_k^1(\vec{f})$, followed by $I_k^{c_2}(\vec{f})$, and $I_k^{a_1}(\vec{f})$.

The formulation in (3.1) is intended to be general. Nevertheless, $B_k^u(\cdot)$ should be designed to reflect the different of clients' traffic demands, signal strengths, and uplink and downlink traffic volume. In later sections, we will show that the convergence of two of our proposed channel allocation algorithms (namely *Local-Coord* and *Global-Coord*) is guaranteed if the weighted interference function has the general form in (3.1). Below we introduce two simplified forms of $W_k^m(\cdot)$ representing practical metrics. The first form, denoted *user-based*, places different weights on the in-situ interference measurements at clients based on the traffic volume and the signal strength at each client. The *user-based* form captures the performance of downlink transmission, which is appropriate for WLANs since traffic measurements show that downlink traffic volume accounts for more than 84% of total (uplink plus downlink) traffic volume [11]. The second form, denoted *AP-based*, includes the interference measurements at APs only. The *AP-based* form can be viewed as a simplified version of the *user-based* one by considering all users have the same traffic volume and signal strength.

1) *User-based*: The *user-based* weighted interference function for \mathbb{Z}_m is defined by

$$W_k^{(C),m}(\vec{f}) = \sum_{l \in \mathbb{L}_m} \frac{Y_{c_l, a_m}}{S_{c_l, a_m}} \cdot I_k^{c_l}(\vec{f}), \quad (3.2)$$

where S_{c_l, a_m} denotes the average received signal power² from a_m to c_l , and Y_{c_l, a_m} denotes the average traffic volume from a_m to c_l . We incorporate the inverse of S_{c_l, a_m} in (3.2) because a client with a stronger S_{c_l, a_m} has higher tolerance to interference and thus should contribute less to the overall weighted interference. Y_{c_l, a_m} is included in (3.2) as a scaling factor, since a client with higher traffic volume should be more important for the weighted interference. In

²Note c_l cannot measure S_{c_l, a_m} directly but can estimate S_{c_l, a_m} as follows. The average in-situ SINR at c_l can be measured at c_l when a_m is transmitting to c_l , and is denoted γ_l . We assume the interference at c_l is the same whether a_m is transmitting to c_l or a_m is idle, i.e., the interference at c_l is always $I_{f_m}^{c_l}(\vec{f})$. Then we estimate $S_{c_l, a_m} = \gamma_l \cdot I_{f_m}^{c_l}(\vec{f})$.

practice, some users may be sampled to reduce the complexity of computing (3.2), i.e., the summation in (3.2) may be over a subset of \mathbb{L}_m .

2) *AP-based*: Only APs measure and compute the average interference, and users do not take interference measurements. The *AP-based* weighted interference function for a_m is defined by

$$W_k^{(A),m}(\vec{f}) = I_k^{a_m}(\vec{f}). \quad (3.3)$$

As seen in (3.3) and (3.2), the computation of the weighted interference involves many components: measurements of interference, signal strength, traffic load, and so on. Simulation results show that measurement algorithms perform better with the *user-based* metric than the *AP-based*. In later sections, when we say that we *measure* or *compute* the weighted interference, we imply that we also measure all the components of the weighted interference.

The three proposed measurement-based algorithms in Section 3.3, along with the AP-based metric in (3.3) and the user-based metric in (3.2), yield six combinations of algorithms. In the simulation section (Section 3.4), we will evaluate the performance of these six combinations, along with the other two proposed site specific algorithms.

3.2.2 The Set of Interfering and Interfered Cells

The interference as seen by users or APs in $\mathbb{Z}_m, m \in \mathbb{M}$ may increase when some APs or users using the same channel are located within the proximity of \mathbb{Z}_m . Suppose a_m currently operates on channel f_m ; we would like to formally define the set of cells that may potentially affect the weighted interference $W_{f_m}^m(\cdot)$.

Definition 3.1. A cell \mathbb{Z}_n is an *interfering cell* of \mathbb{Z}_m if and only if there exists at least one pair of $(u, v), u \in \mathbb{Z}_m, v \in \mathbb{Z}_n$ so that ‘ v ’ induces non-negligible interference on ‘ u ’, given that a_m and a_n operate on the same channel. In other words, a_n or a user in cell \mathbb{Z}_n may interfere with the signal reception at a_m or a user in cell \mathbb{Z}_m , when a_n and a_m are on the same channel. We also say that the cell \mathbb{Z}_n interferes with the cell \mathbb{Z}_m .

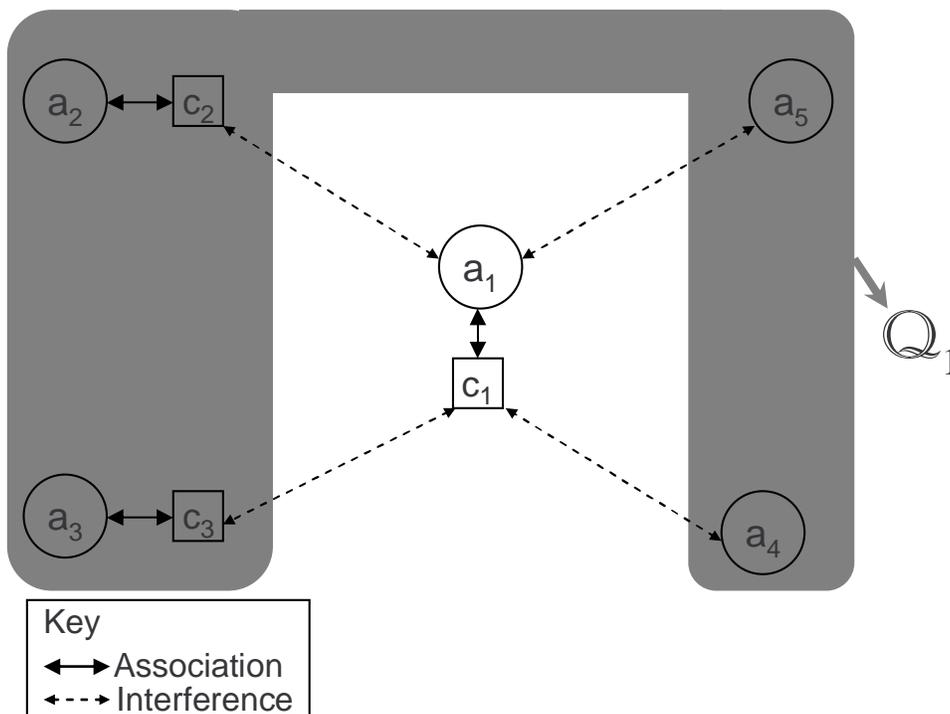


Figure 3.2: Potential interferers for transmissions between the AP a_1 and the user c_1 . The set of interfering cells of \mathbb{Z}_1 is $\mathbb{Q}_1 = \{2, 3, 4, 5\}$, as covered in the gray area.

Definition 3.2. We define \mathbb{Q}_m so that $n \in \mathbb{Q}_m$ if and only if \mathbb{Z}_n is an interfering cell of \mathbb{Z}_m . We call \mathbb{Q}_m the set of interfering cells of \mathbb{Z}_m .

By *non-negligible interference* we mean that the level of interference power that can be detected by RF receivers is non-negligible; for example, this level is more than the noise floor at an RF receiver.

For example, Fig. 3.2 depicts \mathbb{Q}_1 , the set of interfering cells of \mathbb{Z}_1 ; note that four possible types of interference are shown amongst APs and clients; i.e., between a_1 and another AP, a_1 and another user, c_1 and another AP, c_1 and another user. Note that the effect of interference may or may not be *reciprocal*, that is, $m \in \mathbb{Q}_n$ does not imply $n \in \mathbb{Q}_m$. Let us again use Fig. 3.2 as an example, we observe that a_4 's transmission interferes with the signal reception at c_1 , but the uplink transmission from c_1 to a_1 may not interfere the signal reception at a_4

since the transmit power of c_1 may be low. In such case, we have ‘ $4 \in \mathbb{Q}_1$ ’ but ‘ $1 \notin \mathbb{Q}_4$ ’. Hence we have the following converse definition.

Definition 3.3. We define \mathbb{G}_m such that $n \in \mathbb{G}_m$ if and only if $m \in \mathbb{Q}_n$. We call \mathbb{G}_m *the set of cells interfered by \mathbb{Z}_m* . For any $n \in \mathbb{G}_m$, we say that \mathbb{Z}_n is interfered by \mathbb{Z}_m .

Note that both \mathbb{Q}_m and \mathbb{G}_m are defined for the worst case where all APs are on the same channel. The following definition is for the case where APs are on different channels.

Definition 3.4. Suppose the channels on which APs are operating are \vec{f} . The cells interfered by \mathbb{Z}_m that are currently on channel k are defined and given by

$$\mathbb{G}_{m,k}(\vec{f}) \equiv \mathbb{G}_m \cap \mathbb{A}_k(\vec{f}). \quad (3.4)$$

The interfering cells of \mathbb{Z}_m that are currently on channel k are defined and given by

$$\mathbb{Q}_{m,k}(\vec{f}) \equiv \mathbb{Q}_m \cap \mathbb{A}_k(\vec{f}). \quad (3.5)$$

Again we explicitly denote the dependence of $\mathbb{G}_{m,k}(\cdot)$ and $\mathbb{Q}_{m,k}(\cdot)$ on \vec{f} .

The cells that are actually interfered by \mathbb{Z}_m are those operating in the same frequency as a_m , that is, $\mathbb{G}_{m,f_m}(\vec{f}) = \mathbb{G}_m \cap \mathbb{A}_{f_m}(\vec{f})$. Similarly, the cells that interfere with \mathbb{Z}_m are $\mathbb{Q}_{m,f_m}(\vec{f}) = \mathbb{Q}_m \cap \mathbb{A}_{f_m}(\vec{f})$. Note that \mathbb{Q}_m and \mathbb{G}_m depend on the locations of APs and users, as well as on the radio propagation characteristics of the environment; according to the definition, \mathbb{Q}_m and \mathbb{G}_m do *not* vary with \vec{f} , although $\mathbb{G}_{m,k}(\cdot)$ and $\mathbb{Q}_{m,k}(\cdot)$ depend on \vec{f} . The concept of the sets of interfering and interfered cells in our work is similar to the notion of interference set introduced in [2] except that our definitions center on cells rather than clients.

Naturally, the measure of interference should monotonically increase as there are more interfering cells using the same channel. This is formally stated in the following assumption. The following proposition introduces a similar property for the weighted interference function.

Assumption 3.1. For any two channel allocations \vec{f} and \vec{g} , $\forall m \in \mathbb{M}$, $\forall k \in \mathbb{K}$, such that $\mathbb{Q}_{m,k}(\vec{f}) \subseteq \mathbb{Q}_{m,k}(\vec{g})$, we assume $I_k^u(\vec{f}) \leq I_k^u(\vec{g})$ for all $u \in \mathbb{Z}_m$. Furthermore, if \vec{f} and \vec{g} satisfy $\mathbb{Q}_{m,k}(\vec{f}) \subsetneq \mathbb{Q}_{m,k}(\vec{g})$, we assume $I_k^u(\vec{f}) < I_k^u(\vec{g})$ for all $u \in \mathbb{Z}_m$.

Proposition 3.1. For any two channel allocations \vec{f} and \vec{f}' , $\forall m \in \mathbb{M}, \forall k \in \mathbb{K}$, such that $\mathbb{Q}_{m,k}(\vec{f}) \subseteq \mathbb{Q}_{m,k}(\vec{g})$, we have $W_k^m(\vec{f}) \leq W_k^m(\vec{g})$. Furthermore, if \vec{f} and \vec{g} satisfy $\mathbb{Q}_{m,k}(\vec{f}) \subsetneq \mathbb{Q}_{m,k}(\vec{g})$, we have $W_k^m(\vec{f}) < W_k^m(\vec{g})$.

Proof. We will prove the first part, $W_k^m(\vec{f}) \leq W_k^m(\vec{g})$. The proof of the strict inequality can be done in the same manner.

$$W_k^m(\vec{f}) = \sum_{u \in \mathbb{Z}_m} B_k^u(I_k^u(\vec{f})) \quad (3.6)$$

$$\leq \sum_{u \in \mathbb{Z}_m} B_k^u(I_k^u(\vec{g})) \quad (3.7)$$

$$= W_k^m(\vec{g}), \quad (3.8)$$

where (3.6) and (3.8) hold according to the definition in (3.1). Equation (3.7) holds because $B_k^u(\cdot)$ is non-decreasing by definition, and $I_k^u(\vec{f}) \leq I_k^u(\vec{g})$ by Assumption 3.1. \square

Note that having $\mathbb{Q}_{m,k}(\vec{f})$ in Proposition 3.1 captures the spatial characteristics that distinguish wireless from wired communications. In other words, the weighted interference $W_{f_n}^m(\cdot)$ is affected only by the APs in the proximity of a_m ; this characteristic will help us design a scalable and distributed channel-assignment algorithm, namely *Local-Coord*.

Suppose a_m switches from channel k to k' , the cells that see changes in their weighted interference are \mathbb{Z}_m and the cells indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$; hence the weighted interference of the cells indexed by $\mathbb{H}_{m,k,k'}(\vec{f}) \equiv \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})$ are examined by *Local-Coord* if a_m switches from channel k to k' .

Another AP a_n can run *Local-Coord* simultaneously with a_m if the channel switching of a_n induces negligible change of the weighted interference of the cells that may be examined by *Local-Coord*, i.e. \mathbb{Z}_m and the cells indexed by \mathbb{G}_m . We define \mathbb{V}_m as the set of the indices of cells that interfere with \mathbb{Z}_m or the cells indexed by \mathbb{G}_m , i.e. $i \in \mathbb{V}_m$ if and only if there exists $j \in \{m\} \cup \mathbb{G}_m$ such that \mathbb{Z}_i interferes with \mathbb{Z}_j . The cells indexed by \mathbb{V}_m include all the cells that cannot simultaneously change channels with a_m . The notation of \mathbb{V}_m is used for

the distributed protocol of *Local-Coord*. Suppose we are given the locations of all controlled APs and possible locations of clients; then the sets of \mathbb{G}_m and \mathbb{V}_m can be pre-computed and pre-configured in the controlled APs or a central network controller that communicates with the controlled APs, using radio propagation prediction models as described in [10, 40, 41].

3.3 Three Measurement-Based Algorithms

Three proposed algorithms all have an iterative nature. At each point in time (pre-defined, randomly chosen, or determined at runtime), say every 1, 2, or 5 minutes, one *iteration* of channel switching takes place where one or more APs switch their frequency channels according to mechanisms that are specific to the proposed algorithms and are the *weighted interference function* as defined in Section 3.2.1, while other APs stay on their current channels. In hardware, the time needed for channel switching is on the order of milliseconds and is thus negligible as compared to the time interval between two iterations. APs and clients measure and average their in-situ interference between every two successive iterations, and compute the *weighted interference function*. Iterations keep taking place on different AP(s) until the number of iterations reaches a limit or the overall AP frequency allocation reaches a state where no APs will change their channels according to the specified mechanism, which we refer to as *convergence*.

Below we describe the three proposed algorithms that all use the metric $W_k^m(\vec{f})$ to make frequency channel selection decisions. The three algorithms have different levels of coordination among APs. We call them *No-Coord*, *Local-Coord*, and *Global-Coord* to represent no coordination, local coordination, and global coordination, respectively. In the description of the following three algorithms, we suppose that at any point of time, only one AP (say a_m) changes its channel. We relax this assumption for *Local-Coord* to allow multiple APs to switch their channels concurrently; the relaxation makes the proposed *Local-Coord* algorithm scalable. We also study the convergence and the characteristics of convergence points of the three algorithms.

Throughout this chapter, $\vec{f}' \in \mathbb{K}^M$ denotes a vector of channels selected by APs after the representative AP a_m moves from channel f_m to f'_m . Hence \vec{f}' differs from \vec{f} in only the m -th element.

To evaluate the gain of using the interference measurements reported from *users*, we consider a baseline case where only *APs* measure in-situ interference. We refer to these two kinds of interference measurements as *user-based* and *AP-based*, respectively. For each of the three measurement-based algorithms, we compare the user-based with the AP-based. AP-based measurements may not reflect the interference that users see, since users on different locations see different interference. Since user-based interference measurements directly affect the throughput of downlink traffic, which is the majority of WLAN traffic [11, 27], our hypothesis is that the proposed algorithms achieve higher throughput by using the *user-based* measurements, as compared to using the *AP-based*. This hypothesis is corroborated by our simulation results given in Section 3.4.

3.3.1 The No-Coord Algorithm

In each iteration, one or more APs changes its (or their) channels, and each of these APs, without coordinating with any other APs, moves to a new channel where the AP and the users associated with the AP see a lower weighted interference (for simplicity, we say that *the ‘cell’ of the AP* sees a lower interference). In other words, each AP makes a simple local greedy decision for channel selection. Since this algorithm requires no coordination among APs, it is denoted *No-Coord*. In other words, a representative AP a_m will switch from its current channel $f_m = k$ to $f'_m = k'$ only if

$$\text{No-Coord Condition: } W_k^m(\vec{f}) > W_{k'}^m(\vec{f}'). \quad (3.9)$$

3.3.2 The Local-Coord Algorithm

Assume only one AP, represented by a_m , changes its channel in each iteration. If a_m switches from channel k to k' , \mathbb{Z}_m and the cells in $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f}')$ will see changes

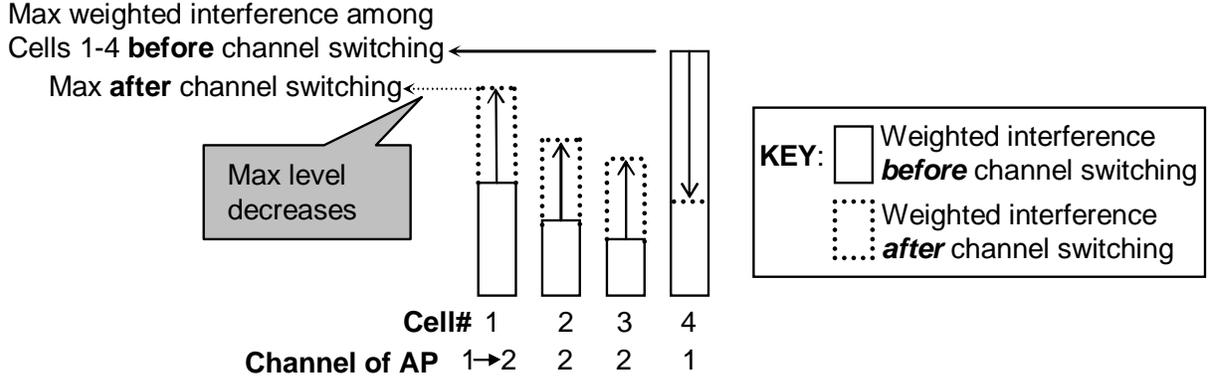


Figure 3.3: The heights of *Solid* and *dotted* bars respectively signify the weighted interference power seen by cells *before* and *after* AP-1 switches from Channel 1 to Channel 2. The max weighted interference seen by Cells 1 – 4 decreases after AP-1 switches to Channel 2.

in their weighted interference. If the max interference seen by these cells decreases after a_m switches to the new channel k' , a_m remains on the new channel; otherwise, a_m returns to the original channel k . Note a_m needs to *locally* coordinate with the APs in $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$ for the channel adjustment; hence this algorithm is denoted *Local-Coord*. The *Local-Coord* Condition is

$$\max_{i \in \mathbb{H}_{m,k,k'}(\vec{f})} W_{f_i}^i(\vec{f}) > \max_{i \in \mathbb{H}_{m,k,k'}(\vec{f})} W_{f_i}^i(\vec{f}'), \quad (3.10)$$

where $\mathbb{H}_{m,k,k'}(\vec{f})$ has been defined in Section 3.2.2. This algorithm is denoted *Local-Coord*, since a_m needs to *locally* coordinate with the APs indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$ via wired backbone network for the channel switching.

For example, Fig. 3.3 depicts the cells that see changes in weighted interference before and after AP-1 switches its channel. Since the max weighted interference seen by Cells 1 – 4 decreases, AP-1 remains on the new channel.

Section 3.3.2 will describe in detail that every iteration of *Local-Coord* has such a decreasing characteristic, or more precisely, the interference seen by APs are *lexicographically decreasing* in each iteration. This decreasing characteristic enables us to prove that this algorithm converges in a finite number of iterations.

Table 3.1: A variable ψ_m used in the protocol for *Local-Coord*.

ψ_m	Channel switching at a_m	Can a_m be locked?
-1	a_m is in the process of switching its channel	No
0	a_m can initiate the process of channel switching	Yes
1 or more	a_m cannot initiate the process of channel switching	Yes

The condition in (3.10) enables us to prove the convergence of the *Local-Coord* algorithm in Theorem 3.2 and show the characteristics of the convergence point in Section 3.3.6. This condition also implies that the decision of a_m is influenced by the weighted interference of the cells in $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$. In other words, a_m needs to *locally* coordinate with the APs in $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f})$ for the channel adjustment. It is why this algorithm is called *Local-Coord*. The nature of local coordination also implies that any cell that does not interfere with the devices in \mathbb{Z}_m , $\mathbb{G}_{m,k}(\vec{f})$, and $\mathbb{G}_{m,k'}(\vec{f})$ can switch its channel at the same time as a_m . We have the following formal definition.

Definition 3.5. We define \mathbb{V}_m as the set of cells that cannot simultaneously change channels with a_m , i.e., $i \in \mathbb{V}_m$ if and only if there exists $n \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})$ so that $i \in \mathbb{Q}_n$.

Since coordination among APs is confined in a local area, multiple APs that are far apart enough can change their channels in the same iteration if a proper inter-AP protocol is employed. We define the *distance* between two cells as the maximum distance between two devices in the two cells respectively. The distance between \mathbb{Z}_m and any cell $\mathbb{Z}_n, n \in \mathbb{V}_m$ is upper bounded because of the decaying nature of radio wave propagation. Therefore, the number of APs that can simultaneously change channels grows linearly (asymptotically) with the number of total APs and depends on spatial density of APs and propagation characteristics between APs. Hence, the *Local-Coord* algorithm is scalable.

Fig. 3.4 presents a protocol for coordination among APs. First we suppose that each AP has an independent random timer that triggers the AP to initiate the process of switching its channel. Part (a) of Fig. 3.4 describes the procedure that a representative AP a_m performs after the timer at a_m is triggered. We say an AP a_m is *locked*, if a_m is not allowed to switch

(a) Suppose a timer triggers a_m to consider initiating a channel switching. Then a_m will do the following procedure.

- 1: **if** $\psi_m = 0$ **then**
- 2: *Phase 1:* Set $\psi_m = -1$ and send requests to lock all APs indexed by \mathbb{V}_m , i.e., $\{a_n : n \in \mathbb{V}_m\}$.
- 3: *Phase 2:* Wait for replies from $\{a_n : n \in \mathbb{V}_m\}$.
- 4: **if** the replies indicate that $\{a_n : n \in \mathbb{V}_m\}$ were all successfully locked by a_m **then**
- 5: a_m switches its channel from k to k' , and stays at k' if (3.10) is satisfied; otherwise, a_m switches back to channel k .
- 6: Send messages to unlock $\{a_n : n \in \mathbb{V}_m\}$.
- 7: **else**
- 8: Send messages to unlock the APs among $\{a_n : n \in \mathbb{V}_m\}$ that were just successfully locked by a_m . (Do not need to unlock the APs that could not be locked by a_m .)
- 9: **end if**
- 10: Set $\psi_m = 0$.
- 11: **end if**

(b) Upon receiving a *locking request* from a_m , a_n will do the following procedure.

- 1: **if** $\psi_n \neq -1$ **then**
- 2: Increase ψ_n by one.
- 3: Reply to a_m that a_n was successfully locked by a_m .
- 4: **else**
- 5: Reply to a_m that a_n could not be locked.
- 6: **end if**

(c) Upon receiving an *unlocking request* from a_m , a_n will decrease ψ_n by one.

Figure 3.4: A protocol for the distributed implementation of *Local-Coord*.

its channel per other APs' requests; if a_m is *unlocked*, a_m may switch its channel. If a_m has been locked by other APs, a_m will ignore this trigger and wait for the next time the timer at a_m is triggered. The cells indexed by \mathbb{V}_m cannot simultaneously change channels with a_m . The key idea of this protocol is that a_m needs to lock all the APs indexed by \mathbb{V}_m before it switches to a new channel, and then unlocks those APs. The procedure to handle locking and unlocking requests are described in parts (b) and (c) of Fig. 3.4, respectively. Phases 1 and 2 of the proposed protocol make sure that all the APs indexed by \mathbb{V}_m can be locked before a_m switches its channel. If any AP indexed by \mathbb{V}_m cannot be locked, a_m cannot switch its channel. When an AP is locked, it cannot initiate a process of channel switching. An AP can be locked for multiple times by other APs. Let ψ_m denote the number of times that a_m has been locked. Only when $\psi_m = 0$ can a_m initiate the process of channel switching. When a_m is in the process of switching its channel (denoted by $\psi_m = -1$), it cannot be locked; we denote this condition by setting $\psi_m = -1$. Table 3.1 describes the significance of ψ_m .

If a distributed protocol is not carefully designed, a problem called *deadlock* may occur. In the context of distributed frequency allocation, a *deadlock* is a situation wherein two or more APs that have initiated the process of switching their channels are waiting for each other to finish before any of these APs proceeds to finish the process of channel switching, and thus none of these APs can ever finish. We have the following theorem.

Theorem 3.1. *Deadlocks do not occur in the distributed protocol described in Fig. 3.4.*

Proof. The deadlock situation arises only when all of the four necessary conditions described in [42] are operative. One of the four conditions is the “no preemption” condition, where resources cannot be forcibly removed from the tasks holding them until the resources are used to completion. In the context of the protocol in Fig. 3.4, a *task* is the channel switching at an AP, and a *resource* is an AP. Phase-3 of the protocol implies that if not all the resources can be locked, the task will release locked resources³. Thus, the “no preemption” condition will not arise. □

³Havender has suggested this so-called “all-or-none” approach for deadlock prevention [43].

3.3.3 The Global-Coord Algorithm

The third measurement-based algorithm considers the *sum* weighted interference over all APs on each channel. An AP will switch to a new channel only if the *sum* interference on the new channel is lower (after the AP switches there) than the *sum* interference on its current channel, i.e., only if the following condition holds.

$$\textit{Global-Coord Condition: } \sum_{n:f_n=k} W_k^n(\vec{f}) > \sum_{n:f_n=k'} W_{k'}^n(\vec{f}'). \quad (3.11)$$

In other words, a_m considers a globally ‘good’ channel k' that has a lower sum weighted interference, even after a_m switches to k' . This algorithm requires global coordination among APs using a central network controller that communicate with all APs, and is thus denoted *Global-Coord*.

