

**Integrated Scheduling and Beam Steering for Spatial Reuse**

by

**Eric William Anderson**

B.A. Carleton College, 2001

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Computer Science

2010

This thesis entitled:  
Integrated Scheduling and Beam Steering for Spatial Reuse  
written by Eric William Anderson  
has been approved for the Department of Computer Science

---

Douglas Sicker

---

Dirk Grunwald

---

Manuel Laguna

---

Timothy Brown

---

Sanjay Shakkottai

Date \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Anderson, Eric William (Ph.D., Computer Science)

Integrated Scheduling and Beam Steering for Spatial Reuse

Thesis directed by Associate Professor Douglas Sicker and Associate Professor Dirk Grunwald

This document describes an approach to integrating antenna selection and control into a time-division MAC scheduling process. I argue that through such integration it is possible to achieve greater spatial reuse and interference mitigation than by solving the two problems separately. Without coupling between the MAC scheduling and physical antenna configuration processes, a “chicken-and-egg” problem exists: If antenna decisions are made before scheduling, they cannot be optimized for the communication that will actually occur. If, on the other hand, the scheduling decisions are made first, the scheduler cannot know what the actual interference and communications properties of the network will be.

This dissertation presents algorithms for optimal spatial reuse TDMA scheduling with reconfigurable antennas. I present and solve the joint beam steering and scheduling problem for spatial reuse TDMA and describe an implemented system based on the algorithms developed. The algorithms described achieve up to a 600% speedup over TDMA in the experiments performed. This is based on using an optimization decomposition approach to arrive at a working distributed protocol which is equivalent to the original problem statement while also producing optimal solutions in an amount of time that is **at worst** linear in the size of the input. This is, to the best of my knowledge, the first actually implemented STDMA scheduling system based on dual decomposition. This dissertation identifies and briefly address some of the challenges that arise in taking such a system from theory to reality.

## Acknowledgements

This dissertation contains the thoughts, effort, and words of the many collaborators whose contributions have made my research possible. I would like first to thank my colleagues and coauthors Anmol Sheth, Brita Munsinger, Douglas Sicker, Dirk Grunwald, Michael Buettner, Richard Han, Christian Doerr, Dola Saha, Gary Yee, Kevin Bauer, Caleb Phillips, Damon McCoy, Harold Gonzales, Greg Grudic, and Markus Breitenbach. This dissertation contains text, figures, and data from the following papers, on which I am but one of several authors [Buettner 07, Anderson 08b, Anderson 09d, Anderson 09c, Anderson 09a, Anderson 09b, Anderson 10b, Anderson 10a]. Thanks also to Tim Brown and Ken Baker for keeping me honest about radio, to my committee members Manuel Laguna and Sanjay Shakkottai for tolerating my regular pestering, and to my advisors Douglas Sicker and Dirk Grunwald for all the things that advisors do. I am especially indebted to Caleb, Kevin, and Mike for their intellectual engagement.

Most of all, I wish to thank my best friend and partner in life, Erin Siffing, for her constant support and patience.

# Contents

Chapter	
<b>1</b>	<b>Introduction</b> . . . . . 1
1.1	Summary of Results . . . . . 2
1.2	Rationale . . . . . 2
1.2.1	Example Scenario . . . . . 3
1.2.2	Empirical Study . . . . . 7
1.3	Overview of Research . . . . . 8
1.3.1	The Joint Beam Selection and Scheduling Problem . . . . . 9
1.3.2	Purpose . . . . . 10
1.4	Definitions . . . . . 10
1.5	Organization . . . . . 10
<b>2</b>	<b>Related Work</b> . . . . . 13
2.1	TDMA and Spatial Reuse . . . . . 13
2.2	Transmission Scheduling for STDMA . . . . . 16
2.2.1	Pairwise link conflict models . . . . . 19
2.2.2	Aggregate Interference Models . . . . . 23
2.2.3	Continuous Link Quality . . . . . 24
2.3	STDMA with Antenna Considerations . . . . . 26
2.3.1	Opportunistic Antenna Reconfiguration . . . . . 26

2.3.2	Scheduling Based on Assumed Antenna Capabilities . . . . .	26
2.3.3	Scheduling Based on Per-Link Antenna Optimization . . . . .	28
2.4	Scheduling Integrated with Other Network Properties . . . . .	30
2.4.1	Lower Layers . . . . .	30
2.4.2	Higher Layers . . . . .	31
2.5	Reconfigurable Antennas in Un-Scheduled Networks . . . . .	32
2.5.1	Random Access One-Hop . . . . .	32
2.5.2	Random Access Packet Relay . . . . .	33
2.5.3	MAC Protocols for Directional Antennas . . . . .	34
2.6	Networking with Fixed Directional Antennas . . . . .	34
2.6.1	Directional Antennas in Mesh Networks . . . . .	35
2.7	Related Problems . . . . .	36
2.7.1	Channel Assignment . . . . .	36
2.7.2	Steerable Antennas Generally . . . . .	36
2.7.3	Cellular Telephony . . . . .	39
2.8	Cross-Layer Optimization in Networking . . . . .	40
2.8.1	Layering as Optimization Decomposition . . . . .	40
2.8.2	Introduction to Mathematical Program Decomposition Techniques . . . . .	42
2.8.3	Optimization-Based Scheduling . . . . .	60
2.8.4	Examples of Decomposition in Wireless Networking . . . . .	61
2.9	Summary . . . . .	62
<b>3</b>	<b>Problem and Formulation</b> . . . . .	<b>64</b>
3.1	Problem Definition . . . . .	64
3.1.1	Formulation . . . . .	64
3.1.2	Extensions . . . . .	68
3.2	Computational Complexity . . . . .	68

<b>4</b>	<b>Decomposition Process</b>	<b>72</b>
4.1	First Decomposition: Dantzig-Wolfe on JBSS-MP . . . . .	72
4.2	Convexity of CLAP . . . . .	78
4.2.1	SINR Constraint . . . . .	78
4.2.2	Objective Function . . . . .	81
4.2.3	Half-Duplex Constraint . . . . .	83
4.2.4	S-V Coupling Constraint . . . . .	83
4.2.5	D-B (Antenna) Coupling Constraint . . . . .	83
4.2.6	Convex Pattern Combination Constraint . . . . .	85
4.2.7	The Convex-Constraint-CLAP Program . . . . .	85
4.2.8	Pseudo-Integral Convex-Constraint-CLAP . . . . .	86
4.3	Second Decomposition: Lagrangian Relaxation on CLAP . . . . .	88
4.4	Block Separability of FARP . . . . .	92
4.5	Second Lagrangian Decomposition on CLAP . . . . .	96
4.6	Economic Interpretation . . . . .	98
4.7	Lagrange Multiplier Updates . . . . .	99
4.7.1	Convergence Properties . . . . .	100
4.8	Complexity Results . . . . .	101
4.9	Summary . . . . .	101
<b>5</b>	<b>Mathematical Issues in System Implementation</b>	<b>104</b>
5.1	Solution Oscillation . . . . .	104
5.2	Partial Pricing . . . . .	107
5.3	Distributed Consensus . . . . .	107
<b>6</b>	<b>Performance Evaluation</b>	<b>109</b>
6.1	Numerical Experiments . . . . .	109
6.1.1	Running time . . . . .	110

6.1.2	Schedule Efficiency . . . . .	110
6.1.3	Alternate Cases . . . . .	113
6.2	Deployed System . . . . .	116
6.3	Performance Problems . . . . .	121
6.3.1	Implementation Issues . . . . .	121
6.3.2	Algorithmic Issues . . . . .	125
6.4	Conclusion . . . . .	126
<b>7</b>	<b>Conclusions and Future Directions</b>	<b>127</b>
7.1	Current Work . . . . .	127
7.2	Conclusion . . . . .	127
	<b>Bibliography</b>	<b>129</b>
	<b>Appendix</b>	
<b>A</b>	<b>Modeling Effects of Directional Antennas</b>	<b>162</b>
A.1	Introduction . . . . .	162
A.2	Background And Related Work . . . . .	163
A.2.1	Path Loss Models . . . . .	164
A.2.2	Fading Models . . . . .	166
A.2.3	Directional Models . . . . .	166
A.3	Method . . . . .	169
A.3.1	Data Collection Procedure . . . . .	169
A.3.2	Commodity Hardware Should Suffice . . . . .	170
A.4	Measurements . . . . .	170
A.4.1	Experiments Performed . . . . .	177
A.4.2	Normalization . . . . .	181



A.4.3	Error relative to the reference . . . . .	182
A.4.4	Observations . . . . .	186
A.5	A New Model of Directionality . . . . .	186
A.5.1	Limitations of Orthogonal Models . . . . .	186
A.5.2	An Integrated Model . . . . .	190
A.5.3	Describing and Predicting Environments . . . . .	194
A.6	Simulation Process . . . . .	196
A.7	Conclusion . . . . .	197
<b>B</b>	<b>Simulation Practices</b>	<b>199</b>
B.1	Introduction . . . . .	199
B.2	Background and Related Work . . . . .	200
B.3	A New Simulation Approach . . . . .	204
B.4	Case Study: Physical Space Security using Smart Antennas . . . . .	205
B.4.1	Implementation . . . . .	206
B.4.2	Simulation . . . . .	206
B.4.3	Analysis . . . . .	207
B.5	Conclusion . . . . .	211
<b>C</b>	<b>The Wide-Area Radio Testbed</b>	<b>216</b>
C.1	Introduction . . . . .	216
C.1.1	Design Goals . . . . .	217
C.1.2	Smart Antenna System . . . . .	218
C.2	Commodity Hardware as a Research Platform . . . . .	220
C.2.1	Received Signal Strength Accuracy . . . . .	220
C.2.2	Transmit Power Precision . . . . .	222
C.2.3	MAC-Layer Flexibility . . . . .	222
C.2.4	Implementing Non-CSMA MACs . . . . .	224

C.2.5	Precise Timing Control . . . . .	227
C.2.6	Time Synchronization . . . . .	228
C.3	Administration and Maintenance Infrastructure . . . . .	228
C.3.1	Centralization . . . . .	229
C.3.2	Management System . . . . .	230
C.3.3	Infrastructure Configuration . . . . .	232
C.3.4	Reliability and Availability . . . . .	233
C.3.5	Remote Repair . . . . .	233
C.3.6	Interchangeable Parts . . . . .	234
C.3.7	Security . . . . .	235
C.4	Deployment Logistics . . . . .	235
C.4.1	Timeline . . . . .	237
C.4.2	Costs . . . . .	238
C.5	Proof-of-Concept Experiments . . . . .	238
C.6	Related Work . . . . .	239
C.6.1	Outdoor Wireless Testbeds . . . . .	241
C.6.2	Indoor Wireless Testbeds . . . . .	242
C.7	Conclusion . . . . .	245
<b>D</b>	<b>Model Code</b>	<b>248</b>
D.1	Model Files . . . . .	249
D.2	Command Files . . . . .	295
D.2.1	On-Line System . . . . .	295
D.2.2	Off-Line Evaluation . . . . .	332

## Tables

### Table

1.1	Definitions used in this document. . . . .	11
2.1	Directional Antenna Categorization (modified from [Li 05a]) . . . . .	35
2.2	Summary of decomposition techniques, modified from [Conejo 06, table 1.29]. . . . .	45
3.1	Notation . . . . .	65
4.1	Constraint matrix structure of CLAP . . . . .	88
4.2	Constant substitutions for LR decomposition. . . . .	92
A.1	Summary of data sets. . . . .	182
A.2	Factors influencing fitted offset values, 16-bin case. . . . .	195
A.3	Summary of Data Derived Simulation Parameters: Gain-offset regression coefficient ( $K_{gain}$ ), offset residual std. error ( $S_{off}$ ), and signal strength residual std. error ( $S_{ss}$ ). 196	
B.1	Summary of Data-Derived Simulation Parameters, repeated from Table A.3 on page 196. 204	
B.2	Physical-layer simulation options . . . . .	207
B.3	Size of 50% vulnerability region, indoor scenario. . . . .	208
B.4	Summary of results for factorial ANOVA on KS-test statistic across all simulation configurations except for the “omni” directivity model. P-values are omitted because all treatments are statistically significant at a level of $p < 2.2 \cdot 10^{-16}$ . Error / residual Df are 9948 indoor, 52428 outdoor. . . . .	210

C.1 Deployment Timeline . . . . . 238

C.2 Cost of labor and parts per WART node. The labor of research group members is  
not considered. . . . . 239

## Figures

### Figure

1.1	Scheduling and beam steering example: Avoidable mutual exclusion. . . . .	4
1.2	“Pie wedge” or flat-topped antenna pattern. . . . .	4
1.3	Interference between neighboring links when greedy antenna patterns are used. Reference lines show theoretical SNR values for $10^{-6}$ BER with BPSK (10.5 dB) and 64-QAM (26.5 dB) modulation schemes. . . . .	7
1.4	Sender-to-sender signal strength on neighboring links when greedy antenna patterns are used. . . . .	8
2.1	Aggregate capacity of interference-limited Gaussian channels. . . . .	15
2.2	Classification tree of interference models used in spatial-reuse scheduling. . . . .	20
2.3	Complexity of cellular and general spatial reuse . . . . .	38
3.1	Size (number of variables) of JBSS-MP as a function of the number of nodes, log/log scale, limited at $10^{60}$ . . . . .	70
3.2	Size (number of variables) of JBSS-MP as a function of the number of nodes, semilog scale, limited at $10^{20}$ . . . . .	71
4.1	Outline of decomposition process. . . . .	73
4.2	Effect of iterating between RMP and column-generation (Prototype 1). Note the different y scales for the objective value and reduced cost. . . . .	77
4.3	Links for which “strong duplex” constraints are defined, relative to link $i \rightarrow j$ . . . . .	87

4.4	Scaling comparison of centralized direct solution and distributed decomposed solution. Figure shows total user CPU time consumed. . . . .	102
4.5	Scaling comparison of centralized direct solution and distributed decomposed solution. Figure shows user CPU time consumed <b>per logical process</b> . . . . .	103
5.1	Lagrange multiplier oscillation example (random scenario snapshot SVN r.1088) . . .	105
6.1	Distribution of number of (minor) iterations necessary in simulations to first reach and optimal solution. . . . .	111
6.2	Distribution of number of (minor) iterations necessary in simulations to first reach and optimal solution – detail. . . . .	112
6.3	Empirical cumulative distribution of achieved speedup (ratio of optimal to TDMA performance) across all simulations. . . . .	114
6.4	Achieved speedup by number of links. . . . .	115
6.5	Grid scenario, 9 nodes . . . . .	116
6.6	Iterations to optimality and to termination for the grid scenario . . . . .	117
6.7	Iterations to optimality and to termination for the clique scenario . . . . .	118
6.8	Trace of algorithm scheduling links $B \rightarrow A$ and $C \rightarrow D$ concurrently, as seen locally at node C. The top strip shows $\bar{\lambda}$ , middle strip shows $\hat{S}$ , and the bottom strip shows the <b>combined</b> gain $\hat{D}_{ij}\hat{D}_{ji}$ . Note that $B \rightarrow D$ and $D \rightarrow A$ are interference if both data links are active. The aligned $x$ axis is time in seconds. . . . .	120
A.1	Sample directional antenna gain pattern displayed on a polar graph . . . . .	162
A.2	Example of two-ray model attenuation, from [Neskovic 00]. . . . .	165
A.3	Illustration of the common path loss model for directional antennas . . . . .	167
A.4	Probability Density Function of percentage of dropped measurement packets in a given angle for all angles and all data sets. . . . .	171

A.5	Linear fit to RSS error observed from commodity cards during calibration. The red (upper) line indicates the regression fit and the black (lower) line is perfect equality.	172
A.6	Comparison of signal strength patterns across different environments and antennas: Parabolic dish indoor environments.	173
A.7	Comparison of signal strength patterns across different environments and antennas: Parabolic dish outdoor environments.	174
A.8	Comparison of signal strength patterns across different environments and antennas: Patch panel indoor environments.	175
A.9	Comparison of signal strength patterns across different environments and antennas: Patch panel outdoor environments.	176
A.10	Receiver side of measurement setup in floodplain	178
A.11	Floor plan of office building used in Array-Indoor-A, Array-Indoor-B, Patch-Indoor-B, Patch-Indoor-C, Parabolic-Indoor-B, and Parabolic-Indoor-C.	179
A.12	Comparison of signal strength patterns across different environments and antennas. Repeated from Figures A.6 to A.9 on pages 173–176 for ease of comparison.	183
A.13	Cumulative Density Functions for the error process (combined across multiple traces) for each antenna type.	185
A.14	Differences between the orthogonal model and observed data in dB: $\hat{P}_{rx} - P_{rx}$ .	188
A.15	Mean error of orthogonal model for each observation point in the Array-Outdoor-A data set. The format is the same as in figures A.14a on page 188 through A.14f on page 188.	189
A.16	Effect of increasing bin count (decreasing bin size) on modeling precision.	192
A.17	Residual error of the discrete offset model with 16 bins.	193
B.1	Physical-layer simulation framework.	201
B.2	Standard simulation model of directional antennas assumes all signals are emitted along a single path.	212

B.3	Example of data striping application . . . . .	213
B.4	CDF plots of application layer performance for simulation configurations: The black line is the <b>observed data</b> , and the red (or grey) lines are the results of repeated simulation runs. The X axis is the proportion of decodable packets, and the Y axis is the cumulative fraction. . . . .	214
B.5	Test-statistic of a two sample Kolmogorov-Smirnov test, run for each simulation configuration against the measured data (smaller values are better). . . . .	215
C.1	Unidirectional Pattern . . . . .	219
C.2	Omnidirectional Pattern . . . . .	219
C.3	Linear fit of reported versus actual signal strength on commodity cards during calibration. . . . .	221
C.4	CSMA/CA Evaluation Apparatus . . . . .	225
C.5	Management box configuration . . . . .	231
C.6	Comparison of available links and link quality between seven testbed nodes using best-steered directional patterns and omnidirectional patterns. Stronger links are indicated with a wider arrow of a darker color. The best links are those with a link of greater than -60 dBm. The worst links plotted are barely above the noise-floor with greater than -95 dBm achieved RSS. . . . .	240



## Chapter 1

### Introduction

The thesis of this dissertation is that integrating physical-layer antenna control with MAC-layer scheduling allows reduced interference and greater spatial reuse in dense wireless networks. Without such integration, a “chicken-and-egg” problem exists: If antenna decisions are made before scheduling, they cannot be optimized for the communication that will actually occur. If, on the other hand, the scheduling decisions are made first, the scheduler cannot know what the actual interference and communications properties of the network will be. In the current state of the art, minimal consideration is given to this integration: The few studies that consider scheduling in the context of steerable antennas optimize the antennas involved in each link for that link **in isolation**, without considering any other links, actual or possible.

I find significant gains by integrating scheduling with antenna reconfiguration. This work does not have the level of direct comparative evaluation I would like, but the available comparisons are quite promising. A simulation analysis of the algorithm developed in this dissertation shows a speedup relative to simple TDMA of up to 600%. In many of these cases, simple TDMA is the highest-reuse schedule that can be shown to be safe without knowing the gains from antenna configuration. Small-scale empirical studies performed on the WART testbed show that that simple techniques such as greedy approaches to antenna steering and scheduling result in substantial interference between neighboring links.

## 1.1 Summary of Results

This dissertation develops a distributed scheduling mechanism for spatial reuse in TDMA networks with reconfigurable antennas. There are three primary contributions: (1) A set of algorithms for integrated beam steering and scheduling with a mathematically sound foundation, (2) an implemented, deployed, and tested MAC system based on those algorithms, and (3) a general decomposition framework for this and related problems. This is the first implemented system for optimization decomposition-based wireless scheduling, though there is a significant body of theory, and Lagrangian coupling between the MAC and higher layers has recently been implemented by others. This work develops a dual-decomposition approach to the underlying problem of identifying optimal **activation sets** of concurrent links, including the configuration of those links.

## 1.2 Rationale

There are significant gains to be had from integrating scheduling with antenna reconfiguration. Many spatial-reuse scheduling algorithms have been proposed (see 2.2), but in general they do not allow for antenna configuration changes. That is, they regard the received power from any given transmitter at any other receiver as either fixed or as a simple function of the transmitted power level. Consequently, if such a scheduling process is used for stations that do have dynamic antennas of some sort, the best the algorithm can do is to assume **one** of the possible configurations and schedule as though that were the **only** option. If an antenna reconfiguration process – however well-designed – occurs after the scheduling process, it may improve the quality of the selected links, but it cannot enable additional links. Conversely, if antenna configuration were to occur before the scheduling process, it could at best make decisions to improve the average quality of all possible combinations of links, but it would have no basis for choosing which subset to prioritize when there are conflicting options. Some level of coupling between the two processes is necessary in order to achieve the network’s full potential capacity. A small number of studies have considered the combination of reconfigurable antennas and spatial reuse scheduling (see 2.3), but they have not

examined the integration question: One paper assumes perfect beam forming (no interference at all) [Cain 03], and the others assume that each node always steers directly at its communicating partner.

### 1.2.1 Example Scenario

Consider the following simplified scenario, shown in 1.1a: Stations A, B, C, and D are arranged in a square. The station in the lower right (D) corner has traffic for the station in the upper left (A), and the station in the upper right (B) has traffic for the station in the lower left (C). This is assuming truly uni-directional communication; any responses, including acknowledgements, from the receiver to the sender would be part of a separate data flow. Suppose that each station has a steerable antenna with an idealized “pie wedge” pattern like that shown in 1.2. Assume that the main lobe width is slightly greater than  $\frac{\pi}{2}$ , and the peak to null ratio (main lobe to back lobe ratio) is 20 dB. Assume also that the difference in path loss between all pairs of stations is  $\leq 5$  dB. Suppose that some minimum signal to interference and noise ratio (SINR) is required for both of the links ( $D \rightarrow A$  and  $B \rightarrow C$ ). Let this value,  $SINR_{min}$  be  $\geq 10$  dB.

Assume reasonable but separate algorithms for link scheduling and antenna configuration. Suppose that the antenna configuration phase occurs first: At configuration time, each node does not know which other (single) node it will be communicating with, but has no information about which other nodes or links are going to be active. One optimal configuration decision would be for each node to point its main lobe directly at the node with which it will be communicating, as depicted in 1.1b. With this antenna configuration, the two links are mutually exclusive. Every node includes every other node in its main lobe, and so the directional antennas do nothing to mitigate interference. The received power of link  $D \rightarrow A$  is  $P_{TxD} * 2 * \text{main lobe gain} * Loss_{DA}$ . The received (interfering) power of link  $B \rightarrow C$  at A is  $P_{TxB} * 2 * \text{main lobe gain} * Loss_{BA}$ , so the SIR is (in log units)  $P_{TxD} - P_{TxB} - (Loss_{DA} - Loss_{BA})$ . Assume that the transmit power at B ( $P_{TxB}$ ) equals the transmit power at D ( $P_{TxD}$ ). Then, the SIR is  $\leq (Loss_{DA} - Loss_{BA})$ , which we here assume is at most 5 dB (and is perhaps more likely -5 dB). Thus, link  $D \rightarrow A$  cannot achieve its necessary

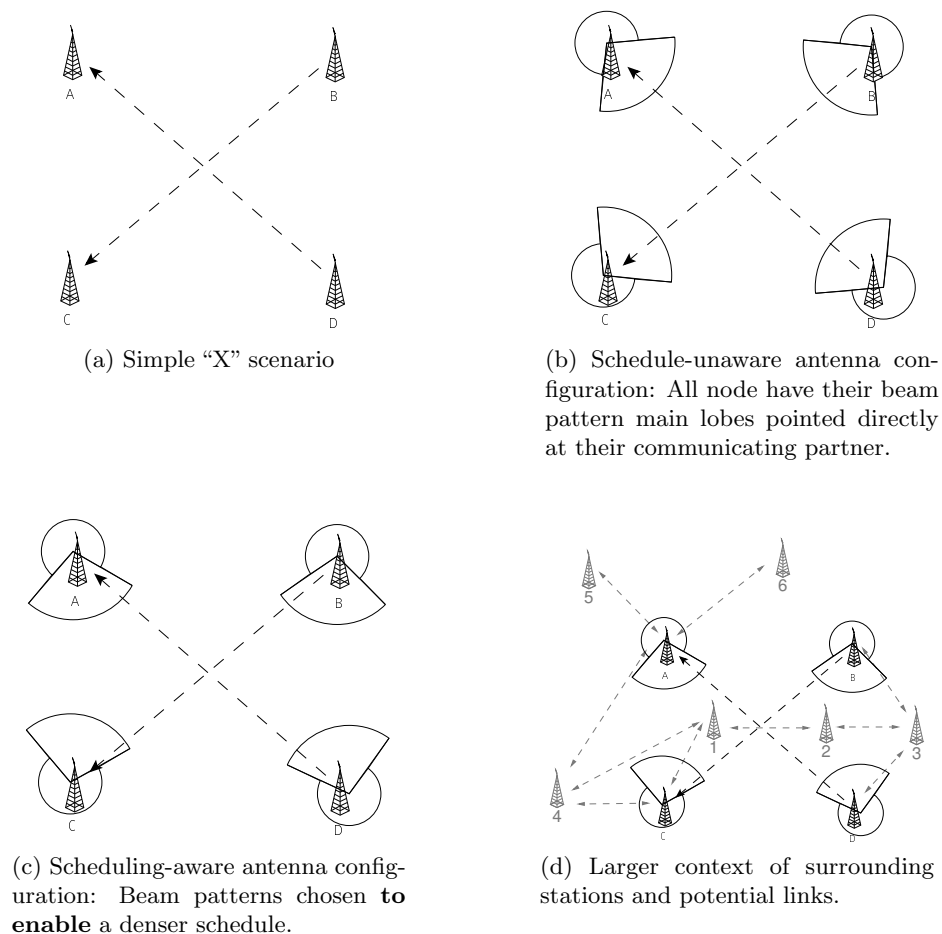
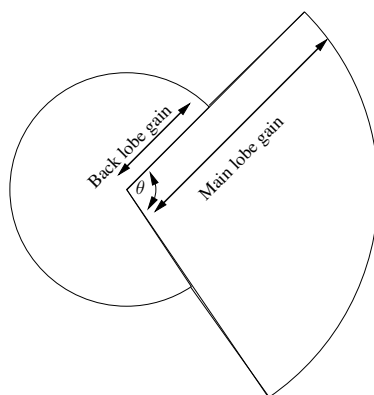


Figure 1.1: Scheduling and beam steering example: Avoidable mutual exclusion.



There is a main lobe having width  $\theta$ , and a side/back lobe covering the remainder of the azimuth. The antenna gain toward any direction is either exactly the “main lobe gain” if the direction is within the arc subtended by the main lobe, or exactly the “back lobe gain” if it is not.

Figure 1.2: “Pie wedge” or flat-topped antenna pattern.

SINR while  $B \rightarrow C$  is operating. By the same argument  $B \rightarrow C$  is also precluded by  $D \rightarrow A$ . If  $P_{TxB} \neq P_{TxD}$ , one of the two links has its SINR reduced by the difference, so it remains impossible to operate both links simultaneously. The scheduling algorithm therefore is forced to schedule the two links for different time slots. The minimally-integrated steering described in 2.3 is equivalent to schedule-unaware antenna configuration in this scenario: The difference between the two is that if any station were participating in multiple potential links, the antenna would be oriented correctly for each link when that link is considered. In this scenario, every station participates in only one potential link, so the difference is moot.

Suppose instead that the scheduling phase occurs before the antenna configuration phase. Assume that the scheduler cannot know how good the best antenna configuration will be, and that it must produce a feasible schedule. The scheduler must then make some conservative estimate of what benefits antenna reconfiguration will deliver, and schedule accordingly. Without violating the assumption that the two processes are separate, the scheduler's estimate cannot be expected to do better than the antenna configuration described above. The scheduler then cannot expect that  $B \rightarrow C$  and  $D \rightarrow A$  can be made mutually compatible, and again must schedule them for different time slots. Once that scheduling decision has been made, the configuration phase cannot do any better than what was described in the previous paragraph.

A jointly-optimized beam pattern is shown in 1.1c. The antennas are configured to minimize the gain for  $B \rightarrow A$  and  $D \rightarrow C$  interference. With this configuration, the links are not mutually exclusive: The SIR for  $D \rightarrow A$  is  $P_{TxD} - P_{TxB} - (Loss_{DA} - Loss_{BA}) + 2(\text{main lobe gain} - \text{back lobe gain})$ . By our earlier assumptions, that reduces to  $2(\text{main lobe gain} - \text{back lobe gain}) \pm 5 = 2(20) \pm 5$  dB. As long as the transmit power  $P_{TxD}$  can be set to at least  $40 - SINR_{min} - 5 = 25$  dB higher than the noise floor, the link can achieve an SINR of  $\geq SINR_{min}$ . By a parallel argument,  $C \rightarrow B$  can also meet the SINR requirement. Because this configuration makes these links compatible, they can both be scheduled in a single time slot, approximately doubling the throughput.

In a scenario this simple, the level of antenna-scheduler integration required to achieve the

improved result is minimal: During antenna configuration, it suffices to know that the scheduler would like to schedule  $C \rightarrow B$  and  $D \rightarrow A$  together if possible. Since those are the only links in the scenario, that seems obvious. If, however, there are even a modest number of possible link combinations, the antenna configuration cannot be simultaneously optimized for all of them. In this case, even if the antenna configuration process has full knowledge of the possible links and their properties, it has almost no information about what subset is useful to optimize. Suppose the “x” scenario is a subset of a larger network, shown in 1.1d. When the greyed-out stations and links are considered, there is no reason to believe that an antenna configuration process, isolated from the scheduler, would arrive at the configuration shown. The links involving stations 1, 2, and 3 would likely have interference problems, and the links connecting 4, 5, and 6 to A and C would likely have poor signal strength. There are any number of reasons why the black links might be the most important to schedule at a given moment, but that information would generally not be available to an isolated beam-forming process.

The scenario described is a simplification of reality, primarily in that the antenna pattern is discrete and the link SINR requirements are given as a simple cutoff rather than a continuous function. These simplifications are made for illustrative purposes only, and comparable situations occur without them.

In the preceding scenario, the **integrated decision process achieves twice the performance of any non-integrated process**, where performance is measured in terms of the number of time slots required to service the given demand. Note that the decision processes discussed do not assume any particular algorithm; rather they represent the best\* decisions possible given the assumed objectives and available information for each category of process. They are thus upper bounds on the performance that can be expected from any algorithms having the type of integration described. It is clear that situations exist in which more thorough integration can provide

---

\* In the case of this artificial pie wedge antenna pattern, there are continuous ranges of angles that have exactly the same gain and therefore there is an infinite set of equally optimal configurations. “Best” in this case means that the chosen configuration is **one of** the optimal choices. With any real antenna gain pattern, one would expect a finite number of positions achieving any particular gain, and thus a finite number of optimal choices – usually one. In general, the single position in any given lobe having the highest gain occurs near the center.

substantial performance improvements.

### 1.2.2 Empirical Study

To understand the effects of this phenomenon on a real network, we conducted an empirical study using a wide-area phased array testbed of seven nodes<sup>†</sup>. Considering all feasible two-link transmission sets (*e.g.*  $\{A \rightarrow B, C \rightarrow D\}$ ) with each link using its independent best (greedy) antenna patterns, we find significant inter-link interference. The distribution of observed signal to interference ratios (SIRs) is shown in figure 1.3. The reference lines mark 10.5 and 26.5 dB, which are theoretical signal to noise (SNR) thresholds<sup>‡</sup> to achieve a bit error rate (BER) of  $10^{-6}$  using two common modulation schemes, BPSK and 64 QAM [Freeman 97]. Pairwise interference is sufficient to preclude BPSK and 64 QAM at this BER in 28% and 74% of cases, respectively.

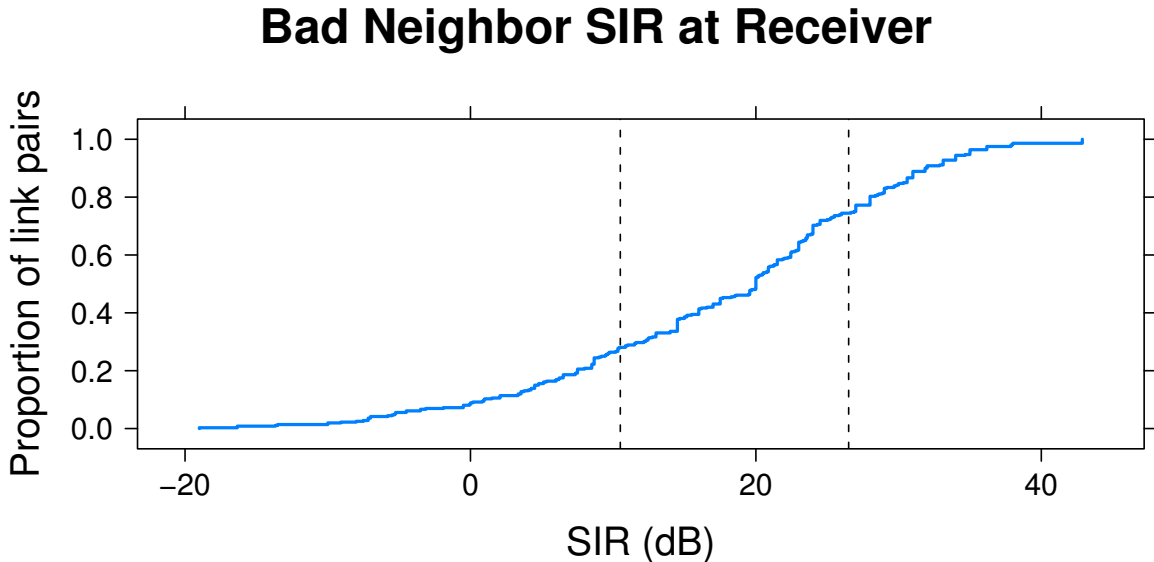


Figure 1.3: Interference between neighboring links when greedy antenna patterns are used. Reference lines show theoretical SNR values for  $10^{-6}$  BER with BPSK (10.5 dB) and 64-QAM (26.5 dB) modulation schemes.

This study also included sender-to-sender signal propagation. This is not directly relevant

<sup>†</sup> This work in particular was done in cooperation with Caleb Phillips.

<sup>‡</sup> These SNR thresholds are roughly comparable with SIR numbers, if the interfering signal is close to Gaussian noise and other sources of noise and interference are negligible.

to TDMA networks but is highly significant for CSMA systems. Figure 1.4 shows an empirical CDF of the power received from any link’s transmitter at the **transmitter** of another link, when both links are using greedy antenna configuration. Note that the cut-off at -95 dBm reflects the minimum signal strength our equipment was able to detect; the actual values could be anything  $\leq -95$  dBm. The reference line at -90 dBm indicates a plausible threshold for carrier detection<sup>§</sup>.

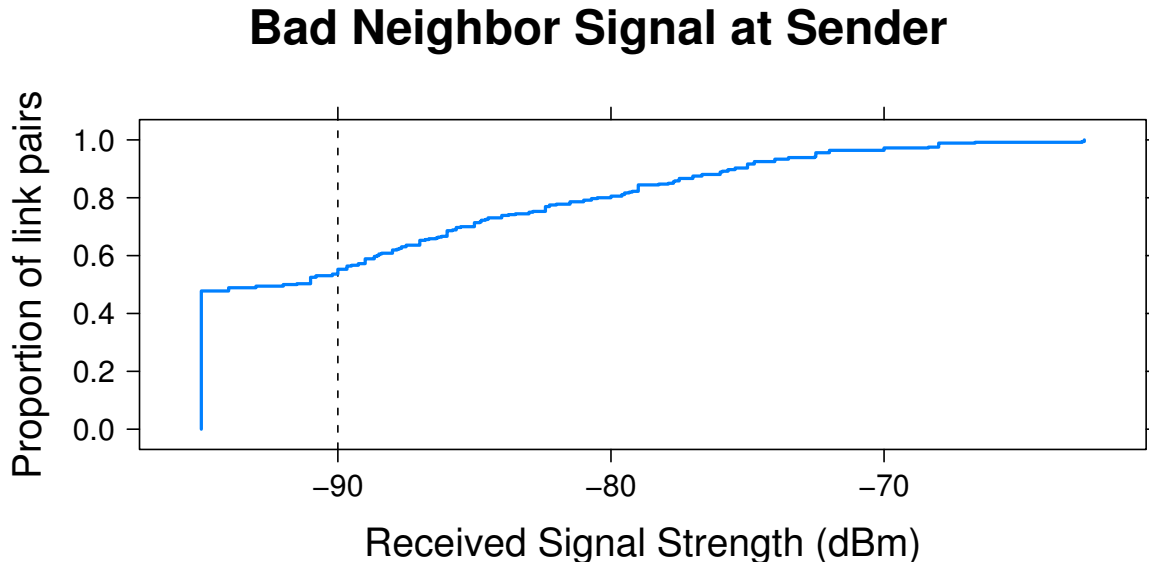


Figure 1.4: Sender-to-sender signal strength on neighboring links when greedy antenna patterns are used.

These sender-sender conflicts could – but in general do not – correspond to interference at the receiver side. This disconnect between the channel as sensed by the transmitter and the channel as experience by the receiver is one of the major reasons why I chose to explore TDM-style MACs rather than CSMA in this work.

### 1.3 Overview of Research

The purpose of this dissertation is to propose and evaluate algorithms for integrated scheduling and physical-layer beam selection, and to characterize the range of options for such integration.

<sup>§</sup> The lowest threshold **mandated** by the IEEE 802.11a,b,g specifications is -82 dBm [IEEE 99, §17.3.10.5] but many devices implement adaptive and/or tunable energy detection thresholds



Section 1.3.1 gives a brief statement of the problem. This is developed more fully in chapter 4.

### 1.3.1 The Joint Beam Selection and Scheduling Problem

The joint beam selection and scheduling problem is a proposed formalization of the problem this research is investigating. It is intended to make the objectives and assumptions more concrete. What follows is a “template” definition: Different solution approaches will formalize the objectives and constraints differently, but all are addressing the problem outlined here.

#### **Joint Beam Selection and Scheduling (JBSS):**

Assume:

- A set of stations, each of which has some possibly infinite set of possible physical-layer configurations.
- A propagation environment with characteristics specific to each combination of sender, sender configuration, receiver and receiver configuration.
- A one-hop link demand  $\geq 0$  for each (sender, receiver set) tuple. There are multiple ways of conceptualizing demand, among them: An infinite workload with relative priorities, a fixed set of resources which must be provided (e.g. rates which must be supported), a function mapping vectors of flow rates to aggregate utility, or a best-effort injection rate. This work will generally consider the first type.

Compute a **joint schedule** consisting of:

- A sequence of time slots having definite lengths.
- An assignment of which nodes may transmit to which other nodes during each slot.
- An assignment of a physical-layer radio configuration to each node for each slot.

Such that:

- The set of communications scheduled for any single slot has acceptable intra-set interference.

- The set of communications scheduled for any single slot meets any other applicable feasibility requirements, such as ensuring that nodes participate in only one concurrent link for each radio interface.
- The given demand is serviced appropriately, for the definition in use.

### 1.3.2 Purpose

This dissertation describes an approach to solving the antenna configuration and scheduling problems together, so that the antenna and scheduling decisions are appropriate for each other. I had originally considered two distinct approaches to integrating the two: **Combined decision space** in which a single decision process is run over the combined space of schedules and configurations, and **iterative refinement**, in which the two problems are considered in alternating phases. In practice, however, the approach developed is both: the process developed iteratively solves for antenna and scheduling components, but they are coupled in a way that the overall properties are well-defined with regard to the combined problem.

This research addresses to some extent the **systems** aspects of the problem as well as the mathematics. This means two things: First, the algorithms proposed are implemented in a deployed, running system, which is deployed on the test bed described in Appendix C on page 216. Second, “real world” aspects of the system are considered.

## 1.4 Definitions

This section provides definitions for terms that are used ambiguously in the literature. Within this document, the terms in Table 1.1 on the next page are used with the definitions given.

## 1.5 Organization

This dissertation is organized as follows: Chapter 2 discusses relevant prior work. Spatial-reuse scheduling with directional antennas is surveyed exhaustively, and salient work in related

<b>Term</b>	<b>Definition</b>
Configurable antenna	Any of the following:
Switched-beam antenna(s)	An antenna or set of antennas providing a finite set of gain patterns from which the user can select one at a time.
Steerable antenna	An antenna having a pattern that is fixed except for rotation, and can be rotated continuously in the azimuth and/or elevation planes. (An antenna that can be rotated only in discrete increments is effectively a switched-beam antenna)
Beam-forming antenna	An antenna having a pattern that can be varied <b>continuously</b> in real-time to optimize some signal property. Especially an antenna that uses pilot tones to maximize the SIR for one or more pre-determined stations.

Table 1.1: Definitions used in this document.

areas is discussed. In particular, spatial reuse generally (sections 2.1 and 2.2) and optimization decomposition (section 2.8 on page 40) are discussed significantly. Chapters 3 and 4 present the mathematical formulations and decomposition. Chapter 5 addresses systems aspects of the mathematical formulation. Finally, performance evaluation is presented in Chapter 6.

There are three appendices addressing research methods. Appendix A discusses modeling antenna effects in real environments, and Appendix B presents simulation methods based on the models developed in Appendix A. Appendix C discusses the Wide Area Radio Testbed which was built to support this work and other phased array antenna research. Finally, Appendix D gives the AMPL models for the optimization problems.

## Chapter 2

### Related Work

This dissertation builds on several areas of research in computer science, radio engineering, and mathematical optimization. The two most immediately related bodies of work are those on transmission scheduling and networking with directional antennas.

#### 2.1 TDMA and Spatial Reuse

One of the most basic medium access protocol ideas is Time Division Multiple Access (TDMA). The core notion is that time is divided into slots, and each slot is assigned exclusively to one transmitter [Hultberg 65, Aein 65]. Generally, consecutive slots are grouped into frames and every station with data to transmit is assigned one or more slots with each frame. This assignment is referred to as a **schedule**. In most cases, the schedule is fixed across a span of many frames [Schwartz 66, Wittman 67].

Time-Division Multiplexing (TDM) of logically separate data streams between the same physical nodes dates back to telegraphy. TDMA differs in that physically-separate stations share a common medium on a time-division basis. The earliest use of TDMA of which the author is aware is in the point-to-multipoint context of ground-to-satellite communication. In this context, many earth stations are attempting to communicate with a single orbiting satellite – or with each other, using the satellite as a repeater – and so the satellite’s radio interface is the primary scarce resource. This is largely the same situation faced by base station-based terrestrial networks, such as cellular telephony, WiFi, and WiMax, so long as a single base station and its associated clients

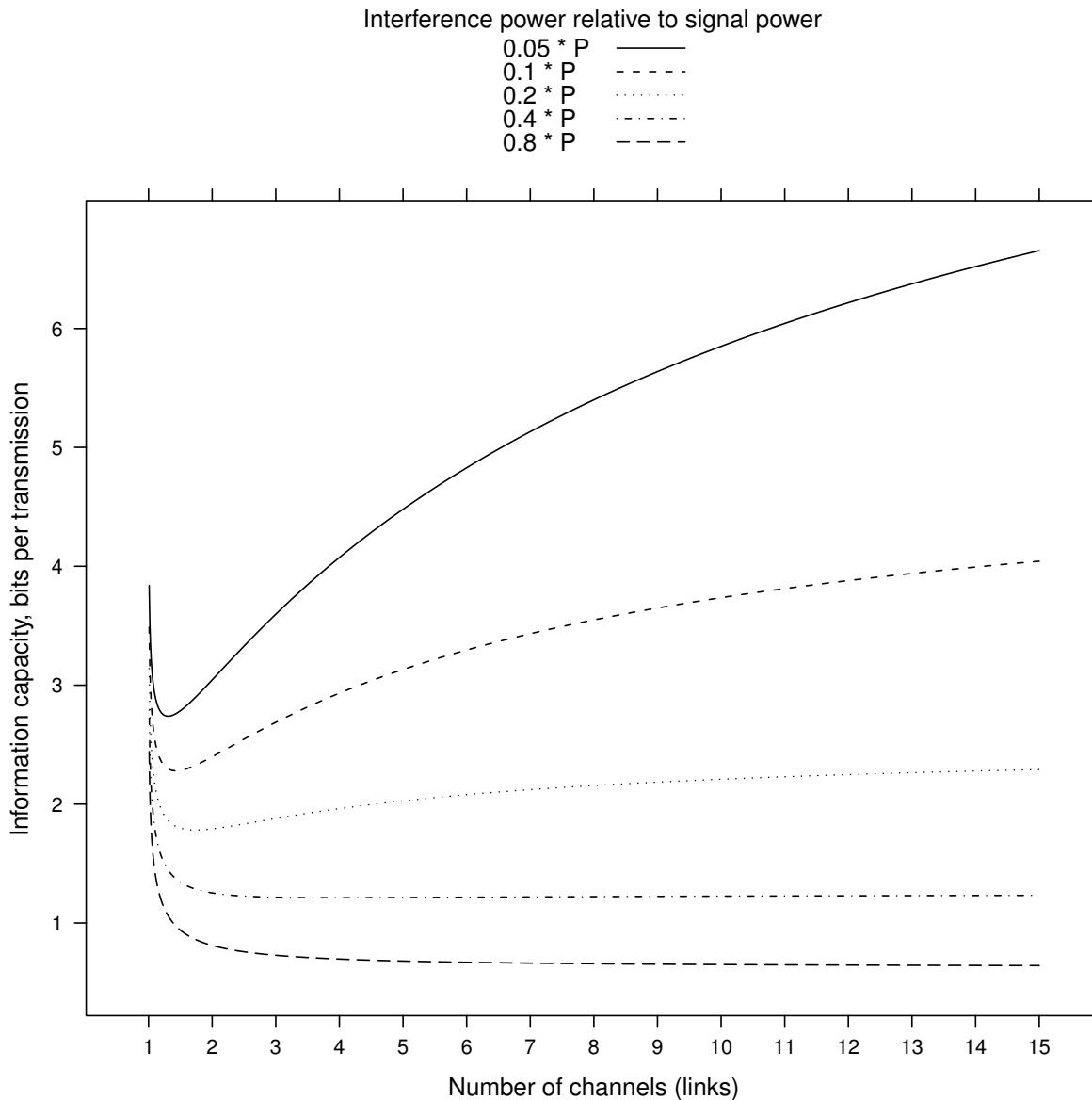
are considered in isolation.

Multipoint-to-multipoint communication is fundamentally different in that no single station is necessarily a bottleneck. Usable spectrum across the set of potential receivers is the primary scarce resource. So long as the receivers have sufficient separation – meaning difference in attenuation of signals from any given transmitter – it is possible for multiple concurrent transmissions to occur without the need for frequency or code division. Spatial-reuse TDMA (STDMA), originally proposed by Nelson and Kleinrock, is an extension to TDMA in which multiple transmitters can be assigned to any given time slot [Nelson 85]. Spatial reuse is fundamentally different from time, frequency, or code division multiple access in that it represents an **increase in channel capacity**, or perhaps a broader definition of the channel, while the others are all techniques for **subdividing a fixed channel capacity**.