3.3.4 Implementation Concerns

Note that in the descriptions of the three proposed algorithms, some terms of weighted interference are unknown before the AP under consideration (say a_m) switches to the new channel. An implementation may require a_m to switch to a new channel by trial, and then require one or more cells to measure and compute their weighted interference after a_m switches to the new channel. Thus, only when all the quantities needed for the channel decisions are known can a_m decide whether switching to the new channel complies with the condition described for each algorithm. If the condition is satisfied, a_m will accept this channel switch and stay on the new channel; otherwise, a_m may switch back to the old channel or try another possible channel to switch to. *No-Coord* requires the weighted interference at cell \mathbb{Z}_m , *Local-Coord* at cells $\{\mathbb{Z}_i : i \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f}')\}$, and *Global-Coord* at all cells. Note that this trial process should not be purely blind guesses, but can be improved by using channel estimation techniques.

3.3.5 Convergence

Theorem 3.2. *Consider a particular realization of the locations of APs and users (i.e., $\mathbb{X}^a, \mathbb{X}^c$) and a weighted interference function of the form of (3.1). Given any set of initial AP channel choices, the channel selection process converges for Local-Coord and Global-Coord in a finite number of steps.*

In proving Theorem 3.2, we will use the following definitions and two lemmas. The proofs of the lemmas are presented in Section 3.6.2.

Definition 3.6. Suppose we have two vectors with the same length N , say $\vec{v} = (v_1, v_2, \dots, v_N)$ and $\vec{v}' = (v'_1, v'_2, \dots, v'_N)$. We sort the elements of \vec{v} in non-increasing order and denote it as $\vec{u} = (u_1, u_2, \dots, u_N)$, and similarly denote \vec{u}' as the non-increasing sorted version of \vec{v}' . We say that \vec{v} *lexicographically dominates* \vec{v}' (or $\vec{v} \succ \vec{v}'$) if there exists some index j , where $N \geq j \geq 1$ for which $u_j > u'_j$ and $u_i = u'_i$ for all $i < j$. We also say that \vec{v} *has a higher lexicographic order than* \vec{v}' .

Definition 3.7. Two vectors \vec{v} and \vec{v}' have *the same lexicographic order* if their non-increasing sorted versions \vec{u} and \vec{u}' are element-wise the same.

Definition 3.8. We say $\vec{v} \succeq \vec{v}'$ if $\vec{v} \succ \vec{v}'$ or \vec{v} and \vec{v}' have the same lexicographic order.

Lemma 3.1 (The lexicographically decreasing nature of *Local-Coord*). *Suppose a_m is a representative AP switching its channel from k to k' according to the Local-Coord Condition in (3.10), and the channels of all the other APs remain unchanged. Let $\vec{f}, \vec{f}' \in \mathbb{K}^M$ respectively denote a vector of channels selected by APs before and after the representative AP a_m moves from channel f_m to f'_m . Hence \vec{f}' differs from \vec{f} in only the m -th element, that is, $f'_n = f_n, \forall n \neq m$, and $f'_m \neq f_m$. Define*

$$\vec{\alpha}(\vec{f}) = \left(W_{f_1}^1(\vec{f}), W_{f_2}^2(\vec{f}), \dots, W_{f_M}^M(\vec{f}) \right) \quad (3.12)$$

$$\vec{\alpha}(\vec{f}') = \left(W_{f'_1}^1(\vec{f}'), W_{f'_2}^2(\vec{f}'), \dots, W_{f'_M}^M(\vec{f}') \right). \quad (3.13)$$

Then we have $\vec{\alpha}(\vec{f}) \succ \vec{\alpha}(\vec{f}')$.

The proof is in Section 3.6.2.

Lemma 3.2 (The lexicographically decreasing nature of *Global-Coord*). *Suppose a_m is a representative AP switching its channel from k to k' according to Global-Coord Condition in (3.11), and the channels of all the other APs remain unchanged. Define*

$$\vec{\beta}(\vec{f}) = \left(\sum_{n:f_n=1} W_1^n(\vec{f}), \sum_{n:f_n=2} W_2^n(\vec{f}), \dots, \sum_{n:f_n=K} W_K^n(\vec{f}) \right) \quad (3.14)$$

$$\vec{\beta}(\vec{f}') = \left(\sum_{n:f'_n=1} W_1^n(\vec{f}'), \sum_{n:f'_n=2} W_2^n(\vec{f}'), \dots, \sum_{n:f'_n=K} W_K^n(\vec{f}') \right). \quad (3.15)$$

Then we have $\vec{\beta}(\vec{f}) \succ \vec{\beta}(\vec{f}')$.

The proof is in Section 3.6.2.

Proof of Theorem 3.2. We will first prove the convergence of *Local-Coord*. We form a directed graph \mathcal{G} with all possible channel vectors \vec{f} as nodes and all channel adjustments that satisfy *Local-Coord* Condition in (3.10) as edges (recall that we assume only one AP switches its channel at any point of time). We will show that this graph is *acyclic*, then we can conclude that the algorithms will converge to one of possibly many sink nodes in \mathcal{G} . Note that the number of nodes in \mathcal{G} is finite, since the number of channel choices is finite. Since \mathcal{G} is acyclic and finite, any initial channel selection will converge to a sink in a finite number of steps. We can show the *acyclic* nature of \mathcal{G} by contradiction: First we note that lexicographic order is a *total order* [44] and thus possesses the transitive property, that is, if $\vec{v} \succ \vec{v}'$ and $\vec{v}' \succ \vec{v}''$, then $\vec{v} \succ \vec{v}''$. Suppose there exists a cycle on \mathcal{G} . Suppose $\vec{f}^0, \vec{f}^1, \vec{f}^2, \dots$ are nodes on this cycle. As we travel through this cycle once, we will see that $\vec{\alpha}(\vec{f}^0) \succ \vec{\alpha}(\vec{f}^1) \succ \vec{\alpha}(\vec{f}^2) \succ \dots \succ \vec{\alpha}(\vec{f}^0)$ according to Lemma 3.1. This implies $\vec{\alpha}(\vec{f}^0) \succ \vec{\alpha}(\vec{f}^0)$ according to the transitive property of lexicographic order, which is a contradiction since $\vec{\alpha}(\vec{f}^0)$ does *not* lexicographically dominate itself (see Definition 3.6).

The proof of *Global-Coord* is similar to that of *Local-Coord* except that the edges of \mathcal{G} are all the channel adjustments satisfying *Global-Coord* Condition in (3.11), and $\vec{\alpha}(\cdot)$

is replaced with $\vec{\beta}(\cdot)$. Based on Lemma 3.2, we can prove the convergence using the same argument for *Local-Coord*. \square

3.3.6 Characterization of Convergence Points

We will characterize the convergence of frequency allocations for *No-Coord*, *Local-Coord*, and *Global-Coord*, respectively. First, we use the well-known Nash equilibria notation [45] for frequency allocation.

Definition 3.9. A vector of frequency allocations denoted by \vec{f} is a *Nash equilibrium* (a concept widely used in game theory [46]), if no single cell can lower its weighted interference by changing only its own channel.

Then, we define local and global minimums with respect to lexicographic order.

Definition 3.10. We say that a vector of frequency allocations denoted by $\vec{f} \in \mathbb{K}^M$ is at a *local lexicographic minimum* with respect to a vector function $\vec{\theta}(\cdot)$, if for any vector of frequency allocations $\vec{f}' \in \mathbb{K}^M$ that differs from \vec{f} in only one element, $\vec{\theta}(\vec{f}') \succeq \vec{\theta}(\vec{f})$ holds true.

We have the following results based on Definitions 3.9 and 3.10.

Theorem 3.3. *Suppose No-Coord converges to a frequency allocation \vec{f} . Then, \vec{f} is a Nash equilibrium.*

Proof. We will prove this by contradiction. Suppose *No-Coord* converges at a frequency allocation \vec{f} , but \vec{f} is *not* a Nash equilibrium. Then there exists at least one AP, say a_m , and one channel f'_m that is different from a_m 's current channel f_m so that a_m can switch from its current channel f_m to f'_m to strictly decrease the weighted interference of cell \mathbb{Z}_m , i.e., $W_{f'_m}^m(\vec{f}') < W_{f_m}^m(\vec{f})$. Then, the frequency allocation should not have converged, since a_m can switch to channel f'_m according to the *No-Coord* condition in (3.9). \square

Note that *No-Coord* does not always converge, although simulation results show that *No-Coord* converges in most cases. Theorem 3.3 is for the cases where *No-Coord* converges; the frequency allocation is converged to a Nash equilibrium. One may *limit the number of iterations* or *specify a minimum gradient slope to rescue and implement No-Coord*.

Below we state a technical assumption useful in proving Theorems 3.4 and 3.5 in Section 3.3.6.

Assumption 3.2. Since the weighted interference in (3.1) takes a continuum of values, it is reasonable to assume that the weighted interference values at different cells or channels are distinct, i.e., $\forall k, j \in \mathbb{K}, \forall m, n \in \mathbb{M}$ such that $k \neq j$ or $m \neq n$, we have $W_k^m(\vec{f}) \neq W_j^n(\vec{f})$ with probability one.

Theorem 3.4. *Suppose Local-Coord converges at a frequency allocation \vec{f} . Then with probability one, \vec{f} is at a local lexicographic minimum with respect to the vector function $\vec{\alpha}(\cdot)$ as defined in (3.12).*

The proof is in Section 3.6.2. □

Theorem 3.5. *Suppose Global-Coord converges at a frequency allocation \vec{f} . Then with probability one, \vec{f} is at a local lexicographic minimum with respect to the vector function $\vec{\beta}(\cdot)$ as defined in (3.14).*

The proof is in Section 3.6.2. □

3.4 Simulation Results

Section 3.4.1 describes the simulation setup, where we consider a scenario with only downlink traffic and another scenario with downlink and uplink traffic. Sections 3.4.2 and 3.4.3 discuss the simulation results for the two scenarios respectively.

No.	Algorithm	Abbr.
1	<i>Local-Coord</i> User-based	<i>Lo-U</i>
2	<i>No-Coord</i> User-based	<i>No-U</i>
3	<i>Global-Coord</i> User-based	<i>Gl-U</i>
4	<i>Local-Coord</i> AP-based	<i>Lo-A</i>
5	<i>No-Coord</i> AP-based	<i>No-A</i>
6	<i>Global-Coord</i> AP-based	<i>Gl-A</i>
7	<i>CFAssign-RaC</i> [2]	<i>CF</i>
8	<i>Leith-Clifford</i> [1]	<i>LC</i>

Table 3.2: Three proposed measurement-based algorithms and two types of weighted interference functions yield six combinations that are shown between the first and the sixth row. The last two rows show two other published algorithms. The last column shows the abbreviation of each algorithm.

3.4.1 Simulation Setup

We will compare our proposed algorithms against *CFAssign-RaC* in [2] and *Leith-Clifford* in [1]. Table 3.2 lists all the algorithms to be compared. The three proposed measurement-based algorithms (presented in Sections 3.3.1, 3.3.2, and 3.3.3, respectively) with two possible weighted interference functions, namely *AP-based* in (3.3) and *user-based* in (3.2) yield six combinations of algorithms, as shown between the first and the sixth row in Table 3.2. Note that the AP-based counterpart of each measurement-based algorithm serves as a baseline to determine the gain of using the more complicated version of *user-based* weighted interference metric. As described in the overview section (Section 3.1), *CFAssign-RaC* has been shown to outperform the frequency allocation algorithms in [20–22], yet it does not consider rogue RF interferers. Leith and Clifford in [1] introduced a first frequency allocation algorithm to deal with rogues, but they did not consider the interference as seen by users. Our proposed algorithm consider the factors neglected by [1, 2, 20–22] and therefore should yield better frequency allocation.

According to [11, 12, 27], the ratio of downlink volume to uplink volume is 5:1 in typical wireless LAN. In the first part of our simulations, we assume all traffic is *downlink*, and we

optimize the frequency allocation for a *worst case* where *all APs are actively transmitting downlink traffic*. It is reasonable that frequency allocation is optimized with respect to this worst case, since in this worst case, frequency allocation is crucial for interference mitigation at users. Section 3.4.2 presents numerical results for this downlink-only scenario. Then, we examine the performance of the optimized frequency allocations in the presence of both downlink and uplink traffic, and present the results in Section 3.4.3. It has been shown in [39] that uplink and downlink capacities in multiple cells are mutually coupled due to inter-cell interference, and no system-level analytic model has been found to model activities of multiple APs. We consider that time is slotted, and propose an approximate probabilistic model where APs independently choose one of the three possible activity states at each time slot. An AP is can be transmitting downlink traffic, receiving uplink traffic, or idle, with probabilities p_d , p_u , and $p_i = 1 - p_d - p_u$, respectively. For any AP that is transmitting downlink traffic or receiving uplink traffic at a certain time slot, a user is randomly chosen (with uniform probability distribution) out of all the users associated with this AP to be the recipient or the sender of the traffic. We fix the ratio of p_d to p_u as 5:1, and simulate eight cases where $p_d + p_u$ (the probability that an AP is active) is $1/8, 2/8, 3/8, \dots, 8/8$, respectively. We intend to see the effect of $p_d + p_u$ on the performance of the proposed algorithms. For each case we simulate 10 independent runs, each with 100,000 time slots. Note that we assume that the activity of each AP is independent from the other APs. This assumption is not true in reality, but it simplifies the simulations and provides a rule-of-thumb for the performance comparison.

To evaluate the performance of various frequency allocation algorithms, we follow the same procedure. For each algorithm, we first assign random frequency channels to APs and run each algorithm mentioned in Table 3.2 until it converges. Then, we compute the performance metric (e.g. user throughput) at the converged frequency allocation, as can be seen in the next subsection. Note that for the *No-Coord* algorithm (either AP-based and user-based) whose convergence is not guaranteed, we limit the number of iterations, where an *iteration* denotes a change of frequency channel at a single AP. In most cases, *No-Coord* converges quickly, and the number of iterations is usually less than the number of APs. Therefore we can limit the

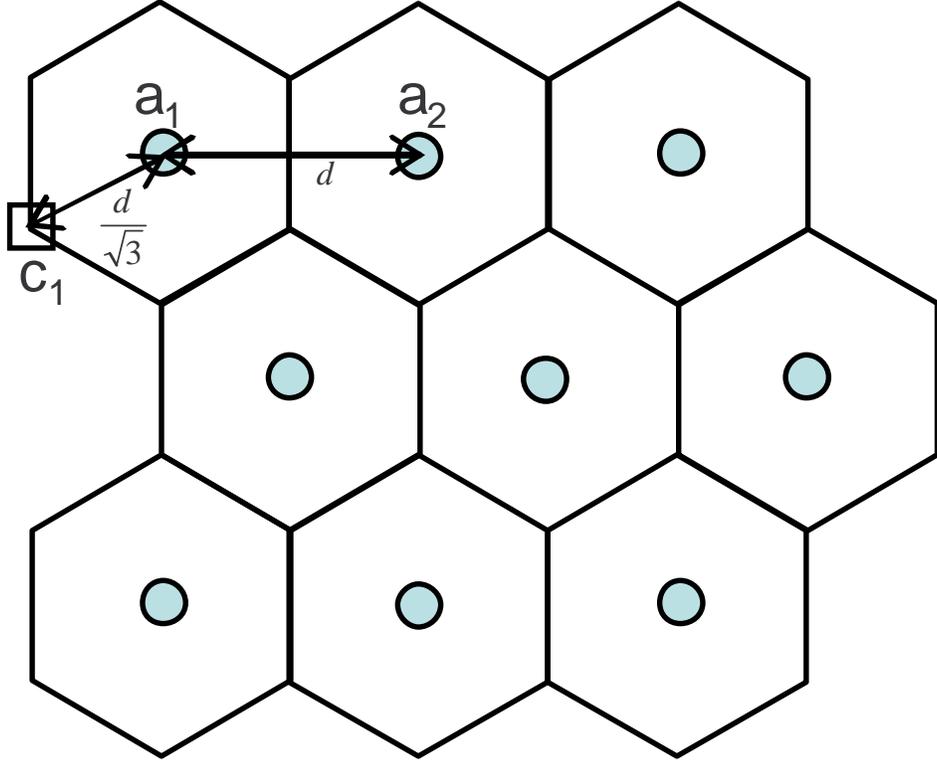


Figure 3.5: The coverage area of each AP is modeled as a hexagon. The separation distance between two adjacent APs is denoted as d ; then the distance from an AP to its farthest user is $\frac{d}{\sqrt{3}}$.

number of iterations to be three or four times the number of APs and determine whether *No-Coord* converges in a given setting of APs, users, and interferers, according to whether the number of actual iterations has reached the limit. In subsequent paragraphs, we will explain in detail the simulation configuration for APs, users, and interferers.

It is known that the coverage area of an AP can be modeled as a hexagon [10]; for brevity, we refer to the coverage of an AP as the *coverage hexagon* of the AP. For example, Fig. 3.5 shows nine APs and their coverage hexagons. Note that there are three rows, each row with three APs; for brevity, we call the AP layout on Fig. 3.5 a *3-by-3* layout. The simulation in this section considers large, medium, or small networks with 100, 49, or 16 APs on a 10-by-10, 7-by-7, or 4-by-4 layout, respectively.

We will explain by using Fig. 3.5 how to determine the distance between adjacent APs. Suppose the separation distance between two adjacent APs a_1 and a_2 is denoted as d . It is obvious that the farthest user (denoted as c_1) from a_1 is located on the vertex of the a_1 's coverage hexagon. From geometry we know that the distance from a_1 to c_1 is

$$d_s = \frac{d}{\sqrt{3}}. \quad (3.16)$$

We want to design d so that the farthest user still has reasonable SINR and throughput. We consider only large-scale path losses without small-scale fading. Let P_{tx} denote the transmit power of a_1 and P_{rx} denote the received power of c_1 . The path loss formula (cf. [10, 47]) is

$$P_{\text{rx}} = P_{\text{tx}} K_0 \left(\frac{d_0}{d_s} \right)^\alpha, \quad (3.17)$$

where α is the path loss exponent, K_0 is a unitless constant which depends on the antenna characteristics and the average channel attenuation, and d_0 is the close-in reference distance which is determined from measurements close to the transmitter. We assume that antenna gains at a_1 and c_1 are both unity; then the value K_0 can be set to the free space path loss at distance d_0 ((cf. [10, 47])):

$$K_0 = \left(\frac{\lambda}{4\pi d_0} \right)^2. \quad (3.18)$$

For microcellular networks, d_0 can be set to 100 meters or 1 meter (cf. [10]); for the simulations in this section, we sets d_0 to 1 m.

We derive d_s (the distance between a_1 and c_1) from (3.17) and (3.18):

$$d_s = \left[\frac{P_{\text{tx}} d_0^{\alpha-2}}{P_{\text{rx}}} \left(\frac{\lambda}{4\pi} \right)^2 \right]^{\frac{1}{\alpha}} \quad (3.19)$$

For the farthest user c_1 to be able to decode desired signal from a_1 , we find from trial and error that most users (especially the users on the edges of hexagons) can have reasonable throughputs from APs when the desired signal at c_1 is three times the noise power at a receiver (denoted as P_n):

$$P_{\text{rx}} \geq 3P_n. \quad (3.20)$$

The thermal noise power (cf. [10]) is modeled as

$$P_n = kT_0B, \quad (3.21)$$

where k is Boltzmann's constant ($k = 1.3806503 \times 10^{-23}$ Joules/Kelvin), T_0 is ambient room temperature (typically taken as 290 K to 300 K), and B is the equivalent bandwidth of the measuring device. We assume $d_0 = 1$ m, $T_0 = 300$ K, $B = 30$ MHz (the bandwidth of IEEE 802.11b/g systems), $\lambda = \frac{1}{8}$ m (the wavelength for the 2.4 GHz carrier frequency of IEEE 802.11b/g), $P_{\text{tx}} = 10$ mW, and $\alpha = 3$ (for indoor environment [10]). In this section, we assume thermal noise is the only constituent of the ambient noise (or noise floor); Section 3.4.2.1 considers realistic noise floor. Now we set $P_{\text{rx}} = 3P_n$ (the equality in (3.20)); then from (3.19) we obtain $d_s = 138.46$ m. Thus from (3.16) the separation distance between two adjacent APs is

$$d = \sqrt{3}d_s = 239.8 \text{ m}. \quad (3.22)$$

We first put APs on a regular 4-by-4, 7-by-7, or 10-by-10 layout, and then randomly move each AP with a small distance (say $0 \sim 5$ meters) to break symmetry, since symmetric AP layouts are uncommon in real world. We say such AP layouts are *uniform*. In order to model deployment error or irregularity, we also consider a *nonuniform* AP layout where each AP is randomly perturbed from the regular 4-by-4, 7-by-7, or 10-by-10 layout by $d_m = d/4$. The separation distance for the nonuniform AP layout needs to be reduced to $d_{\text{nu}} = 0.8d$ in order to guarantee reasonable throughputs on users.

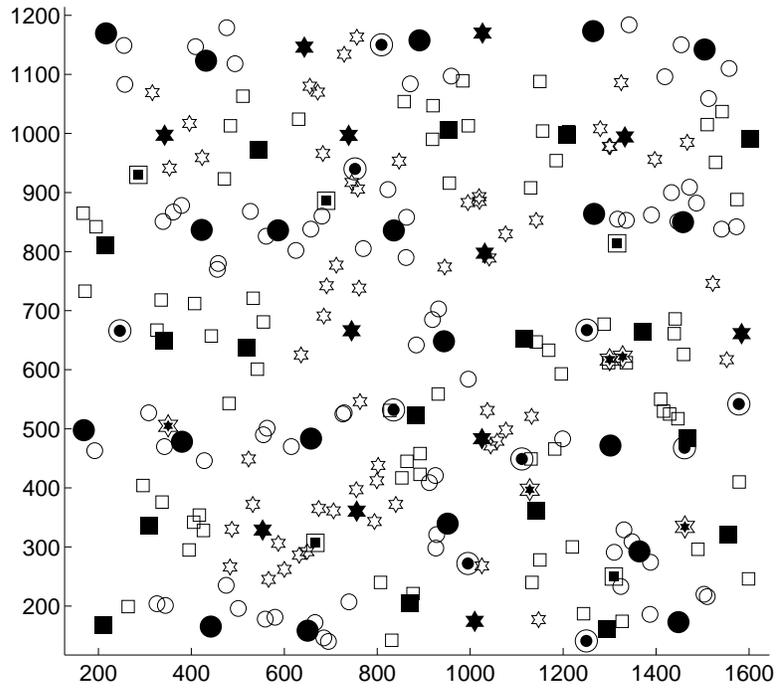
Then we determine the entire area where users may be located. Let A_{min} denote the smallest rectangle area that covers all the APs; A_{min} may cover only 1/4 of the coverage hexagon of the AP on the corner of the AP layout. Let A_{rec} denote a rectangle that is a result of stretching each side of A_{min} by $d/10$ (or $d_{\text{nu}}/10$ for *nonuniform* AP layouts); hence, A_{rec} covers more of the coverage hexagons of corner APs. As we expand A_{min} to become A_{rec} , some region in the expanded rectangle is not covered by any AP's coverage hexagon. In order to limit such uncovered area, we cannot expand A_{min} too much; that is why we choose to expand each side by $d/10$ or $d_{\text{nu}}/10$ for *nonuniform* AP layouts).

Table 3.3: Scenarios considered in the simulations

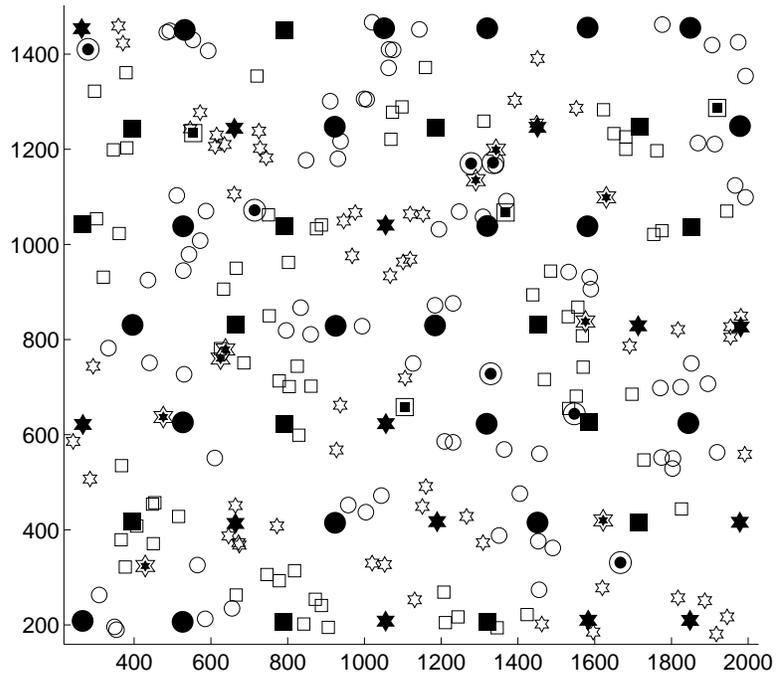
Scenario Number	AP layout	% of rogues	Nonuniform or Uniform
1	4x4	10%	Nonuniform
2	7x7	10%	Nonuniform
3	10x10	10%	Nonuniform
4	4x4	40%	Nonuniform
5	7x7	40%	Nonuniform
6	10x10	40%	Nonuniform
7	4x4	70%	Nonuniform
8	7x7	70%	Nonuniform
9	10x10	70%	Nonuniform
10	4x4	10%	Uniform
11	7x7	10%	Uniform
12	10x10	10%	Uniform
13	4x4	40%	Uniform
14	7x7	40%	Uniform
15	10x10	40%	Uniform
16	4x4	70%	Uniform
17	7x7	70%	Uniform
18	10x10	70%	Uniform

Suppose on average, each AP is associated with four users; hence, a total of 400, 196, or 64 users associate with APs on 10-by-10, 7-by-7, or 4-by-4 layout, respectively. Suppose users are uniformly distributed on A_{rec} . We assume each user is associated with the AP that has the strongest signal strength. We do not employ any load balancing algorithms to balance the load among APs.

We consider the number of rogue RF interferers is low, medium, or high, which denote 10%, 40%, and 70% of the number of controlled APs. For example, for the 10-by-10 AP layout, we consider 10, 40, or 70 rogues, respectively. We assume rogues are randomly placed in the area A_{rec} and do not move. We assume that each rogue interferer stays on a single frequency channel during the period of simulation. For simplicity, we also assume that the transmit



(a) nonuniform



(b) uniform

Figure 3.6: Frequency allocation examples for 49 APs on a 7-by-7 *nonuniform* or *uniform* topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 49 APs; hollow objects denote 196 users; double-layered objects with inner part filled with black denote 20 rogues. The units of X and Y axes are meters.

powers of rogue RF interferers are all 10 mW, which are the same as those of controlled APs. Table 3.3 lists the scenarios considered in the simulations. For each scenario, we randomly generate 10 different cases of the locations of APs, users, and rogues. For each case, we compute users' throughput by using algorithms in Table 3.2; then we sort users' throughput in ascending order. Then we average the 'sorted user throughput' over the 10 different cases for each scenario. Specifically, we mean that we take the lowest throughput from each of the 10 cases and compute the average; then we average the second lowest throughput, and so on. By doing this, we can see a *typical* statistic of users' throughput after averaging over 10 cases. If we do not sort users' throughputs before averaging, the law of large number may smooth (or flatten) the distribution of users' throughput.

Fig. 3.6 depicts examples of *nonuniform* or *uniform* AP topology (7-by-7) with 196 users and 20 rogues. The frequency allocations in Figs. 3.6(a) and 3.6(b) are computed by the *SS-R* algorithm.