It is generally unreasonable for all nodes which have data to communicate to transmit simultaneously. Theoretically, any combination of links is **possible**: If each link is regarded as a Gaussian channel, and all unwanted transmissions arriving at a receiver are assumed to be additive noise, each channel will have a non-zero information capacity. However, many combinations are very poor. Consider a set of links  $\mathfrak{L}_1$  having aggregate information capacity  $C_1$ . Let  $C_l = \frac{1}{2} \log(1 + \frac{P_l}{N_l})$  be the capacity of any link  $l$ , where  $P_l$  and  $N_l$  are the power constraint and noise variance of link  $l$  [Cover 91]. Then  $C_1 = \sum_{l \in \mathfrak{L}_1} C_l$ . Consider adding an additional link  $k$ . The received power from link  $k$ 's transmission at the receiver of every link  $l$  increases the noise variance  $N_l$  by some increment  $N_{kl}$ . This causes a loss of capacity due to interference  $I_{kl} = \frac{1}{2} \log(1 + \frac{P_l}{N_l}) - \frac{1}{2} \log(1 + \frac{P_l}{N_l + N_{kl}})$ . The links in  $\mathfrak{L}_1$  experience a total loss of capacity due to interference  $I_{k\mathfrak{L}_1} = \sum_{l \in \mathfrak{L}_1} I_{kl}$ . Let  $\mathfrak{L}_2$  be the set of links  $\mathfrak{L}_1 \cup \{k\}$  formed by adding  $k$ . Let  $C_k$  be the information capacity of  $k$ , given the noise from all the other links in  $\mathfrak{L}_2$ . The aggregate capacity of  $\mathfrak{L}_2$ ,  $C_2$ , is given by  $C_2 = C_1 + C_k - I_{k\mathfrak{L}_1}$ . Note that  $I_{k\mathfrak{L}_1}$  can easily be greater than  $C_k$ , in which case adding link  $k$  results in a **reduction** in capacity. Note also that as the number of links in  $\mathfrak{L}_1$  increases,  $I_{k\mathfrak{L}_1}$  increases because it is summed over more links, and  $C_k$  decreases because the noise variance  $N_k$  is also summed over more links.

Figure 2.1 shows the effect of spatial reuse in a Gaussian channel and a very simplified scenario

### Aggregate capacity of $n$ interacting interference-limited Gaussian channels



This assumes that every link is a Gaussian channel with capacity given by  $C = \frac{1}{2} \log(1 + \frac{P}{N})$ , where each link creates interference power of  $k * P$  at every other link. Each plotted line shows the aggregate capacity as a function of the number of links for some specific  $k$ .

Figure 2.1: Aggregate capacity of interference-limited Gaussian channels.

in which every link creates the same level of Additive White Gaussian Noise (AWGN) interference for every other link. The scenario is artificial, but it illustrates a few key points: First, even looking at an ideal upper bound with no real-world implementation limitations, there is a diminishing return as the number of links grows, and the maximum capacity can be reached with a relatively small number of links. Second, the ratio of any given transmitter’s power received at the intended destination (signal) to that received at other destinations (interference) determines the maximum aggregate capacity possible. Put differently, the capacity of a set of concurrently-transmitting links depends on the level of RF separation between the links. Guo *et al.* present an analytical model of minimal inter-transmitter spacing for re-use [Guo 03]. When real-world limitations, such as limited modulation options, packet error rate requirements, or minimum flow rates are included, many concurrent link groups become not just inefficient but impossible.

## 2.2 Transmission Scheduling for STDMA

The problem of **transmission scheduling** or **link scheduling** is to choose a sequence of **link sets** such that the links in each set can operate concurrently with acceptably low interference, and the overall sequence meets some predetermined objective. For the purposes of this dissertation, I am concerned with explicit, pre-computed schedules assigning nodes or links to specific time slots. This is in contrast to on-demand contention resolution procedures (including CSMA) which are sometimes described as scheduling. The primary difference between scheduling for ordinary TDMA and STDMA is the complexity of choosing concurrent link sets. Without spatial re-use, the set of such sets is given: Every transmitter (or link) with data to send is a set. With spatial re-use, the possible link sets are the power set of the set of links, so for  $m$  links, there are  $2^m$  possible link sets. Depending on the node degree distribution, for a strongly-connected network with  $n$  nodes,  $n \leq m \leq n^2$ , so the asymptotic complexity of the number of possible link sets is between  $O(2^n)$  and  $O(2^{n^2})$ . In general, identifying the **best** sets is NP-hard, and determining whether a given set of end-to-end flow rates is feasible is NP-complete. Arikian gives a reduction from CLIQUE to the  $\vec{f}$ -feasibility problem in [Arikian 84]. The scheduling problem **without interference**, however, can



be solved in polynomial time [Hajek 88]. Ephremides shows that optimally scheduling **broadcasts**, as opposed to links, is still NP-hard [Ephremides 90]. Sharma *et al.* show that for a simplified K-hop exclusion interference model (where Hajek effectively studied the  $K = 1$  case), optimal scheduling is NP-hard for  $K > 1$  [Sharma 06].

It is worth noting that spatial-reuse scheduling can be done with frequency or code division as well as time division. Although the problems have very similar conceptual structure, they have different implementations [Wittman 67, Ramanathan 97]. This dissertation is focused on time division because it is easier to implement and understand a system which changes antenna characteristics between time slots than one which simultaneously has controllably different characteristics for different frequency bands or different codes. It is conceivable that a phased-array antenna with broadband sampling of every element and digital beam-forming as described in [Godara 04] could duplicate the samples, filter the copies by frequency band or code, and perform separate frequency-domain beam-forming for each stream. Such functions are beyond the capabilities of the hardware employed in this research, and FDMA and CDMA will not be addressed further.

There have been several STDMA-like suggestions in which there is no global scheduling process, but local time-division rules allow spatial reuse: A virtual-circuit establishment algorithm for something very much like STDMA was suggested in [Pond 89]. The interference model is minimal, but no two stations which can communicate with each other can assign themselves the same slot. In [Das 07] the authors propose local-neighborhood priority queueing: The station with the longest queue within a (presumed) interference region gets the channel. This is deemed to be safe only because a prior admission-control layer prevents end-user stations from injecting traffic above their (feasible) allowed rates. A similar technique is explored in [Warrier 08, Warrier 09], which use multiple-priority CSMA/CA to achieve a similar effect. Both of the preceding are implementations of **differential backlog based backpressure** ( $\pi_0$ ) from [Tassiulas 92]. Another contention-resolution process for spatial reuse is given in [Bao 03].

There are three main tasks in creating a spatial re-use schedule:

- (1) Identifying good sets of concurrently-useable links. This accounts for the vast majority of the computational difficulty, and is the main source of difference between STDMA scheduling approaches. This is also the aspect which is primarily responsible for the interaction between scheduling and antenna configuration.
- (2) Choosing how much time to allocate to each set. This can be 0, and probably will be for almost all sets. So long as the overall utility of the network is defined as a linear function of the various flow rates achieved, this reduces to a linear programming problem.
- (3) Choosing the order in which link sets are activated. This is closely-related to scheduling as understood in the wired network and operating system contexts. It has a significant impact on the quality of service (QoS) properties of a network [Zhu 98, Fattah 02, Liao 02, Wallin 03, Kozat 04, Luo 04, Salonidis 04, Salonidis 05, Rangnekar 06, Zou 06a, Zou 06b, Djukic 07b, Djukic 08, Zhang 08] but little on the long-run aggregate throughput. This interacts with antenna configuration only to the extent that some orders may involve more reconfigurations than others. [Liu 01] shows that very fine-grained scheduling can increase performance by letting users claim time slots when their conditions are favorable. A game-theoretic analysis of how often users should test the channel and when they should choose to claim it is given in [Zheng 07].
- (4) Choosing the duration of time allocations. Abstractly, 1 interval of 1 second every 10 seconds 1 ms every 10 ms have the same capacity, but they have very different latency and responsiveness. Due to the effect of delay on TCP and similar congestion control mechanisms, even the capacity – as actually realized – will differ.

The first task is at the core of the work in this dissertation. The way one regards interference largely determines the options for addressing task. A **boolean** approach assumes that for each link there is some threshold of interference below which the link is usable and above which it is not. This effectively corresponds to assuming a fixed signal modulation and some packet error rate beyond which the link isn't worth using. Under a binary model of interference, the best

sets of links are the **maximal** sets, that is those which activate as many links as possible without violating any link's interference constraints. Boolean conflict models can further be subdivided into ones which consider only **pairwise** interference and those which consider **cumulative** interference. A **continuous** approach, on the other hand, regards throughput (or goodput) as a continuous function of the signal strength and interference level. This corresponds to assuming that a link can choose modulation and coding schemes to take advantage of whatever SINR is available, as in the Gaussian channel information capacity discussion above. Many real systems (such as the **802.11 phy** layers) fall in between these two cases, having a finite set of modulation and coding options to choose from. Figure 2.2 shows a classification tree of the interference models used in scheduling research.

### 2.2.1 Pairwise link conflict models

Pairwise conflict models consider interference between pairs of links. A pairwise conflict exists between two links if they cannot both operate simultaneously, assuming that no other transmissions are occurring. Conflict determinations are generally based on the strengths of the intended and interfering signals, or on empirical evidence of interference-based packet loss. In some of the simpler models, conflicts are assumed based on geographic position or minimum path length. When actual propagated signal strength is measured, conflict can be defined in terms of pure SINR or in terms of protocol-specific behaviors. In general, define a SINR requirement for a link  $ij$  as:

$$\frac{\text{received signal from } i \text{ at } j}{\text{noise at } j + \text{interference at } j} \geq \text{threshold } \gamma_1$$

Let  $P_i$  denote the transmit power of node  $i$ ,  $Lb(i, j)$  denote the path loss between nodes  $i$  and  $j$ ,  $N_j$  denote the receiver noise figure at node  $j$ , and  $\gamma_1$  denote the requisite SINR (see table 3.1 on page 65). Simplifying for links  $ij$  and  $kl$ , the preceding inequality becomes:

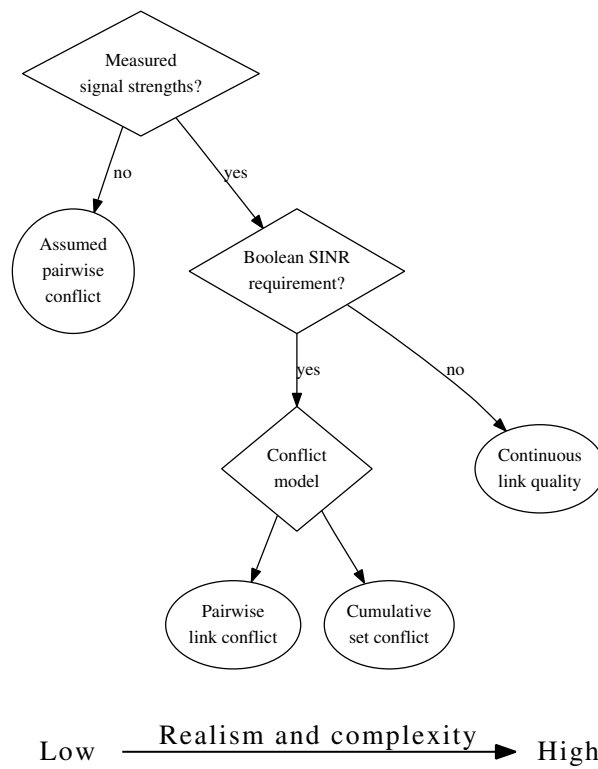


Figure 2.2: Classification tree of interference models used in spatial-reuse scheduling.

$$\frac{\frac{P_i}{Lb(i,j)}}{N_j + \frac{P_k}{Lb(k,j)}} \geq \gamma_1 \quad (2.1)$$

$$\frac{\frac{P_k}{Lb(k,l)}}{N_l + \frac{P_i}{Lb(i,l)}} \geq \gamma_1 \quad (2.2)$$

Some studies add additional pairwise constraints based on CSMA/CA or 802.11 specifically. These generally require that the received power of one transmission at the other **transmitter** be below the threshold required to trigger backoff. Similar requirements relating to RTS/CTS mechanisms may also be considered.

The primary advantage to pairwise conflict models is computational simplicity. Because they are computed over the set of link pairs, having cardinality  $\frac{1}{2}m^2$ , where  $m$  is the number of links, these models do not incur the exponential computational complexity discussed at the beginning of Section 2.2. Because  $\frac{1}{2}n \leq m \leq \frac{1}{2}n^2$ , where  $n$  is the number of nodes, the size of the link-pair set is between  $O(n^2)$  and  $O(n^4)$ . This in turn means that it is frequently feasible to enumerate all of the links and conflicts and use normal graph algorithms to partition the set into conflict-free sets.

The disadvantage is that the cumulative interference from multiple links is not considered. To the extent that interfering signals can be modelled as independent Gaussian processes, interference is additive. Consider the spatial re-use model in Figure 2.1. The leveling-off in information capacity reflects the additive effect of interference. A model which lacks this effect will predict that link groups can grow arbitrarily large while maintaining a linear growth in capacity. Interfering signals are not necessarily actually independent Gaussian processes, but that has been shown to be a good model for at least M-QAM and spread-spectrum CDMA, as discussed in Rappaport, appendix E [Rappaport 01] and Freeman, section 13.7 [Freeman 97].

Pairwise link-conflict models are used in the following papers:

- Scheduling: [Behzad 07], [Chen 06] (the distributed algorithm), [Chlamtac 87], [Das 07] (which uses a 3-hop neighborhood pairwise model), [Djukic 07a], [Djukic 07b], [Koutsonikolas 07], [Kodialam 03], [Kodialam 05], [Sharma 07], [Luo 00],[Salem 05],

[Kozat 04], [Rhee 06, Rhee 09], [Pond 89] and [Lal 04a] (which use a 1-hop neighborhood model), and [Bao 01]. The performance bounds of greedy pairwise algorithms are discussed in [Wu 07].

- Channel assignment:

[Alicherry 05] [Kodialam 05] [Ramachandran 06] [Villegas 05] [Mishra 05] [Mishra 06a] [Mishra 06b].

- Routing:

[Alicherry 05] [Awerbuch 04] [auf der Heide 02] [Kodialam 03] [Kodialam 05] [Wan 01].

- Topology control: [Huang 02] [Li 05b] [Ramanathan 00] [Wattenhofer 03].

- Analysis: [Garetto 05], [Kyasanur 05b]. For the assumption of “**primary**” – pairwise and non-interference-aware – conflict, [Brzezinski 08] provides a characterization of the topologies in which greedy distributed algorithms can be optimal.

Early graph-based algorithms using only local information were proposed by Ephremides and Truong [Ephremides 90]. Shor and Robertazzi extended the same ideas to be traffic-sensitive, that is, to consider link load [Shor 93]. The well-known RAND algorithm for node (rather than edge) scheduling uses a  $k$ -hop interference model for  $k = 2$  [Ramanathan 97, Ramanathan 99]. An equivalent distributed protocol, DRAND, is presented in [Rhee 06, Rhee 09]. Ju and Cai propose two theoretically interesting approaches to **topology-transparent** STDMA scheduling [Ju 98, Ju 99, Cai 03]. They rely on simplified graph models of propagation and interference: Nodes are either neighbors or they are not, and a conflict occurs if and only if two neighbors attempt to communicate (other than with each other) in the same time slot on the same channel. The algorithms proposed use the maximum degree of the graph and group-theoretic techniques to produce a schedule such that every node is assigned **at least one** slot which is not assigned to any neighbor. This guarantee is immune to graph changes so long as the maximum degree does not increase, but the performance cost of this topology-independence is unknown as the authors do

not compare their results with topology-aware techniques. Their work is based on [Chlamtac 94] and [Ju 98], which share similar properties. This work is extended in [Oikonomou 04] to consider letting nodes probabilistically “steal” time slots not assigned to them.

Genetic algorithm-based scheduling is proposed in [Chakraborty 04]. It is interesting in that the author introduces feasibility-preserving mutation and crossover operations specific to STDMA scheduling, and in that the algorithm converges on a solution in a modest number of generations. Unfortunately, the paper uses a very simplistic graph-based model of interference, and the performance results are not compared against any other techniques, so it is difficult to draw any conclusions.

An empirical comparison of graph-based (pairwise-conflict) and interference-based (cumulative-conflict) scheduling is given in [Grönkvist 01]. Behzad revisits this in [Behzad 03]. Balasundaram provides a survey of uses of graph-theoretic algorithms in networking, including wireless, in [Balasundaram 06].

### 2.2.2 Aggregate Interference Models

Aggregate interference models consider the combined effect of interference from all active links on all active links. This is significantly closer to reality than pairwise conflicts, but also much more computationally difficult. Where  $\mathcal{L}$  denotes a set of concurrently active links, let  $(i, j) \in \mathcal{L}$  be the transmitter and receiver of a given link. Using the notation of [Björklund 03], an aggregate interference constraint for  $(i, j)$  would be of the form:

$$SINR(i, j) = \frac{P_i}{Lb(i, j)(Nr_j + \sum_{k \neq i, j | (k, l) \in \mathcal{L}} \frac{P_k}{Lb(k, j)})} \geq \gamma_1 \quad (2.3)$$

This is essentially equivalent to equation (2.1), except that the path-loss term  $Lb(i, j)$  has been reorganized to the denominator, and that the single interference term  $\frac{P_k}{Lb(k, j)}$  is replaced with a summation over all transmitting nodes which are not  $i$  or  $j$ .

This is the interference model used in most of the recent STDMA scheduling literature. Brar gives an algorithm **GreedyPhysical**, which appears to be identical to Grönkvist’s centralized

algorithm, and proves an approximation factor  $\ll O(n \log n)$ .

Interestingly, Gore combines the pairwise-conflict model with aggregate-interference-based criteria [Gore 07]. In particular, it uses a graph-coloring algorithm in which new links are given the “first conflict-free color.”

### 2.2.3 Continuous Link Quality

A continuous model of link quality as a function of interference is the most realistic and general, but also the most complicated. As discussed in Section 2.1, the information-theoretic channel capacity is a continuous function of the signal power limit and noise. If the modulation scheme and power level are fixed, the bit error rate (BER) will be a function from the interference level to the **open** interval  $(0,1)$ . In practice, for any given modulation scheme, there will be a minimum SINR below which real hardware and protocols fail to recognize the existence of a link and so the effective BER is 1. On the other hand, there is no SINR high enough for the BER to actually reach 0. In addition to throughput increasing as a result of diminishing BER on a given modulation, better SINR generally allows more aggressive modulation (more bits per symbol and/or more symbols per second), and so the practically-achievable throughput approximates the theoretical capacity.

It is reasonable to regard scheduling as an optimization problem. Regardless of whether or not one uses an algorithm that is directly rooted in mathematical programming, it is a useful conceptual framework. The goal is to maximize (or minimize) some objective without violating some set of constraints. Using a continuous model of link quality rather than a quality threshold necessarily makes the interference model part of the objective rather than (or as well as) part of the constraints. This implies the existence of a function  $M(\cdot)$  mapping the vector of link qualities  $\vec{q}$  to a vector of capacities or rates  $\vec{c}$ , and a utility function  $U(\cdot)$  mapping vectors of rates to real numbers.  $M$  is given by the properties of any particular communication system, and  $U$  reflects the design objectives of the network. There is no unique correct utility function, but “efficient-but-fair” allocation tends to require a sub-linear function of rate, such as the logarithmic function show in



equation 2.4 [Boche 05, Chiang 05a, Soldati 06].

$$U(\vec{c}) = \sum_i \log(c_i) \tag{2.4}$$

A linear utility function (corresponding to Kaldor-Hicks efficiency) will produce the maximum aggregate throughput, but can easily produce starvation. Consider two links such that increasing the power or time allocated to either necessarily diminishes the throughput of the other. Under a linear utility function, unless the marginal rate of substitution between the two links is exactly one, the highest utility outcome will be to assign **all** of the resources to whichever link has a higher rate per unit of resources. Conversely, a strictly fair utility function (corresponding to Pareto efficiency) such as max-min fairness will slow the entire network to the rate of the slowest link, because it will assign marginal resources to the slowest link, no matter how minor the gain to that link or how great the loss to the other links. Radunović refers to this as the “solidarity” property and observes that it exists any time the capacity region is such that flow rates are fungible [Radunović 04b]. This occurs when the limiting factor on multiple flows is a shared resource (e.g., a shared wired link or shared RF spectrum) which can be flexibly re-assigned. It does not occur when different flows have different bottlenecks, and so reducing the resources allocated to one does not benefit others. Good discussions of fairness and utility in rate allocation in general are found in [Kelly 98, Massoulié 02, La 02, Briscoe 07, Zukerman 08]; wireless networks specifically are discussed in [Huang 01, Tan 05, Eryilmaz 06, Boche 07].

Note that a number of proposals use an explicit  $U(\vec{r})$  objective, but still use a simpler interference model (e.g., [Chen 06] is based on pair-wise conflicts). The work in [Eryilmaz 06] discusses the integration of **back-pressure scheduling** [Tassiulas 92] with congestion control and routing in the context of wireless dynamics. This is in principle open to sophisticated interference models in that the process of identifying compatible **activation sets** is left open, as is the mechanism for finding rate vectors meeting the stated objectives. Eryilmaz and Srikant note in particular that they do not consider distributed scheduling.

Some work exists on routing and TDMA scheduling in **wide-band** channels using a contin-

uous quality model [Radunović 04a]. The information capacity of a wide-band AWGN channel is a **linear** function of the SINR, rather than logarithmic as is the case for narrowband channels. Combined with the assumption of continuously-variable coding, this leads to properties distinctly different from the other systems considered.

Two papers by Zhu and Corson discuss the **protocol** aspects of STDMA scheduling [Zhu 01b, Zhu 01a].

## 2.3 STDMA with Antenna Considerations

This is the set of research closest to this dissertation. None of these closely integrate the antenna configuration with the scheduling process, nor give serious consideration to decisions in that space.

### 2.3.1 Opportunistic Antenna Reconfiguration

A minimal level of integration is to perform scheduling with no assumption of antenna reconfigurability, and have a separate process configure the antennas for whatever sets of nodes end up being active together.

Jorswieck gives analytical models of the potential value of opportunistic beamforming as a function of the distribution of the stations given [Jorswieck 07]. This does not explicitly address scheduling, but it establishes desirable properties for a set of stations to have.

### 2.3.2 Scheduling Based on Assumed Antenna Capabilities

One set of papers assumes idealized high-level effects of using directional antennas, rather than deal with the actual RF gains of specific antenna configurations. Such approaches are computationally much easier, but the assumptions are often incorrect. Cain *et al.* assume that their antennas have an effectively perfect directionality (“very narrow or zero beamwidth”) and therefore there will be no interference [Cain 03]. They then apply a graph-coloring based scheduling algorithm based on that by Ma and Lloyd [Ma 98], with only the constraint that each node may

have only one link active at a given time. (See also [Liu 98, Lloyd 02] for more discussion of the scheduling protocol).

Sundaresan *et al.* present a scheduling approach for adaptive arrays based on their “degrees of freedom” (DoF) [Sundaresan 06]. For a  $K$ -element array, it is possible to define  $K - 1$  positions having distinct relative power levels, so it is said to have  $K - 1$  degrees of freedom. If one DoF is used to specify the main beam direction (look direction) for the communicating partner, that leaves  $K - 2$  DoF which can be allocated to suppressing interference [Godara 04, section 2.4]. If there is no requirement of a look direction, all  $K - 1$  DoF can be allocated to creating nulls, but then there is no assurance that the intended communication partner will have good (or even any) gain. The Sundaresan algorithms are based on a pairwise interference model, with the extension that each node is assumed to be able to “null out”  $K - 1$  interfering links. This model is a substantial simplification:  $K - 1$  DoF only allows the gain to be independently controlled in  $K - 1$  only for very specific sets of directions: The directions must induce mutually orthogonal antenna vectors for the arrays in question [Wirth 01, section 10.1], which is equivalent to saying that each direction must lie on a zero (absolute null) of all of the antenna patterns the array would produce if it were beamforming toward any of the other directions. For general directions, it is possible to **reduce** or **increase** the gain in  $K - 1$  directions, but the magnitude of the change may or may not be significant. Additionally, in the presence of multi-path propagation, any given interference source may produce signals arriving from several discrete directions and/or spread across a continuous range of directions.