We use the empirical model in [11, 12, 40] to relate throughput to SINR:

$$r_l(\gamma_l) = T_{\max} (1 - e^{-A_e(\gamma_l - \gamma_0)}), \quad (3.23)$$

where the three constants T_{\max} , A_e , and γ_0 denote peak throughput, slope of throughput variation, and the cutoff SINR, respectively, as described in [11, 12]. Note that the model in (3.23) captures the downlink throughput of a client c_l when all other clients associated with the same AP are idle, and the received SINR of this client c_l is γ_l . In our simulation, we use a time division multiplexing (TDM) model for medium access. Hence, at any point of time, an AP is sending data to only one client, and the SINR at this client can be computed by considering interference from all other APs on the same channel. Hence, the model in (3.23) is valid, as long as we multiply the throughput in (3.23) by the time fraction that AP allocates to Client c_l .

According to the instruction in [11, 12], we set $T_{\max} = 40$ (Mbps), $A_e = 0.11$, and $\gamma_0 = 0$ (dB). Note that the throughput expression in (3.23) denotes the achievable throughput between c_l and a_m when c_l is the *only* client that is associated with AP a_m . When multiple

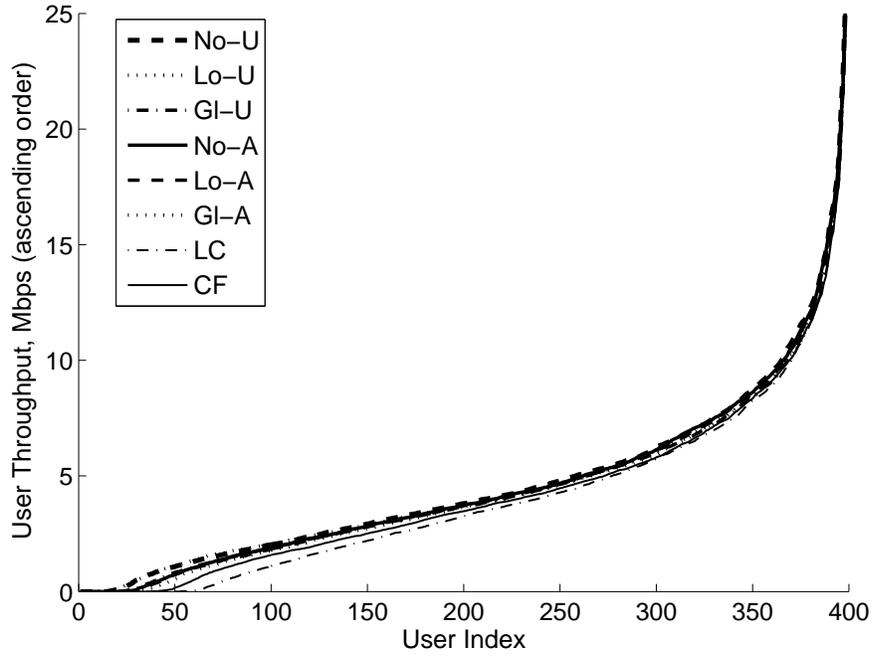
	No-U	Lo-U	Gl-U	No-A	Lo-A	Gl-A	<i>LC</i>	<i>CF</i>
75P	6.27 7.69%	5.92 1.71%	6.06 4.04%	6.15 5.68%	6.14 5.41%	6.10 4.85%	5.75	5.82
50P	3.81 9.34%	3.73 7.21%	3.69 5.96%	3.76 8.00%	3.72 6.83%	3.66 5.14%	3.27	3.48
25P	2.03 27.5%	2.04 27.9%	1.91 19.8%	1.89 18.9%	1.85 16.0%	1.79 12.4%	1.12	1.59
20P	1.69 48.2%	1.77 55.4%	1.51 32.3%	1.54 35.0%	1.47 29.0%	1.40 23.3%	0.607	1.14
15P	1.32 151%	1.37 160%	1.08 105%	1.05 98.6%	1.03 95.1%	0.92 74.3%	0.0445	0.527
10P	0.855 7480%	0.878 7690%	0.491 4250%	0.426 3680%	0.378 3250%	0.231 1950%	0	0.0113
5P	0.119	0.121	0.005	0	0	0	0	0
3P	0	0	0	0	0	0	0	0
mean	4.79 10.1%	4.64 6.78%	4.58 5.35%	4.68 7.58%	4.63 6.46%	4.56 4.99%	4.13	4.35

Table 3.4: Comparison of the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on) and the mean of *users' throughputs* in Mbps using site-specific prediction based algorithms, measurement-based algorithms, *LC* [1], or *CF* [2]. APs are on a 10-by-10 uniform AP layout with 400 users and 10 rogue RF interferers. The percentages indicate the throughput gains over *CF*; the 5 and 3 percentiles are not compared to *CF*, since *CF* yields zero throughputs at these points. The 50 percentile of course corresponds to the median.

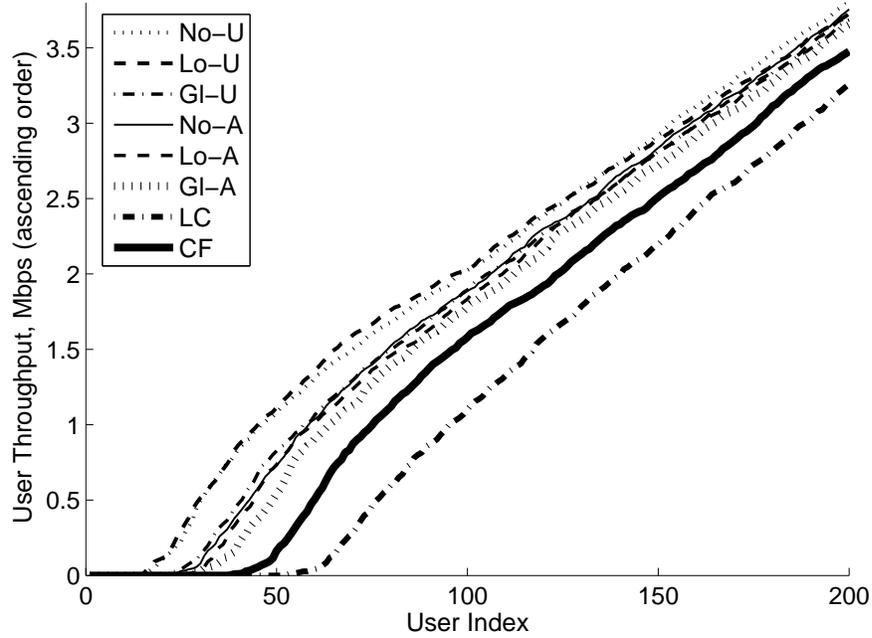
clients are sharing an AP, we assume each client shares an equal amount of time. In other words, if the number of clients associated with AP a_m is L_m , the throughput of client c_l (c_l is one of the clients associated with a_m) is $\frac{r_l(\gamma_l)}{L_m}$.

3.4.2 Results and Discussion for Downlink-Only Scenarios

First, we present the simulation results for a particular scenario with a 10-by-10 nonuniform AP layout, 400 users, and 10 rogues, which is chosen from Table 3.3. Our simulation results show that all the other scenarios in Table 3.3 have similar trends of performance, ex-



(a) All users



(b) 200 users with lower throughputs

Figure 3.7: *User throughput (in Mbps) comparison in a setting with APs on a uniform 10-by-10 layout, 400 users, and 10 rogue RF interferers. Subfigure (b) is part of (a) with only 200 users with the lowest throughputs.*

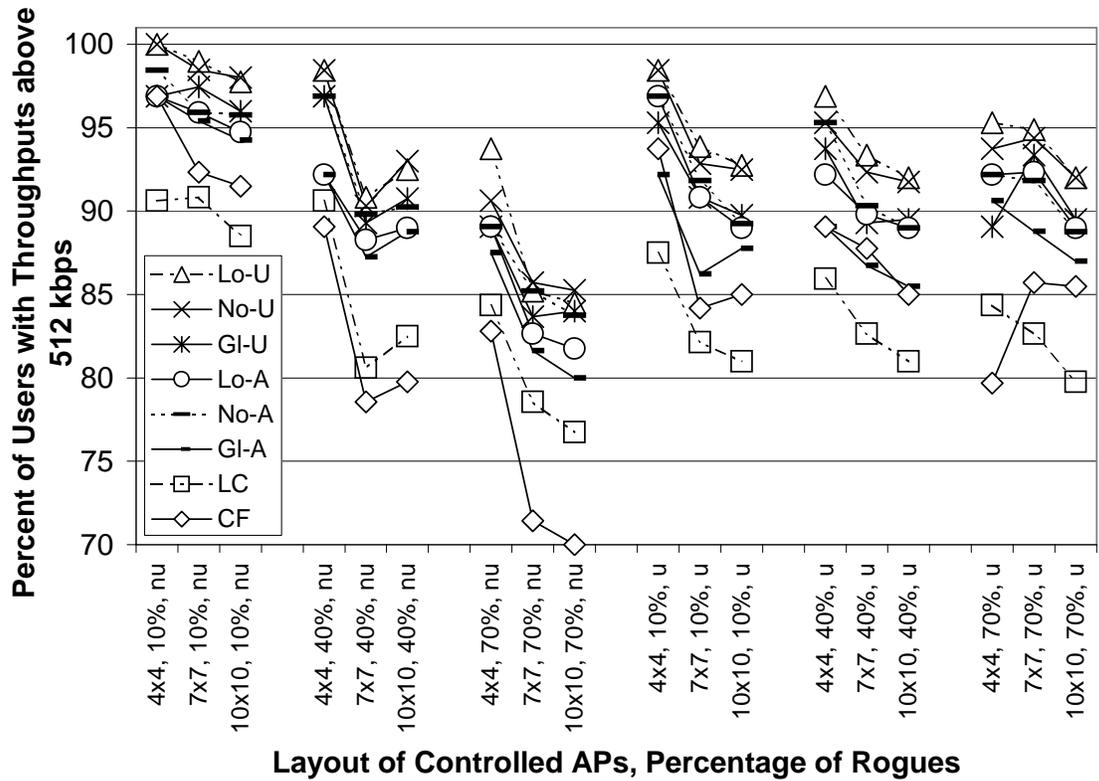


Figure 3.8: Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted ‘nu’ and ‘u’, respectively.

cept that gains over the CF algorithm become larger for scenarios with more rogues. The number of orthogonal channels (K) is set to 3 to represent 802.11b/g; other larger values of K produce very similar trends as to the results and figures shown in this subsection, making our approach applicable to cellular networks and 802.11a. We assume each AP can source up to 54 Mbps per the 802.11g standard. Fig. 3.7(a) shows users' throughputs resulted from different frequency allocation algorithms. Note that the throughputs of 400 users are sorted in ascending order (the x-axis indicates user index, from the first user with the lowest throughput to the four-hundredth user with the highest throughput). Recall that in Section 3.4.1 we mentioned that all the algorithms compute users' throughputs for 10 different cases of locations of APs, users, and rogues. Then, users throughputs are sorted in ascending order for each case and averaged. Hence, each curve on Fig. 3.7(a) is an average of 10 cases. We see difference of throughputs among the curves (each curve denotes the result from an algorithm) on Fig. 3.7(a). We zoom in the left part of Fig. 3.7(a), i.e., the 200 users with lower throughputs, and show the enlarged plot in Fig. 3.7(b) in order that we can better compare the performance between algorithms.

In order to quantitatively compare the performance difference between site-specific knowledge-based algorithms, measurement-based algorithms, LC [1], or CF [2], Table 3.4 compares the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on), and the mean of *users' throughputs* in Mbps. The 50 percentile of course corresponds to the median. Note that the *mean* of users' throughput is much higher than the median because some users that are located close to their associated APs have very large throughputs and thus dominate the mean, as can be seen on the very right side of Fig. 3.7(a). Since the *mean* is biased by such users, we believe that the *median* may better reflect a typical value of user throughput.

If the number of rogues becomes larger (e.g. 40 or 70), the LC algorithm, which takes rogues into account, will perform better than CF , which does not consider rogues, yet our proposed site-specific knowledge-based and measurement-based algorithms perform even better. $Lo-U$ yields the highest 25, 20, 15, 10, and 5 percentiles in the user throughput among

all proposed measurement-based algorithms.

If the goal of system design is to bring up users' throughput to a certain level (e.g. 512 kbps), *Lo-U* may be the best choice. One can draw a horizontal line on 512 kbps on Fig. 3.7(a) or Fig. 3.7(b) and then find out the percentage of users that have throughputs above 512 kbps; Fig. 3.8 shows such percentages for all the scenarios listed in Table 3.3. Among all the proposed measurement-based algorithms, *Lo-U* enables most users to have throughputs above 512 kbps. The second best is *No-U*. When site specific knowledge is not available, the measurement-based *Lo-U* is a very good option. *LC* and *CF* perform poorer than our proposed algorithms.

Although *Global-Coord* uses global coordination among APs, it does not perform very well. It may be because the channel adjustment condition of *Global-Coord* is trying to lower the sum weighted interference on every channel. Two situations cause the sum weighted interference on a channel to be low: first, the number of APs and users on this channel is reduced; second, the interferences experienced by APs and users are reduced. Obviously, the second situation is desirable to enhance the performance, i.e., users' throughput. However, *Global-Coord* fails to distinguish these two situations by using only a single criterion in (3.11). In other words, *Global-Coord* may not minimize the interference that users or APs experience, but rather minimize the number of APs or users on some particular channel. Therefore, *Global-Coord* does not perform as well as *No-Coord*, *Local-Coord*, and site-specific knowledge-based algorithms, although *Global-Coord* still outperforms *LC* and *CF* in most cases.

3.4.2.1 Results with Realistic Noise Floor

In Section 3.4.1, the noise floor consists of thermal noise only. In normal radio frequency environment, however, the noise floor is above the thermal noise due to citizens band radios, radars, lightning, power lines, microwave ovens, electrostatic discharges, and other factors. According to [48], it is reasonable to assume that the noise floor is *10 dB* above the thermal noise in (3.21) (in Section 3.4.1). Because of the increase of the noise floor, the communication distance between an AP and a client decreases. We shorten the separation distance between

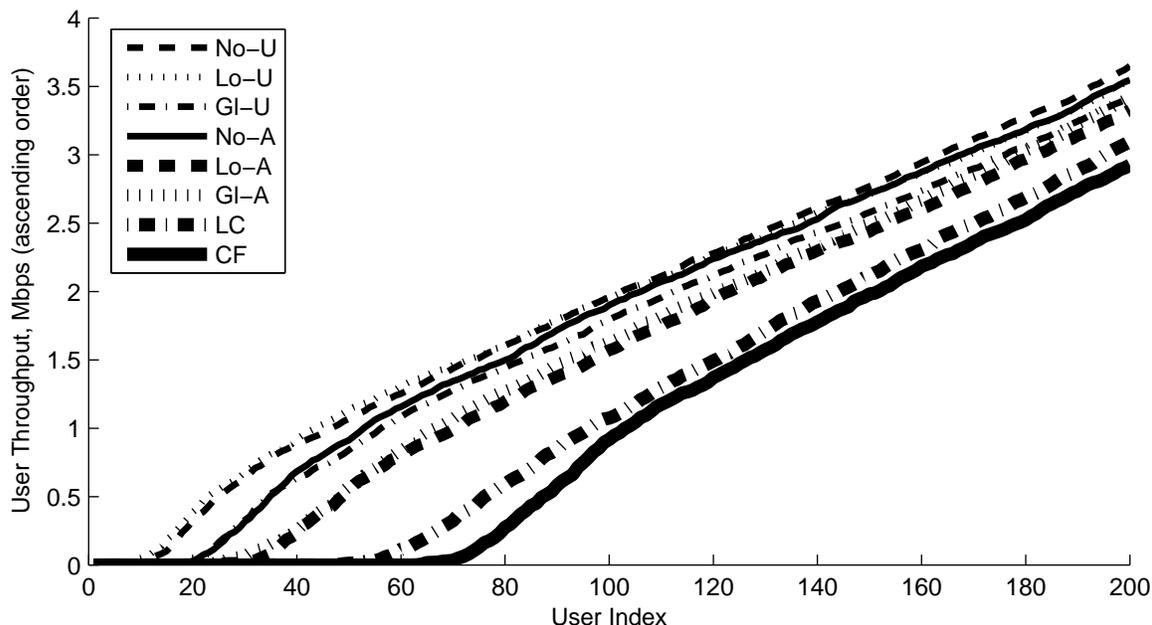


Figure 3.9: *User throughput (in Mbps) comparison in a setting with APs on a uniform 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with lower throughputs are shown. Noise floor is 10dB above the thermal noise.*

APs to be 106 meters (the original separation distance is shown in (3.22) when the noise floor is set as the thermal noise.).

Fig. 3.9 shows that *Lo-U* outperforms the best published work, *LC*, by 13% and 14.3% for mean and median user throughputs respectively, and 81%, 168%, and 1011% for 25, 20, and 15 percentiles of user throughputs, respectively. Fig. 3.9 also shows that *No-U* outperforms *LC* by 15% and 18% for mean and median respectively, and 81%, 167%, and 965% for 25, 20, and 15 percentiles of user throughputs, respectively; our algorithms yield significant throughput gains *especially for users with low throughputs*. Compared with the results with ideal thermal noise assumption, the throughput gains of our proposed algorithms are higher when the noise floor is 10 dB above the thermal noise. This is because the throughput curves used in the simulations, i.e., the expression in (3.23), are *concave*. When realistic noise floor is considered, SINR seen by clients become lower, and the slope corresponding to the SINR on the throughput curves in (3.23) become higher. Since the slope affects the throughput gains, we see a higher

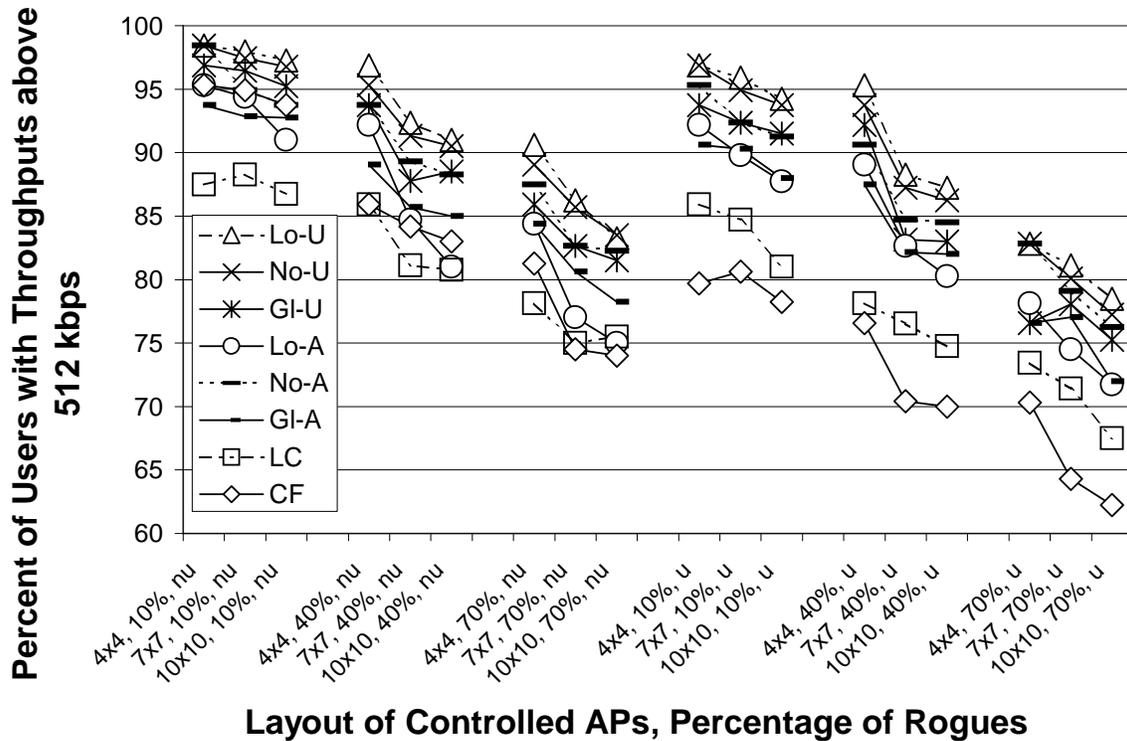


Figure 3.10: Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted ‘nu’ and ‘u’, respectively. Noise floor is 10dB above the thermal noise.

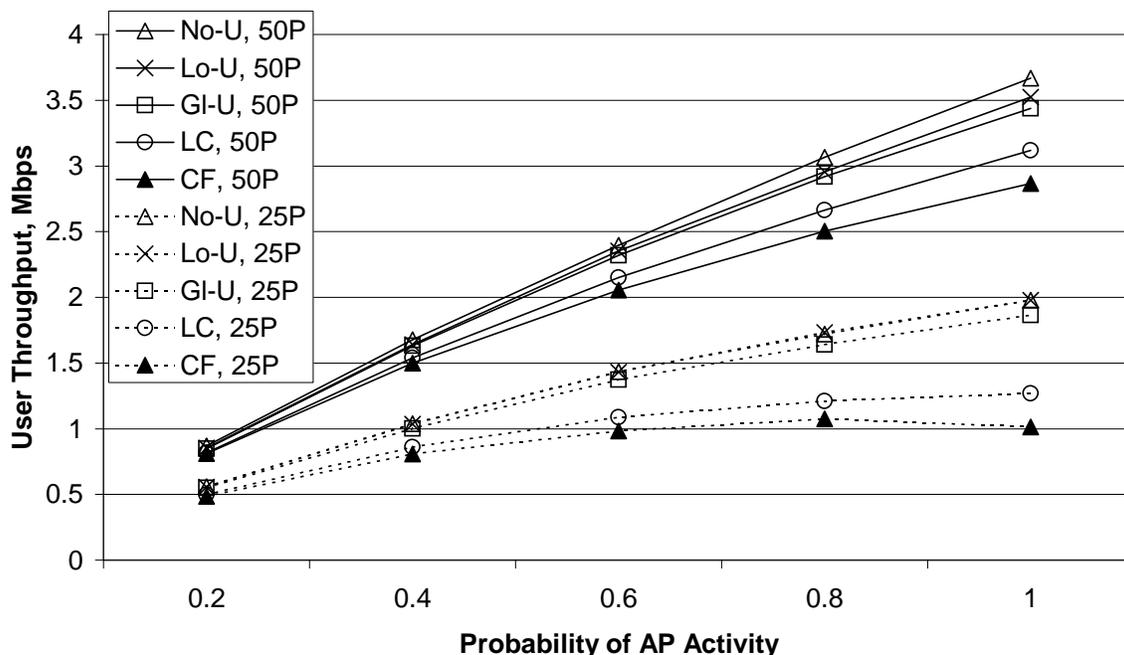


Figure 3.11: 50 and 25 percentiles of users' throughputs (50P and 25P) respectively, including both downlink and uplink traffic, for 400 users on a 10-by-10 uniform AP layout with 10 rogues.

throughput gains with realistic noise floor than merely the thermal noise.

Fig. 3.10 shows the percentage of users with throughputs larger than 512 Mbps when noise floor is set to be 10 dB above the thermal noise. Note that our proposed algorithms outperform other published algorithms (represented by *CF* and *LC*) for all cases. Note that the gains of our proposed algorithms over *LC* or *CF* are *slightly higher* in Fig. 3.10 than in Fig. 3.8 by up to 5.5%; it implies that our proposed algorithms can *work well in real world*, since the noise floor in real world is approximately 10 dB above the thermal noise [48].

3.4.3 Results and Discussion for Downlink-and-Uplink Scenarios

We present the simulation results for a particular scenario with a 10-by-10 uniform AP layout, 400 users, and 10 rogues, which is chosen from Table 3.3. The noise floor is also set to be 10 dB above the thermal noise. Simulation results of all the other scenarios in Table 3.3

have similar trends of performance as in Fig. 3.11. Fig. 3.11 shows the 25 and 50 percentiles of users' throughputs resulted from *Lo-U*, *No-U*, *Gl-U*, *LC*, and *CF*, including both downlink and uplink. As the probability of AP activity increases, we see a stable increase of the throughputs. Therefore, when a network has higher traffic load and the AP is more active, better frequency allocations are crucial for interference mitigation and throughput improvement. *Lo-U* yields the highest throughput. Note Fig. 3.11 includes both downlink and uplink traffic; downlink or uplink alone has similar trends of performance.

3.5 Conclusions of This Chapter

The three proposed algorithms substantially outperform all other published ones irrespective of the numbers of controlled APs and rogue interferers, nonuniform or uniform AP placement, and the level of AP activity. Among the three algorithms, *Local-Coord* is the best in uplifting the throughputs of users that suffer low throughputs. A distributed protocol is also introduced to make *Local-Coord* scalable. Furthermore, the convergence of *Local-Coord* is guaranteed. Therefore *Local-Coord* should be the best algorithm for WLAN frequency allocation. However, if coordination among APs can not be realized as required in *Local-Coord*, *No-Coord* is also a good option, since it does not need coordination among APs. Although *No-Coord* is not guaranteed to converge, simulations show that it converges in most cases and has comparable throughput gain as *Local-Coord*, and practical way to implement is given.

3.6 Appendix

3.6.1 Modeling Measured Interference

The measured interference at an AP or a user is a function of the loads/activities of interfering APs or users. We define $J_{u,v}^k$, $u, v \in \mathbb{X}$ as the interference power that v causes on u given that v is active and is on channel $k \in \mathbb{K}$; $J_{u,v}^k$ depends on v 's transmit power and the path loss between u and v , which is determined by the environment. Let t_u ($u \in \mathbb{X}$) denote the average time fraction (over a unit time) that u is actively transmitting data; again u can be

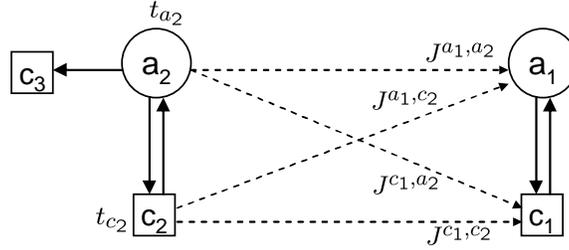


Figure 3.12: The activities of AP a_2 and user c_2 affect the interference on a_1 and c_1 .

an AP or a user. In Figure 3.12, we model the average interference at a_1 as $J_{a_1,a_2}^k t_{a_2} + J_{a_1,c_2}^k t_{c_2}$, and the average interference at c_1 as $J_{c_1,a_2}^k t_{a_2} + J_{c_1,c_2}^k t_{c_2}$. Note that t_{a_2} is the time fraction of the transmissions from a_2 to *all* users associated with it (c_2 and c_3 , in this example). On the other hand, the activities at a_2 and c_2 also affect data transmissions on a_1 and its users. In [39], Bonald et al has modeled such multi-cell wireless data networks as mutually-coupled processor-sharing queues.

3.6.2 Proofs of Lemmas 3.1 and 3.2, and Theorems 3.4 and 3.5

We will use the following lemma to prove Lemmas 3.1 and 3.2, and Theorems 3.4 and 3.5.