ROMA is based on the assumption that interference can be predicted using simple spatial rules [Bao 02b]. Specifically, it is assumed that side-lobes are inconsequential, that each node can form (and use) to some  $K$  main lobes concurrently, and that main-lobe gain is effectively zero at an angular distance of one half-power beamwidth from the center of the lobe. Based on these assumptions, a geometric model is used to produce pairwise conflict information. A related protocol, but without directional antenna support, is presented in [Coupechoux 05].

The TDMA MAC proposed in [Deopura 07] is very much like ROMA. A graph of pair-wise

conflict between links is produced based on geometrical assumptions about the coverage area of each antenna pattern. The antenna patterns for any given link are assumed to be given (simplistic per-link antenna optimization).

The collision-free MAC in [Lin 04] uses a very light-weight scheduling algorithm in which interference is entirely ignored. The set of all possible links is pruned to make a planar graph, and then partitioned so that every node has at most one communication partner during each interval. Nodes are thus able to steer (or otherwise configure) their antennas for the appropriate link, but there is no particular assurance that the set of simultaneously-active links will be compatible from an interference perspective.

### 2.3.3 Scheduling Based on Per-Link Antenna Optimization

These papers assume that the antennas involved in each link are configured for that link in isolation. This is the case described in the introduction (section 1.2).

Sundaresan *et al.* present a scheduling and conflict model based on pairwise link conflicts. They present a framework which allows conflicts (and resource constraints) to be identified in different ways for different antenna technologies, and then uses consistent representations and algorithms across all of them [Sundaresan 04, Sundaresan 07]. Graph-based algorithms are used to identify non-conflicting groups of links. The conflict-identification phase assumes per-link antenna configuration for steerable/switched antennas, but allows **pairs** of links to be configured jointly for adaptive antennas.

#### 2.3.3.1 Grönkvist Algorithm Extensions

A series of papers, mostly by Marvin Sánchez Garache and Karin Dyberg, have examined using smart antennas with variants of the Grönkvist algorithm. They all share the same basic approach to antenna-scheduling integration, although they consider a wide variety of other factors. The first of these is [Sánchez 99]. This considers spatial TDMA with the centralized Grönkvist algorithm and 4-element circular array antennas on all the nodes. The original algorithm is enhanced by

pre-configuring the antennas in each link to achieve maximum gain to their communicating partners (Equation (3) in that paper), and using those patterns in determining link-set admissibility.

A more thorough evaluation of the same algorithm is given in [Dyberg 02a] and its corresponding methodology report, [Dyberg 02b]. That work examines the effects of varying the beam-forming strategy, the number of antenna elements, and the terrain. The beam-forming options considered were: Sender and receiver use isotropic antennas; sender uses isotropic antenna, receiver uses adaptive beamforming; sender uses beam steering and receiver uses adaptive beamforming; and sender and receiver both use beam steering. Note that adaptive beamforming by the sender is not considered, and the effects of adaptive beamforming are not fed back into the STDMA algorithm. Both of these reflect the fact that receiver beamforming for actual reference signals (wanted and interference) is relatively easy, but beamforming for hypothetical scenarios is not. The adaptive algorithm used (in section 5.5.2) is based on minimizing the error between observed beamformer output and pre-determined reference signal.

Sánchez introduces a cross-layer routing and scheduling approach for STDMA with smart antennas in [Sánchez 02c] and [Sánchez 02b]. The STDMA algorithm and the interaction between scheduling and antenna configuration are not described in detail but appear to be the same as in the previous work. The novelty in this work is the routing-scheduling interaction, which provides one model for non-formal cross-layer optimization: The joint routing and scheduling is based on a two-pass interleaving. In the first pass, one set of routes  $r_1$  is computed on the graph of all possible links, and then a schedule is created based on the traffic load resulting from those routes. In the second pass, a new set of routes  $r_2$  is computed on the graph of links defined by the first-pass schedule, and then a second schedule is computed based on the new traffic loads. Whichever route/schedule combination has the best predicted performance is the one actually used. In his Licentiate thesis, Sánchez presents the same algorithm in more detail, along with CSMA/CA analysis and deployment-scenario evaluation similar to that of Dyberg [Sánchez 02a].

The dissertation [Garache 08] presents extensive analysis on (Generalized) STDMA, including: Integration with routing, effects of variable modulation rates, and the effects of varying beam

widths. The integration with beam forming is essentially the same as in the previous papers: Each link is configured in isolation, and the gain resulting from these configurations are used in the scheduling process. (Figure 5.5 in [Garache 08]). The algorithm is extended to incorporate power control (Figure 6.3) and variable modulation (Figure 8.1). Optimization-based scheduling – including power control and rate selection, **but not antenna configuration** – is compared with the heuristic algorithms. The effect of antenna configuration ( $F_{ml}$  in equations 7.11, 7.18, and others) is taken as a fixed input to the scheduling process. The optimization process used is that of [Johansson 06]. Much of the same information is given in condensed form in [Sánchez 07]. The optimization work, which the authors refer to as Joint Routing, Resource Allocation, and Scheduling (JRRAS) is also presented in [Xiao 04, Soldati 04, Soldati 08].

## 2.4 Scheduling Integrated with Other Network Properties

There is a variety of research on “scheduling-plus- $x$ ,” where  $x$  is some other controllable property of the network. This section may overlap somewhat with the discussion of cross-layer optimization generally in section 2.8.

### 2.4.1 Lower Layers

After explicitly combining scheduling with antenna control, the most nearly-related work is that which combines scheduling with other physical-layer controls. Channel (or code, or subcarrier) assignment and transmit power control seem to be the most widely-researched aspects of physical-layer reconfiguration.

#### 2.4.1.1 Channel Assignment

Channel assignment and scheduling are very closely-related problems: A frequency band, time slice, or code assignment can be regarded as defining a “logical channel,” such that two activities interact if and only if they occur in the same logical channel. Then, the generic assignment problem consists of selecting sets of transmissions which can proceed concurrently. In fact, several techniques

have been designed to be channelization-aspect-agnostic [Chlamtac 87, Ramanathan 99, Luo 04]. The abstraction is not perfect: Frequency- and code-division channels are less orthogonal than time-division ones. Frequency bands and codes must generally be assigned from a finite pool in fixed quanta, while time can often be divided more flexibly. A node which is active in a given time slot can generally observe others, while the same cannot be said of frequencies. For precisely the same reason, a frequency-division channelization may allow truly continuous operation while time-division does not [Wittman 67].

Because of these similarities, the mechanisms for slot-plus-channel assignment are essentially the same as those for pure slot assignment (scheduling). Examples of time-and-frequency scheduling are [Lin 07], [Liu 07], [Deb 08], and [Wang 08].

#### 2.4.1.2 Power Control

A large number of papers discuss integrating scheduling with power control. This is fundamentally different from integration with beam steering because the effects of power control are **linear and uniform**, making it a much simpler problem. It is, however, possible that similar conceptual approaches may be useful. For example, [Madan 05, Madan 06] iterates between successive adjustments to the scheduling process and the physical-layer (power) control process, which is one of the strategies this work discusses for beam selection. Other papers which may be addressed in the dissertation but do not require further discussion for now include: [Behzad 07], [Cruz 02], [ElBatt 02a], [Jarquín 02], [Kozat 04], [Miao 06], [Radunović 03], [Radunović 04a], [Soldati 06], [Soldati 08], [Moscibroda 06], [Somarriba 03], [Somarriba 07], [Sommariba 04], and [Yu 04].

#### 2.4.2 Higher Layers

There is a significant body of work on integrating scheduling with routing and/or congestion control. There are conceptual parallels in the approaches to cross-layer coupling, but the problems are distinctly different from those I consider here. My work assumes that routing is given – the per-link loads have already been established – but it could reasonably be integrated with existing

joint scheduling and routing approaches.

Some examples of joining scheduling with higher-layer concerns are listed here, but they will not be discussed further. Routing:

[Bonuccelli 04], [Chen 06], [Chen 05], [Eryilmaz 06], [Kodialam 03], [Lin 07], [Radunović 03], [Radunović 04a], [Radunović 04a], [Sánchez 02c, Sánchez 02b], [Sommariba 04], [Wang 08], [Yu 04].  
Congestion and admission control: [Chen 06], [Chen 05], [Eryilmaz 06], [Lin 06b], [Sharma 07], [Soldati 06], [Soldati 08].

## 2.5 Reconfigurable Antennas in Un-Scheduled Networks

This section addresses salient work in using reconfigurable antennas without a scheduling process. This is primarily concerned with random-access MAC protocols such as CSMA/CA.

### 2.5.1 Random Access One-Hop

Ward and Compton propose a mechanism for adaptive antenna use in random-access one-hop networks in which CSMA is infeasible because the client nodes cannot all hear each other [Ward 92]. Their approach, like that subsequently adopted by [Singh 05], is based on the base station reactively beamforming toward received signals, thereby strengthening the power capture effect and reducing the likelihood that a collision will result in packet loss. The authors propose an “acquisition preamble,” which is in practice very similar to the PLCP preamble used in 802.11, to help the base station recognize and quickly beamform for incoming packets. Directional antennas for 802.11 hot spots are considered in [Otyakmaz 04].

“Interference Mitigation in WLAN networks using client-based smart antennas” [Desautel 02] is actually MAC-agnostic. It discusses a low cost two-element beamformer intended for use in WLAN clients. The paper evaluates its interference rejection capabilities based on different channel models.



### 2.5.2 Random Access Packet Relay

The use of steerable antennas as a means of avoiding interference, particularly jamming, in multi-hop networks was proposed by Sussman in [Sussman 80]. The Rapidly Deployable Radio Network (RDRN) [Evans 99] is an early prototype system. The RDRN is neither scheduled nor random-access; the architecture is based on dedicated point-to-point links between all stations. The combination of beam steering (with multiple independent beams at each station) and channel assignment is used to achieve spatial reuse.

Comparison of switched-beam and adaptive beamforming, assuming CSMA/CA with RTS-CTS: [Radhakrishnan 02]. Longer version: [Lal 04b].

The original article on the capacity improvement from using directional antennas [Zander 90] follows the reasoning of [Takagi 84]. Zander argues that the optimal connectivity increases linearly with the antenna gain and the expected forward progress per hop increases as roughly the square root of the gain.

It has been shown that, under the assumption of no traffic locality, the expected end-to-end capacity of a multihop packet relay network grows sub-linearly in the number of nodes, meaning that per-node capacity approaches 0 as the number of nodes goes to infinity [Gupta 00]. Two recent articles evaluate the effect of directional antennas on the asymptotic capacity of relay networks [Yi 03, Peraki 03]. Yi *et al.* show a constant improvement in capacity  $O(\frac{1}{\beta^2})$ , where  $\beta$  is the beam width. The constants change depending on whether one is considering a random or worst-case topology. As a constant-factor improvement, these results don't change the asymptotic result. Peraki *et al.* show an asymptotic improvement, but only with the assumption that nodes can send and decode multiple concurrent beams, as long as their communicating partners are not co-linear. The total improvement is from  $\Theta(\sqrt{n/\log n})$  to  $\Theta(\sqrt{n \log n})$ . With the additional assumption that nodes can generate **arbitrarily precise** beams (but with bounded power), the limit becomes  $\Theta(\sqrt{n \log^{\frac{3}{2}} n})$  [Peraki 03, section 4.4]. With unbounded power density and arbitrarily precise beams, the capacity grows linearly with the network size, but each node must be able to generate

at least  $\Theta\left(n^{\frac{1}{3}}\right)$  resolvable beams.

### 2.5.3 MAC Protocols for Directional Antennas

An early proposal for integrating steerable or sectored antennas with CSMA/CA is given in [Ko 00]. The authors compare the use of directional and omnidirectional antennas for the RTS-CTS exchange, with the goal of establishing only the necessary region of exclusion for other nodes. The findings support the use of directional RTS and omnidirectional CTS.

The WACnet MAC is essentially direction-aware CSMA/CA [Bandyopadhyay 01]. Communicating nodes beamform towards each other, and nodes not involved null-steer toward active transmitters based on RTS/CTS messages. Assuming that null-steering is successful, those nodes are then free to send or receive their own data.

Smart-ALOHA is a CSMA-based MAC for adaptive array antennas. It relies on the receiving node being able to compute DoA information for incoming signals based on a short preamble tone (via the MUSIC algorithm for example), and on senders having accurate cached information about the direction in which to beam-form for potential receivers [Singh 05]. Smart-ALOHA has the advantage of simplicity, but requires capabilities not present in low-end reconfigurable antennas. This paper also contains a good summary of prior MACs for directional antennas, including those presented in [Bao 02b], [Bellofiore 02c], [Choudhury 02], [Elbatt 02b], [Fahmy 02],[Bandyopadhyay 01], [Choudhury 06], and [Choudhury 05].

## 2.6 Networking with Fixed Directional Antennas

DMesh [Das 06] is designed for 802.11 with multi-radio nodes and fixed directional antennas. Directional antennas are used in pairs for point-to-point links. Every link is greedily assigned a channel on creation, and then reassignment is done periodically. The order of channel reassignment is not stated. A link is preferentially given a conflict-free channel, or else the least heavily loaded of the channels on which there is a conflict. The authors define both measurement-based (M-DCA) and geometry-based (C-DCA and A-DCA) criteria for identifying conflicts. Interestingly, all of the

Implementation	Analog beamforming Digital beamforming
Flexibility	Switched beam Steerable Adaptive
Purpose	Transmit beamforming Receive beamforming Joint Tx & Rx beamforming

Table 2.1: Directional Antenna Categorization (modified from [Li 05a])

explicitly-coordinated conflict avoidance (A-DCA, C-DCA, and the omni-directional OCA) schemes significantly out-performed the measurement-based scheme. Routing is a (slightly) modified form of OLSR.

### 2.6.1 Directional Antennas in Mesh Networks

A survey of issues in the use of steerable antennas in CSMA-based multihop networks is given in [Li 05a]. The paper provides a useful typology of steerable antennas, which I reproduce as Table 2.1.

Vilzmann evaluates the effects of very simple beam-forming strategies on pairwise communication among randomly-placed nodes [Vilzmann 05]. Their analysis includes estimates of the **expected SIR** for links, as other links are added. Specifically, they consider the impact on each link when the  $n$  **least-interfering** other links are active, as a function of  $n$ .

Jaikaeo evaluates various beam-steering strategies in the context of multicast with a CSMA-based MAC [Jaikaeo 03]. The paper compares directional and omnidirectional patterns at the sender and receiver, with and without ACKs. For almost all metrics, directional reception with directional transmission (DRTD) and directional reception with omnidirectional transmission (DROT), without ACKs, were the best configurations. For every measure except delay, the two were very close to each other in performance, and significantly better than any other options.

The paper [Rashid-Farrokhi 98a, Rashid-Farrokhi 98b] presents an integrated algorithm for beamforming and power control, so that the adaptive array base station and (fixed set of) omnidi-

rectional clients are optimized as a single system.

## 2.7 Related Problems

### 2.7.1 Channel Assignment

This subsection discusses papers which consider channel assignment outside the context of time scheduling.

Channel and **code assignment** are analogous problems which arise in frequency division and code division systems. **Subcarrier allocation** is similar to channel assignment and occurs in OFDM systems[Hottinen 06]. Bao presents a distributed code assignment algorithm in [Bao 02a].

The paper [Kyasanur 05a] gives a general discussion of ways to make mesh networking more physical-layer aware. The main mechanisms discussed are **multi-channel routing** and **spatial backoff**. In multi-channel routing, the link-layer protocol tries to assign nearby links to different channels, and the routing protocol tries to select paths with high channel diversity. Spatial backoff is more directly related to co-channel spatial reuse. The core idea is for nodes to adapt their CSMA/CA carrier-sensing threshold and modulation scheme to optimize aggregate throughput. Kyasanur also addresses multi-channel networking in several other papers: [Kyasanur 05b, Chereddi 06, Kyasanur 06a, Kyasanur 06b].

### 2.7.2 Steerable Antennas Generally

An excellent overview on the applications of adaptive antenna systems is provided in [Godara 97a, Godara 97b]. Bellofiore *et al.* discuss the use of smart antennas in mesh networks from the perspective of the antenna designer [Bellofiore 02c, Bellofiore 02a, Bellofiore 02b]. In particular, they derive conclusions about the types of antenna patterns and signal processing capabilities which contribute to system capacity. Their analysis assumes a specific 802.11-like MAC, but likely extends to other protocols.

Babich *et al.* investigate the performance of several directionality-aware MAC techniques,

using more rigorous RF modeling than is common [Babich 06]. Major findings are that on the receiver side, directional antennas perform significantly worse than adaptive arrays, and that the process of identifying the direction of arrival (DOA) of an interfering signal and steering a null at it is very error-prone. Both of these effects are related to the multipath and angular spread characteristics of the channel.

Sakr and Todd discuss random access SDMA for a single base station [Sakr 00]. This work is oriented toward exploiting spatial diversity **at a single base station** for multiple concurrent up-links, not disjoint links in a wide area. This is also the model generally employed in cellular telephony (see section 2.7.3.) An 802.11-compatible MAC for multi-beam base station SDMA is presented in [Wang 07]. Their system assumes a multi-radio base station with either a sectored antenna or suitable smart antenna serving 802.11 (DCF) clients with omni-directional antennas. The work mainly consists of developing 802.11-compatible mechanisms to induce clients to do time-division duplexing (TDD) because the base station cannot be expected to receive and transmit at the same time, even in different sectors.

Shad *et al.* present algorithms for TDMA-SDMA in a similar context [Shad 01]. As with STDMA, optimal scheduling is shown to be NP-complete by reduction to graph coloring (which reduces to 3-SAT.) Lower-complexity algorithms with reasonable performance are presented. Another set of algorithms for scheduling in cellular SDMA, incorporating support for bit-rate selection, is presented in [Kim 05]. Kuehner *et al.* discuss “dynamic slot allocation,” which is near real-time scheduling, in the cellular SDMA context [Kuehner 01].

Macedo and Sousa evaluate the impact of physical-layer parameters, especially the modulation scheme, on SDMA capacity in an indoor environment [Macedo 98]. They assume a sectored-antenna base-station with a variable number of ports and a relatively simple “first fit” scheduling algorithm. Their findings suggest the capture threshold as a key determinant of spatial reuse, and they propose an alternative OFDM designed to reduce this threshold in high-multipath environments.

Intra-cell SDMA is fundamentally different from inter-cell or non-cellular (e.g., mesh) spatial

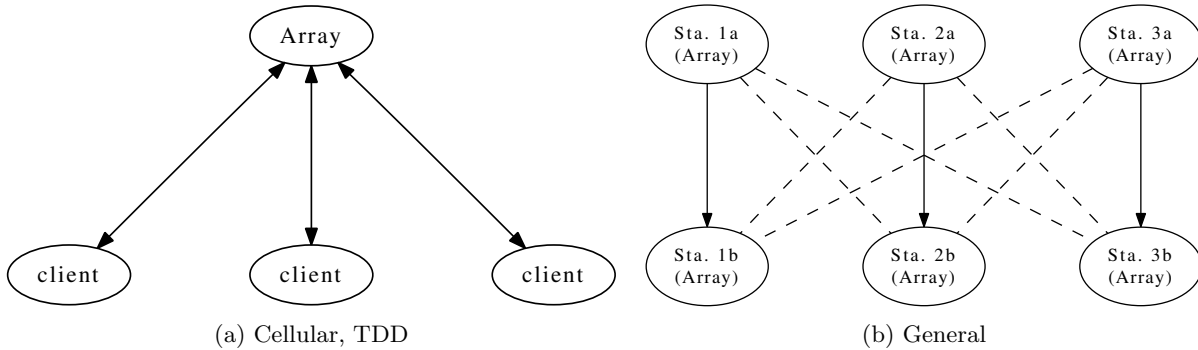


Figure 2.3: Complexity of cellular and general spatial reuse

reuse. In the intra-cell case, all of the co-channel concurrent links share a common endpoint, which has several significant implications: 1) For  $n$  concurrent links, the number of channels of interest varies as  $O(n)$ , not  $O(n^2)$ . 2) The number of concurrent links is bounded by the spatial-division capabilities of the base station, not the by the size of the network. 3) All of the necessary channel information is inherently available at a single location. In particular, knowing the spatial channel covariance matrices for all clients vis-a-vis the single antenna array provides a basis for both choosing a spatially-separable set of stations and forming an optimal set of beams for those stations [Kuehner 01, Galvan-Tejada 01, Zhang 01, Koutsopoulos 03, Jorswieck 07]. The synchronization and communication requirements make aggregating the channel information in the general case difficult if not infeasible. Figure 2.3 shows an example for  $n = 3$  links. Note that the inter-client channels are not of interest in the cellular case as long as the system uses time or frequency division duplexing so the situation in which one client is transmitting while another is receiving on the same frequency does not occur.

One interference-aware (or collision-aware) routing approach is given in [Roy 03]. It is effectively a link metric in which the cost of a link is based on the number of **currently active** neighbors known to be the coverage area of the nodes involved, as configured for that link. Other approaches to using directional antennas to improve routing are discussed in [Saha 03, Saha 04], and [Choudhury 05].

### 2.7.3 Cellular Telephony

Much of the early work on using steerable or smart antennas to manage interference is in the area of cellular telephony. The problems are similar to those faced in wireless networking, with a few key distinctions. The cellular research generally assumes smart antennas (and behavior) at the base station with simple handsets nodes [Swales 90]. In some cases, each cell is assumed to be making entirely local decisions, while in other cases base stations coordinate to reduce inter-cell interference. In neither case, however, is there inter-cell communication. Much of the empirical (*e.g.* [Li 97]) and analytical ([Naguib 94, Petrus 98, Ho 98, Matsumoto 97, Asakura 99]) research on the effectiveness of smart, switched, and multibeam antennas was also done in this context. Of particular interest, [Winters 94, Winters 99] compares simple steering with adaptive beamforming.

Much of the beam-forming work in cellular systems falls into the category of **single-station** SDMA, where the goal is to allow that station to support multiple concurrent up- or down-links. For example, [Razavilar 00] addresses the possible increase in capacity (decrease in blocking) from adding SDMA to frequency division.

“Multicell time-division beam scanning” is proposed in [Zander 92]. Time-division beam scanning is TDMA in which the base station switches its beam configuration to optimize communication with the station active during any given slot. There may be, but do not need to be, multiple beams for SDMA as well. The authors develop estimates of the level of inter-cell interference as a function of beam width and the number of stations served per time slot.

Another approach to using directional antennas to mitigate inter-cell interference is Quasi-Static Resource Allocation with Interference Avoidance (QRA-IA) [Chawla 99]. This is designed for a cellular architecture, but with an emphasis on data networks, not telephony. The authors emphasize managing downlink interference because downlink traffic tends to dominate in end-user data networks. The essence of QRA-IA is to divide the TDMA frame into sub-frames and have every base station **not transmit** in any given direction for one sub-frame out of every frame. Client terminals then identify the sub-frame or sub-frames in which they experience the least

interference, because their direction is being spared by whichever base station typically causes the most interference. By selectively scheduling clients to receive data in their preferred sub-frame, they experience diminished interference.

A centralized approach with a similar motivation is given in [Veronesi 04, Veronesi 06]. The main novel technique is **nulls preallocation**: Each base station identifies the stations in adjacent cells for which it creates the most interference, and reserves some of its beam-forming degrees of freedom for steering nulls at those stations. Base stations all broadcast their lists of nulled-out stations to adjacent cells, and then every station uses its neighbors null lists in making local beam-steering and slot allocation decisions.

## 2.8 Cross-Layer Optimization in Networking

This section will discuss cross-layer optimization in general, and especially focus on the “layering as optimization decomposition” school of thought. It is critical to note that “optimization” is a very ambiguous term in computer science. Its meanings range from “making something better” to “finding the best possible solution to some problem which is sufficiently well-defined to allow an absolute measure of goodness.” “Optimization” is also used to refer to the application of mathematical programming, regardless of whether this produces an optimal solution to the underlying problem.

Similarly, “cross-layer optimization” can refer to any technique which violates traditional layering abstractions to improve performance, e.g., [Clark 90] and [Baldo 08], or to approaches which are optimal in some sense but are not based on mathematical programming [Ghaderi 09].