Lemma 3.3. *Suppose two vectors $\vec{v} = (v_1, v_2, \dots, v_N)$ and $\vec{v}' = (v'_1, v'_2, \dots, v'_N)$ differ in at least one element. Assume all elements in \vec{v} are distinct, and so are those in \vec{v}' . Let \mathbb{D} denote indices where \vec{v} and \vec{v}' differ, i.e., $\mathbb{D} = \{i : v_i \neq v'_i\}$. Then we have $\vec{v} \succ \vec{v}'$ if $\max_{i \in \mathbb{D}} v_i > \max_{i \in \mathbb{D}} v'_i$.*

Proof. Denote $\vec{u} = (u_1, u_2, \dots, u_N)$ and $\vec{u}' = (u'_1, u'_2, \dots, u'_N)$ as the non-increasing sorted versions of \vec{v} and \vec{v}' , respectively. Suppose v_{m^*} is the largest among the elements in \vec{v} that differ from \vec{v}' , i.e., $m^* = \operatorname{argmax}_{m \in \mathbb{D}} v_m$. Let Ψ denote the set of indices where the elements in \vec{v} are larger than v_{m^*} ; note all these elements come from those that are equal in \vec{v} and \vec{v}' , since the different elements are not larger than v_{m^*} . Hence $\Psi = \{i : v_i = v'_i > v_{m^*}\}$. The first $\Gamma = |\Psi|$ elements in \vec{u} are the those in \vec{v} indexed by Ψ , and $u_{\Gamma+1} = v_{m^*}$. Note the elements in

\vec{v}' that differ from \vec{v} are all less than v_{m^*} according to the assumption stated in the lemma. Hence, the elements in \vec{v}' that are greater than v_{m^*} are those in \vec{v}' indexed by Ψ , which are also the first Γ elements in \vec{u}' . Consequently we have $u'_{\Gamma+1} \leq v_{m^*}$. Note $u'_{\Gamma+1} = v_{m^*} = u_{\Gamma+1}$ cannot hold; otherwise, m^* would not have been in \mathbb{D} . Thus we have $u'_{\Gamma+1} < v_{m^*} = u_{\Gamma+1}$. Finally since the first Γ elements in \vec{u} and \vec{u}' are equal and $u_{\Gamma+1} > u'_{\Gamma+1}$, $\vec{v} \succ \vec{v}'$ holds according to the definition of lexicographic dominance in Definition 3.6. \square

Proof of Lemma 3.1. Recall that we assume a representative AP a_m switches its channel from $f_m = k$ to $f'_m = k'$ according to *Local-Coord* Condition in (3.10), and the channels of all the other APs remain unchanged. Let $\vec{f}, \vec{f}' \in \mathbb{K}^M$ respectively denote a vector of channels selected by APs *before* and *after* the representative AP a_m moves from channel k to k' . Hence \vec{f}' differs from \vec{f} in only the m -th element, that is, $f'_n = f_n, \forall n \neq m$, and $f'_m \neq f_m$. We define

$$\vec{\alpha}(\vec{f}) = \left(W_{f_1}^1(\vec{f}), W_{f_2}^2(\vec{f}), \dots, W_{f_M}^M(\vec{f}) \right) \quad (3.24)$$

$$\vec{\alpha}(\vec{f}') = \left(W_{f'_1}^1(\vec{f}'), W_{f'_2}^2(\vec{f}'), \dots, W_{f'_M}^M(\vec{f}') \right). \quad (3.25)$$

Note that each element (say, the n -th) of the vector $\vec{\alpha}(\vec{f})$ signifies the weighted interference that the AP a_n and its users see on their current channel f_n . Each element in $\vec{\alpha}(\vec{f}')$ in (3.25) denotes the weighted interference as seen by each AP and its users after the representative AP a_m switches to channel k' .

If a_m switches from channel k to k' , cell \mathbb{Z}_m and some surrounding cells that are on either k or k' will experience changes in their weighted interference, i.e., the cells in $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f}')$. All other cells will not experience such changes. Specifically, between $\vec{\alpha}(\vec{f})$ and $\vec{\alpha}(\vec{f}')$, the different elements are the m -th and those indexed by $\mathbb{G}_{m,k}(\vec{f})$ and $\mathbb{G}_{m,k'}(\vec{f}')$. According to Lemma 3.3, it suffices to show that the maximum of these different elements in $\vec{\alpha}(\vec{f})$ is greater than the maximum of those in $\vec{\alpha}(\vec{f}')$. Recall the i -th element of $\vec{\alpha}(\vec{f})$ is $W_{f_i}^i(\vec{f})$ and that of $\vec{\alpha}(\vec{f}')$ is $W_{f'_i}^i(\vec{f}')$, for all $i \in \mathbb{M}$. Therefore, *it suffices to show that*

$$\max_{i \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f}')} W_{f_i}^i(\vec{f}) > \max_{i \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f}')} W_{f'_i}^i(\vec{f}'). \quad (3.26)$$

Equation (3.26) is equal to the *Local-Coord* condition in (3.10). Hence, the proof is done. \square

Proof of Lemma 3.2. Recall that we assume a representative AP a_m switches its channel from $f_m = k$ to $f'_m = k'$ according to *Global-Coord* Condition in (3.11), and the channels of all the other APs remain unchanged. Let $\vec{f}, \vec{f}' \in \mathbb{K}^M$ respectively denote a vector of channels selected by APs *before* and *after* the representative AP a_m moves from channel k to k' . Hence \vec{f}' differs from \vec{f} in only the m -th element, that is, $f'_n = f_n, \forall n \neq m$, and $f'_m \neq f_m$. We define

$$\vec{\beta}(\vec{f}) = \left(\sum_{n:f_n=1} W_1^n(\vec{f}), \sum_{n:f_n=2} W_2^n(\vec{f}), \dots, \sum_{n:f_n=K} W_K^n(\vec{f}) \right) \quad (3.27)$$

$$\vec{\beta}(\vec{f}') = \left(\sum_{n:f'_n=1} W_1^n(\vec{f}'), \sum_{n:f'_n=2} W_2^n(\vec{f}'), \dots, \sum_{n:f'_n=K} W_K^n(\vec{f}') \right). \quad (3.28)$$

The j -th element in $\vec{\beta}(\vec{f})$ denotes the sum weighted interference over all cells on channel j . When a_m switches from channel k to k' , the cells on channels k and k' will see changes of weighted interference. All the other cells do not see such changes. Therefore, between $\vec{\beta}(\vec{f})$ and $\vec{\beta}(\vec{f}')$, only the k -th and the k' -th elements are different. According to Lemma 3.3, it suffices to show that the maximum of the k -th and k' -th elements in $\vec{\beta}(\vec{f})$ is greater than that of $\vec{\beta}(\vec{f}')$, i.e., it suffices to show that

$$\max_{j \in \{k, k'\}} \sum_{n:f_n=j} W_j^n(\vec{f}) > \max_{j \in \{k, k'\}} \sum_{n:f'_n=j} W_j^n(\vec{f}'). \quad (3.29)$$

First, we consider the sum weighted interference on channel k *before* and *after* the representative AP a_m switches to channel k' , i.e., we compare $\sum_{n:f_n=k} W_k^n(\vec{f})$ and $\sum_{n:f'_n=k} W_k^n(\vec{f}')$. When a_m switches to channel k' , the sum weighted interference on channel k loses one positive term that is contributed by a_m and its users, and moreover, some surrounding cells on channel k that were originally interfered by a_m and its users see less interference. Hence we conclude that the sum interference on channel k decreases after a_m switches to channel k' . Formally

we write

$$\sum_{n:f_n=k} W_k^n(\vec{f}) = W_k^m(\vec{f}) + \sum_{n:f_n=k, n \neq m} W_k^n(\vec{f}) \quad (3.30)$$

$$\geq W_k^m(\vec{f}) + \sum_{n:f'_n=k, n \neq m} W_k^n(\vec{f}') \quad (3.31)$$

$$> 0 + \sum_{n:f'_n=k, n \neq m} W_k^n(\vec{f}') \quad (3.32)$$

$$= \sum_{n:f'_n=k} W_k^n(\vec{f}'), \quad (3.33)$$

where (3.30) holds by taking one term $W_k^m(\vec{f})$ out of the summation. When a_m has switched from k to k' , some surrounding cells on channel k may see one less interfering cell (i.e., \mathbb{Z}_m), and all the other farther cells see the same interfering cells (since the cell \mathbb{Z}_m is far from them and does not induce noticeable interference on them). In general, the set of interfering cells for any cell on channel k either remains the same or has one less element, after a_m switches to channel k' , i.e., $\mathbb{Q}_{n,k}(\vec{f}') \subseteq \mathbb{Q}_{n,k}(\vec{f})$, for all $n \in \{n : f_n = k, n \neq m\}$. We apply Proposition 3.1 and obtain that $W_k^n(\vec{f}) \geq W_k^n(\vec{f}'), \forall n \in \{n : f_n = k, n \neq m\}$; therefore, (3.31) holds. Since $W_k^m(\vec{f}) > 0$ by definition, (3.32) holds. The $n \neq m$ part of the subscript is not needed in (3.33) since AP a_m has switched to channel k' and does not operate on channel k , i.e., $f'_m = k' \neq k$.

If $\sum_{n:f'_n=k} W_k^n(\vec{f}')$ is the larger one among the two terms on the right-hand side (RHS) of (3.29), we are done with the proof of (3.29). This is because we know the maximum of the LHS terms is greater than or equal to the first LHS term, $\sum_{n:f_n=k} W_k^n(\vec{f})$, which is then greater than the first RHS term, $\sum_{n:f'_n=k} W_k^n(\vec{f}')$, by (3.33). In general, however, we have to show the LHS of (3.29) is greater than the second RHS term, i.e., it suffices to show that

$$\max_{j \in \{k, k'\}} \sum_{n:f_n=j} W_j^n(\vec{f}) > \sum_{n:f'_n=k'} W_{k'}^n(\vec{f}'). \quad (3.34)$$

Recall that the *Global-Coord* condition that a_m would switch from k to k' is that the sum weighted interference on channel k' after a_m switches to channel k' is lower than the sum weighted interference on channel k when a_m is on channel k , which we rewrite here

$$\sum_{n:f_n=k} W_k^n(\vec{f}) > \sum_{n:f'_n=k'} W_{k'}^n(\vec{f}'). \quad (3.35)$$

Note that the maximum of the LHS two terms in (3.34) is greater than or equal to the first LHS term, $\sum_{n:f_n=k} W_k^n(\vec{f})$, which is also greater than the RHS term in (3.34) according to the *Global-Coord* condition in (3.35). Formally, we write

$$\begin{aligned} \max_{j \in \{k, k'\}} \sum_{n:f_n=j} W_j^n(\vec{f}) &\geq \sum_{n:f_n=k} W_k^n(\vec{f}) \\ &> \sum_{n:f'_n=k'} W_{k'}^n(\vec{f}'). \end{aligned}$$

Hence, we have proven (3.34) and can conclude that $\vec{\beta}(\vec{f}) \succ \vec{\beta}(\vec{f}')$. \square

Proof of Theorem 3.4. Let $\vec{f}, \vec{f}' \in \mathbb{K}^M$ respectively denote a vector of channels selected by APs *before* and *after* the representative AP a_m moves from channel $k = f_m$ to $k' = f'_m$. Hence \vec{f}' differs from \vec{f} in only the m -th element, that is, $f'_n = f_n, \forall n \neq m$, and $f'_m \neq f_m$. It is equivalent to prove that $\vec{\alpha}(\vec{f}') \succeq \vec{\alpha}(\vec{f})$ holds with probability one for every AP $a_m, m \in \mathbb{M}$ and every new channel k' ($k' \neq k$), according to the definition of local lexicographic minimum in Definition 3.10.

Recall from the proof of Lemma 3.1 that $\vec{\alpha}(\vec{f})$ differs from $\vec{\alpha}(\vec{f}')$ only in the elements indexed by $\{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})$. In order to prove that $\vec{\alpha}(\vec{f}') \succeq \vec{\alpha}(\vec{f})$ holds with probability one, *it suffices to show that*

$$\max_{i \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})} W_{f_i}^i(\vec{f}) < \max_{i \in \{m\} \cup \mathbb{G}_{m,k}(\vec{f}) \cup \mathbb{G}_{m,k'}(\vec{f})} W_{f'_i}^i(\vec{f}'). \quad (3.36)$$

holds with probability one, according to Lemma 3.3.

Since *Local-Coord* converges at \vec{f} , no AP can move to a new channel so that the *Local-Coord* condition in (3.10) is satisfied. In other words, for every AP $a_m, m \in \mathbb{M}$ (say, it is currently on channel k) and every new channel k' ($k' \neq k$), the converse of (3.10) holds. The equality in the converse of (3.10) holds with probability zero according to Assumption 3.2. Therefore, the converse of (3.10) without the equal sign, which is the same as (3.36), holds with probability one. Thus, the proof is done. \square

Proof of Theorem 3.5. Let $\vec{f}, \vec{f}' \in \mathbb{K}^M$ respectively denote a vector of channels selected by APs *before* and *after* the representative AP a_m moves from channel k to k' . Hence \vec{f}' differs from \vec{f} in only the m -th element, that is, $f'_n = f_n, \forall n \neq m$, and $f'_m \neq f_m$. It is equivalent to prove that $\vec{\beta}(\vec{f}') \succeq \vec{\beta}(\vec{f})$ holds with probability one for every AP $a_m, m \in \mathbb{M}$ and every new channel k' ($k' \neq k$), according to the definition of local lexicographic minimum in Definition 3.10.

Recall from the proof of Lemma 3.2 that $\vec{\beta}(\vec{f})$ differs from $\vec{\beta}(\vec{f}')$ in only the k -th and the k' -th elements. In order to prove that $\vec{\beta}(\vec{f}') \succeq \vec{\beta}(\vec{f})$ holds with probability one, *it suffices to show that*

$$\max_{j \in \{k, k'\}} \sum_{n: f'_n = j} W_j^n(\vec{f}') > \max_{j \in \{k, k'\}} \sum_{n: f_n = j} W_j^n(\vec{f}), \quad (3.37)$$

holds with probability one, according to Lemma 3.3.

First, we consider the sum weighted interference on channel k' *before* and *after* a_m switches to channel k' , i.e., we compare $\sum_{n: f_n = k'} W_{k'}^n(\vec{f})$ and $\sum_{n: f'_n = k'} W_{k'}^n(\vec{f}')$. When a_m switches to channel k' , the sum weighted interference on channel k' adds one positive term that is contributed by a_m and its users, and moreover, some surrounding cells on channel k' see more interference from a_m and its users. Hence we conclude that *the sum interference on channel k' increases after a_m switches to channel k'* . Formally we write

$$\sum_{n: f'_n = k'} W_{k'}^n(\vec{f}') = W_{k'}^m(\vec{f}') + \sum_{n: f'_n = k', n \neq m} W_{k'}^n(\vec{f}') \quad (3.38)$$

$$\geq W_{k'}^m(\vec{f}') + \sum_{n: f_n = k', n \neq m} W_{k'}^n(\vec{f}) \quad (3.39)$$

$$> 0 + \sum_{n: f_n = k', n \neq m} W_{k'}^n(\vec{f}) \quad (3.40)$$

$$= \sum_{n: f_n = k'} W_{k'}^n(\vec{f}), \quad (3.41)$$

where (3.38) holds by taking one term $W_{k'}^m(\vec{f}')$ out of the summation. When a_m has switched from k to k' , some surrounding cells on channel k' may see one more interfering cell (i.e., \mathbb{Z}_m), and all the other farther cells see the same interfering cells (since the cell \mathbb{Z}_m is far from

them and does not induce noticeable interference on them). In general, the set of interfering cells for any cell on channel k' either remains the same or has one more element after a_m switches to channel k' , i.e., $\mathbb{Q}_{n,k'}(\vec{f}) \subseteq \mathbb{Q}_{n,k'}(\vec{f}')$, for all $n \in \{n : f'_n = k', n \neq m\}$. We apply Proposition 3.1 and obtain that $W_{k'}^n(\vec{f}) \leq W_{k'}^n(\vec{f}'), \forall n \in \{n : f'_n = k', n \neq m\}$; therefore, (3.39) holds. Since $W_{k'}^m(\vec{f}') > 0$ by definition, (3.40) holds. The $n \neq m$ part of the subscript is not needed in (3.41) since AP a_m has switched to channel k' and does not operate on channel k , i.e., $f_m = k \neq k'$.

Since *Global-Coord* converges at \vec{f} , no AP can move to a new channel so that the *Global-Coord* condition in (3.11) is satisfied. In other words, for every AP $a_m, m \in \mathbb{M}$ (say, it is currently on channel k) and every new channel k' ($k' \neq k$), the converse of (3.11) holds. We write the converse of (3.11) below without the equality and claim that it still holds with probability one, since Assumption 3.2 assumes that the equality holds with probability zero.

$$\sum_{n:f'_n=k'} W_{k'}^n(\vec{f}') > \sum_{n:f_n=k} W_k^n(\vec{f}) \quad (3.42)$$

Then we have

$$\max_{j \in \{k, k'\}} \sum_{n:f'_n=j} W_j^n(\vec{f}') \geq \sum_{n:f'_n=k'} W_{k'}^n(\vec{f}') \quad (3.43)$$

$$> \max_{j \in \{k, k'\}} \sum_{n:f_n=j} W_j^n(\vec{f}) \quad (3.44)$$

with probability one, where (3.43) holds because the RHS is one of the two terms of the LHS, (3.44) holds by (3.42) and (3.41). \square

Chapter 4

Site Specific Knowledge for Improving Frequency Allocations in Wireless LAN and Cellular Networks

This chapter¹ exhibits the substantial gains that result from applying *site specific knowledge* to frequency allocation in wireless networks. Two site-specific knowledge-based frequency allocation algorithms are introduced, and are shown to outperform all other published work. We assume a central network controller communicates with all APs, and has site specific knowledge which enables the controller to differentiate the sources of RF interference at every AP or user. By predicting the power from each interference source, the controller optimizes user throughputs. Our algorithms yield significant throughput gains, especially for users with low throughputs, e.g., this work reveals algorithms that outperform the measurement-based algorithms in Chapter 3 by up to 3.75%, 11.8%, 10.2%, 18.2%, 33.3%, and 459% for 50, 25, 20, 15, 10, and 5 percentiles of user throughputs, respectively, and outperform all other published work on frequency allocations even more. Simulations also show consistently high gains irrespective of nonuniform or uniform AP topology, the level of AP activity, and the number of controlled APs, rogue interferers, and available frequency channels.

In this chapter we use site specific knowledge to improve on-going frequency allocation in WLANs. We consider WLANs formed by APs and their clients (e.g. users). When the number of orthogonal frequency channels is limited relative to the number of APs, some nearby APs inevitably use the same channel and induce co-channel interference. Judicious channel reuse mechanisms are necessary to reduce such interference, particularly for the case of mobile users, such as in enterprise voice over IP networks or in cellular networks.

¹Part of the work in this chapter has been submitted to *IEEE Vehicular Technology Conference 2007* [41].

A number of WLAN frequency allocation schemes have been proposed thus far. The work in [21] defines the effective channel utilization of an AP as the fraction of time at which the channel is used for the AP’s data transmission or is sensed busy due to interference from other APs; then, the maximum effective channel utilization among all APs is minimized. AP placement and frequency allocation are jointly optimized in [22] with the same objective of minimizing the max channel utilization as in [21]. The frequency allocation problem is modeled as a minimum-sum-weight vertex-coloring problem in [20] where vertices are APs, and the weight of each edge between two APs denotes the number of clients that are associated with either one of these two APs and are interfered by the other AP. The work in [2] minimizes the number of clients whose transmissions suffer channel conflicts; a client associated with an AP suffers conflicts if other clients or other APs interfere with the client or the AP under consideration. The definition of channel conflict in [2] is more comprehensive than those in [20–22]; the work in [2] has been shown to outperform [20–22].

However, none of [2, 20–22] presents mechanisms to detect and reduce the negative impact from *rogue interferers*, which refer to intentional or unintentional RF interferers in noncooperative networks, microwave ovens, or other RF devices that also operate on the unlicensed bands as WLAN. Only the work in [1] and Chapter 3 (also in [37]) handle rogue interferers. In [1], each AP senses interference and independently selects a channel whose measured interference power is below a predefined threshold, without coordinating with other APs. The work in Chapter 3 (also in [37]) assumes that clients periodically report in situ interference measurements to their associated APs, and presents three iterative algorithms that use the reported measurements to minimize interference seen by clients. In each iteration, these three different algorithms reduce the overall interference (computed by some weighted function defined in Chapter 3, which is also in [37]) seen in a single cell, a group of nearby cells, or all cells, respectively, where a *cell* means an AP and its associated users. The second algorithm with local coordination among nearby APs has been shown to be the best among the three in Chapter 3 (also in [37]).

Most traffic in WLANs is downlink [11,12,25]; hence, maximizing downlink throughput

and signal-to-interference-and-noise ratio (SINR) seen by *users* are key to proper network design. The work in [1, 21, 22] minimizes the interference at *APs* rather than minimizes that at *users*, as is done in [2, 20, 37]. The work in [1, 2, 20] use a binary model for interference, which has been shown to be inferior than physical model, as used in Chapter 3 (also in [37]). Since the measurement-based algorithms in Chapter 3 also have the ability to deal with rogue interferers, they have been shown to outperform [1, 2, 20–22]. Despite their success, the measurement-based algorithms in Chapter 3 (also in [37]) can still be improved if we assume that a central network controller has and uses site specific knowledge to optimize frequency allocation of each AP and each user. The advantage of using site specific knowledge is to predict *a priori* path loss between any pair of AP and user, when the user’s location is obtained via GPS (Global Positioning System) or other known position location technologies², as described in Chapter 1.

Note that the central controller must know the active transmitters at any point in time in order to predict correct interference at all times; this information may be too costly to obtain, but time sampling may be done. Since downlink traffic presently dominates WLAN traffic, this chapter considers a case where all APs are actively transmitting downlink traffic. It is reasonable to assume that frequency allocation is optimized with respect to this most active case, since in this case, frequency allocation is crucial for interference mitigation at users. Simulations show that our algorithms also perform well in scenario with both downlink and uplink traffic and with different levels of AP activities.

We present system models, notation, and assumptions, followed by the details of the two algorithms based on site specific knowledge. Then we show by simulations that our algorithms substantially outperform the others in [1, 2, 20–22, 37].

²Several indoor position location approaches, based on signal strength sensing, are widely known today and used in some WLANs [23, 24]. Other triangulation methods can also be used to locate a client. Modern cellular handsets are equipped with GPS chips or other position location technologies. State-of-the-art GPS can work not only outdoors but also indoors; various vendors, e.g. Metris and SnapTrack, provide indoor GPS solutions.

4.1 Notation and Assumptions

We use the same notation as the previous chapter for consistence. We briefly review the notation used in this chapter. Suppose M APs, indexed by $\mathbb{M} = \{1, 2, \dots, M\}$, operate on K orthogonal frequency channels, indexed by $\mathbb{K} = \{1, 2, \dots, K\}$. We index users (or clients) by $\mathbb{L} = \{1, 2, \dots, L\}$. We denote the identity of an AP and a client by a_m ($m \in \mathbb{M}$) and c_l ($l \in \mathbb{L}$), respectively. We assume for this work the locations of the APs and the clients do not vary with time, and no APs or users are at the same locations, although the algorithms given here also apply for mobile APs and/or clients. We assume every user is associated with a single AP. Let f_m ($f_m \in \mathbb{K}$) denote the channel that a_m operates on, and let $\vec{f} = (f_1, f_2, \dots, f_M)$ denote the channels of all M APs. We define *the set of interfering cells* of a_m and the users associated a_m as $\mathbb{Q}_{m, f_m}(\vec{f})$, so that $n \in \mathbb{Q}_{m, f_m}(\vec{f})$ if and only if a_n operates on channel f_m , and a_n or a user associated with a_n induces non-negligible interference at a_m or a user associated with a_m .

We assume that the central network controller periodically (say every 5 minutes) requires the APs to stop transmitting for a short duration of time (say, one second). In this duration, APs take turns in requiring all users associated with them to perform measurements of background interference, which includes both noise floor of the RF environment and rogue interference from RF devices outside the controlled network. Note that each user needs to measure the background interference for all available frequency channels. The users then feedback to APs these measured background interference. Site specific knowledge along with measurements of background interference make the estimations of SINR at users or APs much more accurate. Let σ_l denote the background interference measured at client c_l .

4.2 Site-Specific Knowledge-Based Algorithms

4.2.1 The Site-Specific SINR (SS-S) Formulation

First, we optimize the sum of utility functions for all the users' SINR, assuming all APs are actively transmitting downlink traffic (but not uplink traffic). That is, we optimize

the following problem over $\vec{f} \in \mathbb{K}^M$, which is denoted *Site Specific SINR* or *SS-S* in the rest of this chapter:

$$\max \sum_{l \in \mathbb{L}} U(\gamma_l) \quad (4.1)$$

$$\gamma_l = \frac{S_{c_l, a_m}}{\sigma_l + \sum_{n \in \mathbb{Q}_{m, f_m}(\vec{f})} S_{c_l, a_n}} \quad (4.2)$$

where a_m in (4.2) denotes the AP with which c_l is associated, σ_l denotes background interference power that client c_l measures (as described in Section 4.1), γ_l denotes the SINR at user c_l as shown in (4.2), S_{c_l, a_m} denotes the average received signal power from a_m to c_l . Note that the objective in (4.1) is not optimizing ‘sum SINR’, since such an objective may favor users that are closer to APs and may cause users which are further away to suffer low SINR. A fair SINR distribution can be achieved if we optimize the sum of utility functions in (4.1), where the utility function $U(\cdot)$ in (4.1) can be any function that is concave, continuously differentiable, and strictly increasing. For example, Mo and Walrand have proposed a class of utility functions that capture different degrees of fairness parameterized by q [49], which takes positive integer values:

$$U(\gamma_l) = \begin{cases} (1 - q)^{-1} \gamma_l^{(1-q)}, & \text{if } q \neq 1 \\ \log \gamma_l, & \text{if } q = 1 \end{cases}, \gamma_l \in (0, \infty). \quad (4.3)$$

This family of utility functions is concave, continuously differentiable, and strictly increasing. Intuitively, as q increases, the degree of fairness increases, but the sum of SINR decreases. The trade-off between sum SINR and the individual SINR of users that are further away from a serving AP can be observed. By increasing the degree of fairness, we imply that users that are further from APs have increasingly higher SINR (which is needed to provide high throughput to distant users). The work in [49] shows that if $q \rightarrow \infty$, the formulation in (4.1) becomes a special case that achieves max-min fairness. At max-min fairness, the degree of fairness is the highest; however, the sum SINR is the lowest. Simulation results in Section 4.3 show that $q = 2$ may be a good parameter for the trade-off between the sum SINR and the SINR for farther users, and is a topic for further research.