### 2.8.1 Layering as Optimization Decomposition

The central notion of “Layering as Optimization Decomposition” is that the components of a network can be regarded as combining algorithmically and not just mechanically. Historically, there have been well-defined and proven algorithms for the various tasks involved in a network. There has been a much weaker understanding of the function of the network as a whole and the significance of



dividing that function into particular sub-tasks. To the extent that these tasks can be described as optimization problems – whether or not they were originally formulated that way – their combined effects can be modeled as a composition of optimization sub-problems. This perspective makes it possible to **reverse-engineer** the objectives and constraints which effectively describe an existing system [Chiang 05a, Chiang 06b, Chiang 06a]. These address wireless scheduling specifically, and develop the broader notion of network layers as **computational elements** coupled together to solve some global objective, whether by design or by accident. Of particular import for scheduling is work by Tan, Palomar, and Chiang which shows that many non-convex functions of interest, such as interference-limited Shannon capacity, are log-convex when transferred to the logarithmic domain, and therefore admit equivalent convex formulations [Tan 07, Tan 09].

In the forward direction, cross-layer (or multi-layer) designs can be formulated based on optimization decomposition. Once the overall objective and the fundamental constraints are enumerated, established decomposition techniques can often be used break the problem down into manageable components [Chiang 07]. Solving a mathematical program is, intuitively, a serial computation process. Many decomposition techniques make sense as a way of reducing the total amount of computation – that the original task is replaced by multiple new tasks is conceptually incidental. Networking is intuitively distributed; we take it as given that computation occurs at physically disparate locations and the notion of layered architectures is deeply internalized among both researchers and practitioners. What is interesting is that the lines along which the optimization problem is decomposed often map to the ways in which the networking task can be divided. Additionally, the optimization formalism and the associated conceptual tools often suggest decompositions which may not be apparent from the underlying problem. The first explicit treatment of utility maximization and corresponding dual problems in networking of which I am aware is Kelly’s work on rate adaptation [Kelly 98].

### 2.8.2 Introduction to Mathematical Program Decomposition Techniques

Excellent reviews of price-directive and resource-directive decomposition methods are given in [Rockafellar 93] and [Molina 79], respectively. A good text on optimization decomposition generally is [Conejo 06]. A tutorial on cross-layer optimization in wireless networking specifically is given in [Lin 06a].

Optimization decomposition is not limited to linear programs, but they make an easy context for illustrative examples. Consider an LP in standard form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{2.5}$$

Decompositions are generally based on exploiting special structure in the constraint matrix  $A$ . As shown below,  $a_{ij}$  is the coefficient of variable  $i$  in constraint  $j$ .

$$\begin{aligned} & \left\{ x_1 \quad \cdots \quad x_n \right\} \\ & \begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \end{aligned}$$

A constraint matrix with block-diagonal structure is trivially decomposable. Each block represents an independent problem. The computational complexity of linear programming is a non-trivial problem (see [Megiddo 87], [Bertsimas 97], and [Todd 02]), but the best worst-case complexity is  $O(n^{3.5}L)$ , where  $n$  is the number of variables and  $L$  is the length of the input in bits. Consequently, splitting one  $n$ -variable program into  $r$  programs of  $\frac{n}{r}$ -variables each would not only allow a parallel solution but also reduce the total work by a factor of  $r^{2.5}$ . Consider the example below, where the empty places represent 0 values:

$$A = \begin{bmatrix} a_{11} & a_{21} & & & & \\ a_{12} & a_{22} & & & & \\ & & a_{33} & a_{43} & & \\ & & a_{34} & a_{44} & & \\ & & & & a_{55} & a_{65} \\ & & & & a_{56} & a_{66} \end{bmatrix}$$

$$Ax = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} a_{33} & a_{43} \\ a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} a_{55} & a_{65} \\ a_{56} & a_{66} \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}$$

$$c^T x = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_3 & c_4 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} c_5 & c_6 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}$$

In this case, the original problem becomes (2.6). This trivial decomposition is possible because every constraint and every variable involves only one block in matrix  $A$ . The objective functional involves variables from all blocks, but they are simply added, and **min** is distributive over addition.

$$\begin{aligned} \min \quad & c_{1\dots 2}^T x_{1\dots 2} & \min \quad & c_{3\dots 4}^T x_{3\dots 4} & \min \quad & c_{5\dots 6}^T x_{5\dots 6} \\ \text{s.t.} \quad & A_{1\dots 2} x_{1\dots 2} = b_{1\dots 2} & \text{s.t.} \quad & A_{3\dots 4} x_{3\dots 4} = b_{3\dots 4} & \text{s.t.} \quad & A_{5\dots 6} x_{5\dots 6} = b_{5\dots 6} \\ & x_{1\dots 2} \geq 0 & & x_{3\dots 4} \geq 0 & & x_{5\dots 6} \geq 0 \end{aligned} \quad (2.6)$$

The more normal case, which motivates most decomposition research, is that a problem provides almost-block-diagonal structure. When the coefficient matrix is block-diagonal except for a small number of rows or columns with non-zeros corresponding to multiple blocks, these are termed **complicating constraints** or **complicating variables**, respectively.

$$\begin{array}{ccc}
\text{Complicating constraint} & & \text{Complicating variable} \\
A = \begin{bmatrix} a_{i1} \cdots a_{j1} & a_{k1} \cdots a_{l1} & a_{m1} \cdots a_{n1} \\ \text{block 1} & & \\ & \text{block 2} & \\ & & \text{block 3} \end{bmatrix} & A = \begin{bmatrix} a_{1i} \\ \vdots & \text{block 1} \\ a_{1j} \\ a_{1k} \\ \vdots & \text{block 2} \\ a_{1l} \\ a_{1m} \\ \vdots & \text{block 3} \\ a_{1n} \end{bmatrix} & (2.7)
\end{array}$$

The goal of decomposition techniques is to transform a problem with a difficult structure into several problems – or occasionally one – with an easier structure. Table 2.2 on the following page shows the applicability of various common decomposition techniques to various scenarios.

### 2.8.2.1 Dantzig-Wolfe Decomposition

The prototypical decomposition scheme is the (primal) Dantzig-Wolfe method [Dantzig 60]. A very brief illustration, closely taken from [Dantzig 63, section 23.1], is given here. Denote a two-block “complicating constraint(s)” problem as follows: Note first that  $\min_x c_1x_1 + \cdots + c_nx_n$  is equivalent to  $\max_{x_0} \text{s.t. } P_0x_0 + c_1x_1 + \cdots + c_nx_n = 0$  where  $P_0$  is unit.  $X$  and  $Y$  denote the variables occurring in the first and second blocks,  $A_1$  and  $A_2$  denote the coefficients corresponding to those blocks, and  $b_1$  and  $b_2$  are the constraint right-hand-sides (RHSs) for the constraints in those blocks.  $\bar{A}_1$  and  $\bar{A}_2$  are the coefficients of the complicating constraints for the variables  $X$  and  $Y$ , and  $\bar{b}$  is the RHS.

Technique	Problem			
	Continuous linear		Continuous non-linear	
	Complicating constraints	Complicating variables	Complicating constraints	Complicating variables
Dantzig-Wolfe	✓		✓	
Lagrangian relaxation	*		✓	
Augmented Lagrangian	✓		✓	
Optimality conditions <sup>c</sup>	*		✓	
Benders		✓		✓
Technique	Non-continuous linear		Non-continuous non-linear	
	Complicating constraints	Complicating variables	Complicating constraints	Complicating variables
	Complicating constraints	Complicating variables	Complicating constraints	Complicating variables
Dantzig-Wolfe	✓ <sup>b</sup>		✓ <sup>a</sup>	
Lagrangian relaxation	*	✓	✓	
Augmented Lagrangian	*	✓	✓	
Optimality conditions <sup>c</sup>			✓	
Benders		✓		✓

\* Partial, occasional, or minimal applicability.

<sup>a</sup> Known with names other than Dantzig-Wolfe.

<sup>b</sup> Known as “branch and price.”

<sup>c</sup> “Optimality conditions decomposition” introduced in [Conejo 02].

Table 2.2: Summary of decomposition techniques, modified from [Conejo 06, table 1.29].

$$\begin{aligned}
\max \quad & x_0 \\
\text{s.t.} \quad & A_1 X = b_1 \\
& A_2 Y = b_2 \\
& P_0 x_0 + \bar{A}_1 X + \bar{A}_2 Y = \bar{b} \\
& X, Y \geq 0
\end{aligned} \tag{2.8}$$

Regard the LP (2.8) as solving  $\max_{x_0} P_0 x_0 + \bar{A}_1 X + \bar{A}_2 Y = \bar{b}$ , subject to additional constraints:

$$\mathcal{L}_1 : A_1 X = b_1$$

$$\mathcal{L}_2 : A_2 Y = b_2$$

So long as the set denoted by  $\mathcal{L}_1$  is convex and bounded, any point  $X$  satisfying  $\mathcal{L}_1$  can be written as a convex combination of the extreme points satisfying  $\mathcal{L}_1$ , by Minkowski's theorem. Note also that the reverse holds: Any such convex combination of extreme points is in the set, and also that any convex combination of **any** points in the set – extreme or not – is also another point in the set. Therefore, given the finite set of extreme points  $\{X_1, X_2, \dots, X_K\}$ , the constraint  $\mathcal{L}_1 : A_1 X = b_1$  is exactly equivalent to:

$$X = \sum_{i=1}^K \lambda_i X_i \tag{2.9}$$

$$\sum_{i=1}^K \lambda_i = 1 \tag{2.10}$$

$$\lambda \geq 0; \tag{2.11}$$

Constraint (2.9) states that  $X$  is a linear combination of extreme points; (2.10) and (2.11) require that combination to be convex. If the sum of any subset of multipliers were to fall outside the range  $(0, 1)$ , that would specify a point outside the convex hull of the points  $X_i$ . The constraint  $\mathcal{L}_2$  can be similarly replaced. Let  $\mu$  designate its multipliers.  $X$  can be replaced with  $\sum_{i=1}^K \lambda_i X_i$  and  $Y$  can be replaced with  $\sum_{i=1}^L \mu_i Y_i$  throughout the program, leading to the linear program given in (2.12), which is equivalent to the original program (2.8).

$$\begin{aligned}
\max \quad & x_0 \\
\text{s.t.} \quad & P_0 x_0 + \sum_{i=1}^K \lambda_i (\bar{A}_1 X_i) + \sum_{j=1}^L \mu_j (\bar{A}_2 Y_j) = \bar{b} \\
& \sum_{i=1}^K \lambda_i = 1 \\
& \sum_{j=1}^L \mu_j = 1 \\
& \lambda, \mu \geq 0
\end{aligned} \tag{2.12}$$

This can be rewritten slightly using the following definitions:  $S_i \triangleq \bar{A}_1 X_i$ ,  $T_j \triangleq \bar{A}_2 Y_j$  to a canonical form known as the **extremal problem** or **full master problem**, shown in (2.13).

$$\begin{aligned}
\max \quad & x_0 \\
\text{s.t.} \quad & P_0 x_0 + \sum_{i=1}^K S_i \lambda_i + \sum_{j=1}^L T_j \mu_j = \bar{b} \\
& \sum_{i=1}^K \lambda_i = 1 \\
& \sum_{j=1}^L \mu_j = 1 \\
& \lambda, \mu \geq 0
\end{aligned} \tag{2.13}$$

The constraint blocks  $A_1 X = b_1$  and  $A_2 Y = b_2$  **do not appear** in this master problem. Rather, they are accounted for implicitly by the sets  $\{X_1, \dots, X_K\}$  and  $\{Y_1, \dots, Y_L\}$ . The sub-problems of **generating the extremal points** effectively replace those blocks of constraints.

### 2.8.2.2 Delayed Column Generation

Observe that the initial problem (2.8) was a single program with two blocks which would be independent if not for a (hopefully small) set of complicating constraints. The decomposition described thus far produces three separate programs: A program with no dependencies for finding the extreme points of  $\mathcal{L}_1$ ; a separate program, also with no dependencies, for finding the extreme points of  $\mathcal{L}_2$ ; and a much-simplified master program which depends on the results of both. This decomposition is a success except that  $\mathcal{L}_1$  and  $\mathcal{L}_2$  may have very many extreme points. (In the problem of link scheduling, as formulated by myself and others [Björklund 03, Yu 04, Johansson 06], every feasible subset of the set of links is an extreme point. This implies that the number of extreme

points is potentially  $O(2^n)$  where  $n$  is the number of links in the network.) When there are a large number of extreme points, merely enumerating them can be impractical, and solving an LP with that many variables can be effectively impossible.

As mentioned earlier, any set of points from a convex region define a convex hull fully contained by that region. If that set of points is the complete set of extreme points of the original region (and that region is a polytope, which any linearly-constrained system is), then the two regions are identical. If the set of points in question is a proper subset of the original extremal points, then their convex hull will be a proper subset of the original region, but will extend to its limit at some vertices and faces.

The essence of **column generation** is to identify a set of extreme points significantly smaller than the whole set, but still large enough that their convex hull includes the optimum of the original region. The process is also referred to as **delayed column generation**, because these extreme points are not found *a priori* but rather iteratively generated until the set is shown to be sufficient. Note that “column,” “variable,” and “extreme point” are effectively interchangeable terms in this context: Every extreme point of  $\mathcal{L}_1$  is an  $X_i$  with a corresponding multiplier variable  $\lambda_i$  and a corresponding column in the constraint coefficient matrix of the master problem. Likewise  $\mathcal{L}_2$ ,  $Y_i$  and  $\mu_i$ .

Dantzig describes a procedure for generating new columns, and removing (“pricing out”) existing ones, based on the simplex method of solving the master problem [Dantzig 60]. The criterion by which one would select one of the existing columns to bring into the basis also characterizes a desirable new column to generate. Considering the first subproblem  $\mathcal{L}_1$ : Let  $s^0 = - \begin{bmatrix} 0, \delta_i, \dots, \delta_k \end{bmatrix}$  be a row vector which identifies  $S_1, \dots, S_k$  in the simplex basis of the master problem [Dantzig 60, eq. (18)]. Let  $\pi^0$  denote the first row of the simplex multipliers (pricing vector) associated with solving the master problem with the same set  $X_{1\dots k}$ . The column  $i$  having the lowest relative cost  $\pi^0 S_i - s^0$  is the best (vis-a-vis  $X$ ) to enter the basis. Correspondingly, the best new column is  $S^* = \bar{A}_1 X^*$ , where  $X^*$  is the solution to (2.14). This is the **column-generation subproblem** for variables  $X$  or constraint  $\mathcal{L}_1$ :



$$\begin{aligned} \min_X \quad & (\pi^0 \bar{A}_1)X \\ \text{s.t.} \quad & A_1 X = b_1 \end{aligned} \tag{2.14}$$

The best new column vis-a-vis  $Y$ ,  $T^*$  is found analogously. This process also provides a stopping condition: The relative cost – also referred to as “reduced cost” – is an upper bound on how much the master problem objective can be improved by adding the column  $S^*$  or  $T^*$  to the problem. If the lowest relative cost  $\pi^0 S^* - s^0$  (or  $\pi^0 T^* - t^0$ ) is 0 or greater, then there is no feasible column (extreme point) which could be added to the master problem to improve its solution, which is to say that the current solution is optimal. The master problem solved over the subset of extreme points is referred to as the **restricted master problem**. It looks exactly like the full master problem (2.13) on page 47, except that the sets  $\{X_1, \dots, X_K\}$  and  $\{Y_1, \dots, Y_L\}$  are different.

The discussion of decomposition thus far has assumed that the goal is to exploit block-diagonal structure. As mentioned earlier, in section 2.8.2 on page 42, a magic decomposition of an  $n$ -variable problem into  $r$  completely separate  $\frac{n}{r}$ -variable problems could reduce the overall complexity by  $O(r^{2.5})$ , assuming Karmarkar’s worst-case complexity in all cases. When there are complicating constraints or variables, the decomposition is not perfect. The subproblems presented above do have  $\frac{n}{r}$  variables (if the blocks happen to be of equal size), but they must be solved repeatedly, and the restricted master problem has some minimal complexity remaining.

Decomposition need not involve block-diagonal structure. The Dantzig-Wolfe procedure is effectively performing two decomposition steps at once: First separating the complicating constraints from the easy part, and second dividing the easy part into one sub-problem per block. The same “complicating part / easy part” decomposition can be applied regardless of why the “easy part” is easy. In many cases, there is some other special structure to the problem which lets the easy part be solved using algorithms other than general-purpose linear program (or integer linear program) solving. The first actual application of column generation was given by Gilmore and Gomory, who applied it to the Cutting Stock problem [Gilmore 61]. They solve their specific subproblem as a

knapsack problem.

### 2.8.2.3 Lagrangian Relaxation

This subsection provides a very minimal introduction to Lagrangian relaxation (LR). For a more thorough discussion, see [Rockafellar 93] and [Conejo 06, sections 4.2, 4.4, and 5.3]. This introduction borrows heavily from both of those works.

Consider a mathematical program, potentially linear but often non-linear, of the form in (2.15) below. Lagrangian relaxation is a process for constructing and solving a dual problem for a general non-linear primal problem. If the primal problem is convex, and the primal and dual are both feasible, the Lagrangian dual problem exhibits **strong duality**, meaning that the optimal objective values of the two problems are the same. This is the same duality property that linear programs exhibit.

$$\begin{aligned}
 \min_x \quad & f(x) \\
 \text{s.t.} \quad & a(x) = 0 \\
 & b(x) \leq 0 \\
 & c(x) = 0 \\
 & d(x) \leq 0
 \end{aligned}
 \left. \vphantom{\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & a(x) = 0 \\ & b(x) \leq 0 \\ & c(x) = 0 \\ & d(x) \leq 0 \end{aligned}} \right\} \text{Complicating constraints} \tag{2.15}$$

If problem (2.15) were unconstrained, the optimality of any given point  $x$  could be checked by verifying that  $x$  was a global minimum of  $f(x)$ . Because  $f(x)$  is assumed to be convex, checking that  $x$  is a local minimum would suffice. The first- and second-derivative tests give criteria for evaluating non-degenerate points: The gradient (which is simply the Jacobian for the case of a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ )  $\nabla f(x)$  must equal 0, and the Hessian (the matrix of all second-order partial derivatives)  $H(f)$  must be positive definite.

The Karush-Kuhn-Tucker (KKT) optimality conditions are a generalization of the same criteria to the case of constrained optimality. Ignore for the moment the (so far artificial) distinction between the complicating constraints and others, and assume there are only two vectors of con-

straints:  $h(x) = 0$  and  $g(x) \leq 0$ . The first-order KKT conditions are:

Stationarity:

$$\nabla f(x) + \sum_{j=1}^l \lambda_j \nabla h_j(x) + \sum_{i=1}^m \mu_i \nabla g_i(x) = 0 \quad (2.16)$$

Primal feasibility:

$$g_i(x) \leq 0 \quad (2.17)$$

$$h_j(x) = 0 \quad (2.18)$$

Dual feasibility:

$$\mu_i \geq 0 \quad (2.19)$$

Complementary slackness:

$$\mu_i g_i(x) = 0 \quad (2.20)$$

The stationarity condition, equation (2.16), states that the gradients of equality constraints  $\nabla h_i(x)$ , the gradients of the (active) inequality constraints  $\nabla g_j(x)$ , and the gradient of the objective function  $\nabla f(x)$  are **linearly independent**. Phrased differently, at the optimal point, the vector sum of the (scaled) gradients of the active constraints is exactly the inverse of the gradient of the objective function. The multipliers ( $\lambda$  and  $\mu$ ) are termed the **Lagrange multipliers**. The interpretation of the Lagrange multipliers as dual values is consistent with our intuition from linear program duality: Consider an infinitesimal relaxation of constraint  $h_j$  at the optimal point,  $x^{*t}$ . The change in  $x$ ,  $\Delta x = x^{*t+1} - x^{*t}$  projected onto  $\nabla h_j(x)$  has a magnitude of  $\epsilon$ . Assuming that no other constraint is relaxed, and therefore the projection of  $\Delta x$  onto the other gradients is 0, the projection onto  $\nabla f(x)$  has magnitude  $\lambda_j$ , which is exactly what is expected.

The **Lagrangian function**, or just **Lagrangian**,  $\mathcal{L}$  is defined as equation (2.21) on the next page. Note that  $\lambda^T h(x) = \sum_{j=1}^l \lambda_j h_j(x)$ , and that  $\mathcal{L}$  considers the objective and constraint

functions directly, rather than their gradients.

$$\mathcal{L}(x, \lambda, \mu) \triangleq f(x) + \lambda^T h(x) + \mu^T g(x) \quad (2.21)$$

In general, Lagrangian relaxation is based on relaxing (or “dualizing”) only the complicating constraints. Returning to the problem given in (2.15), the Lagrangian is (2.22).

$$\mathcal{L}(x, \lambda, \mu) \triangleq f(x) + \lambda^T c(x) + \mu^T d(x) \quad (2.22)$$

Using the Lagrangian, the **dual function** of problem (2.15), written  $\phi(\lambda, \mu)$ , is (2.23):

$$\phi(\lambda, \mu) \triangleq \begin{cases} \min_x & \mathcal{L}(x, \lambda, \mu) \\ \text{s.t.} & a(x) = 0 \\ & b(x) \leq 0 \end{cases} \quad (2.23)$$

Two problems follow from the dual function. First, the **relaxed primary problem** (RPP) consists of minimizing the dual function for a fixed multiplier estimate  $(\bar{\lambda}, \bar{\mu})$ , (equation (2.24)). Second, the dual problem is defined as (2.25) below:

$$\begin{aligned} \min_x & \mathcal{L}(x, \bar{\lambda}, \bar{\mu}) \\ \text{s.t.} & a(x) = 0 \\ & b(x) \leq 0 \end{aligned} \quad (2.24)$$

Finding the **Lagrangian dual problem** (DP) (2.25) depends on solving for  $\phi(\cdot)$  analytically.

$$\begin{aligned} \max_{\lambda, \mu} & \phi(\lambda, \mu) \\ \text{s.t.} & \mu \geq 0 \end{aligned} \quad (2.25)$$

In general, the DP need not be solved directly. The **phase 1** Lagrangian relaxation algorithm iterates between solving the RPP (often in further-decomposed form) and incrementally updating the multipliers. The multipliers can be updated by a variety of techniques which will not be discussed in detail here. One such approach is based on finding a **subgradient** of the dual function

at each iteration. One such subgradient is given by the extent to which the (dualized) constraints are not satisfied. At time  $t$ , let  $s^t$  be a subgradient given by equation (2.26):

$$s^t = \begin{bmatrix} c(x^t) \\ d(x^t) \end{bmatrix} \quad (2.26)$$

Then, where  $\|\cdot\|$  denotes the  $\ell^2$  norm and  $k^t$  is the **step size** at time  $t$ , the update procedure is given by (2.27):

$$\begin{bmatrix} \lambda \\ \mu \end{bmatrix}^{t+1} \leftarrow \begin{bmatrix} \lambda \\ \mu \end{bmatrix}^t + k^t \frac{s^t}{\|s^t\|} \quad (2.27)$$

#### 2.8.2.4 Additional Lagrangian Relaxation Decomposition Variants

Two techniques mentioned in table 2.2 on page 45 are variants on Lagrangian relaxation: Augmented Lagrangian decomposition is based on adding a quadratic penalty term to the Lagrangian function to reduce or eliminate the **duality gap** when the primal function is non-convex [Rockafellar 74].