As described in Section 4.1, we assume that the APs or the users in the controlled network periodically measure the background interference; hence, σ_l in (4.2) is known. We assume the central network controller has site specific knowledge and locations of all APs and users, and can predict signal power for any pair of AP and user, and can compute S_{c_l, a_n} for all c_l, a_n in the denominator of (4.2). Then all the quantities in the optimization problem in (4.1) are known, yet measurement-based algorithms in Chapter 3 (also in [37]) may need to take a longer time to learn each individual component of S_{c_l, a_n} in (4.2) and thus are not able to solve (4.1) without adequate time of learning (as described in Chapter 1). Because the optimization in (4.1) is a combinatorial problem, there is no fast algorithm (polynomial-time algorithms) that can solve (4.1) [50]. Therefore, we propose an efficient heuristic in Section 4.2.3 that can find the locally optimal solution of (4.1); simulations show that the algorithm in 4.2.3 outperforms the measurement-based algorithms in Chapter 3 (also in [37]) as well as all other frequency allocation algorithms in [1, 2, 20–22].

4.2.2 The Site-Specific Rate (SS-R) Formulation

The formulation in (4.1) in Section 4.2.1 strives to provision fair SINR across users. From the users' perspective, however, *throughput* may be a better metric than SINR for users' performance. Below we formulate another problem that aims at provisioning fair throughput across users, and this formulation may be denoted *Site Specific Rate* or *SS-R*.

$$\max \sum_{l \in \mathbb{L}} U(\chi_l) \tag{4.4}$$

$$\chi_l = \frac{r_l(\gamma_l)}{L_m}, \tag{4.5}$$

where L_m denotes the number of clients that are associated with a_m , χ_l denotes the throughput of c_l from a_m (c_l is associated with a_m), $r_l(\gamma_l)$ denotes the long-term average data rate that c_l receives from a_m if c_l is the only user associated with a_m ; r_l depends on the SINR seen at user c_l , i.e., γ_l , as defined in (4.2). $r_l(\gamma_l)$ may also be viewed as the achievable capacity between c_l and a_m . We assume that the AP a_m evenly distributes its resource (e.g. time) amongst its

L_m users and therefore has the denominator in (4.5). There are several ways to model $r_l(\gamma_l)$; for example, we may use Shannon capacity

$$r_l(\gamma_l) = \log_2(1 + \gamma_l) \quad (4.6)$$

or an empirical model, e.g., such as introduced in [11, 12, 40] to relate throughput to SINR:

$$r_l(\gamma_l) = T_{\max} (1 - e^{-A_e(\gamma_l - \gamma_0)}), \quad (4.7)$$

where the three constants T_{\max} , A_e , and γ_0 denote peak throughput, slope of throughput variation, and the cutoff SINR, respectively, as described in [11, 12]. Section 3.4.1 has justified the usage of this model.

4.2.3 A Local Optimization Algorithm for *SS-S* and *SS-R*

The optimization problems in (4.1) and (4.4) are combinatorial; solving them exhaustively requires exponential computation time (exponential in the number of APs). Hence, we present an iterative local optimization procedure that yields rapid and nearly-optimal solutions of (4.1); the same procedure can also solve (4.4). At the beginning of each iteration, a frequency allocation \vec{f} is given, and at the end of the iteration, we find a better frequency allocation \vec{g} that improves the objective in (4.1); \vec{f} and \vec{g} may differ in several elements, which means that the channels of several APs may change. During each iteration, we do the following steps. First, we select an AP, say a_m . We find $V - 1$ other APs that produce the strongest interference on a_m , assuming these $V - 1$ other APs and a_m are on the same channel; for example, $V = 7$ implies that we find 6 other APs that are in the vicinity of a_m so that they will likely interfere with a_m 's clients the most. We try all possible K^V permutations of channels for these V APs, while fixing the channels at the other $M - V$ APs. We can find the best out of the K^V permutations so that (4.1) is maximized and is strictly larger than the value before this iteration; then we change the corresponding V elements in \vec{f} and thus form \vec{g} . If these V APs have operated on the best channel allocation before this iteration, we have $\vec{f} = \vec{g}$; in this case, another AP (instead of a_m) and its $V - 1$ neighboring APs

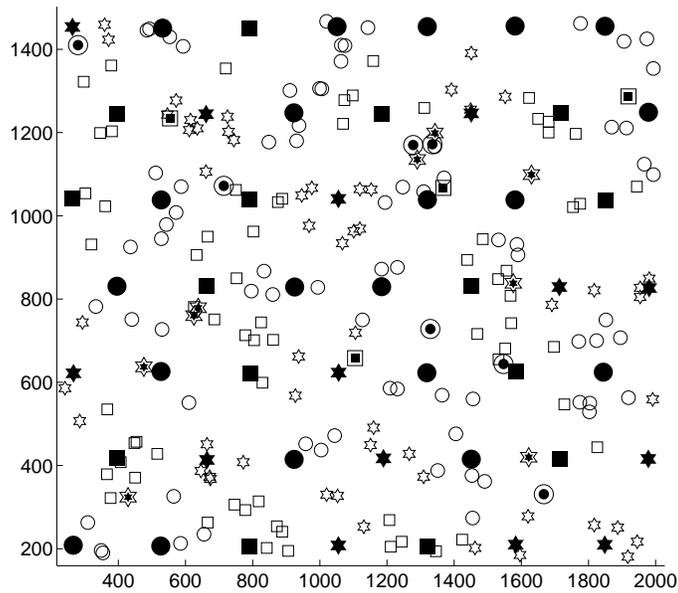
will be selected to restart this iteration. The iterative algorithm runs until every set of V neighboring APs reaches the best frequency allocation. This iterative algorithm converges in a finite number of steps, since the number of channel permutations is finite, and each iteration strictly increases the objective in (4.1). In practice, one may limit the number of iterations or specify a minimum gradient slope due to time constraints. We expect that the channel allocation found by this local optimization algorithm will be close to the optimum if V is large enough, since the exhaustive search can explore more possible allocations with a larger V . Nevertheless, simulations in Section 4.3 shows that this local optimization algorithm with $V = 7$ outperforms all other algorithms [1, 2, 20–22, 37].

The algorithm proposed above solves the *SS-S* formulation in (4.1) and the *SS-R* in (4.4). When it is applied to solve *SS-S*, we refer to the algorithm as the *SS-S* algorithm; similarly, when the algorithm is used to maximize throughput (rate), we refer to it as the *SS-R* algorithm.

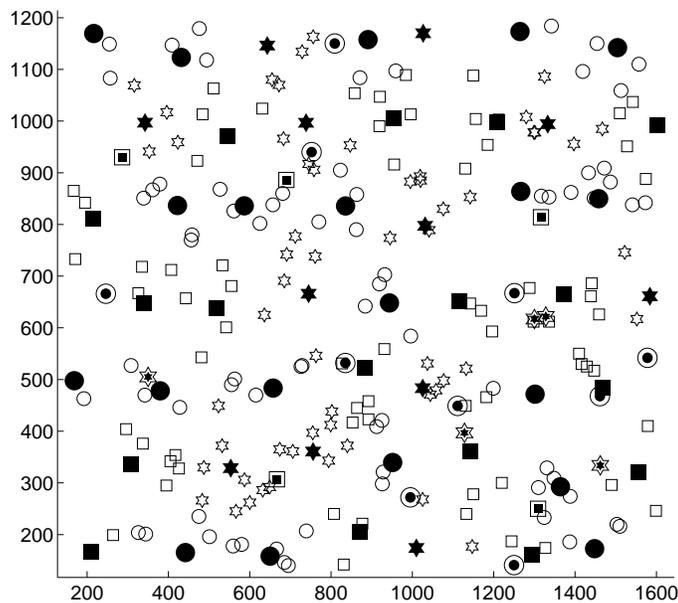
4.3 Simulation Setup and Results

The algorithm in [2], denoted as *CF*, has been shown to outperform [20–22]. Hence, we compare our proposed algorithms against *CF*, the algorithm in [1], and the two better measurement-based algorithms *Lo-U* and *No-U* in Chapter 3 (also in [37]).

The simulation setup in this chapter is the same as that in Section 3.4.1. For convenience, we review some relevant elements of the setup below. First, we consider a saturated network where all APs are transmitting downlink traffic. We set the number of orthogonal channel (K) to 3 to represent 802.11b/g; other larger values of K produce very similar results as shown in Fig. 4.2 and Fig. 4.3, making our approach applicable to cellular networks and 802.11a. We consider three network sizes, three levels of rogue interference, and two network topologies, and thus have eighteen combinations ($3 \times 3 \times 2$), as shown in the x-axis of Fig. 4.3. The three network sizes include a 4-by-4 AP layout with 64 users, a 7-by-7 layout with 196 users, and a 10-by-10 layout with 400 users; the numbers of users are chosen so that every AP



(a) uniform



(b) nonuniform

Figure 4.1: Frequency allocation examples for 49 APs on a 7-by-7 *nonuniform* or *uniform* topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 49 APs; hollow objects denote 196 users; double-layered objects with inner part filled with black denote 20 rogues. The units of X and Y axes are meters.

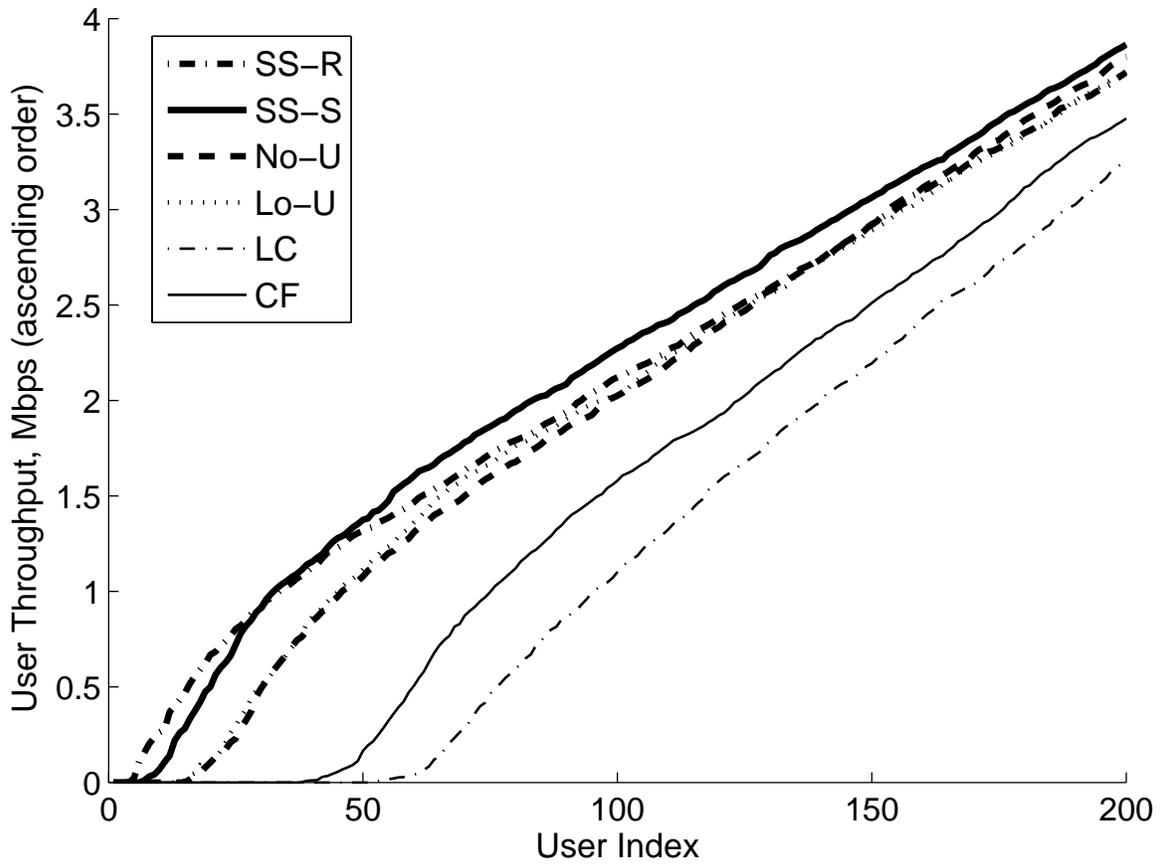


Figure 4.2: *User throughput (in Mbps) comparison in a setting with APs on a uniform 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with lower throughputs are shown.*

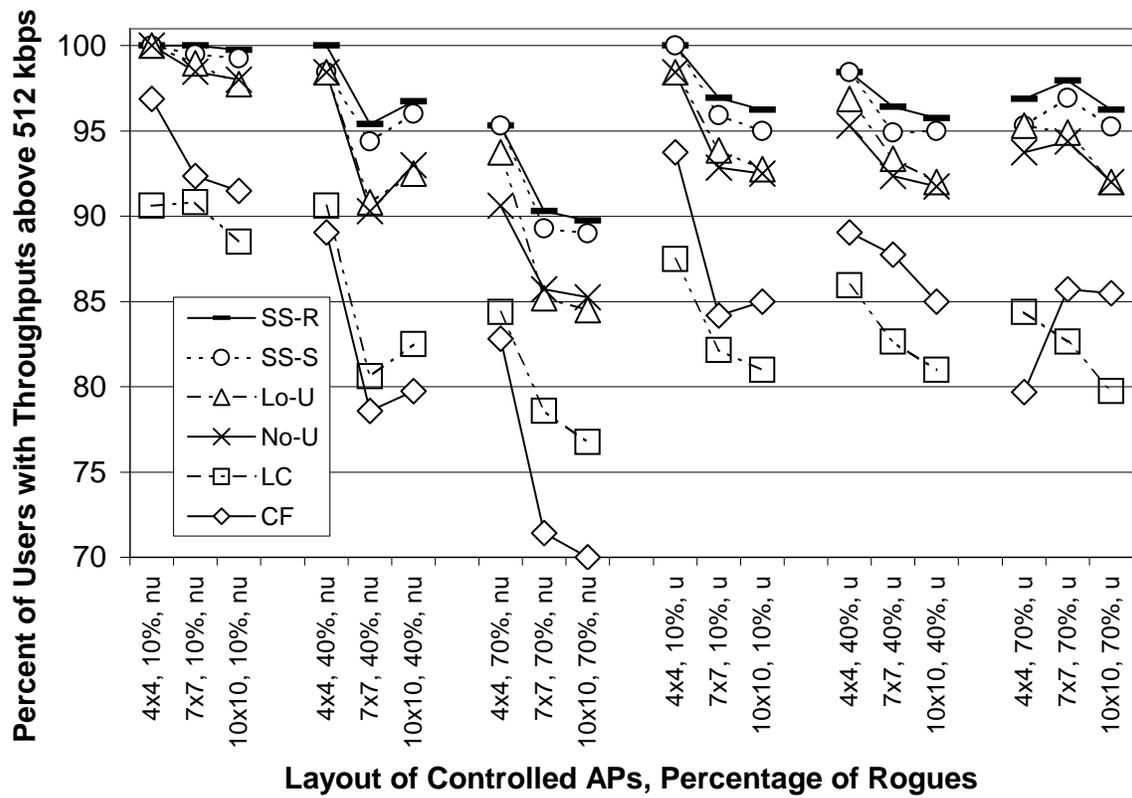


Figure 4.3: Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted ‘nu’ and ‘u’, respectively.

	SS-R	SS-S	No-U	Lo-U	<i>LC</i>	<i>CF</i>
75P	5.98 2.72%	6.18 6.11%	6.27 7.69%	5.92 1.71%	5.75	5.82
50P	3.73 7.00%	3.87 11.1%	3.81 9.34%	3.73 7.21%	3.27	3.48
25P	2.13 33.5%	2.28 43.1%	2.03 27.5%	2.04 27.9%	1.12	1.59
20P	1.80 57.8%	1.95 71.6%	1.69 48.2%	1.77 55.4%	0.607	1.14
15P	1.48 181%	1.62 207%	1.32 151%	1.37 160%	0.0445	0.527
10P	1.13 9950%	1.17 10200%	0.855 7480%	0.878 7690%	0	0.0113
5P	0.676	0.532	0.119	0.121	0	0
3P	0.385	0.180	0	0	0	0
mean	4.71 8.33%	4.84 11.3%	4.79 10.1%	4.64 6.78%	4.13	4.35

Table 4.1: Comparison of the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on) and the mean of **users’ throughputs** in Mbps using site-specific prediction based algorithms, measurement-based algorithms, *LC* [1], or *CF* [2]. APs are on a 10-by-10 uniform AP layout with 400 users and 10 rogue RF interferers. The percentages indicate the throughput gains over *CF*; the 5 and 3 percentiles are not compared to *CF*, since *CF* yields zero throughputs at these points. The 50 percentile of course corresponds to the median.

is associated with four users in average. We consider low, medium, and high interference from rogue interferers, where the ratio of the number of rogue interferers to the number of APs is 10%, 40%, and 70%, respectively. We consider a uniform topology where APs are regularly located as illustrated in Fig. 4.1(a), and a nonuniform topology, where APs are perturbed from the uniform layout with a small random distance (up to 25% of separation), as shown in Fig. 4.1(b). The transmit powers at APs are 10 mW. Noise floor is the thermal noise.

For example, Fig. 4.2 considers 100 controlled APs with 10 rogues, and Table 4.1 compares the 75, 50, 25, 20, 15, 10, 5, and 3 percentiles (denoted as 75P, 50P, and so on) and the mean of *users’ throughputs* in Mbps in the case of 100 controlled APs with 10 rogues.

CF is known to be the best prior to our previous work in Chapter 3 (also in [37]). Fig. 4.2 shows that $SS-R$ and $SS-S$ outperform CF by 8.33% and 11.3% in terms of mean user throughput, 7.00% and 11.1% in terms of median, 33.5% and 43.1% in terms of 25-percentile, and 181% and 207% in terms of 15-percentile user throughputs. $Lo-U$ is known to be the best overall algorithm, especially in uplifting throughputs for users with low throughputs. Nevertheless, Fig. 4.2 shows that $SS-S$ outperforms $Lo-U$ by 3.75%, 11.8%, 10.2%, 18.2%, 33.3%, and 340%, and $SS-R$ outperforms $Lo-U$ by 0.00%, 4.41%, 1.69%, 8.03%, 28.7%, and 459%, in terms of 50, 25, 20, 15, 10, and 5 percentiles of user throughputs, respectively. $SS-R$ yields the highest 5 and 3-percentile throughput. Generally, $SS-R$ sacrifices the users with higher throughput to improve the users with lower throughput. Although $SS-R$ is worse than $SS-S$ for users with high throughput, $SS-R$ is still better than $Lo-U$, the best algorithm in the literature. Our algorithms yield significant throughput gains especially for users with low throughputs. We assume each AP can source a max of 54 Mbps per the 802.11g standard. Fig. 4.3 shows that our algorithms enable more users to operate above 512 kbps irrespective of the number of APs and rogues; this trend is true for other throughput thresholds, as well. Fig. 4.3 shows that $SS-R$ accommodates up to 18% and 7% more users than CF and $Lo-U$, respectively.

In the first part of our simulations, we assume all traffic is *downlink*, and we optimize the frequency allocation for the *most active case* where *all APs are actively transmitting downlink traffic*. It is reasonable that frequency allocation is optimized with respect to this most active case, since in this case, frequency allocation is crucial for interference mitigation at each user. Then, we examine the performance of the optimized frequency allocations in the presence of both downlink and uplink traffic. It has been shown in [39] that uplink and downlink capacities in multiple cells are mutually coupled due to inter-cell interference, and no system-level analytic model has been found to model activities of multiple APs. We consider that time is slotted, and propose an approximate probabilistic model where APs independently choose one of the three possible activity states at each time slot. An AP can be transmitting downlink traffic, receiving uplink traffic, or idle, with probabilities p_d , p_u , and $p_i = 1 - p_d - p_u$,

respectively. For any AP that is transmitting downlink traffic or receiving uplink traffic at a certain time slot, a user is randomly chosen (with uniform probability distribution) out of all the users associated with this AP to be the recipient or the sender of the traffic. We fix the ratio of p_d to p_u as 5:1 [11, 12, 25], and simulate eight cases where $p_d + p_u$ (the probability that an AP is active) is $1/8, 2/8, \dots, 8/8$, respectively. We intend to see the effect of $p_d + p_u$ on the performance of the proposed algorithms. The assumption that the activity of each AP is independent from the other APs simplifies the simulations and provides a rule-of-thumb for the performance comparison.

4.3.1 Simulation with Realistic Noise Floor

Note that the simulation results presented before this section are based on the assumption that the noise floor consists of *thermal noise* only. In this subsection we consider that the noise floor is 10 dB above the thermal noise, which properly represents the real world [48].

Fig. 4.4 shows users' throughputs in a saturate network where all the APs are transmitting downlink traffic; the setup for Fig. 4.4 is similar to that for Fig. 4.2 except the separation between APs is now 106 meters and the noise floor is changed. The noise floor is 10 dB above the thermal noise for Fig. 4.4, whereas the noise floor consists of only thermal noise for Fig. 4.2.

The gains of our proposed algorithms over *LC* are higher in Fig. 4.4 than in Fig. 4.2, which shows that our proposed algorithms can *work well in real world*.

Fig. 4.5 shows the percentage of users with throughputs larger than 512 Mbps when noise floor is set to be 10 dB above the thermal noise. Note that our proposed algorithms outperform other published algorithms (represented by *CF* and *LC*) for all cases. Note that the gains of our proposed algorithms over *LC* or *CF* are *slightly higher* in Fig. 3.10 than in Fig. 3.8 by up to 9.4%; it implies that our proposed algorithms can *work well in real world* with the realistic noise floor.

Fig. 4.6 shows that our algorithms consistently yield throughput gains (including both

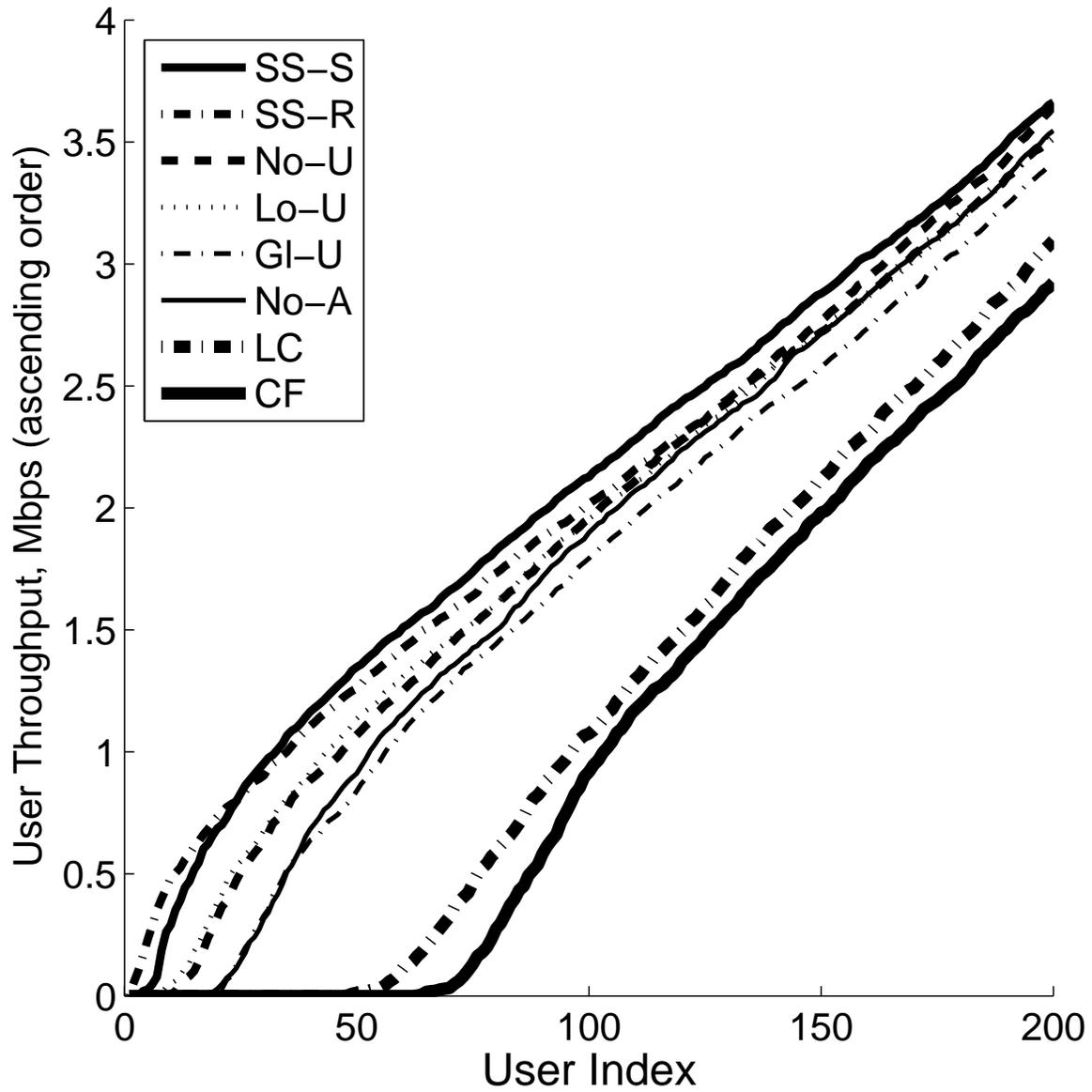


Figure 4.4: *User throughput (in Mbps)* comparison with different levels of noise floor in a setting with APs on a uniform 10-by-10 layout, 400 users, and 10 rogue RF interferers. Only the 200 users with the lowest throughputs are shown.

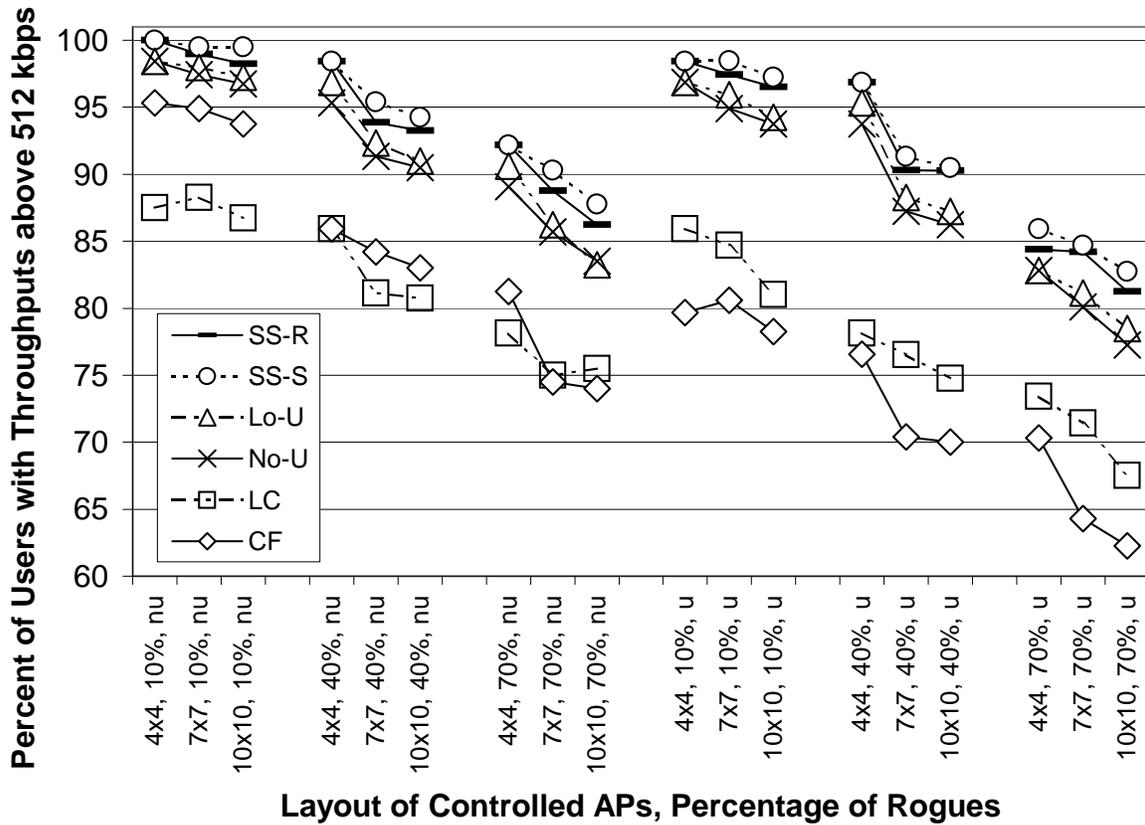


Figure 4.5: Percent of users that have throughputs higher than 512 kbps. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted ‘nu’ and ‘u’, respectively. Noise floor is 10dB above the thermal noise.

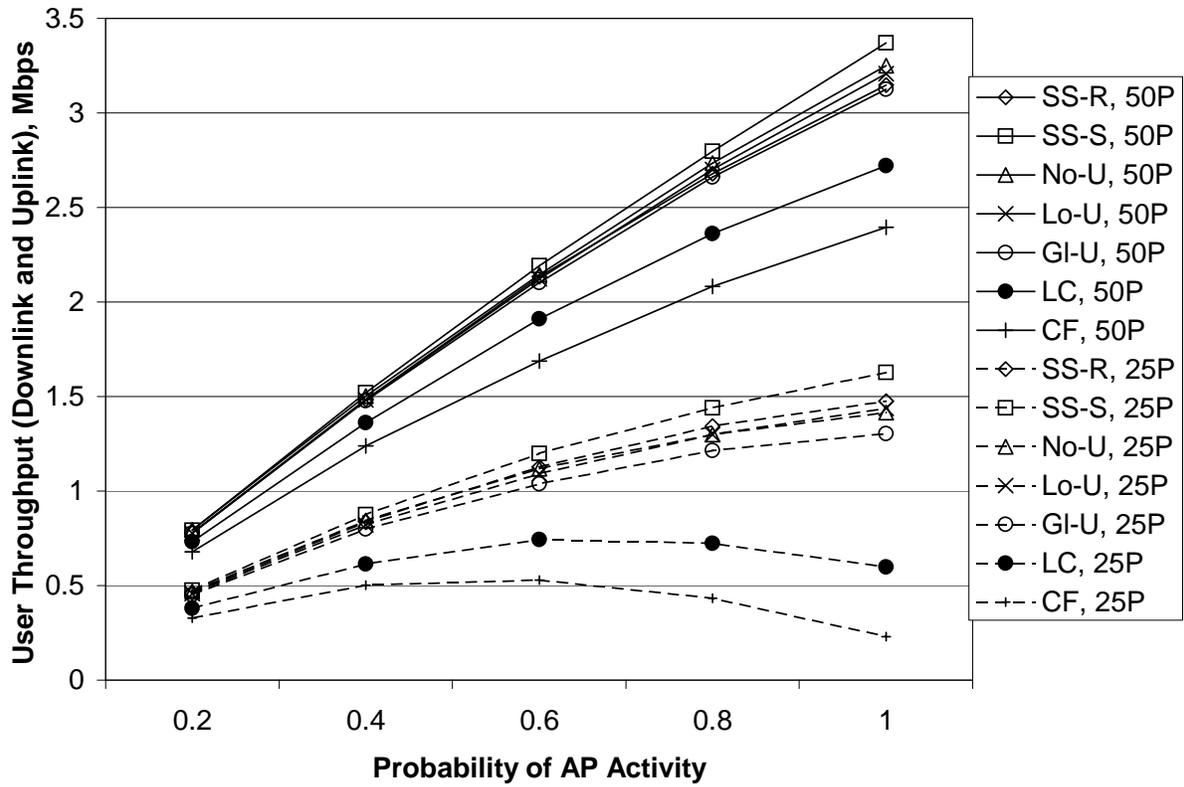


Figure 4.6: 50P and 25P signify 50 and 25 percentiles of users' throughputs respectively, including both downlink and uplink traffic. Noise floor is 10dB above the thermal noise

downlink and uplink) irrespective of the probability of AP activity; particularly the gains are high when APs are highly active, and the network traffic load is heavy. In Fig. 4.6 we still see the same trend as in Fig. 4.4 that *SS-S* and *SS-R* performs well in providing higher throughputs for users who suffer low throughputs.

4.4 Conclusions of This Chapter

A central network controller with site specific knowledge can predict the path loss between any AP and client, and therefore predicts the impact of SINR and throughput on every AP and user when the channel of any AP is changed. This site specific knowledge leads to vast network improvements which we have demonstrated by using two specific-specific algorithms that use fairness parameters. These algorithms substantially outperform all other published ones irrespective of the numbers of controlled APs and rogue interferers, nonuniform or uniform AP placement, and the level of AP activity. Our proposed algorithms are particularly useful when the traffic load of the network is high and APs are highly active. The two algorithms, *SS-S* and *SS-R*, are better in uplifting the throughputs of users that suffer low throughputs when particular utility functions are chosen. Judicious selection of utility function is a topic of future research. We believe that site specific knowledge is also useful for other wireless communication problems in both cellular networks and WLANs, which will be validated by ongoing and future work.

Chapter 5

Power Control with Site Specific Knowledge for Maximizing Throughput of the Network

In Chapters 3 and 4, we optimize frequency allocations to minimize co-channel interference and maximize the throughput of the network, assuming the transmit power of APs and users are fixed. Increasing the transmit power of an AP can potentially increase the downlink throughput from this AP to its clients, but also induces larger interference on nearby APs or clients that are on the same channel, thereby lowering their throughputs. Since most traffic in wireless LANs is downlink, we focus on controlling the transmit powers of APs. In a network with multiple APs and clients, optimizing transmit powers requires knowledge of path gains between APs and clients, which can be estimated by using site specific knowledge. A central network controller that communicates with all controlled APs and has site specific knowledge can optimize transmit powers at APs to maximize and balance the throughputs of all clients in the network.

The *main contribution* of this chapter¹ is a transmit power control scheme that works seamlessly with the best frequency allocation algorithms to date (i.e., the algorithms in Chapter 4) to further improve users' throughputs, e.g., we improve the 25, 10, 5, and 3 percentiles of users' throughputs by up to 4.2%, 9.9%, 38%, and 110%, and save power by 20%. Section 5.1 contrasts our work with prior related work. Section 5.2 introduces notation and assumptions used in this chapter, and Section 5.3 describes the formulation, algorithms, and implementation concerns for our proposed transmit power control problem. Then, Section 5.4 shows simulation results, followed by conclusions for this chapter.

¹Part of the work in this chapter has been submitted to *IEEE Globecom 2007* [51].

5.1 Contrast with Prior Related Work

We have reviewed prior work in [28–32] in Chapter 2.3; we now contrast these with our work on power control.

Recall that [28] assumes that a central network controller knows which APs and clients are actively sending data (downlink or uplink, respectively), and optimizes transmit powers for these active APs and clients. Whenever the set of active APs and clients changes, the central network controller has to know the new set and optimize the power and rates again. Obviously, the overhead of this scheme is considerable. Unlike [28], we will only consider optimizing transmit power for *downlink* transmissions. Recall that in Chapters 3 and 4 we have shown that the frequency allocations optimized for the downlink-only case also perform well in networks with both uplink and downlink traffic, as long as downlink traffic dominates. By doing so, the complexity of the algorithm is reduced.

The work in [29–32] enables clients to meet minimum SINR requirements, but does not mention the characteristics of clients that have high SINRs. In this chapter, we achieve proportional fairness for the SINR distribution for all clients, thereby yielding significant throughput gains, especially for users that suffer low throughputs.

Like the algorithms in Chapters 3 and 4, the work in this chapter has the ability to mitigate the impact of rogue interference, i.e., interference from outside the controlled network. By contrast, [28–32] do not mention mitigating the negative impact of rogue interference.

5.2 Notation and Assumptions

We shall use the same notation as in the previous chapters for consistency, but we briefly review the notation used in this chapter below.

Suppose M APs, indexed by $\mathbb{M} = \{1, 2, \dots, M\}$, operate on K orthogonal frequency channels. We index users (or clients) by $\mathbb{L} = \{1, 2, \dots, L\}$. We denote the identity of an AP and a client by a_m ($m \in \mathbb{M}$) and c_l ($l \in \mathbb{L}$), respectively. We assume no APs or users are at the same locations. We assume every user is associated with a single AP; $m(l)$ (depending on

l) denotes the index of the AP with which c_l is associated, i.e. $a_{m(l)}$ is associated with c_l . Let f_m ($f_m \in \mathbb{K}$) denote the channel that a_m operates on, and let $\vec{f} = (f_1, f_2, \dots, f_M)$ denote the channels of all M APs. Let P_n denote the transmit power of AP a_n .

We assume that the central network controller periodically (say every 5 minutes) requires the APs to stop transmitting for a short duration of time (say, one second). In this duration, APs take turns in requiring all users associated with them to perform measurements of background interference, which includes noise floor of the RF environment and interference from rogue RF devices outside the controlled network. Note that each user needs to measure the background interference for all available frequency channels. The users then feedback these measurements to the APs. Site specific knowledge along with measurements of background interference make the estimations of SINR at users or APs much more accurate. Let σ_l denote the background interference measured at client c_l . The ability to deal with rogue interferers is critical since WLANs share unlicensed bands; however, [28–32] do not address the negative impact of rogue RF interferers.

The RF channel gain between any AP and client can be predicted by using site specific knowledge [4,9,11,12,52]; let $h_{l,n}$ denote the RF channel gain (the inverse of path loss) between the client c_l and the AP a_n , i.e., $h_{l,n}$ is defined as the ratio of the received power at c_l divided by the transmit power of a_n if no other RF interference or noise exists in the environment.

5.3 Transmit Power Control Problem

Recall that Chapter 4 assumes the transmit power of each AP is fixed (denoted P_0) and maximizes the sum of utility of each user’s SINR in a case where all APs are actively transmitting downlink traffic, since downlink traffic volume presently dominates WLAN traffic [11,12,25]. More precisely, Chapter 4 maximizes the following optimization problem² over all

²Note that we slightly modify the expression of (4.1) in Chapter 4 to point out the significance of the *transmit power*.

possible frequency allocations $\vec{f} \in \mathbb{K}^M$.

$$\begin{aligned} & \max_{\vec{f}} \sum_{l \in \mathbb{L}} U(\gamma_l) & (5.1) \\ \text{subject to } & \gamma_l = \frac{h_{l,m(l)}P_0}{\sigma_l + \sum_{n:f_n=f_{m(l)}, n \neq m(l)} h_{l,n}P_0} \\ & \vec{f} \in \mathbb{K}^M \end{aligned}$$

The SINR at client c_l is denoted γ_l in (5.1); the denominator of γ_l in (5.1) consists of background interference (denoted σ_l) and co-channel interference from other APs on the same channel $\sum_{n:f_n=f_{m(l)}, n \neq m(l)} h_{l,n}P_0$. Let \vec{f}^\sharp denote an optimal frequency channel vector for (5.1), which can be found by using the algorithm in Chapter 4, for example.

In this chapter, we fix the frequency channel vector as an optimal one, i.e., \vec{f}^\sharp , and control APs' transmit powers to further improve clients' throughputs. Simulation results in Section 5.4 show the throughput gains achieved by employing transmit power control, as compared with using fixed power. We intend to solve the following problem.

$$\begin{aligned} & \max P_1, P_2, \dots, P_M \sum_{l \in \mathbb{L}} U(\gamma_l) & (5.2) \\ \text{subject to } & \gamma_l = \frac{h_{l,m(l)}P_{m(l)}}{\sigma_l + \sum_{n:f_n^\sharp=f_{m(l)}^\sharp, n \neq m(l)} h_{l,n}P_n} \\ & P_{\min} \leq P_n \leq P_{\max}, \forall n \in \mathbb{M}. \end{aligned}$$

The variables in (5.2) are all M APs' transmit powers $\{P_n : n \in \mathbb{M}\}$. The transmit power ranges between P_{\min} and P_{\max} , which are specific to hardware of APs, e.g., $P_{\min} = 1\text{mW}$ and $P_{\max} = 100\text{mW}$ may be reasonable values for IEEE 802.11a/b/g APs [53]. The objective in (5.2) does not maximize *sum SINR*, since maximizing sum SINR may favor users with high RF channel gains from their associated APs and cause users with low channel gains to suffer very low SINR. The work in [49] can be applied to show that the SINR distribution of clients exhibits q -proportional fairness in SINR, if the utility function in (5.2) has the following form

in (5.2), where q is a *fairness* parameter that captures degrees of fairness.

$$U(\gamma_i) = \begin{cases} (1 - q)^{-1} \gamma_i^{(1-q)}, & \text{if } q \neq 1 \\ \log \gamma_i, & \text{if } q = 1 \end{cases}, \gamma_i \in (0, \infty). \quad (5.3)$$

Generally, as q gets larger, the SINR distribution becomes *fairer*, i.e., the difference of SINR between clients becomes smaller; especially the SINRs of the users that have low channel gains from APs become larger. In the mean time, however, the average SINR becomes lower as q gets larger. Trade-off between fairness and average SINR can be adjusted by changing q . The work in [49] shows that if $q \rightarrow \infty$, the distribution of clients' SINR achieves max-min fairness. Before presenting a key theorem to characterize and solve (5.2), we introduce *geometric programming*.

5.3.1 Background on Geometric Programming

We first present some definitions; then, the description of geometric programming follows. This section is based on the work in [54, 55].

Let x_1, \dots, x_n denote n real positive variables, and $x = (x_1, \dots, x_n)$ a vector with components x_i . A real valued function g of x , with the form

$$g(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \quad (5.4)$$

where $c > 0$ and $a_i \in \mathbf{R}$, is called a *monomial function*, or a *monomial* (of the variables x_1, \dots, x_n).

A sum of one or more monomials, i.e., a function of the form

$$g(x) = \sum_{j=1}^J c_j x_1^{a_{1j}} x_2^{a_{2j}} \cdots x_n^{a_{nj}}, \quad (5.5)$$

where $c_j > 0$ and $a_{ij} \in \mathbf{R}$, is called a *posynomial function*, or a *posynomial* (with J terms, in the variables x_1, \dots, x_n).

According to [54,55], if an optimization problem has the following form, it is a *geometric program*.

$$\begin{aligned}
& \text{minimize } e_0(x) \\
& \text{subject to } e_i(x) \leq 1, \quad i = 1, \dots, n \\
& \quad \quad \quad g_i(x) = 1, \quad i = 1, \dots, p
\end{aligned} \tag{5.6}$$

where $g_i(x)$ are monomials, $e_i(x)$ are posynomials, and x_i are the optimization variables. n and p denote the number of inequality and equality constraints, respectively. There is an implicit constraint that the variables are positive, i.e., $x_j > 0$. The problem in (5.6) is referred to as a *geometric program in standard form*.

5.3.2 Algorithms for Transmit Power Control

Theorem 5.1. *If the fairness parameter ‘ q ’, as introduced in (5.3), is an integer and $q \geq 2$, the optimization problem in (5.2) can be converted to a geometric program in standard form.*

Proof. If $q = 2$, we have $U(\gamma_l) = -1/\gamma_l$. Hence maximizing the objective in (5.2) is equivalent to minimize the following expression:

$$\sum_{l \in \mathbb{L}} \frac{1}{\gamma_l} = \sum_{l \in \mathbb{L}} \left\{ \frac{\sigma_l + \sum_{n: f_n = f_{m(l)}, n \neq m(l)} h_{l,n} P_n}{h_{l,m(l)} P_{m(l)}} \right\} \tag{5.7}$$

$$\begin{aligned}
& = \sum_{l \in \mathbb{L}} \left\{ \frac{\sigma_l}{h_{l,m(l)}} P_{m(l)}^{-1} \right. \\
& \quad \left. + \frac{1}{h_{l,m(l)}} \sum_{n: f_n = f_{m(l)}, n \neq m(l)} h_{l,n} P_n P_{m(l)}^{-1} \right\}, \tag{5.8}
\end{aligned}$$

which is a *posynomial* in P_1, P_2, \dots, P_M . Similarly, when $q = 3, 4, \dots$, maximizing the objective in (5.2) can still be written as minimizing a *posynomial*. Below we present a proof for any integer value of q that satisfies $q \geq 2$. For convenience, let $i = q - 1$. Since $U(\gamma_l) = (-i)^{-1} \gamma_l^{-i}$, maximizing $\sum_{l \in \mathbb{L}} U(\gamma_l)$ is equivalent as minimizing $-\sum_{l \in \mathbb{L}} U(\gamma_l)$, which is equivalent as the

following expressions:

$$-\sum_{l \in \mathbb{L}} U(\gamma_l) = \sum_{l \in \mathbb{L}} \frac{1}{i} \cdot \frac{1}{\gamma_l^i} \quad (5.9)$$

$$= \frac{1}{i} \sum_{l \in \mathbb{L}} \left\{ \frac{\sigma_l + \sum_{n: f_n=f_m(l), n \neq m(l)} h_{l,n} P_n}{h_{l,m(l)} P_{m(l)}} \right\}^i \quad (5.10)$$

$$= \frac{1}{i} \sum_{l \in \mathbb{L}} \frac{P_{m(l)}^{-i}}{h_{l,m(l)}^i} \left\{ \sigma_l + \sum_{n: f_n=f_m(l), n \neq m(l)} h_{l,n} P_n \right\}^i \quad (5.11)$$

$$= \frac{1}{i} \sum_{l \in \mathbb{L}} \frac{P_{m(l)}^{-i}}{h_{l,m(l)}^i} \sum_{j=0}^i \binom{i}{j} \sigma_l^{i-j} \left[\sum_{n: f_n=f_m(l), n \neq m(l)} h_{l,n} P_n \right]^j \quad (5.12)$$

where (5.12) holds by binomial theorem [56], and $\binom{i}{j} = \frac{i!}{j!(i-j)!}$. Note that the term

$$\left[\sum_{n: f_n=f_m(l), n \neq m(l)} h_{l,n} P_n \right]^j$$

in (5.12) is a posynomial (for any $j = 0, 1, \dots, i$) by multinomial theorem [57]. Since summation of posynomials scaled by positive numbers is still a posynomial, the term

$$\sum_{j=0}^i \binom{i}{j} \sigma_l^{i-j} \left[\sum_{n: f_n=f_m(l), n \neq m(l)} h_{l,n} P_n \right]^j$$

in (5.12) is a posynomial. Then it is straightforward to see that the whole term in (5.12) is a posynomial; therefore, the objective in (5.2) is a posynomial.

After examining the objective function, we check the optimization constraints in (5.2), which can be rewritten as

$$P_{\min} P_n^{-1} \leq 1 \quad (5.13)$$

$$(1/P_{\max}) P_n \leq 1, \quad (5.14)$$

which complies with the *standard form* of geometric programs, as described in Section 5.3.1. \square

A geometric program, such as (5.2), can be transformed into a convex program. Efficient algorithms exist to solve geometric and convex programs (see [54, 55]); these algorithms are called *geometric optimization algorithms*. Any such geometric optimization algorithm can be used to solve our problem formulation in (5.2). A central network controller that has site specific knowledge and communicates with all the controlled APs can perform the geometric optimization algorithms to solve the transmit power control problem in (5.2).

5.3.3 Implementation Concerns: Block Processing, Overhead, and Discrete Power Levels

Note that the problem formulation in (5.2) needs the knowledge of *path gains* between APs and clients. Since clients may be moving, joining, or leaving the network, the path gains vary over time. We assume that *block processing* is used for obtaining path gains, i.e., path gains are sampled and updated periodically. When the path gains are updated, optimal transmit powers at APs, i.e., the solution to (5.2) must be recomputed. The period of sampling path gains and recomputing transmit powers is a design choice and could be the same as the period that frequency allocation algorithms are performed, as described in Chapter 4 (say 1, 2, or 5 minutes).

Simulations show that the computation time needed for solving (5.2) is on the order of seconds in the MATLAB programming language. Implementation in low-level languages such as C or Assembly may reduce the computation time to be tens of milliseconds, which are much less than the period of sampling path gains and performing transmit power control. Hence, the *overhead* is negligible.

Note that the transmit power considered in (5.2) takes a continuum of values between P_{\min} and P_{\max} . In practice, however, the transmit power takes *discrete* values. We may quantize the optimal transmit power obtained from solving (5.2). Quantization clearly loses the optimality. Nevertheless, if the separation between discrete power levels is small enough, the quantization loss may be negligible. Therefore, we would like to determine a practical separation of power levels; results in Section 5.4.2 show that a separation of 2.5dB or 4dB is

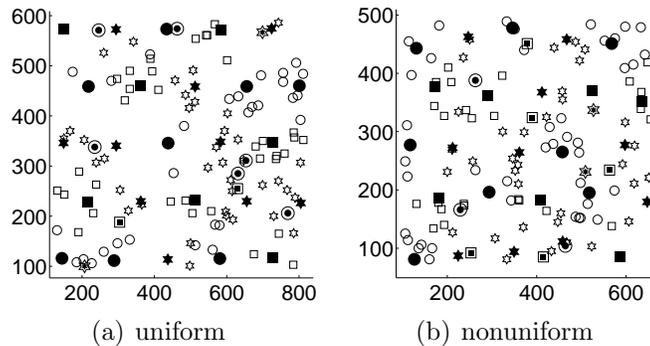


Figure 5.1: Frequency allocation examples for 25 APs on a 5-by-5 *nonuniform* or *uniform* topology. Three kinds of objects (squares, stars, and circles) signify three orthogonal frequency channels. Filled back objects denote 25 APs; hollow objects denote 100 users; double-layered objects with inner part filled with black denote 10 rogues. The units of X and Y axes are meters.

a good option.

5.4 Simulation Setup and Results

Section 5.4.1 describes the simulation setup, and Section 5.4.2 presents and discusses the simulation results.

5.4.1 Simulation Setup

The frequency allocation algorithm in Chapter 4 has been shown to outperform all other published work on WLAN frequency allocations. However, in Chapter 4, all APs use a constant transmit power. The results in Chapter 4 are considered as our *baseline* case, but we set the transmit power of every AP to be the maximum power (100 mW), as opposed to 10 mW, as used in Chapter 4; this adjustment is based on the data sheet in [53], which states that the transmit power of APs ranges between 1 and 100 mW. We compare the baseline case with the optimal transmit power obtained by solving (5.2). Users' throughputs are the metric for comparison. Note that for both the baseline case and our power control case, we use the optimal frequency channel vector obtained by solving (5.1), i.e., the vector $\vec{f}^{\#}$ mentioned in

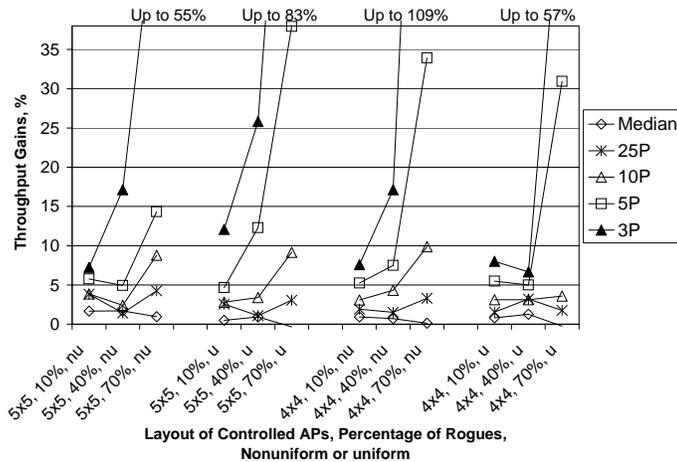


Figure 5.2: Gains of median, and 25, 10, 5, and 3 percentiles (denoted 25P, 10P, 5P and 3P) of users’ throughputs when transmit power control is employed, as compared with using constant transmit power of 100 mW at every AP. The x-axis represents the layout of controlled APs and the percentage of rogue APs compared to the controlled APs. *Nonuniform* and *uniform* AP layouts are denoted ‘nu’ and ‘u’, respectively.

(5.2).

We consider 2 network sizes, 3 levels of rogue interference, and 2 network topologies, and thus have 12 combinations ($2 \times 3 \times 2$), as shown in the x-axis of Fig. 5.2. The 2 network sizes include a 4-by-4 AP layout with 64 users and a 5-by-5 layout with 100 users; the number of users are chosen so that every AP is associated with 4 users on average. We consider low, medium, and high interference from rogue interferers, where the ratio of the number of rogue interferers to the number of APs is 10%, 40%, and 70%, respectively. We consider a uniform topology where APs are regularly located as illustrated in Fig. 5.1(a), and a nonuniform topology, where APs are perturbed from the uniform layout with a small random distance (up to 25% of separation), as shown in Fig. 5.1(b). The separation between adjacent APs is 106 meters, which is the same as the setup in Chapters 3 and 4. Noise floor is set to be 10 dB above the thermal noise to properly represent the RF environment [48]; the thermal noise is modeled as kT_0B , where k is Boltzmann’s constant ($k = 1.3806503 \times 10^{-23}$ Joules/Kelvin), T_0 is ambient room temperature (typically taken as 300 Kelvin), and B is the equivalent bandwidth of the measuring device ($B = 30$ MHz for the bandwidth of IEEE 802.11b/g

Table 5.1: *Power saving* for different network sizes, rogue interference, and network topologies.

	5x5, nu	5x5, u	4x4, nu	4x4, u
10% Rogue	19.2%	19.3%	17.3%	15.5%
40% Rogue	20.7%	19.0%	19.9%	16.6%
70% Rogue	20.4%	17.5%	18.8%	16.5%

systems). We consider saturated networks where all APs are transmitting downlink traffic. For the numerical results shown in this section, the fairness parameter q is set to 2. Higher values of q uplift throughputs of the users that suffer low throughputs, while sacrificing the high-throughput users. Judicious selection of the fairness parameter q depends on application requirement and is a topic of ongoing and future research. We set the number of orthogonal channels (K) to 3 to represent 802.11b/g; other larger values of K produce very similar trends as to those shown in Fig. 5.2, making our approach applicable to cellular networks and 802.11a.

5.4.2 Simulation Results and Discussions

Fig. 5.2 shows the throughput gains of using optimal transmit power, as compared with using constant power, i.e., the baseline case. The results of each of the 12 combinations shown on Fig. 5.2 are averaged from 10 randomly generated networks. Although the work in Chapter 4 has been shown to be able to improve the throughputs of users with poor throughputs, the results in this section show that transmit power control can improve even more. Throughputs of the users that suffer low throughputs are greatly uplifted, i.e., our transmit power control algorithm improves the algorithm in Chapter 4 by up to 4.24%, 9.87%, 37.9%, and 109% for the 25, 10, 5, and 3 percentiles of clients' throughputs. The median and 75, 60, 20, and 15 percentiles of clients' throughputs are improved by up to 1.69%, 1.46%, 1.97%, 5.74%, and 5.29% (these percentiles are not shown on Fig. 5.2 due to lack of space). Our results show that transmit power control built upon frequency allocations allows more users to have satisfactory quality of service.