Optimality-condition decomposition (OCD) is a method for efficient solution of Lagrangian relaxation problems with multiple complicating constraints [Conejo 02]. The core insight is to retain the complicating constraints in the decomposed relaxed primal problems (DPPs) so that **at every iteration** of the DPPs, sensitivity information from each subproblem can be used to update the multipliers on every other subproblem. Consider the minimization problem (2.28) below:

$$\begin{aligned} \min_{x_1, x_2} \quad & f(x_1, x_2) \\ \text{s.t.} \quad & h_1(x_1, x_2) = 0 \\ & h_2(x_1, x_2) = 0 \\ & c_1(x_1) = 0 \\ & c_2(x_2) = 0 \end{aligned} \quad (2.28)$$

A standard LR decomposition dualizes  $h_1(\cdot)$  and  $h_2(\cdot)$  into the objective function, giving the relaxed primal problem (2.29) below:

$$\begin{aligned} \min_{x_1, x_2} \quad & f(x_1, x_2) + \bar{\lambda}_1^T h_1(x_1, x_2) + \bar{\lambda}_2^T h_2(x_1, x_2) \\ \text{s.t.} \quad & c_1(x_1) = 0 \\ & c_2(x_2) = 0 \end{aligned} \tag{2.29}$$

The RPP (2.29) is separable by  $x_1$  and  $x_2$ . A normal LR approach would be to solve it as two DPPs, one for each set of variables. The DPP for  $x_1$  treats the current best estimate of  $x_2$  – denoted  $\bar{x}_2$  – as a constant and solves for  $x_1$ . The DPP for  $x_2$  works similarly. The problems then are (2.30) below:

$$\begin{aligned} \min_{x_1} \quad & f(x_1, \bar{x}_2) + \bar{\lambda}_1^T h_1(x_1, \bar{x}_2) + \bar{\lambda}_2^T h_2(x_1, \bar{x}_2) \\ \text{s.t.} \quad & c_1(x_1) = 0 \end{aligned} \tag{2.30a}$$

$$\begin{aligned} \min_{x_2} \quad & f(\bar{x}_1, x_2) + \bar{\lambda}_1^T h_1(\bar{x}_1, x_2) + \bar{\lambda}_2^T h_2(\bar{x}_1, x_2) \\ \text{s.t.} \quad & c_2(x_2) = 0 \end{aligned} \tag{2.30b}$$

The decomposition in (2.30) allows the problem to be evaluated separately in terms of  $x_1$  and  $x_2$ . Each subproblem is solved to optimality, and then a subgradient is computed as in equation (2.26) on the previous page:

$$s = \begin{bmatrix} h_1(x_1, x_2) \\ h_2(x_1, x_2) \end{bmatrix}$$

The optimality condition decomposition method avoids solving each DPP to completion and then separately updating the multipliers by partially retaining the complicating constraints in the DPPs. Because they are part of the DPPs, every update step **within the solution of each DPP** produces a subgradient vector for the complicating constraints. The OCD DPPs for (2.30) could be (2.31) below:

$$\begin{aligned} \min_{x_1} \quad & f(x_1, \bar{x}_2) + \bar{\lambda}_1^T h_1(x_1, \bar{x}_2) \\ \text{s.t.} \quad & c_1(x_1) = 0 \end{aligned} \tag{2.31a}$$

$$h_2(x_1, \bar{x}_2) = 0$$

$$\begin{aligned} \min_{x_2} \quad & f(\bar{x}_1, x_2) + \bar{\lambda}_2^T h_2(\bar{x}_1, x_2) \\ \text{s.t.} \quad & c_2(x_2) = 0 \end{aligned} \tag{2.31b}$$

$$h_1(\bar{x}_1, x_2) = 0$$

After each iteration of the subproblem, a subgradient may be found in the usual way, using the values of  $x_1$  and  $x_2$  from (2.31a) and (2.31b), respectively. The updated multipliers are then the estimates input to the next iteration. I do not consider the OCD method for my algorithms for purely practical reasons: It requires modifying the internals of the solver(s) used for the subproblems at a rather deep level.

### 2.8.2.5 Benders Decomposition

One major decomposition technique that is specifically oriented toward the “complicating variables” case is Benders decomposition. Benders decomposition and Dantzig-Wolfe decomposition are dual to each other. Consider the minimization problem (2.32) below.

$$\begin{aligned} \min_{x,y} \quad & f(x, y) \\ \text{s.t.} \quad & c(x) \leq 0 \\ & d(x, y) \leq 0 \end{aligned} \tag{2.32}$$

The decision variables are divided into subsets  $x$  and  $y$ , where  $x$  are the complicating variables. That is to say, if the variables  $x$  are assumed to be fixed to some estimate  $\bar{x}$ , the resulting problem (2.33) is further decomposable, or otherwise significantly easier to solve.

$$\begin{aligned} \min_y \quad & f(\bar{x}, y) \\ \text{s.t.} \quad & d(\bar{x}, y) \leq 0 \end{aligned} \tag{2.33}$$

To exploit this problem structure, an **auxiliary function**  $\alpha(x)$  is defined in (2.34) to consider the complicating variables.

$$\alpha(x) \triangleq \begin{cases} \min_y & f(x, y) \\ \text{s.t.} & d(x, y) \leq 0 \end{cases} \quad (2.34)$$

The original problem can then be written in terms of  $\alpha(x)$  as problem (2.35):

$$\begin{aligned} \min_x & \alpha(x) \\ \text{s.t.} & c(x) \leq 0 \end{aligned} \quad (2.35)$$

Using the above definitions, the Benders decomposition procedure can be stated. Let  $t$  be the number of the current iteration, let  $x^t$  and  $y^t$  be the values of  $x$  and  $y$  found in iteration  $t$ , and let  $\lambda^t$  be the dual values associated with the constraints  $x = x^t$  in the subproblem iteration  $t$ . Initialization consists of choosing a feasible initial estimate  $x = x^0$  satisfying  $c(x^0) \leq 0$ . The algorithm then consists of iteratively solving the subproblem and master problem until a convergence test is satisfied.

The **Benders decomposition subproblem** is given by (2.36) below:

$$\begin{aligned} \min_y & f(x, y) \\ \text{s.t.} & d(x, y) \leq 0 \\ & x = x^t \end{aligned} \quad (2.36)$$

The outputs of the subproblem at iteration  $t$  are: The  $y$  values  $y^t$ ; the **dual price vector**  $\lambda^t$ , which is the price of the constraints  $x = x^t$ ; and the objective value  $f(x^t, y^t)$ . Note that  $x^t$  was initialized to a value satisfying  $c(x^t) \leq 0$ , and all updates to  $x^t$  will maintain the property. Consequently, the constraint  $x = x^t$  dominates the constraint  $c(x) \leq 0$ . This means that the subproblem (2.36) is strictly more constrained than the original problem (2.32). Consequently every solution to the subproblem is also a feasible solution to the original problem and  $f(x^t, y^t)$  constitutes an upper bound  $z_{up}^t$  on the optimal solution  $z^*$ .



The **Benders decomposition master problem** is given by (2.37) below. The new variable  $\alpha$  is a scalar and not the function  $\alpha(\cdot)$ . The **constant**  $\lambda_k^i$  or  $x_k^i$  denotes the  $k$ -th element of the  $\lambda$  or  $x$  vector found in iteration  $i$ .

$$\begin{aligned} \min_{x, \alpha} \quad & \alpha \\ & \alpha \geq f(x^i, y^i) + \sum_{k=1}^n \lambda_k^i (x_k - x_k^i) \quad \forall i = 1, \dots, t-1 \\ & c(x) \leq 0 \end{aligned} \tag{2.37}$$

The outputs of the master problem at iteration  $t$  are  $x^t$  and  $\alpha^t$ . Note that the master problem is a **relaxation** of the original problem, and so the solution value  $\alpha^t$  provides a lower bound  $z_{down}^t$  on the optimal solution to the original problem. Combined with the upper bound obtained from the subproblem, this makes it possible to compute the **optimality gap** in equation (2.38):

$$\frac{|z_{up}^t - z_{down}^t|}{|z_{down}^t|} \tag{2.38}$$

If the gap after subproblem iteration  $t$  is  $\leq \epsilon$ , computation terminates.

### 2.8.2.6 Geometric Programming

Geometric programming is not strictly a decomposition technique, but it gives rise to a transformation technique which facilitates subsequent decomposition. Geometric programs (GPs) are a class of optimization problem which exploits a special structure in the objective and constraint functions, like the better known linear programs, quadratic programs, cone programs, and semidefinite programs, among others. The fundamental property of interest is that GPs involve sets and functions which have a convexity property defined using the **weighted geometric mean**, rather than the usual weighted arithmetic mean.

A familiar definition of a convex set is that, given any two points in that set, all points on the line segment between those points are also in the set. A more formal definition of  $x$  “being on

the line segment between points”  $x_1$  and  $x_2$  is:

$$x = \theta x_1 + (1 - \theta)x_2$$

$$0 \leq \theta \leq 1$$

This gives exactly that  $x$  is a weighted arithmetic mean of  $x_1$  and  $x_2$ . Fundamental results in convex geometry such as Carathéodory’s theorem and Jensen’s inequality are similarly defined. An analogous definition is that all  $x$  on the arc of weighted geometric means between  $x_1$  and  $x_2$  are in the set:

$$x = x_1^\theta x_2^{(1-\theta)}$$

$$0 \leq \theta \leq 1$$

Functions and sets which are convex under this definition are the elements of geometric programming. For a general introduction, the reader is referred to the seminal work on GP [Duffin 67] and subsequent texts [Ecker 80, Boyd 04b, Chiang 05b, Boyd 07].

A property of particular practical interest is that many geometrically convex functions and sets can be bijectively mapped to functions and sets which are convex in the conventional sense. This transformation provides a means of identifying a convex equivalent for some seemingly non-convex problems, as well as allowing GPs to be solved using general-purpose convex programming techniques. This technique is applicable for **posynomial functions** which define either a minimization objective or the left-hand side of a  $\leq$  inequality constraint, and **monomial functions** which define the left-hand side of an equality constraint. This transformation will be explained briefly, mostly following the notation of [Boyd 04a, Boyd 07]: Consider the following program, where each doubly-indexed function, *e.g.*  $f_{0_2}()$ ,  $g_{2_1}()$ , is a monomial function of  $x$ , *e.g.*  $f_{0_1}(x) \equiv c_{0_1} x_1^{a_{0_1 1}} x_2^{a_{0_1 2}} x_3^{a_{0_1 3}} \dots$ , where the coefficients  $c_{0_1} \in \mathbb{R}_{++}$  and  $a_{0_1 1}, a_{0_1 2}, \dots \in \mathbb{R}$ .

$$\begin{aligned}
\min_x \quad & f_0(x) \equiv f_{0_1}(x) + f_{0_2}(x) + \cdots \\
\text{s.t.} \quad & f_1(x) \equiv f_{1_1}(x) + f_{1_2}(x) + \cdots \leq 1 \\
& f_2(x) \equiv f_{2_1}(x) + f_{2_2}(x) + \cdots \leq 1 \\
& g_1(x) \equiv f_{3_1}(x) = 1 \\
& g_2(x) \equiv f_{4_1}(x) = 1
\end{aligned}$$

For every variable  $x_i$ , we introduce a new variable  $y_i = \log(x_i)$ . Now,  $f(x) = f(e^y)$ . Taking the natural log of the left- and right-hand sides of the objective and each constraint and distributing through, each monomial function is replaced with:

$$\tilde{f}_{0_1}(y) \equiv e^{a_{0_1 1}y_1 + a_{0_1 2}y_2 + a_{0_1 3}y_3 + \cdots + \log(c_{0_1})}$$

Using the above definition, the complete program is replaced with the following, which is referred to as the convex form of the geometric program:

$$\begin{aligned}
\min_y \quad & \tilde{f}_0(y) \equiv \log \left( \tilde{f}_{0_1}(y) + \tilde{f}_{0_2}(y) + \cdots \right) \\
\text{s.t.} \quad & \tilde{f}_1(y) \equiv \log \left( \tilde{f}_{1_1}(y) + \tilde{f}_{1_2}(y) + \cdots \right) \leq 0 \\
& \tilde{f}_2(y) \equiv \log \left( \tilde{f}_{2_1}(y) + \tilde{f}_{2_2}(y) + \cdots \right) \leq 0 \\
& \tilde{g}_1(y) \equiv \log \left( \tilde{f}_{3_1}(y) \right) = 0 \\
& \tilde{g}_2(y) \equiv \log \left( \tilde{f}_{4_1}(y) \right) = 0
\end{aligned}$$

Note that  $\log \left( \tilde{f}_{3_1}(y) \right)$  is just  $a_{3_1 1}y_1 + a_{3_1 2}y_2 + a_{3_1 3}y_3 + \cdots + \log(c_{3_1})$  and similarly for any other equality constraints. It is important to note that the original (posynomial form) equality constraints must be monomial functions, as these produce affine functions after the transformation.

This transformation is of particular bearing on this dissertation because several important functions in wireless communication, such as signal to interference and noise ratio (SINR) and Shannon capacity, are non-convex but can be expressed as (generalized) geometric programs. This fact has been used in a number of algorithms for wireless network power control [Foschini 93,

Julian 02, Kandukuri 02, Tan 07, Tan 09], and is used in the convexity analysis in this dissertation in section 4.2 on page 78.

### 2.8.3 Optimization-Based Scheduling

Almost all work in optimization-based scheduling involves, at least implicitly, the notion of the **capacity region** as a convex hull of basic rate vectors. The idea is implicit in Arikian's formulation of  $\vec{f}$ -feasibility [Arikian 83, Arikian 84], but I will follow the discussion of [Toumpis 00, Toumpis 03]. The combination of physics and protocol requirements allows some set of **transmission schemes**, where a transmission scheme is a description of which nodes are transmitting what information to which other nodes, and how, at a given instant. The rate of information flow associated with each scheme can be described by a **rate matrix**. The rate matrix is a square matrix  $R = r_{ij}$ , where  $r_{ij}$  is defined by equation (2.39).

$$r_{ij} = \begin{cases} r & \text{if node } j \text{ receives information from node } i \text{ at rate } r, \\ -r & \text{if node } j \text{ transmits information to node } i \text{ at rate } r, \\ 0 & \text{otherwise.} \end{cases} \quad (2.39)$$

The same information often denoted by a **rate vector**, where each scalar element of the vector is the rate on some directional link  $ij$ . Toumpis and Goldsmith refer to the rates achievable by any given transmission scheme as a **basic** rate matrix. Assume that over some duration, the network is in scheme 1 with rate matrix  $R_1$  for a fraction of time  $\alpha$ , and in scheme 2 with rate matrix  $R_2$  for a fraction of time  $(1 - \alpha)$ . (The assumption that the two fractions sum to 1 implies that switching delay between schemes is negligible.) The cumulative rate is  $\alpha R_1 + (1 - \alpha)R_2$ . By extension, for  $k$  schemes given time  $\alpha^k$  each such that  $\sum_k \alpha_k = 1, \alpha_k \geq 0$ , the cumulative rate is  $\sum_k \alpha_k R_k$ . Therefore, the capacity region – the set of feasible rate matrices (or vectors) – is exactly the convex hull of basic rate matrices (or vectors). Note that the vertices (extrema points) of the hull are all basic, but not all basic rate matrices are necessarily vertices.

As the convex hull of any set of points is a convex polytope, the feasible rate region is

continuous and bounded by a set of linear constraints. This means that, **given the set of extremal basic rate vectors**, and assuming the overall network utility is a function of the flow rates, the scheduling problem is only as complex as the chosen utility function. The complications, both noted by Arıkan, are that the number of basic extremal points can be exponentially large, and identifying those extremal points is NP-hard under any reasonable interference model.

Consequently, all of the optimization-based scheduling approaches of which I am aware decompose the problem into: (1) A relatively simple master problem which is responsible for allocating time between basic rate vectors (transmission sets) to maximize the overall utility, and (2) one or more subproblems which identify **an approximation** of the extremal basic rate vectors.

#### 2.8.4 Examples of Decomposition in Wireless Networking

Xiao *et al.* present the Simultaneous Routing and Resource Allocation (SRRA) problem, which is a joint optimization approach to routing and something resembling scheduling [Xiao 04]. The authors make – and acknowledge – the simplifying assumption that link capacities can be determined completely by sender-local decisions. While this abstracted view of network and physical constraints does not correspond with any real system, it enables a very clean and logical development of techniques central to multi-layer optimization in networks. Xiao’s paper presents hierarchical dual decomposition using subgradient solution methods, and the coupling of routing and scheduling by per-node capacity prices.

[Xiao 04] is of excellent tutorial value because it discusses convex optimization and decomposition in general, and presents a problem crossing many traditional layers and involving several nested decompositions using different techniques. The scheduling and resource allocation component makes some significantly unrealistic assumptions in order to simplify the problem, which limits the usefulness of their specific solution. Another useful overview is [Lin 06a], although its coverage of scheduling is limited.

Following [Chiang 05a], [Yuan 05] provides a decomposition-based approach to routing, power control and network coding. One interesting contribution is that the paper compares a convex-

optimization approach to power control (based on approximating rate in the log domain as in [Chiang 05a]) with a game-theoretic equilibrium approach.

Björklund introduced the first optimization formulation of wireless scheduling of which I am aware: a **linear** program for STDMA link scheduling based on a Dantzig-Wolfe decomposition and column generation (section 2.8.2.2 on page 47) [Björklund 03]. The initial problem is to, over the set of all possible sets of concurrently-active links, determine how much time to assign to each such set. This program is formulated as identifying the shortest list of sets such that each set meets the physical SINR requirements of the underlying technology, and the combination of all the sets activates each link for long enough to meet its traffic requirements. This formulation is used as the basis for the work in this dissertation as well.

[Madan 05] uses an optimization-driven approach to choosing which links to activate, and then finds the minimum power levels to maintain an acceptable SINR on each link. The optimization formulation is a MILP to minimize the maximum sum of cross-link gain on active links.

## 2.9 Summary

Sections 2.1 and 2.2 are intended to provide a conceptual framework for the problem of spatial-reuse TDMA scheduling. Particularly important is the way groups of transmissions interfere with each other, and how those interactions are modelled. The model has an almost determining impact on the design of any scheduling system. The simple models underestimate interference, but admit low-complexity solutions, while the most accurate models lead to substantial algorithmic challenges.

Section 2.3 presents the prior work most directly related to this dissertation: STDMA scheduling with some form of antenna consideration. Several such approaches exist, but I argue that the coupling between scheduling decisions and antenna decisions is still largely unexplored. In particular, all of the existing schemes either use assumed suboptimal antenna configurations when making scheduling decisions (which is a conservative approximation), or make those decisions without knowing whether an antenna configuration that makes the schedule feasible actually exists (an aggressive approximation), or both. It is easy to identify scenarios, such as the X configuration

from figure 1.1a on page 4, in which these limitations produce substantially suboptimal results.

Section 2.4 discusses the integration of scheduling with factors other than antenna configuration. Integration with higher layers in the traditional stack (e.g. routing and congestion control) seems to be an essentially orthogonal problem. Integration with other low-level parameters, like power and rate control, is more similar to the problem at hand. The problems turn out to have substantial differences preventing techniques designed for one from being directly applicable to the others. The problems of power and rate control have nice properties which antenna configuration lacks: A change in the power level of any transmitter has a simple consistent effect on all receivers. If the transmit power increases by factor  $c$ , the signal strength of the intended transmission will increase by  $c$ , and the interference created by that transmission at every other receiver will increase by  $c$ . A change in the rate (modulation scheme) on any given link has no expected effect on any other link. Neither of these is at all true for antenna changes.

Sections 2.5 through 2.7 discuss less closely-related work having to do with resource assignment or antenna issues in contexts other than spatial-reuse TDMA. Lastly, section 2.8 reviews the mathematical programming optimization decomposition techniques which underlie much of this research.

## Chapter 3

### Problem and Formulation

This chapter presents an optimization-based algorithm and analysis for joint scheduling and antenna configuration. It is based on the link-assignment scheduling approach presented in [Björklund 03], and this work will use that notation, which is explained in table 3.1. This work considers **link** scheduling, rather than **node** scheduling, because antenna configuration is dependent on both the sender and the receiver.

#### 3.1 Problem Definition

Informally, the problem is to identify multiple **sets of links**, with antenna configurations for each node, such that (a) every link in a given set can be active concurrently, and (b) the combination of all sets includes every link on which there is a traffic demand. Demands consist of quanta of information to move on each link, and the objective is to minimize the amount of time (**i.e.**, the length of the schedule) required to meet those demands.

The size and complexity of the problem make direct solution computationally infeasible: The number of possible configurations is exponential in both the number of nodes and the number of antenna options per node. Further, the complete formulation is a mixed-integer non-linear program (MINLP) of degree 3.

##### 3.1.1 Formulation



symbol	context	interpretation
$A$	*	The set of all links
$N$	*	The set of all nodes
$L_A$	*	The set of all concurrently-feasible link sets
$L_A^t$	*	The generated subset of $L_A$ at time $t$
$L_A^0$	RMP	The generated subset of $L_A$
$x_l$	RMP	Number of slots assigned to link set $l \in L_A$
$S_{ijl}$	RMP	Link $(i, j)$ is active in link set $l$
$q_{ij}$	RMP	Demand (in slots) for link $ij$
$S_{ij}$	CLAP	Activation of link $ij$ (in current set)
$M_{ij}$	CLAP	Constant s.t. ineq. (4.6) holds when $S_{ij} = 0$
$V_i$	CLAP	Node $i$ is active (in current link set)
$P_i$	CLAP	Transmit power of node $i$
$\gamma_1$	CLAP	Desired SINR threshold
$N_r$	CLAP	Receiver noise level
$D_{ij}$	CLAP	Directivity of node $i$ in the direction of node $j$
$Lb(i, j)$	CLAP	Path loss from node $i$ to node $j$
$G_{ikp}$	CLAP	Gain for node $i$ using pattern $p$ , toward node $k$
$B_{jp}$	CLAP	Beam (antenna, pattern, ...) $p$ used at node $j$

Table 3.1: Notation

[JBSS-MP]

$$\min \sum_{l \in L_A} x_l \quad (3.1)$$

$$\text{s.t.} \quad \sum_{l \in L_A} S_{ijl} x_l \geq q_{ij} \quad \forall i, j \quad (3.2)$$

$$\sum_{j: (i,j) \in A} S_{ijl} + \sum_{j: (j,i) \in A} S_{jil} \leq 1 \quad \forall i, l \quad (3.3)$$

$$\left. \begin{aligned} & \frac{P_{il} D_{ijl} D_{jil}}{Lb(i, j) N_r} S_{ijl} + \\ & \gamma_1 (1 + M_{ijl}) (1 - S_{ijl}) \geq \\ & \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_{kl} D_{kjl} D_{jkl}}{Lb(k, j) N_r} V_{kl} \right) \end{aligned} \right\} \quad \forall i, j, l \quad (3.4)$$

$$S_{ijl} \leq V_{il} \quad \forall i, j, l \quad (3.5)$$

$$\sum_{p \in P} B_{jpl} = 1 \quad \forall j, l \quad (3.6)$$

$$D_{ik} = \sum_{p \in P} G_{ikp} B_{ipl} \quad \forall i, k, l \quad (3.7)$$

$$x_l \geq 0 \quad \forall l \in L_A \quad (3.8)$$

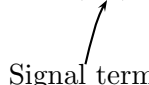
$$S_{ijl}, B_{jpl} \in \{0, 1\} \quad (3.9)$$

The objective, (3.1), is to minimize the number of activated link sets. For each link set  $l$  in the universe of possible concurrent link sets  $L_A$ ,  $x_l$  is a variable indicating the amount of time for which  $l$  is active. Constraint (3.2) specifies that the schedule must “cover” the demand.  $S_{ijl}$  is a boolean variable indicating whether link  $ij$  is active in link set  $l$ , and  $q_{ij}$  is the demand for link  $ij$ , measured in time. The constraint therefore requires that the total time for which link sets containing  $ij$  are activated is sufficient. Constraint (3.3) specifies that in any given link set  $l$ , every node  $j$  may be active in **at most** one link. This precludes duplex operation, as well as transmitting to or receiving from multiple partners.

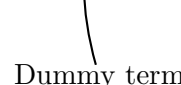
Constraint (3.4) specifies that minimum SINR requirements are met, taking antenna configuration into account. The formulation of this constraint is patterned after Björklund, and can be

somewhat unintuitive. See [Björklund 06, Chapter 3, eq. (3.12), and Appendix B].

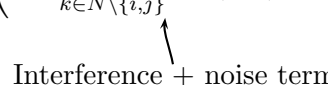
$$\frac{P_{il}D_{ijl}D_{jil}}{Lb(i,j)N_r}S_{ijl} + \gamma_1(1 + M_{ijl})(1 - S_{ijl}) \geq \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i,j\}} \frac{P_{kl}D_{kjl}D_{jkl}}{Lb(k,j)N_r}V_{kl} \right) \forall_{i,j,l} \quad (3.10)$$



Signal term



Dummy term



Interference + noise term

Ignoring the “dummy” term, the constraint specifies that if the link  $ij$  is used, the received signal strength must exceed the combined interference and noise level at  $j$  by factor  $\gamma_1$ . Note that the signal term is a product of 0-1 variable  $S_{ijl}$ , and the dummy term is a product of  $(1 - S_{ijl})$ . The effect of the dummy term is to ensure that the constraint is satisfied when  $S_{ijl} = 0$ . This means that the constraint is effectively a no-op when the link  $ij$  is not selected. For any given  $ij$ , when  $S_{ijl} = 1$ , the constraint reduces to inequality (3.11). Considering a given link in a given link set  $l$ , the  $l$  subscripts are removed for clarity.

$$\frac{P_i D_{ij} D_{ji}}{Lb(i,j)N_r} S_{ij} \geq \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i,j\}} \frac{P_k D_{kj} D_{jk}}{Lb(k,j)N_r} V_k \right) \quad (3.11)$$

The left-hand side gives the received SNR linear units:  $P_i$  is node  $i$ 's transmit power.  $D_{ij}$  and  $D_{ji}$  are the directional gain of node  $i$  toward node  $j$  and vice-versa.  $Lb(i,j)$  is the path loss between nodes  $i$  and  $j$ , and  $N_r$  is the receiver noise figure. The noise figure is regarded as a constant. Note that while this formulation uses a single  $N_r$  for all nodes, having a different noise figure for each receiver does not change the complexity of the solution. The right-hand side is the sum of the contribution above the noise floor of received interfering signals plus 1. The 0-1 variable  $V_k$  specifies that node  $k$  is (or may be) transmitting in the given time slot.

Constraint (3.5) couples the decision variables  $S_{ijl}$  and  $V_{il}$  so that if any link  $i,j$  is selected, the variable  $V_{il}$  reflects that  $i$  is transmitting. The  $V$  variable is used in (3.4) to identify sources of interference. The 0-1 variable  $B_{jpl}$  indicates whether node  $j$  uses beam pattern  $p$  in link set  $l$ . Constraint (3.6) specifies that in any given link set each node must select exactly one beam pattern. Constraint (3.7) couples the otherwise free directional gain variables  $D_{jil}$  and  $D_{jkl}$  to the choice of antenna beam  $B_{jpl}$ . Constraints (3.8) and (3.9) specify positivity and 0-1 requirements for variables.

### 3.1.2 Extensions

The joint beam steering and scheduling problem generalizes several other joint scheduling problems. In particular, transmit power and receiver sensitivity control are achieved by relaxing constraint (3.6) to allow fractional antenna gain as below, which is mathematically equivalent:

$$\sum_{p \in P} B_{jpl} \leq 1 \quad \forall_{j,l}$$

Additionally, selection from a finite set of modulation schemes is achieved by considering multiple “logical” links for each physical link, with a different  $\gamma_{1,r}$  for each rate  $r$  and adding a rate constant  $R_r$  to constraint (3.2) as follows:

$$\sum_{l \in L_A} S_{ijlr} R_r x_l \geq q_{ij} \quad \forall_{i,j}$$

Neither extension is considered further in this dissertation, but both are consistent with the decompositions presented. Adding rate selection is computationally similar to increasing the number of links, while adding power control does not increase the complexity at all.

## 3.2 Computational Complexity

The master problem (JBSS-MP) is complete, but computationally intractable. First, the program is mixed-integer cubic, meaning that the objective and/or constraints involve polynomials of degree 3 and a mixture of free and integer variables. There are a number of computational techniques for efficiently solving convex programs having degrees 1 or 2 (referred to as linear and quadratic programs), but cubic programs are as complex as non-linear programs in general. There is no obvious way to reformulate the cubic terms ( $D_{ijl}D_{jil}S_{ijl}$  and  $D_{kjl}D_{jkl}V_{kl}$ ) away as they are the fundamental determinants of SINR and are all real decision variables. The size of the problem is also vast. The subscript  $l$  indexes the set of all possible sets of links, having dimension  $2^m$  for  $m$  links. Several of the variables are indexed over  $L_A \times A \times A$ , meaning there are  $\Theta(n^2 2^m)$  variables and similarly many constraints. Concretely, that means over  $10^{15}$  variables for singly-connected (1 edge per node) topology of 40 nodes. Figures 3.1 on page 70 and 3.2 on page 71 show the

problem growth visually. Both are artificially trimmed in the  $y$  dimension to show more detail at the bottom. For 30 nodes, the fully-connected scenario has  $10^{300}$  variables.

It is important to note that the solution process is **again** super-linear, and indeed often exponential, in the number of variables. Polynomial-time algorithms exist for linear programs with continuous variables – in general, the easiest class of mathematical programs – but it has been shown that, even for a restricted subclass and even for solving to a constant approximation factor, the problem is P-complete [Efrimidis 08]. The recent solution of an LP with  $10^9$  variables on the BlueGene supercomputer is, to the best of my knowledge, the largest LP ever **directly** solved [Gondzio 06]. Moreover, solving general polynomial programs to a constant approximation factor is NP-complete [Bellare 95]. Given that it is impractical to explicitly enumerate, let alone directly solve, JBSS-MP, subsequent comparisons of computational complexity will consider only the **implicitly enumerated**, or column-generation, form.

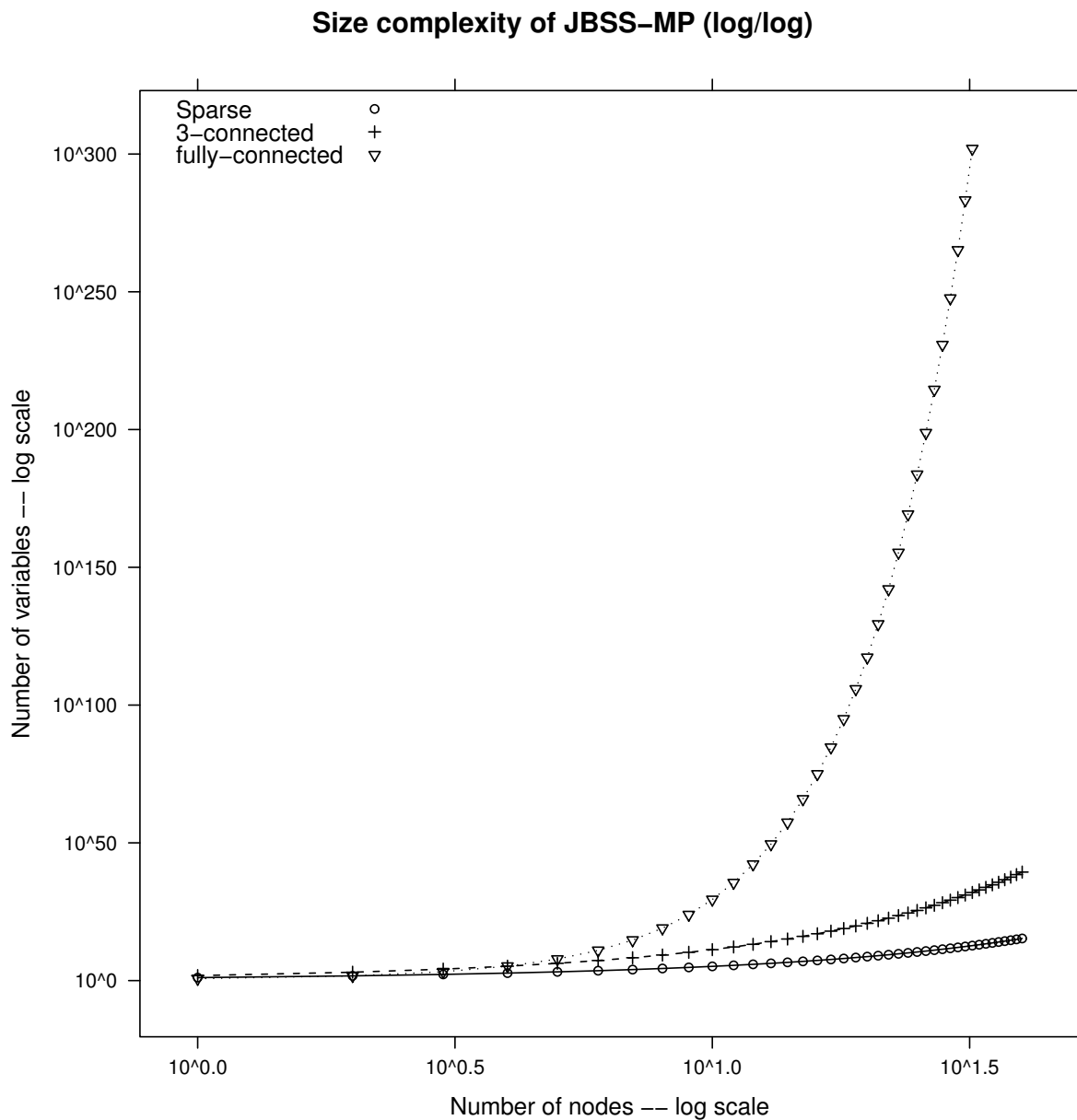


Figure 3.1: Size (number of variables) of JBSS-MP as a function of the number of nodes, log/log scale, limited at  $10^{60}$

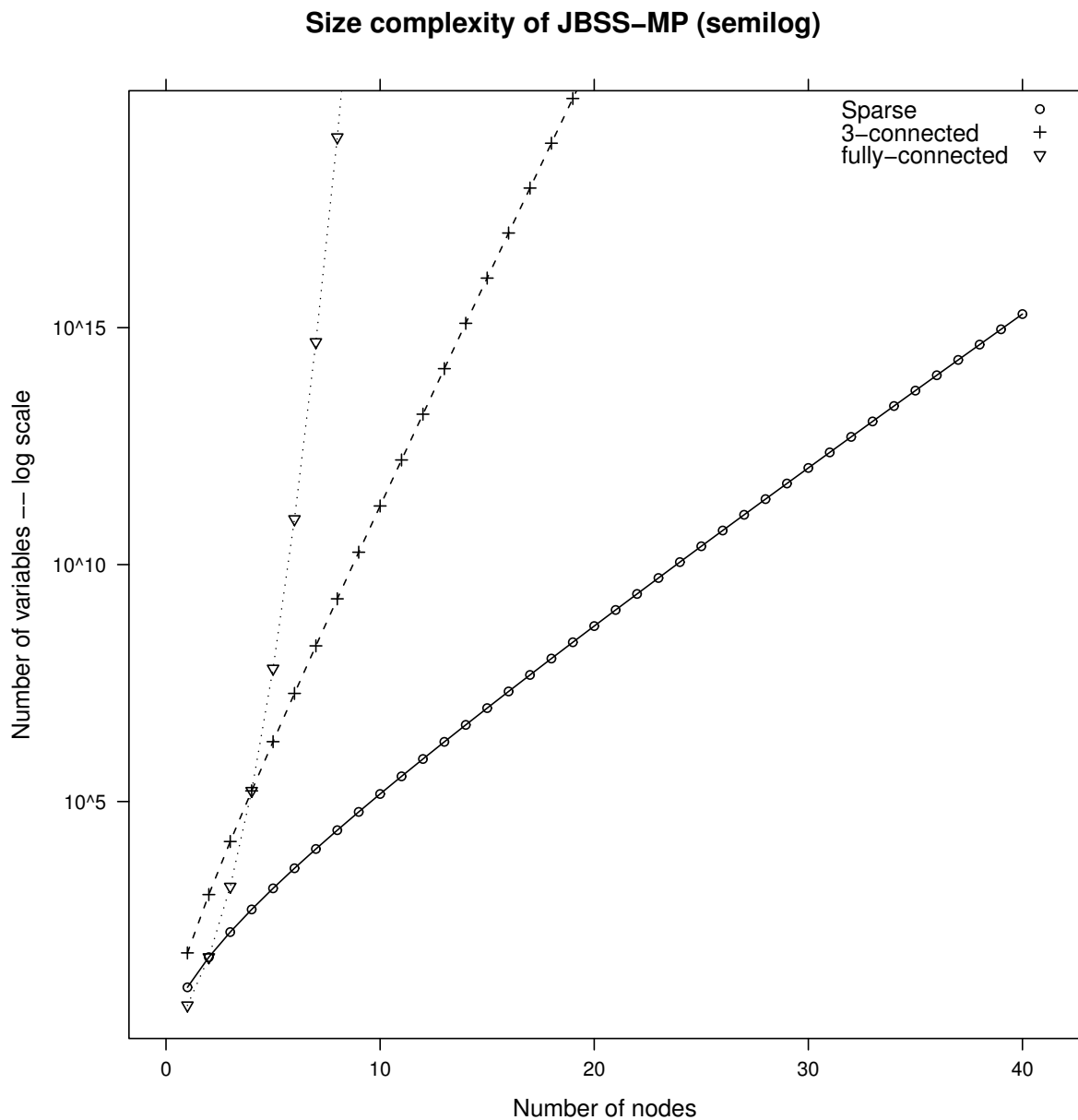


Figure 3.2: Size (number of variables) of JBSS-MP as a function of the number of nodes, semilog scale, limited at  $10^{20}$

## Chapter 4

### Decomposition Process

This chapter describes a distributed and decomposed mathematical solution to the integrated beam steering and scheduling problem. To work around the computational difficulties in this problem, I apply multiple decompositions to the problem into multiple smaller interacting problems. The basic formulation is the Joint Beam Steering and Scheduling Master Problem (JBSS-MP). This is decomposed into a Restricted Master Problem (RMP) and the Configuration and Link Activation Problem (CLAP). After working through two intermediate forms, CLAP is decomposed into the Fixed-link Antenna Reconfiguration Program (FARP), and the Relaxed Primal Fixed-antenna Link Activation Problem (RP-FLAP). FARP and RP-FLAP are separable, and are split into distributed, per-node versions, the Single Node Antenna Reconfiguration Problem (SNARP) and Single Node Relaxed Primal FLAP (SNRP-FLAP). Lastly, SNRP-FLAP is transformed to reduce oscillations, producing the Single-node Dual Quadratic FLAP (SDQ-FLAP). This process is shown schematically in figure 4.1 on the following page

#### 4.1 First Decomposition: Dantzig-Wolfe on JBSS-MP

The first restructuring, which is part of [Björklund 03] and also appears in [Garache 08, Johansson 06], [Yu 04], and [Chiang 07], is **Dantzig-Wolfe decomposition**. The problem is split into a **restricted master problem (RMP)** which operates on only a special subset of the variables present in the original, and **subproblems** which are linked by the master problem. In particular, to avoid having to enumerate (and consider) all possible groups of nodes, I apply the technique



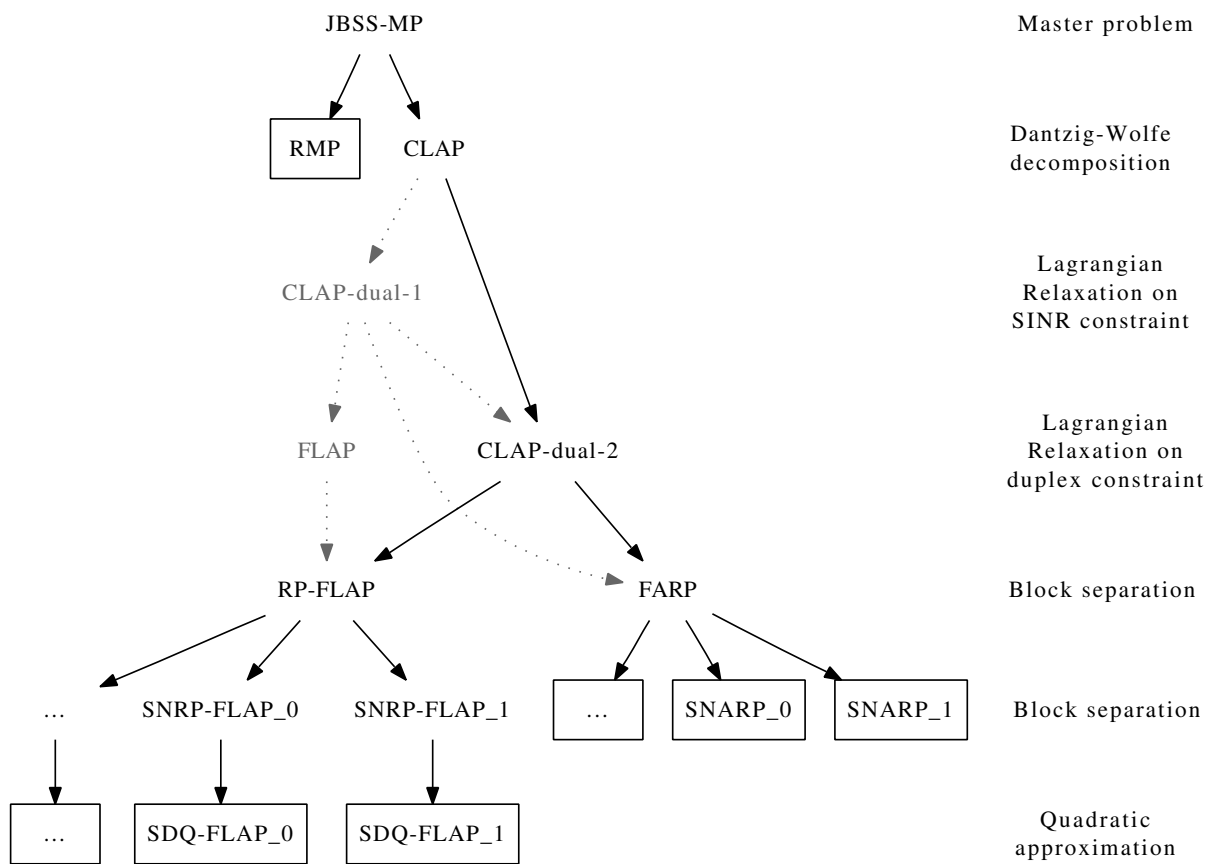


Figure 4.1: Outline of decomposition process.

of **column generation** [Gilmore 61, Gilmore 63] (also called **delayed column generation** and **implicit enumeration**.) The objective function of JBSS-MP (equation (3.1) on page 66) is quite simple; the complexity lies in defining the region of feasible values. By definition, the feasible space of a bounded system of linear constraints is a convex polytope. By Minkowski's Theorem, any point in the polytope can be formed by a linear combination of its extreme points. Note that while the JBSS-MP is not in general bounded by linear constraints, the constraints involving the variables  $x$  in which the decomposition is performed are all linear. As the optimal solution must be a feasible point, it can be written as a linear combination of extreme points. Column generation uses this fact to preserve the correctness of the original problem while working with a reduced set of points by considering only extreme points, and identifying them iteratively on an as-needed basis.

The RMP becomes (4.1)-(4.3) below, where the feasibility requirements for the link sets are implicit in  $l \in L_A$ . The time allocated to each feasible point  $l$  is denoted  $x_l$ , and the activation level of each link  $i, j$  in each  $l$  – an output of the subproblem, not a decision variable here – is denoted  $S_{ijl}$ . The set of feasible points defining the RMP's approximation polytope in iteration  $t$  is  $L_A^t$ .

[RMP]

$$\min \sum_{l \in L_A^t} x_l \quad (4.1)$$

$$\text{s.t.} \quad \sum_{l \in L_A^t} S_{ijl} x_l \geq q_{ij} \quad \forall_{ij} \quad (4.2)$$

$$x_l \geq 0 \quad \forall_{l \in L_A^t} \quad (4.3)$$

The column set  $L_A^t$  must be seeded with a set of columns s.t. RMP is feasible. The RMP is then solved to optimality. In general, the unconstrained global optimum may or may not be within the feasible region. If it is, then it is equal to the constrained optimum and no constraints are binding. If not, the constrained optimum will be at an extreme point of the feasible polytope, and the constraints generating that point will be binding. The binding constraints have, by definition,

non-zero dual costs associated with them. It can be seen that no finite  $x$  minimizes (4.1), so this problem is of the latter type. The AMPL specification of RMP is given in Listing D.1 on page 250.

The dual prices from the solution of the RMP form the inputs to the configuration and link assignment problem (CLAP). After each iteration of solving the RMP, CLAP uses them to attempt to generate a new extreme point (column) which extends the feasible region from the current optimum along an improving (negative for minimization) gradient of the objective function. A point having that property is referred to as a **negative reduced cost column**. If a reduced cost column **consistent with the constraints of the original problem** exists, then adding it to the pool  $L_A^l$  allows the solution to RMP to improve while ensuring that the solution is feasible for the original problem. Conversely, if no (feasible) reduced cost column exists, then the current optimal RMP solution is also optimal for the original problem and the optimization process is done. Expressions (4.4) - (4.10) on the following page present an optimization program for the **best** (minimum) reduced cost column. Note that this discussion drops the  $l$  subscripts because this sub-problem occurs in the context of creating a new link set, so  $l$  is in effect implicit.

[CLAP]

$$\max_S \bar{\beta}^T S \quad (4.4)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} S_{ji} \leq 1 \quad \forall_i \quad (4.5)$$

$$\left. \begin{aligned} & \frac{P_i D_{ij} D_{ji}}{Lb(i,j) N_r} S_{ij} + \\ & \gamma_1 (1 + M_{ij}) (1 - S_{ij}) \geq \\ & \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i,j\}} \frac{P_k D_{kj} D_{jk}}{Lb(k,j) N_r} V_k \right) \end{aligned} \right\} \quad \forall_{i,j} \quad (4.6)$$

$$S_{ij} \leq V_i \quad \forall_i \quad (4.7)$$

$$D_{ik} = \sum_{p \in P} G_{ikp} B_{ip} \quad \forall_{i,k} \quad (4.8)$$

$$\sum_{p \in P} B_{jp} = 1 \quad \forall_j \quad (4.9)$$

$$S_{ij}, B_{jp} \in \{0, 1\} \quad (4.10)$$

The parameters  $\bar{\beta}_{ij}$  are the inputs from the RMP and the decision variables  $S_{ij}$  define the column returned to RMP. The variables  $B_{ip}$  define the beam selections for use with the link set  $S_{ij}$ .  $V_i$  and  $D_{ij}$  are used internally to couple the SINR constraint, inequality (4.6), with  $S_{ij}$  and  $B_{ip}$ , respectively. The constraints are the same as in JBSS-MP (3.2 - 3.9), except that there is no  $l$  variable subscript, as only one link set is in consideration. A direct (centralized) specification of CLAP is given in Listing D.2 on page 251.

The problem RMP is trivial, but CLAP retains most of the original complexity of JBSS-MP. Crucially, however, it is no longer dimensioned over the set of all possible sets of links: For  $n$  nodes, the number of variables and the number of constraints are both  $\Theta(n^2)$ . The column-generation design allows a smaller problem to be solved many times rather than solving the larger problem once. Figure 4.2 on the following page shows an example of the progress of the column-generation scheme. In this case, the algorithm terminates after 15 RMP-CLAP iterations.