In addition to throughput gains, our transmit power control also *saves the transmit*

Table 5.2: Percentiles of throughput gains with continuous power levels, or 2, 2.5, 4, 5, or 10 dB of separation between discrete power levels.

	continuum	2dB	2.5dB	4dB	5dB	10dB
25P	3.28%	3.00%	2.92%	3.40%	3.73%	1.45%
20P	5.74%	5.17%	3.78%	5.59%	5.46%	1.43%
10P	9.87%	9.87%	8.91%	9.68%	10.3%	3.86%
5P	33.9%	27.4%	32.7%	31.5%	13.9%	0.985%
3P	109%	103%	109%	65.9%	24.1%	4.52%

power. The intuition is that the inter-cell interference is reduced by lowering the transmit power of some APs; thus, some clients' throughputs are uplifted. Table 5.1 shows that the saving of power expenditure for each of the 12 combinations; the transmit power is reduced by about 20%, i.e., the average transmit power is about 80mW instead of 100mW.

Quantization: For practical implementation, we have to quantize the transmit power level. We study several different values of separation between discrete power levels, namely, 2, 2.5, 4, 5, and 10 dB. For example, if the separation is 4 dB, the actual transmit power levels are 0, 4, 8, 12, 16, and 20 dBm (recall the maximum and minimum transmit power levels are 1 and 100 mW, which are equivalent to 0 and 20 dBm, respectively). We consider the case of 4x4 AP layout, 11 rogue interferers, and nonuniform AP topology, and compute the 25, 20, 10, and 5 percentiles of throughput gains with continuous or discrete power levels, as shown in Table 5.2. Table 5.2 shows a large drop of 3 and 5 percentiles of throughput gains from 2.5dB to 5dB. Therefore, 2.5dB or 4dB is a practically good option for separation between discrete power levels. Other cases of AP layouts, rogue interference, and topology produce very similar trends as to those shown in Table 5.2, making our choice of 2.5dB or 4dB applicable for various network conditions.

5.5 Conclusions of This Chapter

A central network controller with site specific knowledge can predict the path loss between any AP and client, and therefore predicts the impact of SINR and throughput on every AP and user when the transmit power of any AP is changed. This site specific knowledge leads to vast network improvements which we have demonstrated by using a transmit power control algorithm, which can work seamlessly with site-specific based frequency allocation algorithms. Practical discrete power levels are given, i.e., 2.5dB or 4dB separation. Our power control scheme is better at uplifting the throughputs of users that suffer low throughputs when particular utility functions are chosen. We believe that site specific knowledge is also useful for other wireless communication problems in both cellular networks and WLANs, which will be validated by ongoing and future work.

Chapter 6

Load Balancing for Wireless Data Networks

This chapter¹ presents an efficient iterative load-balancing algorithm for time and bandwidth allocation among access points (APs) and users subject to heterogeneous fairness and application requirements. The proposed load-balancing algorithm can work seamlessly with the frequency allocation algorithms in Chapters 3 and 4, and the transmit power control algorithm in Chapter 5, and can be viewed as an add-on to frequency allocations and transmit power control to further improve user throughputs. Frequency allocations are performed during a longer time scale (say 5 minutes) to optimize average throughputs of users, whereas load balancing is performed when any user joins or leaves, whose time scale is often shorter than that for frequency allocations (say 5, 30, or 60 seconds).

The proposed load-balancing algorithm can be carried out either at a central network controller with site-specific propagation predictions, or in a decentralized manner. The algorithm converges to maximum network resource utilization from any starting point, and usually converges in 3 to 9 iterations in various network conditions including users joining, leaving, and moving within a network and various network sizes. Such a fast convergence allows real-time implementations of our algorithm. Simulation results show that our algorithm has merits over other schemes especially when users exhibit clustered patterns. Our algorithm, when assuming multiple radios at each user, achieves 48% gain of median throughput as compared with the max-min fair load-balancing scheme (also with the multi-radio assumption) while losing 14% of fairness index; we also achieve 26% gain of median throughput and 52% gain of fairness index over the Strongest-Signal-First scheme (which assumes each user has only

¹Part of the work in this chapter has been presented in *IEEE Vehicular Technology Conference, Melbourne, Australia, May 2006* [58].

a single radio). When only a single radio is used, our algorithm is similar to the max-min fairness scheme, and is still better than SSF with 44% gain of 25-percentile throughput and 37% gain of fairness index.

6.1 Introduction

Cellular networks and converged cellular/WLAN networks are sure to proliferate as multi-mode devices enter the enterprise and homes. People consider increasing the capacity of WLAN or microcellular networks by increasing AP density and assigning proper non-overlapping frequency channels to APs. As the number of APs to which a user can connect increases, an algorithm that efficiently associates users to APs becomes critical for bandwidth and quality of service (QoS) management. However, the default Strongest-Signal-First (SSF) approach used in 802.11 products, in which each user chooses an AP with the strongest signal, results in unevenly distributed loads among APs and poor performance [26].

In addition, mobile users need to re-associate and re-authenticate with the network as they leave the coverage of one AP and enter into the next. Hence, vendors have introduced centralized controllers or switches wherein mobility, handoffs, QoS, reliability, and security are overlaid on the existing WLAN infrastructure. The role of WLAN or cellular switches is evolving to have network-layer controls over the AP's normal processing in physical layers today. The implementations and limitations of centralized switches vary with vendors. For example, the wireless LAN services module with a single Cisco catalyst 6500 switch can support up to 6000 wireless LAN users and 300 APs [59]. Other vendors such as Trapeze, Aruba, Meru, and Symbol have similar architectures.

Suppose a central switch or network controller has the knowledge of the walls, building layouts, and obstacles in the surrounding environment, as well as the locations of all APs and users. Then, site-specific prediction models can provide the central switch with accurate and detailed RF predictions and real-time optimization from measurements for path loss, throughput, as well as the received SINR and the achievable capacity for each wireless link

[5,12,14–17]. Because the number of disjoint channels is limited, APs reuse the same frequency channel, thus causing co-channel interference. Note that the prediction of capacity and SINR takes into account the co-channel interference. Nevertheless, such interferences are negligible when the number of disjoint frequency bands is large enough (e.g., IEEE 802.11a systems have 12 non-overlapping channels) or CDMA is used for single-cell reuse [10]. Past research [5, 12, 14–17] on site-specific propagation modeling has reported good agreements with measurements of received signal strength intensity (RSSI), and end-user throughput can be estimated to high degree of accuracy [12, 16, 17]. The average and the standard deviation of the RSSI difference between site-specific predictions and measurements are less than 1 and 5 dB, respectively [5]. The correlation coefficient between predicted and measured throughput is 85% [12].

In order to better balance loads, vendors such as Cisco, Trapeze, Aruba, Meru, and Symbol have introduced central switches to have network-layer controls (e.g. load balancing and handoffs) over the AP’s normal processing in physical layers today.

This chapter presents a load-balancing algorithm that can be carried out either in a distributed way with some message exchange between APs and mobile users, or at a central switch with site-specific predictions (such predictions can provide the central switch with detailed RF parameters, the received SINRs, and estimate the achievable capacity for each wireless link; see [11, 14, 15] and references therein). The centralized version with the site specific predictions does not have the overhead of message exchange as in the distributed version of the proposed algorithm.

Several heuristic load-balancing schemes have been presented. As described in Chapter 2, the work in [26, 33–35] outperform schemes with little or no load balancing, but are not shown to be optimal. To our best knowledge, the only work that achieves some form of optimality in load balancing is [36], which achieves max-min fairness of user bandwidth. Our work extends [36] and can achieve different degrees of fairness.

This chapter considers a network with multiple APs and users, as depicted in Fig. 6.1 and tries to answer a fundamental question: *which AP(s) should be connected with a particular user, and how much time should the specific AP(s) allocate to this user in order to*

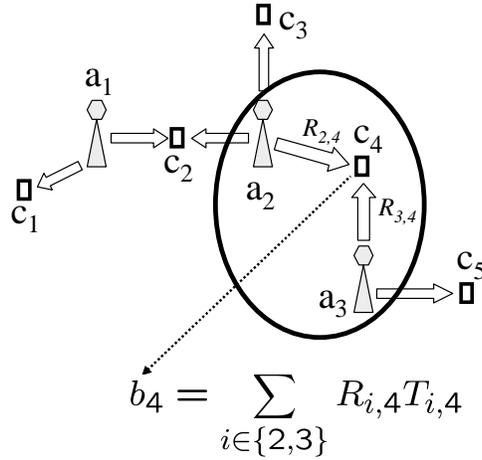


Figure 6.1: A simple network with 3 APs and 5 users. Three APs, denoted by $a_1, a_2,$ and a_3 , use disjoint channels. Five users are denoted by $c_1 - c_5$. $\mathbf{R}_{m,l}$ denotes the long-term transmission rate between a_m and user c_l . $\mathbf{T}_{m,l}$ denotes the time fraction allocated to c_l over the RF channel of a_m . The aggregate rate that c_4 receives from a_2 and a_3 are denoted b_4 .

achieve optimal network utilization subject to heterogeneous fairness and application requirements. Section 6.2 describes the system model and notation in detail. Section 6.3 presents the formulation and an iterative algorithm for the optimal allocation of channel usage time. Simulation results are presented in Section 6.4.

6.2 System Model and Notation

A WLAN card today can only choose one AP. Nevertheless, The max-min load-balancing work in [36] has suggested a model where each user possesses multiple radio interfaces at different frequency bands and thus can communicate with multiple APs simultaneously. We adopt the same assumption in [36]: we assume a multi-radio capability that allows multiple channels to be received and decoded in parallel by each user. It is suggested in this chapter that the multiple-radio assumption simplifies the computation to be efficient (the problem formulation is convex). Our approach can also be used for multi-radio APs. Our algorithm allows up to an unlimited number of radios on a user; however, 2 to 4 radios suffice in practice, since a user in an actual WLAN or microcellular network is usually surrounded

by at most 4 APs.

We assume that users exhibit a *quasi-static* mobility pattern (a model that has been adopted in [36]) where users can move from place to place, but they tend to stay in the same physical places for long periods of time [27]; the median of such a staying period is about 20 minutes according to the WLAN measurements in [27]. This model allows us to consider long-term averaged link capacities over a time scale of about 20 minutes (denoted as T_{AVG}); the proposed load-balancing algorithm is executed based on average link capacities. Due to interference or changes in user applications, user locations, or transmission states, link capacities may change. Our algorithm may use an exponential average of link capacities, which have higher weights on recent or current capacities and lower weights on past capacities. We expect the resource re-allocation will not occur too often due to the quasi-static mobility model. Moreover, this chapter will show that the proposed load-balancing algorithms are very fast and are suitable for real-time computing. The notation in Chapter 3, which is given in Section 3.2, will still be used in this chapter. We will introduce new notation below.

The link capacity $\mathbf{R}_{m,l}$ (e.g. throughput) between an AP a_m , and a user, c_l , is determined by the peak throughput for a single (unshared) user, and also determined from predicted, measured, or optimized throughput estimates based on site specific information. As described in Chapter 4, *SINR* at every user can be predicted using site specific prediction techniques (e.g. those in [4, 9, 11, 12, 52]), given the knowledge of the surrounding propagation environment, building layouts, the locations and electrical properties of physical objects, transmit powers of APs, and the locations and frequency channels of APs and users. The work in [11, 12] presented an empirical model to relate *throughput* to *SINR*; hence, $\mathbf{R}_{m,l}$ can be predicted using site specific knowledge.

For the case where multiple users share a single AP over an RF channel, the throughput between the AP, a_m , and a user, c_l , is a fraction (the time fraction of channel usage) of the link capacity, that is, $\text{Throughput}_{m,l} = \mathbf{T}_{m,l}\mathbf{R}_{m,l}$, where $\mathbf{T}_{m,l}$ is the fraction of channel usage time between a_m and c_l . Let \mathbb{M}_l denote the set of indices of APs from which the user c_l receives positive throughput, i.e. $\mathbb{M}_l = \{m : m \in \mathbb{M}, \mathbf{R}_{m,l} > 0\}$. During a T_{AVG} interval, even though

users may join/leave the network, or RF noise sources may emit interfering signals, the effects of these transient events on link throughputs are quantized and sampled every T_{AVG} (e.g. block processing is used). In the beginning of every T_{AVG} interval, our iterative load balancing algorithm re-adjusts the time/bandwidth resource allocation over all users and APs.

The algorithm converges to optimum in merely 3 to 9 iterations irrespective of network sizes, although the computation time of each *iteration* grows linearly with the number of users multiplied by the number of APs controlled by the switch. On a 2GHz Intel Pentium computer with Windows XP, each iteration in MATLAB takes 30 milliseconds for a network with 36 APs and 300 users. Code implemented in assembly or C language would be much faster and is very suitable for real-time implementations of our algorithms on hardware/firmware, as contemplated in [14, 15].

With the above mentioned assumptions, the real throughput that a user experiences mainly depends on the channel usage time allocated from the APs to this user. For instance, in Fig. 6.1, suppose a_2 and a_3 allocate $\mathbf{T}_{2,4} = 20\%$ and $\mathbf{T}_{3,4} = 40\%$ of their time (over disjoint channels 1 and 2, respectively) to c_4 , respectively. The total bandwidth that c_4 obtains is $b_4 = \frac{20}{100}\mathbf{R}_{2,4} + \frac{40}{100}\mathbf{R}_{3,4}$; the bandwidths of other users can be computed in a similar way. we consider an infinite backlog of packets (full and ready queues on every channel) for every user. Hence a user's throughput is the same as the bandwidth allocated to her. We maximize the sum utility of throughput, which means maximizing $\sum_{l=1}^3 U_l(b_l)$ over the channel usage time in this example. If utility functions are properly chosen, users will be allocated different notions of fair allocation when the network reaches maximum sum utility [49].

We made the assumption that all APs are under the control of a network switch. However, some rogue APs or RF noise sources may emit interfering signals in the coverage area of the controlled APs. In this case, some controlled APs or overlay sensors can detect signals from rogue APs. With detected signal parameters and site specific knowledge, position location techniques can locate the rogue APs [14, 15]. Then, AP channel assignments are changed so that the APs near the rogue APs operate at orthogonal RF channels in order to eliminate most interference from rogue APs. Then, the switch will predict SNR and link

capacities between users and controlled APs using site specific models for the rogue locations and transmit properties, and apply our algorithm to find the optimal resource allocation accordingly. This chapter assumes the frequency band of each AP has been properly assigned [14, 15], and focuses on finding the optimal bandwidth/time allocation in a fully-controlled network.

With an assigned allocated frequency channel, each AP serves its user by time sharing. The fraction of time resource dedicated for payload transmissions between users and an AP, a_m , over an RF channel is denoted as T_m^{frac} ($0 \leq T_m^{\text{frac}} \leq 1$) (e.g., it ranges from 59% to 88% in 802.11a). The subscript m in T_m^{frac} is used, since the payload time fractions may differ from AP to AP. We suppose that each user shares her utility function to all the APs that transmit signals strong enough to reach her. Then, each AP allocates its time resource (over its assigned RF channel) to users based on the information of the utility functions of all the users within its coverage area, based on site specific knowledge [11, 14, 15]. In this chapter, utility functions are assumed to be concave, continuously differentiable, and strictly increasing [54] for simplicity of analysis. User c_l is said to be within the coverage of AP a_m if $\mathbf{R}_{m,l} > 0$; otherwise, $\mathbf{R}_{m,l} = 0$. Each entry in the rate matrix can be predicted from a site-specific prediction engine [11, 14, 15]. Within a unit time period, suppose AP a_m allocates a time fraction $\mathbf{T}_{m,l}$ (over the assigned RF channel of AP a_m) to user c_l ($0 \leq \mathbf{T}_{m,l} \leq 1$). The actual bandwidth that user c_l gets from AP a_m is $\mathbf{T}_{m,l}\mathbf{R}_{m,l}$. Since allocating time between a_m and c_l does not improve any throughput when the capacity between a_m and c_l is zero, we assert that

$$\mathbf{T}_{m,l} = 0 \text{ if } \mathbf{R}_{m,l} = 0, \forall m \in \mathbb{M}, \forall l \in \mathbb{L}. \quad (6.1)$$

6.3 Maximum Sum Utility with Time Allocation

The optimal AP-user association can be formulated as the sum utility maximization problem in (6.2) over time resources from APs on different RF channels to users.

$$\begin{aligned}
& \max \sum_{l \in \mathbb{L}} U_l \left(\sum_{m \in \mathbb{M}_l} \mathbf{T}_{m,l} \mathbf{R}_{m,l} \right) \\
& \text{subject to } \sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} \leq T_m^{\text{frac}}, \forall m \in \mathbb{M}, \\
& \mathbf{T}_{m,l} = 0, \text{ if } \mathbf{R}_{m,l} = 0, \forall m \in \mathbb{M}, \forall l \in \mathbb{L} \\
& \text{over } \mathbf{T}_{m,l} \geq 0, \forall m \in \mathbb{M}, \forall l \in \mathbb{L}
\end{aligned} \tag{6.2}$$

It is hard to find a closed-form expression of the optimal channel usage time allocation for (6.2). Nevertheless, if the optimization is over the time resources of only a single AP (over one channel), assuming the other APs' time allocations are fixed, closed-form expressions for each AP's optimal time allocation have closed-form expressions, shown in (6.13) which are solutions to formulation (6.4). Theorem 6.1 discussed below shows that the original multiple AP problem in (6.2) reaches the optimum if and only if the time allocation from every AP simultaneously has the closed-form expressions as in (6.13). Hence, the optimization of the multiple-AP problem can be done by successively optimizing each AP's time resources, as presented in the algorithm in Fig. 6.2 as an efficient iterative algorithm. Our derivation and proofs extend [60] to a wide class of utility functions (beyond logarithmic) for different degrees of fairness and application needs. The sole constraint in (6.2) means that the total channel usage time used at each AP is upper bounded. The objective is to maximize the network utility $\sum_{l \in \mathbb{L}} U_l(\sum_{m \in \mathbb{M}_l} \mathbf{T}_{m,l} \mathbf{R}_{m,l})$. Mo and Walrand have proposed a class of utility functions that capture different degrees of fairness and model applications with heterogeneous needs parameterized by q_l [49]:

$$U_l(b_l) = \begin{cases} (1 - q_l)^{-1} b_l^{(1-q_l)}, & \text{if } q_l \neq 1 \\ \log b_l, & \text{if } q_l = 1 \end{cases}, b_l \in (0, \infty). \tag{6.3}$$

The parameter q_l has an index l because each user c_l may have a different application/fairness requirement. This family of utility functions is concave, continuously differentiable, and

strictly increasing [49]. The sum of concave functions is still a concave function; hence, problem (6.2) is convex since a concave function is to be maximized over a convex constraint set [54]. The work in [49] shows that if $q_l \rightarrow \infty$, the formulation in (6.2) becomes a special case that achieves max-min fairness, as studied in [36]. Within every T_{AVG} , \mathbf{R} remains constant after block processing, and the optimal sum utility and \mathbf{T} will be determined accordingly.

Suppose the sum utility optimization is performed over the channel usage time resources of only AP a_m , $\mathbf{T}_{m,\bullet} = [\mathbf{T}_{m,1}, \mathbf{T}_{m,2}, \dots, \mathbf{T}_{m,L}]$, assuming that the time allocations from the other APs to users are fixed. Then the formulation in (6.2) is reduced to

$$\begin{aligned} & \max \sum_{l \in \mathbb{L}_m} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l}) \\ \text{subject to } & \sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} \leq T_m^{\text{frac}}, \end{aligned} \quad (6.4)$$

$$\mathbf{T}_{m,l} = 0, \text{ if } \mathbf{R}_{m,l} = 0, \forall l \in \mathbb{L}$$

$$\text{over } \mathbf{T}_{m,l} \geq 0 \quad \forall l \in \mathbb{L},$$

$$\text{where } c_{m,l} = \sum_{n \in \mathbb{M}_l \setminus \{m\}} \mathbf{T}_{n,l} \mathbf{R}_{n,l} \text{ are fixed.}$$

The objective should have been $\sum_{l \in \mathbb{L}} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l})$, but we note that

$$\sum_{l \in \mathbb{L}} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l}) = \sum_{l \in \mathbb{L}_m} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l}) + \sum_{l \in \mathbb{L} \setminus \mathbb{L}_m} U_l\left(\sum_{n \in \mathbb{M}_l} \mathbf{T}_{n,l} \mathbf{R}_{n,l}\right). \quad (6.5)$$

If $l \in \mathbb{L} \setminus \mathbb{L}_m$, then $m \notin \mathbb{M}_l$. Hence, $\mathbf{T}_{n,l}$ is fixed for all $n \in \mathbb{M}_l$, which also implies that the second term in (6.5) is fixed. Therefore, maximizing $\sum_{l \in \mathbb{L}} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l})$ is equivalent to maximize the first term in (6.5), i.e., maximizing $\sum_{l \in \mathbb{L}_m} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l})$.

Denote by λ_m the Lagrange multiplier for the constraint in (6.4). Then, the Lagrangian [54] is given by

$$L(\mathbf{T}_{m,\bullet}, \lambda_m) = \sum_{l \in \mathbb{L}_m} U_l(\mathbf{T}_{m,l} \mathbf{R}_{m,l} + c_{m,l}) - \lambda_m \left(\sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} - T_m^{\text{frac}} \right). \quad (6.6)$$

Since utility functions $U_l(\cdot)$ are increasing, it is natural to exhaust the time resource for maximizing sum utility [54]; therefore, at the maximum of (6.4), we have $\sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} = T_m^{\text{frac}}$.

Then, the sufficient and necessary optimality conditions (KKT conditions) [54] for (6.4) can be written as:

$$\mathbf{R}_{m,l}U_l'(\mathbf{T}_{m,l}\mathbf{R}_{m,l} + c_{m,l}) = \lambda_m \quad \text{if } \mathbf{T}_{m,l} > 0, \quad \forall l \in \mathbb{L} \quad (6.7)$$

$$< \lambda_m \quad \text{if } \mathbf{T}_{m,l} = 0, \quad \forall l \in \mathbb{L} \quad (6.8)$$

$$\sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} = T_m^{\text{frac}} \quad (6.9)$$

$$\mathbf{T}_{m,l} \geq 0, \quad \forall l \in \mathbb{L}; \quad \lambda_m > 0. \quad (6.10)$$

It is obvious that no time is allocated to links with zero capacity (i.e. $\mathbf{T}_{m,l} = 0$ if $\mathbf{R}_{m,l} = 0$). Therefore, we focus on deriving the optimal $\mathbf{T}_{m,l}$ for $\mathbf{R}_{m,l} > 0$. For general utility functions, the optimal time fraction can be derived from (6.7):

$$\mathbf{T}_{m,l} = \left\{ \frac{1}{\mathbf{R}_{m,l}} U_l'^{-1} \left(\frac{\lambda_m}{\mathbf{R}_{m,l}} \right) - \frac{c_{m,l}}{\mathbf{R}_{m,l}} \right\}^+. \quad (6.11)$$

While closed-form solutions of $\mathbf{T}_{m,l}$ do not exist for general utility functions, they can be obtained for the family of utility functions in (6.3), for which (6.7) becomes

$$\frac{\partial L}{\partial \mathbf{T}_{m,l}} = \frac{\mathbf{R}_{m,l}}{(\mathbf{T}_{m,l}\mathbf{R}_{m,l} + c_{m,l})^{q_l}} - \lambda_m \quad (6.12)$$

Equating (6.12) with zero gives the optimal time allocation (note that for completeness we include the case where $\mathbf{T}_{m,l} = 0$):

$$\mathbf{T}_{m,l} = \begin{cases} \left\{ \lambda_m^{(-\frac{1}{q_l})} \mathbf{R}_{m,l}^{(\frac{1}{q_l}-1)} - \frac{c_{m,l}}{\mathbf{R}_{m,l}} \right\}^+, & \text{if } \mathbf{R}_{m,l} \neq 0 \\ 0, & \text{if } \mathbf{R}_{m,l} = 0 \end{cases} \quad (6.13)$$

In (6.11) and (6.13), the notation $\{x\}^+$ is needed because $\mathbf{T}_{m,l}$ is nonnegative: $\{x\}^+ = x$ if $x \geq 0$ and $\{x\}^+ = 0$ otherwise. By substituting (6.13) or (6.11) into $\sum_{l \in \mathbb{L}_m} \mathbf{T}_{m,l} = T_m^{\text{frac}}$ in (6.9), λ_m for each AP a_m can be numerically solved [54, 60]. In each iteration of our algorithm, finding the time resources of each AP requires solving a single-variable (λ_m) polynomial equation with L terms; hence, the time complexity of each iteration is $O(ML)$. If the parameter $q_l = 1$, the expression of $\mathbf{T}_{m,l}$ in (6.13) is the water-filling expression, where the constant λ_m^{-1} is known as the water-filling level [60].

Theorem 6.1. $\{\mathbf{T}_{m,l} : m \in \mathbb{M}, l \in \mathbb{L}\}$ is an optimal solution to (6.2) if and only if $\{\mathbf{T}_{m,1}, \mathbf{T}_{m,2}, \dots, \mathbf{T}_{m,L}\}$ is the solution in (6.13) for AP a_m with the time allocation from the other APs $\{\mathbf{T}_{n,l} : \forall n \neq m, \forall l\}$ fixed, for all $m = 1, 2, \dots, M$.

Proof. This proof is inspired by Theorem 1 in [60]. First, the “only if” part is proven by contradiction. Suppose at the optimum of (6.2), $\{\mathbf{T}_{m,l} : m \in \mathbb{M}, l \in \mathbb{L}\}$ is an optimal time allocation, but there exists an AP a_m such that $\{\mathbf{T}_{m,1}, \mathbf{T}_{m,2}, \dots, \mathbf{T}_{m,L}\}$ does not satisfy the single-AP water-filling condition in (6.13). Fix all other time fractions $\{\mathbf{T}_{n,l} : n \in \mathbb{M}, n \neq m, l \in \mathbb{L}\}$ and let $\{\mathbf{T}_{m,1}^W, \mathbf{T}_{m,2}^W, \dots, \mathbf{T}_{m,L}^W\}$ be the single-AP water-filling time vector computed by (6.13), where the constants $\{c_{m,l} : l \in \mathbb{L}\}$ that are needed to compute the water-filling expression are obtained from the other fixed time fractions $\{\mathbf{T}_{n,l} : \forall n \neq m, \forall l\}$. Since the time fractions $\{\mathbf{T}_{n,l} : \forall n \neq m, \forall l\}$ are fixed, the multi-AP optimization problem in (6.2) is reduced to a single-AP problem in (6.4). Thus, changing $\{\mathbf{T}_{m,1}, \mathbf{T}_{m,2}, \dots, \mathbf{T}_{m,L}\}$ to $\{\mathbf{T}_{m,1}^W, \mathbf{T}_{m,2}^W, \dots, \mathbf{T}_{m,L}^W\}$ increases the sum utility objective of the single-AP problem, as well as the multiple-AP problem, thus contradicting the optimality of $\{\mathbf{T}_{m,l} : m \in \mathbb{M}, l \in \mathbb{L}\}$.

The proof of the “if” part is given here. Since the formulation in (6.2) is convex, the KKT conditions for (6.2) are necessary and sufficient for optimality. Hence, it suffices to prove that if $\{\mathbf{T}_{m,1}, \dots, \mathbf{T}_{m,L}\}$ is a single-AP water-filling time allocation according to (6.13) for all m , the KKT conditions for (6.2) hold.

If $\{c_{m,l} : m \in \mathbb{M}, l \in \mathbb{L}\}$ are properly defined according to (6.4), the KKT conditions for (6.2) are the same as those from (6.7) to (6.10) except for the fact that the former conditions are for all APs but the latter ones are for a single AP. Hence, the single-AP water-filling time allocations in (6.13) satisfy the KKT conditions for the multiple-AP optimization problem. This concludes the proof of the “if” part. \square

As described in Theorem 6.1, the time allocations from each AP to users can be solved by (6.13), assuming time allocations from the other APs are fixed. Hence, the optimal time allocation for the multiple-AP optimization problem (6.2) can be found by an iterative algorithm (see Fig. 6.2).

- 1: **Given** a rate matrix $\{\mathbf{R}_{m,l}, \forall m, l\}$.
- 2: Start with a valid time allocation $\{\mathbf{T}_{m,l}, \forall m, l\}$.
- 3: **repeat**
- 4: **for** each AP $m = 1, 2, \dots, M$ **do**
- 5: Compute $\{c_{m,l}, \forall l\}$ by (6.4).
- 6: Compute $\{\mathbf{T}_{m,l}, \forall l\}$ by (6.13) or (6.11).
- 7: **end for**
- 8: **until** the sum utility converges
- 9: **Output** $\{\mathbf{T}_{m,l}, \forall m, l\}$.

Figure 6.2: An iterative algorithm to solve (6.2)

Theorem 6.2. *The algorithm in Fig. 6.2 results in an optimal sum utility and causes $\{\mathbf{T}_{m,l}, \forall m, l\}$ to converge to an optimal time allocation for Formulation (6.2).*

Proof. This proof is similar to that of Theorem 2 in [60]. At each water-filling step in the algorithm in Fig. 6.2, the optimal time allocations from one AP to users are found while regarding the time allocations from other APs as fixed. The sum utility objectives of the multi-AP problem and the single-AP problem are the same except that in the single-AP optimization, only the time fractions from one AP can be changed; therefore, the multi-AP sum utility objective is non-decreasing within each water-filling step. The sum utility objective is bounded because every time fraction is between 0 and 1. Hence, the sum utility converges to a limit.

The time fractions from each AP, $\{\mathbf{T}_{1,\bullet}, \mathbf{T}_{2,\bullet}, \dots, \mathbf{T}_{M,\bullet}\}$, also converge. For the single-AP optimization problem, the water-filling solution is unique; hence, at a water-filling step for AP a_m , the time allocation $\mathbf{T}_{m,\bullet}$ either strictly increases the sum utility or remains the same. At the convergence limit, all $\mathbf{T}_{m,\bullet}$'s are simultaneously single-AP water-filling expressions. According to Theorem 6.1, such a time allocation is the optimal one for the multiple-AP optimization problem. Note that the proof holds for any initial time allocation. \square

The algorithm in Fig. 6.2 can be carried out in a decentralized manner: each AP a_m computes the optimal time allocation $\{\mathbf{T}_{m,l} : l \in \mathbb{L}\}$ only for those users who are in the

coverage of this AP. For the computation of each user's $\mathbf{T}_{m,l}$, a constant $c_{m,l}$ needs to be known, which in turn requires the knowledge of the bandwidth that this user c_l receives from APs other than AP a_m . In a realistic WLAN setup, a user is under the coverage of no more than 4 APs; hence, the computation of $c_{m,l}$ at each user is efficient. APs sequentially perform such decentralized computing. When the sum utility converges, a control message may be sent to APs to stop the decentralized computing.

6.4 Simulation Results

In this section, we compare the throughput and fairness performance of our maximum utility (denoted as *MaxUtil*) scheme with the max-min fairness scheme in [36], denoted as *MaxMin*, and the Strongest-Signal-First scheme in current 802.11 implementations. We consider a simplified scenario of free-space propagation model where no obstacles exist in the vicinity of APs. It is clear that our algorithm can utilize site specific information, which will be considered in future work. We consider different percentages (between 1% and 5%) of users joining, leaving, or moving within the network; hence, the link capacities change over time. We sample \mathbf{R} for every T_{AVG} , and within this time interval, \mathbf{R} is fixed. Two kinds of user distributions, namely *uniform* and *cluster* (or hotspot), are considered. First, users are uniformly distributed in a 600 meters by 600 meters square that encompasses the 36 APs. Second, we consider that a hotspot at the center attracts more people: users are distributed in a circle-shaped area centered at the middle of the APs with a radius of 250 meters. Users are randomly located on this circle based on their uniformly generated polar coordinates (the distance from the center and the polar angle are uniformly distributed between $(0, 250)$ and $(0, 2\pi)$, respectively). From the viewpoint of the Cartesian coordinate, the user density is higher near the center than near the circumference of the circle. Each point on the figures is an average over 100 independent runs. In the SSF case, each user (whose transceiver can handle only a single channel) associates with the strongest AP, and then each AP evenly distributes its time resources to the associated users. Simulations show that the number of iterations (mostly between 3 and 9) does not grow with the number of users. Our algorithm

converges quickly even for large networks.

Figs. 6.3 and 6.4 show the medians and the 25-percentiles of user throughputs, respectively. Table 6.1 presents fairness indices (see [3] for this metric) for cases with 400 users; scenarios with different number of APs and users are omitted, since their fairness index values are similar to those in Table 6.1. Both *MaxUtil* and *MaxMin* assume that each user has multiple radios. For fair comparisons with SSF, we also compute single-radio results by properly rounding multi-AP time allocation; *MaxMin-R* denotes the results produced by the rounding method in [36]. The *MaxUtil-R* results were obtained by a different rounding method: we first compute normal multiple-radio time allocation; then, if any user indeed uses multiple APs, this user simply chooses the AP that supplies her with the most bandwidth. Finally, if any AP has any time resource remained not allocated, this AP allocates the remaining time proportionally to its associated users. For example, if the rate matrix $\mathbf{R} = \begin{bmatrix} 7 & 5 & 6 & 3 \\ 4 & 1 & 4 & 4 \end{bmatrix}$ and all users' utility parameters, q , are 1, then the optimal time fraction (allowing multi radios) is $\mathbf{T} = \begin{bmatrix} 0.417 & 0.417 & 0.166 & 0 \\ 0 & 0 & 0.375 & 0.625 \end{bmatrix}$. Each user chooses only one single AP; then the time matrix becomes $\mathbf{T} = \begin{bmatrix} 0.417 & 0.417 & 0 & 0 \\ 0 & 0 & 0.375 & 0.625 \end{bmatrix}$. Then, since the first AP has time fraction (16.6%) remained, the remaining time is proportionally distributed to users 1 and 2; finally the time matrix for the single-radio case is $\mathbf{T} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.375 & 0.625 \end{bmatrix}$.

A trade-off between throughput and fairness can be seen in multi-radio cases *MaxUtil* and *MaxMin*. Our *MaxUtil* has very good performance in *cluster* case: in Fig. 6.3(b), *MaxUtil* exhibits about 48% higher median throughput over *MaxMin* while sacrificing only 14% of fairness as in Table 6.1. It is because *MaxMin* tends to achieve absolute fairness (its fairness index is almost 100% as in Table 6.1) by sacrificing throughput (giving more time resource to users with poor link capacities). Our *MaxUtil* trades throughput with fairness; even in *uniform* case in Fig. 6.3(a), *MaxUtil* yields 9% higher median throughput than *MaxMin* while losing 2% of fairness as in Table 6.1. Our algorithm, with multiple radios at each user, outperforms SSF by 26% and 52% in terms of median throughput and fairness index, respectively, as in

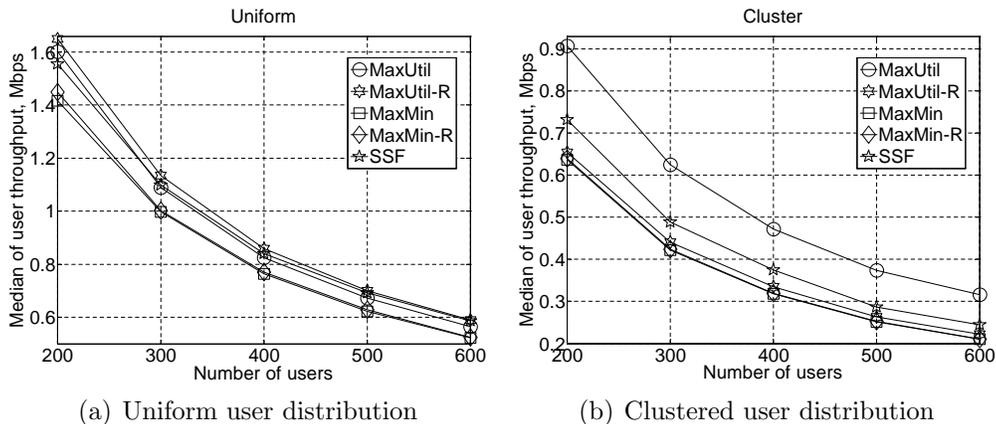


Figure 6.3: The median of user throughput.

Table 6.1: Fairness index (cf. [3]) of user throughput allocation for two kinds of user distributions (cluster and uniform) in a network with 36 APs and 400 users. (Unit: %)

	MaxMin	MaxMin-R	MaxUtil	MaxUtil-R	SSF
Cluster	99.6	97.9	85.7	71.4	34.2
Uniform	100	99.3	98.2	95.7	85.5

Fig. 6.3(b) and Table 6.1.

Surprisingly, the single-radio scheme *MaxUtil-R* yields worse median throughput than SSF, mainly because our rounding method (as presented in the numerical example above) makes users choose stronger APs, thereby causing unbalanced loads on APs. The rounding method in [36] may be modified to be imposed upon *MaxUtil* for better rounding performance; this is a subject for future research. Nevertheless, *MaxUtil-R* yields similar 25-percentile user throughputs as *MaxMin-R*, and is 44% and 17% higher than SSF in cluster and uniform cases, respectively (as seen in Fig. 6.4). Moreover, Table 6.1 indicates that SSF has poor fairness indices as compared with all other schemes (37% lower than *MaxUtil-R* in cluster case, for example). In summary, our method, *MaxUtil-R*, outperforms SSF in terms of 25-percentile throughput and fairness index with small sacrifice of median throughput.

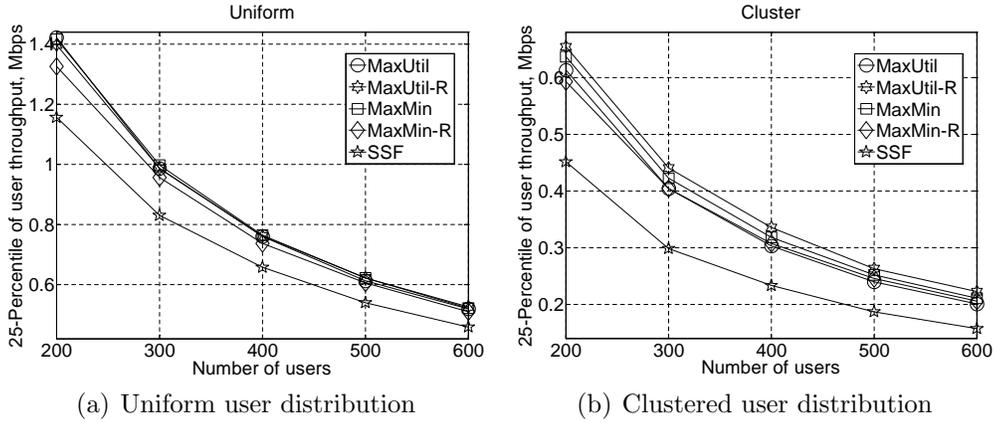


Figure 6.4: The 25-percentile of user throughput.

6.5 Conclusions of This Chapter

We find analytical expressions for the optimal channel usage time allocation and present a fast iterative algorithm to achieve the optimum. Simulation results show that when users are clustered, our utility maximization formulation yields substantial throughput gain over both the max-min scheme in [36] and the SSF scheme, which is currently being used in WLAN products. When users are uniformly distributed in space, our max utility scheme is similar as the scheme in [36], and achieves better fairness than SSF. Regardless of the number of APs or users in a network, the convergence of the sum utility is fast in various network conditions such as users joining, leaving, or moving within the network. Therefore, the iterative algorithm has good scalability and can be implemented in real time.

Chapter 7

Conclusions

In this dissertation, we have examined the benefit of applying *site specific knowledge* to frequency allocation, transmit power control, and load balancing in wireless networks. A central network controller equipped with site specific knowledge is able to differentiate the sources of RF interference at every AP or client. By predicting the power from each interference source, the controller has a *bird's eye* view of the entire wireless network that consists of multiple APs and clients; therefore, the controller can perform centralized optimization for frequency allocation, transmit power control, or load balancing.

We also present measurement-based frequency allocation algorithms that can be used when site specific knowledge is available; the algorithms require that a subset of APs and clients measure their in-situ interference power at all available channels. We have presented three different algorithms that adjust APs' frequency channels based on the measured interference. We have shown in Chapter 3 that the proposed measurement-based algorithms achieve substantial throughput gains over all other published work on frequency allocations in wireless networks. Nevertheless, it may take a long time for measurement-based algorithms to learn the interference power between any transmitter and any receiver. Since site specific knowledge is able to quickly predict each individual interference component *a priori*, site-specific knowledge-based algorithms can better mitigate the negative impact from strong RF interference sources. Simulation results in Chapter 4 corroborate our hypothesis, and show that the our site-specific knowledge-based frequency allocation algorithms perform even better than the measurement-based algorithms.

In Chapters 3 and 4, we optimize frequency allocations to minimize co-channel interference and maximize the throughput of the network, assuming the transmit power of APs

and users are fixed. Chapter 5 studies the benefit of adjusting the transmit power to further reduce co-channel interference and maximize network throughputs. Increasing the transmit power of an AP can potentially increase the downlink throughput from this AP to its clients, but also induces larger interference on nearby APs or clients that are on the same channel, thereby lowering their throughputs. Since site specific knowledge enables the central network controller to predict path losses between every AP and client, we formulate a centralized transmit power control problem in Chapter 5 in order to optimize clients' throughputs in the entire wireless network. We have shown that the formulated problem is a geometric program; thus, off-the-shelf algorithms may be used to optimize transmit powers efficiently, and real-time implementation is possible. Simulation results show that we improve the 25, 10, 5, and 3 percentiles of users' throughputs by up to 4.2%, 9.9%, 38%, and 110%, and save power by 20% with transmit power control, as compared with using fixed transmit powers.

In Chapter 6, we present an efficient load-balancing algorithm that optimally associates clients with APs. This algorithm is useful when most clients are within coverage of more than one AP, since in such a case clients may not simply choose the AP with the strongest signal but instead should take traffic loads into account. The proposed load-balancing algorithm is very efficient and produces little overhead. Our algorithm can be carried out either at a central network controller with site-specific propagation predictions, or in a decentralized manner. Our algorithm yields substantial throughput gain over the state of the art.

Note that practical ways to implement the proposed algorithm in real time are given for each algorithm in this dissertation. We are familiar with the example that vehicle drivers can find optimal route to the destination if the vehicle is equipped with a GPS. Site specific knowledge to wireless network management is like GPS is to driving. We believe that site specific knowledge can be used more extensively to solve and optimize other wireless communication systems in both cellular networks and WLANs.

Bibliography

- [1] D. J. Leith and P. Clifford, “A self-managed distributed channel selection algorithm for WLANs,” in *Proceedings of International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Apr. 2006, pp. 1 – 9.
- [2] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, “A client-driven approach for channel management in wireless LANs,” in *Proceedings of IEEE INFOCOM*, Apr. 2006, pp. 1 – 12.
- [3] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” DEC, Research Report TR-301, 1984.
- [4] S. Y. Seidel and T. S. Rappaport, “914 MHz path loss prediction models for indoor wireless communications in multifloored buildings,” *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207 – 217, Feb. 1992.
- [5] —, “Site-specific propagation prediction for wireless in-building personal communication system design,” *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 879 – 891, November 1994.
- [6] L. Piazzzi and H. L. Bertoni, “Achievable accuracy of site-specific path-loss predictions in residential environments,” *IEEE Transactions on Vehicular Technology*, vol. 48, no. 3, pp. 922 – 930, May 1999.
- [7] A. Verstak, J. He, L. T. Watson, T. S. Rappaport, C. R. Anderson, N. Ramakrishnan, C. A. Shaffer, K. Bae, J. Jiang, and W. H. Tranter, “S⁴W: globally optimized design of wireless communication systems,” in *Proc. International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Apr. 2002, pp. 173 – 180.

- [8] G. Durgin, T. S. Rappaport, and H. Xu, “Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz,” *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1484 – 1496, Nov. 1998.
- [9] M. Hassan-Ali and K. Pahlavan, “A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 1, pp. 112 – 124, Jan. 2002.
- [10] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.
- [11] C. Na, J. K. Chen, and T. S. Rappaport, “Measured traffic statistics and throughput of IEEE 802.11b public WLAN hotspots with three different applications,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3296 – 3305, Nov. 2006.
- [12] —, “Hotspot traffic statistics and throughput models for several applications,” in *Proc. IEEE Globecom*, vol. 5, 29 Nov. - 3 Dec. 2004, pp. 3257 – 3263.
- [13] R. R. Skidmore, A. Verstak, N. Ramakrishnan, T. S. Rappaport, L. T. Watson, J. He, S. Varadarajan, C. A. Shaffer, J. Chen, K. K. Bae, J. Jiang, and W. H. Tranter, “Towards integrated PSEs for wireless communications: Experiences with the S⁴W and SitePlanner® projects,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 20 – 34, Apr. 2004.
- [14] T. S. Rappaport and R. R. Skidmore, “System and method for predicting network performance and position location using multiple table lookups,” U.S. Patent Appl., no. 20040259555, Dec. 2004.
- [15] —, “System and method for automated placement or configuration of equipment for obtaining desired network performance objectives and for security, RF tags, and bandwidth provisioning,” U.S. Patent Appl., no. 20040236547, Nov. 2004.

- [16] B. E. Henty and T. S. Rappaport, “Throughput measurements and empirical prediction models for IEEE 802.11b wireless LAN (WLAN) installations,” Virginia Tech, Technical Report MPRG 01-08, 2001.
- [17] T. S. Rappaport, B. E. Henty, and R. R. Skidmore, “System and method for design, tracking measurement, prediction and optimization of data communication networks,” U.S. Patent No. 6,973,622, December 6, 2005.
- [18] W. Zhang, “Fast two-dimensional diffraction modeling for site-specific propagation prediction in urban microcellular environments,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 428 – 436, Mar. 2000.
- [19] M. F. Iskander and Z. Yun, “Propagation prediction models for wireless communication systems,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 662 – 673, Mar. 2002.
- [20] A. Mishra, S. Banerjee, and W. Arbaugh, “Weighted coloring based channel assignment for WLANs,” *ACM Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 19 – 31, 2005.
- [21] K. K. Leung and B.-J. Kim, “Frequency assignment for IEEE 802.11 wireless networks,” in *Proceedings of IEEE Vehicular Technology Conference*, vol. 3, Oct. 2003, pp. 1422 – 1426.
- [22] Y. Lee, K. Kim, and Y. Choi, “Optimization of AP placement and channel assignment in wireless LANs,” in *Proc. IEEE Conf. on Local Computer Networks (LCN)*, Nov. 2002, pp. 831 – 836.
- [23] R. R. Skidmore and T. S. Rappaport, “System and method for measuring and monitoring wireless network performance in campus and indoor environments,” U.S. Patent no. 7,096,160, August 22, 2006.

- [24] P. Bahl and V. N. Padmanabhan, “A software system for locating mobile users: Design, evaluation, and lessons,” Microsoft, Tech. Rep., Apr. 2000.
- [25] C. Na and T. S. Rappaport, “Measured wireless LAN public hotspot traffic statistics,” *IEE Electronics Letters*, vol. 40, no. 19, pp. 1202 – 1203, Sep. 2004.
- [26] A. Balachandran, P. Bahl, and G. M. Voelker, “Hot-spot congestion relief in public-area wireless networks,” in *Proc. Fourth IEEE Workshop on Mobile Computing Systems and Applications*, June 2002, pp. 70 – 80.
- [27] D. Kotz and K. Essien, “Analysis of a campus-wide wireless network,” in *Proc. the Eighth Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*. ACM Press, September 2002, pp. 107 – 118.
- [28] M. Chiang and J. Bell, “Balancing supply and demand of bandwidth in wireless cellular networks: utility maximization over powers and rates,” in *Proc. IEEE INFOCOM*, vol. 4, March 2004, pp. 2800 – 2811.
- [29] G. J. Foschini and Z. Miljanic, “A simple distributed autonomous power control algorithm and its convergence,” *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 6441 – 6446, April 1993.
- [30] M. Xiao, N. B. Shroff, and E. K. P. Chong, “A utility-based power-control scheme in wireless cellular systems,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 210 – 221, April 2003.
- [31] S. V. Hanly, “An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1332 – 1340, sep 1995.
- [32] R. D. Yates and C.-Y. Huang, “Integrated power control and base station assignment,” *IEEE Transactions on Vehicular Technology*, vol. 44, no. 3, pp. 638 – 644, Aug. 1995.

- [33] H. Velayos, V. Aleo, and G. Karlsson, “Load balancing in overlapping wireless LAN cells,” in *Proc. IEEE International Conference on Communications*, vol. 7, Jun. 2004, pp. 3833 – 3836.
- [34] V. Aleo, “Load distribution in IEEE 802.11 cells,” Master’s thesis, KTH, Royal Institute of Technology, Stockholm, Sweden, 2003.
- [35] Y. Fukuda, T. Abe, and Y. Oie, “Decentralized access point selection architecture for wireless LANs,” in *Proc. Wireless Telecommunications Symposium*, May 2004, pp. 137 – 145.
- [36] Y. Bejerano, S.-J. Han, and L. E. Li, “Fairness and load balancing in wireless LANs using association control,” in *Proc. ACM MobiCom*, Sep. 2004, pp. 315 – 329.
- [37] J. K. Chen, G. de Veciana, and T. S. Rappaport, “Improved measurement-based frequency allocation algorithms for wireless networks,” submitted to *IEEE Globecom*, Washington DC, USA, Nov. 2007.
- [38] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, “Self-management in chaotic wireless deployments,” in *Proceedings of ACM MobiCom*, Aug. 2005, pp. 185 – 199.
- [39] T. Bonald, S. Borst, N. Hegde, and A. Proutière, “Wireless data performance in multi-cell scenarios,” in *Proceedings of ACM SIGMETRICS*, Jun. 2004, pp. 378 – 387.
- [40] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74 – 80, Oct. 2003.
- [41] J. K. Chen, T. S. Rappaport, and G. de Veciana, “Site specific knowledge for improving frequency allocations in wireless LAN and cellular networks,” submitted to *IEEE Vehicular Technology Conference*, Baltimore MD, USA, Oct. 2007.
- [42] E. G. Coffman, M. Elphick, and A. Shoshani, “System deadlocks,” *ACM Computing Surveys (CSUR)*, vol. 3, no. 2, pp. 67 – 78, Jun. 1971.

- [43] J. W. Havender, “Avoiding deadlock in multi-tasking systems,” *IBM Systems Journal*, vol. 2, pp. 74 – 84, 1968.
- [44] (2007, Apr.) Lexicographical order. [Online]. Available: http://en.wikipedia.org/wiki/Lexicographical_order
- [45] A. Fabrikant, C. Papadimitriou, and K. Talwar, “The complexity of pure Nash equilibria,” in *Proc. thirty-sixth annual ACM symposium on Theory of computing*, Jun. 2004, pp. 604 – 612.
- [46] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.
- [47] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [48] E. N. Skomal and J. Albert A. Smith, *Measuring the Radio Frequency Environment*. Van Nostrand Reinhold, 1985.
- [49] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556 – 567, Oct. 2000.
- [50] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2001.
- [51] J. K. Chen, T. S. Rappaport, and G. de Veciana, “Site specific knowledge for improving transmit power control in wireless networks,” submitted to *IEEE Globecom*, Washington DC, USA, Nov. 2007.
- [52] R. R. Skidmore, T. S. Rappaport, and A. L. Abbott, “Interactive coverage region and system design simulation for wireless communication systems in multifloored indoor environments: SMT Plus,” in *Proc. IEEE International Conference on Universal Personal Communications*, vol. 2, Oct. 1996, pp. 646 – 650.
- [53] “Data Sheet of Cisco Aironet 1100 Series Access Point,” Cisco, San Jose, CA, USA.

- [54] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [55] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, “A tutorial on geometric programming,” Stanford University, Tech. Rep., 2006. [Online]. Available: http://www.stanford.edu/~boyd/gp_tutorial.html
- [56] (2007, Apr.) Binomial theorem. [Online]. Available: http://en.wikipedia.org/wiki/Binomial_theorem
- [57] (2007, Apr.) Multinomial theorem. [Online]. Available: http://en.wikipedia.org/wiki/Multinomial_theorem
- [58] J. K. Chen, T. S. Rappaport, and G. de Veciana, “Iterative water-filling for load-balancing in wireless LAN or microcellular networks,” in *IEEE Vehicular Technology Conference*, vol. 1, May 2006, pp. 117 – 121.
- [59] “Data Sheet of Cisco Catalyst 6500 Series Wireless LAN Services Module,” Cisco, San Jose, CA, USA.
- [60] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, “Iterative water-filling for Gaussian vector multiple-access channels,” *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 145 – 152, Jan. 2004.

Vita

Jeremy Kang-pen Chen received the B.S. degree from National Taiwan University, Taipei, Taiwan, Republic of China, in 2001 and the M.S. and Ph.D. degrees from The University of Texas at Austin, in 2004 and 2007 respectively, all in electrical engineering. He will join QUALCOMM Flarion Technologies, Inc., in Bridgewater, New Jersey, USA in May 2007.

Since 2003, he has been a research assistant in the Wireless Networking and Communications Group (WNCG) at UT Austin. He has extensive experiences applying analytical skills to solve cross-layer network design problems, performing RF propagation and network traffic measurements, algorithm analysis and design, and software engineering. He held summer internships with Wireless Valley Communications, Inc. (now Motorola, Inc.), Austin, Texas, in 2004 and Intumit Technology, Taipei, Taiwan in 2000. During 1998 and 2001, he was an undergraduate research assistant at Academia Sinica, Taipei, Taiwan, Republic of China, where he co-developed a cross-platform Chinese-input software package called *Chewing*, which won the Taiwan Free Software Community Awards in 2003 and is now widely used in the Chinese community. He has won the Asian Championship and world final's 10th place (out of 1,400) in the International Collegiate Programming Contest held by the Association for Computing Machinery (ACM) in 1999. He received the Silver Medal in the International Olympics in Informatics in 1997. He ranked the first place (out of approximately 50,000) in the Annual Joint College Entrance Exam in Republic of China (Taiwan) in 1997.

Permanent address: 11F, 252, FU-HE ROAD, YUNG-HE CITY,
TAIPEI COUNTY 23450, REPUBLIC OF
CHINA (TAIWAN)

This dissertation was typeset with \LaTeX^\dagger by the author.

[†] \LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's \TeX Program.