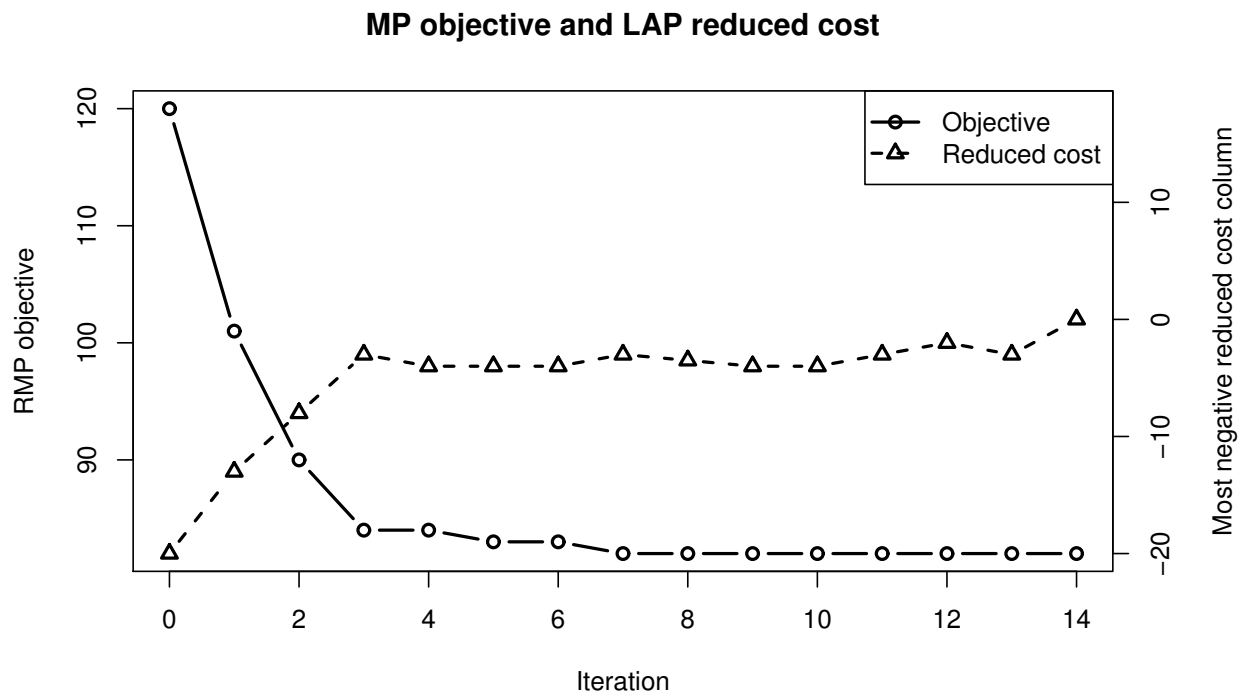


Figure 4.2: Effect of iterating between RMP and column-generation (Prototype 1). Note the different y scales for the objective value and reduced cost.

## 4.2 Convexity of CLAP

Many desirable optimization properties depend on the convexity of the feasible set and objective function. Knowing that a problem is convex makes it easy to infer **global** properties from **local** information; in fact local optimality suffices to show global optimality. This section will show that the constraint structure – but not objective – of CLAP has **abstract** convexity and will present an explicitly convex formulation: Convex-Constraint-CLAP. The requirement that certain variables take on integer or boolean values is irreducibly non-convex, however there is a large class of objective functions for which they prove unnecessary.

This discussion will begin with the SINR constraint (4.6), which is non-convex in its current form. The techniques introduced here, especially re-formulation as part of a geometric program (GP) in convex form, will be applied throughout. For reviews of geometric programming, including posynomial forms and the **log-sum-exp** transformation to convex form, see [Ecker 80, Boyd 04b, Chiang 05b, Boyd 07] and [Boyd 04a, §3.5, 4.5]. The formulation of Shannon capacity as a GP was introduced in [Foschini 93] and revisited in [Chiang 05b, Boyd 07, Tan 07, Tan 09], among others.

### 4.2.1 SINR Constraint

Constraint (4.6) can be re-written as follows: The “dummy” component of (4.6) served to keep the scheduling formulation, sans antenna consideration, in the linear domain. However, the  $(1 - S_{ij})$  component (in combination with an  $S_{ij}$  component elsewhere) prevents straightforward translation to a posynomial formulation. The following form (4.11) is closer to our natural understanding of SINR, and is easily re-written in posynomial form. The parameter  $\alpha > 0$  allows one to “tune” the function’s response to values of  $S$  between 0 and 1, without changing the value at those points. This can be thought of along the lines of an augmented Lagrangian.

$$\frac{P_i D_{ij} D_{ji}}{NrLb(i, j)} \geq S_{ij}^\alpha \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} V_k \frac{P_k D_{kj} D_{jk}}{NrLb(k, j)} \right) \quad \forall_{i, j} \quad (4.11)$$

Dividing by the (original) left-hand side and then switching left for right to make the inequality a  $\leq$  yields ineq. 4.12 on the next page:

$$\frac{S_{ij}^\alpha NrLb(i, j) \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_k V_k D_{jk} D_{kj}}{NrLb(k, j)} \right)}{P_i D_{ji} D_{ij}} \leq 1 \quad \forall_{i, j} \quad (4.12)$$

This constraint will be further manipulated, however its properties will first be demonstrated while it still closely resembles the original.

**Proposition 4.2.1.** *The left-hand side of inequality (4.12) is a posynomial functional in  $S$ ,  $V$ , and  $D$ , if  $\alpha \in \mathbb{Z}$*

*Proof.* Note that the elements of  $N_r$ ,  $P$ ,  $M$ , and  $Lb$  are scalar constants having values always  $\in \mathbb{R}_+$ . These terms are consequently monomial. Similarly, the elements of  $S$ ,  $V$ , and  $D$  are scalar variables having values  $\in \mathbb{R}_+$  and are monomial functions. Monomials are closed under multiplication, division, and exponentiation to integer powers, so the terms  $S_{ij}^\alpha \gamma_1$ ,  $P_k V_k D_{jk} D_{kj}$ ,  $NrLb(k, j)$ ,  $P_i D_{ij} D_{ji} S_{ij}$ , and  $NrLb(i, j)$  are also monomial.

A sum of monomials is posynomial, as is a posynomial divided by a monomial. Let  $m$  and  $p$  denote any monomial and any polynomial, respectively, in the variables discussed, and  $\rightarrow$  denote inferring a new proposition about multiple terms. By repeated inference, the LHS of (4.12) can be shown to be a posynomial:

$$\frac{m(m + \frac{m}{m} + \frac{m}{m} + \dots)}{\frac{m}{m}} \rightarrow m \frac{m + m + m + \dots}{m} \rightarrow m \frac{p}{m} \rightarrow p$$

□

**Proposition 4.2.2.** *Inequality (4.12) is a geometric program constraint in standard form for any set of variables  $X \supseteq \{S, V, D\}$ .*

*Proof.* Ineq. (4.12) is of the form  $f(S, V, D) \leq 1$ ,  $f$  posynomial by Proposition 4.2.1. For any variable  $x \in X \setminus \{S, V, D\}$ , let  $f'(S, V, D, x) = f(S, V, D) * x^0$ . The term  $x^0$  is an integer power of a monomial, and therefore a monomial, and therefore a posynomial. The set of posynomials is closed under multiplication, so  $f' = f(S, V, D) * x^0$  is a posynomial functional of  $S, V, D$ , and  $x$ .

Consequently,  $f'(S, V, D, x) \leq 1$  is a GP constraint in standard form for the variables  $S, V, D$ , and  $x$  [Boyd 04b].

Moreover,  $x^0 = 1 \forall x$ , so  $f(S, V, D) * x^0 = f'(S, V, D, x) = f(S, V, D) \forall_{S, V, D, x}$ . Since it is identical to the constraint written in  $f'$ ,  $f(S, V, D) \leq 1$  is also a GP constraint in standard form for the variables  $S, V, D$ , and  $x$ . By induction on  $x' \in X \setminus \{S, V, D, x\}$ , the result holds for all of  $X$ .  $\square$

**Proposition 4.2.3.** *The **log-sum-exp** transformation of inequality (4.12) defines a convex region in any variables  $X \supseteq \{S, D, V\}$ .*

*Proof.* By Proposition 4.2.2, Ineq. (4.12) is a GP constraint in standard form for  $X$ . It follows immediately that the transformed version is a **geometric constraint in convex form**, which is, in fact, convex [Boyd 07, §2.5].  $\square$

In order to facilitate this transformation, we first algebraically simplify (4.12) to the following (4.13). It is easily verified that the LHS of (4.13) is still posynomial (it is a simple sum of monomials) and that it is exactly equal to (4.12).

$$\frac{Nr\gamma_1 Lb(i, j) S_{ij}^\alpha}{P_i D_{ij} D_{ji}} + \sum_{k \in N \setminus \{i, j\}} \frac{P_k V_k \gamma_1 D_{jk} D_{kj} Lb(i, j) S_{ij}^\alpha}{P_i D_{ij} D_{ji} Lb_{kj}} \leq 1 \quad \forall_{i, j} \quad (4.13)$$

The transformation is presented here: For any variable  $x$ , let  $\tilde{x}$  denote the natural log of  $x$ . The inequality becomes:

$$\log \left( e^{-\tilde{P}_i - \tilde{D}_{ij} - \tilde{D}_{ji} + \alpha \tilde{S}_{ij}} Nr\gamma_1 Lb(i, j) + \sum_{k \in N \setminus \{i, j\}} \frac{e^{\tilde{P}_k + \tilde{V}_k + D_{jk} + D_{kj} - D_{ij} - D_{ji} + \alpha \tilde{S}_{ij} \gamma_1 Lb(i, j)}}{P_i Lb(k, j)} \right) \leq 0 \quad \forall_{i, j} \quad (4.14)$$

The inequality (4.14) constitutes part of a convex program in the log-domain variables  $\tilde{D}, \tilde{V}$ , and  $\tilde{S}$ . Any value  $\tilde{x}$  satisfying these inequalities corresponds directly to an  $x : \tilde{x} = \log(x)$  satisfying (4.11).



### 4.2.2 Objective Function

Let us consider the remaining components of CLAP in sequence: The objective (4.4) constitutes a **concave maximization** in the linear domain as long as the constants  $\bar{\beta} \geq 0$ , which is true by definition. Because  $\bar{\beta}$  are positive, this is a posynomial functional. Transformed to convex GP form, the objective becomes:

$$\max_{\tilde{S}} \sum_{ij} \log \left( e^{\tilde{S}_{ij} + \log(\bar{\beta}_{ij})} \right)$$

The value  $\log(0)$  is generally taken to be either  $-\infty$  or undefined. For the purpose of any numerical solution process, it is undefined and constitutes a problematic input. Since  $\beta$  can easily be 0, we write the objective as (4.15), where  $\epsilon > 0$ .

$$\max_{\tilde{S}} \sum_{ij} \log \left( e^{\tilde{S}_{ij} + \log(\bar{\beta}_{ij} + \epsilon)} \right) \quad (4.15)$$

An analogous issue occurs in multiple places in this formulation: Geometric programs are in general defined only over the strictly positive domain  $\mathbb{R}_{++}^n$ . This does not limit us in any meaningful way, as they are well defined for arbitrarily small  $\epsilon$ , and the natural domains of our variables are all  $\geq 0$ . In general,  $\epsilon$  must be chosen to be small enough not to alter the solution materially, but large enough to avoid numerical instability or rounding to 0. In the context of CLAP, there are no variables for which it is important to maintain a distinction between a value of 0 and very nearly 0, so a constraint of the form  $\epsilon < x \leq 1$  is functionally equivalent to  $0 \leq x \leq 1$ . The variables  $S$ ,  $V$ , and  $B$  are ultimately 0 – 1 integers, so values near  $\epsilon$  can safely be interpreted as 0.

This transformation, however, produces a (strictly) convex maximization problem. The linear-domain objective is affine, meaning that it is both convex and concave, and therefore well-behaved as either a maximization or minimization objective. A strictly convex maximization objective, on the other hand, is equivalent to a strictly concave minimization objective, which is undesirable. It is possible to define many convex objective functions which are equivalent to our original **in the discrete case**, but they behave badly in the continuous case. This is not merely a technicality: In the way we generally understand it, the value of “activating” a link is essentially

binary – a partially-activated link is either an undefined concept, or a rather low value case.\* A continuous version of the problem is acceptable, so long as the optimal solutions happen to fall on discrete values (the integrality gap is 0). I am not aware of any convex function of the logarithm of link activations which has this property, and indeed I strongly suspect that none is possible.

**Definition 4.1** (0-1 Dominance Property). *Consider an minimization objective function  $f(x_b, x_c) : X \rightarrow \mathbb{R}$  defined on a set  $X \subset \mathbb{R}^{n+m}$ , where  $x_b \subset \mathbb{R}^n$ ,  $x_c \subset \mathbb{R}^m$ . Say that  $f$  has 0-1 dominance with regard to  $x_b$  and  $X$  if and only if: For any  $x_b, x_c \in X$ ,  $x_b \notin \{(0, \epsilon], 1\}^n$ ,  $\exists x'_b, x'_c \in X$ ,  $x_b \in \{(0, \epsilon], 1\}^n$  s.t.  $f(x'_b, x'_c) \leq f(x_b, x_c)$ .*

**Conjecture 4.2.1.** *For any set  $X$  such that  $X' = (\log(x_b), x_c) \forall x_b, x_c \in X$  is closed and convex, define  $f'(\log(x_b), x_c) : X' \rightarrow \mathbb{R} \equiv f(x_b, x_c)$ . Conjecture:  $\nexists f, f'$  s.t.  $f$  has 0-1 dominance with regard to  $x_b$  and  $f'$  is convex.*

The argument for this conjecture proceeds from the fact that  $\left. \frac{\delta \log(x)}{\delta x} \right|_1 = 1$  and  $\lim_{x \rightarrow +0} \left. \frac{\delta \log(x)}{\delta x} \right|_x = +\infty$ . For any  $f'$  which is negatively linear or super-linear in  $\log(x_b)$ , and a feasible space which allows a linear “trade off” between variables  $x_{b1} \approx 0$  and  $x_{b2} \approx 1$ , moving from  $f'(\approx 0, \approx 1, \dots)$  to  $f'(\approx 0 + \delta, \approx 1 - \delta, \dots)$  will lower (improve) the objective function. A real proof requires expressing the notion of a “trade off” formally (which follows from the KKT conditions), and generalizing beyond the linearity assumptions.

Because of the above limitation, I refer the formulation developed in this section as Convex-Constraint-CLAP to emphasize that the objective function of CLAP is not preserved in this transformation.

---

\* Note that activating a link for a fraction of the total time is perfectly reasonable, but in this context we are discussing which links are active in a given slice of time (or frequency, or code space).

### 4.2.3 Half-Duplex Constraint

The duplex constraint (4.5) is already a posynomial inequality. Transforming to the log domain gives:

$$\log \left( \sum_{j \in N \setminus \{i\}} e^{\tilde{S}_{ij}} + e^{\tilde{S}_{ji}} \right) \quad (4.16)$$

### 4.2.4 S-V Coupling Constraint

The first coupling constraint, (4.7), can be re-written as below:

$$\frac{S_{ij}}{V_i} \leq 1 \quad \forall_{ij}$$

Transformed into the log domain, that gives:

$$\log \left( e^{\tilde{S}_{ij} - \tilde{V}_i} \right) \leq 0 \quad \forall_{ij} \quad (4.17)$$

### 4.2.5 D-B (Antenna) Coupling Constraint

The antenna coupling constraint, equation (4.8), does not admit a trivial transformation to a GP, as it is naturally a (non-monomial) posynomial **equality**. This difficulty can be side-stepped by formulating the constraint **directly** in the log domain, rather than writing it in the linear domain and then transforming it. Consider the following equality (4.18) where the constant  $\tilde{G}_{ikp}$  denotes the antenna gain in log units of node  $i$  toward node  $k$ , when using antenna pattern index  $p$ :

$$-\tilde{D}_{ik} + \sum_p B_{ip} \tilde{G}_{ikp} = 0 \quad \forall_{ik} \quad (4.18)$$

**Proposition 4.2.4.** *The left-hand side of equality (4.18) defines an affine function.*

*Proof.* Noting that  $\tilde{G}_{ikp}$  is a constant for all  $i, k$ , and  $p$ , the result is immediate.  $\square$

**Proposition 4.2.5.** *Equality (4.18) defines a valid constraint for a convex program in the variables  $\tilde{D}$  and  $B$ .*

*Proof.* An equality of the form  $f(x) = b$ , where  $f(x)$  is affine and  $b$  is a constant, defines a hyperplane. Given any convex optimization program  $\mathcal{P}$ , its feasible set is convex by definition. The

intersection of a convex set with a hyperplane is a (possibly empty) convex set. By induction over all  $ik$ , adding (4.18) to any otherwise valid convex optimization program produces a valid convex optimization program.  $\square$

Note that while (4.18) is not produced by converting a GP to convex form, its semantics are consistent with the objective and constraints produced that way. In particular, it avoids the pitfall in mixed linear geometric programming of using the same variable in the linear and logarithmic domains (see [Boyd 07]).<sup>†</sup> Constraint (4.18) is not directly interchangeable with (4.8). Consider the simplified case below:

$$\begin{aligned} k_1 x_1 + k_2 x_2 &= k_3 \\ \log(k_1) x_1 + \log(k_2) x_2 &= \log(k_3) \end{aligned}$$

These equations do not in general have the same solution sets. However, when we add the assumptions  $x_1, x_2 \in \{0, 1\}$ ,  $\sum x \leq 1$ ,  $k \in \mathbb{R}_{++}^3$ , we quickly derive  $k_3 \in \{k_1, k_2\}$  and  $\log(k_3) \in \{\log(k_1), \log(k_2)\}$  respectively. Over the given domains, these have the same solution sets.

**Proposition 4.2.6.** *The optimization program with constraint (4.18) has at least one optimal solution value in common with the same program constrained instead by (4.8).*

*Proof.* Following the proof of proposition 4.4.2 on page 95,  $\exists$  a minimizer  $B^*$  for which  $B_{ip} \in [0, 1] \forall i, p$  and  $\sum_p = 1 \forall i$ . Let  $p^1$  denote the index of  $B^*$  having value 1 for any given  $i$ .  $B^*$  satisfies (4.8) iff  $G_{ikp^1} = D_{ik} \forall i, k$ .  $B^*$  satisfies (4.18) iff  $\log(G_{ikp^1}) = \log(D_{ik}) \forall i, k$ , which is true exactly when  $G_{ikp^1} = D_{ik} \forall i, k$ .  $\square$

This means that if there is a unique optimal solution, it is the same regardless of which version of the constraint one uses. If there are multiple optimal solutions, any solution not satisfying both constraints could be discarded post hoc. Consequently, the two forms are equivalent for all purposes of interest here.

---

<sup>†</sup> As an aside, using the same variable in both domains is impossible because  $x$  and  $\tilde{x} \triangleq \log(x)$  are **not** the same variable. The “ $\triangleq$ ” exists in the modeler’s head but not in the actual program, and therein lies the problem. Further, adding a constraint of the form  $\log(x) = \tilde{x}$  is problematic precisely because  $\log(x)$  is not affine.

#### 4.2.6 Convex Pattern Combination Constraint

**Proposition 4.2.7.** *Constraint (4.9) defines a valid constraint for a convex program in the variables  $B$ .*

*Proof.* Proof parallels that of Propositions 4.2.4 to 4.2.5 on page 83. The LHS is affine and the RHS is constant. The variables appearing,  $B$ , appear only in the linear domain throughout the program.  $\square$

#### 4.2.7 The Convex-Constraint-CLAP Program

Consider now the following program:

Convex-Constraint-CLAP:

$$\begin{aligned}
 \max \quad & f_0 \quad \text{any concave objective} \\
 \text{s.t.} \quad & \text{SINR constraint (4.14)} \\
 & \text{Single pattern constraint (4.9)} \\
 & \text{Duplex constraint (4.16)} \\
 & \text{S-V Coupling constraint (4.17)} \\
 & \text{D-B Coupling constraint (4.18)}
 \end{aligned}
 \tag{4.19}$$

**Proposition 4.2.8.** *Convex-Constraint-CLAP (4.19) is a family of convex programs.*

*Proof.* The objective function must be convex minimization (or concave maximization), and constraints (4.14) through (4.17) are  $\leq$ -type convex inequalities. This follows directly from their construction as a geometric program in convex form. All variables appearing in these terms have natural domains within  $\mathbb{R}_{++}^n$ . Constraints (4.18) and (4.9) are affine equality constraints. No variable appears both directly and in a log-transformed analogue. The objective and constraints are all well-defined over the problem's intended domain.  $\square$

**Proposition 4.2.9.** *The program Convex-Constraint-CLAP is equivalent to CLAP if  $f_0 = (4.15)$  or  $f_0(S, \dots) = (4.15)$  on the restricted set  $S \in \{0, 1\}^n$  and  $S$  is so constrained.*

*Proof.* Equivalence is a loosely-defined notion in this context. In this case, we mean the following: (1) For any instance  $\mathcal{P}$  of CLAP, there is a procedure for producing a program  $\mathcal{P}'$  which is an instance of Convex-Constraint-CLAP. (2) For any  $X'^*$  which is a set of optimal values for  $\mathcal{P}'$ , there is a procedure for producing a set  $X^*$  which are optimal values for  $\mathcal{P}$  within arbitrarily small error  $\epsilon$ . For part (1), such a procedure is defined in the discussion of each functional in CLAP above. For part (2), the procedure is as follows: for any variable  $\tilde{x}$  which was converted to the logarithmic domain, let  $x = e^{\tilde{x}}$ . If  $X^*$  is not a singleton, remove any  $x^* \in X^*$  violating any constraint of  $\mathcal{P}$ . Proposition 4.2.6 guarantees that  $X'^*$  contains at least one element which is feasible for  $\mathcal{P}$ , but allows the possibility of others which are not. For the other constraints, it follows directly from the definitions of their convex equivalents that an identical feasible region under the transformation described is defined in  $\mathcal{P}$  and  $\mathcal{P}'$ . It similarly follows that the objective functions are identical under the same transformation.  $\square$

#### 4.2.8 Pseudo-Integral Convex-Constraint-CLAP

Notwithstanding the integrality properties of FLAP and FARP, Convex-Constraint-CLAP and the continuous relaxation of CLAP both admit non-integer solutions. In particular, non-integer values of  $S$  (and consequently  $V$ ) are possible and are observed in practice. This is important when Convex-CLAP is being solved directly, as fractional values of  $S$  undermine the semantics of the duplex constraint by allowing solutions like  $S_{12} = 0.6$ ,  $S_{32} = 0.4$ .

One way of handling this is to explicitly constrain  $S \in \{0, 1\}^n$  and solve the resulting mixed-integer program, but that approach forgoes the advantages of convexity. There is, however, a convexity-preserving solution.

For every pair of links which “conflict” under the duplex constraint (*e.g.*  $k \rightarrow i$  and  $i \rightarrow j$ ),

define the **Strong Duplex Property** as follows:

$$S_{ki}S_{ij} \leq 0$$

The strong duplex property is satisfied if and only if one or both of the links is completely off. The types of “conflicting” links enumerated by  $\forall_{i,j,k,l: (k \in \{i,j\} \vee l \in \{i,k\}) \wedge (i,j) \neq (k,l)}$  are shown visually in Figure 4.3 below.

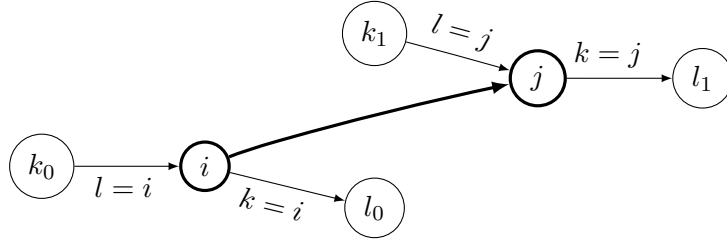


Figure 4.3: Links for which “strong duplex” constraints are defined, relative to link  $i \rightarrow j$ .

Writing this in standard GP form gives (4.20):

$$S_{ij}S_{kl} + 1 \leq 1 \quad \forall_{i,j,k,l: (k \in \{i,j\} \vee l \in \{i,k\}) \wedge (i,j) \neq (k,l)} \quad (4.20)$$

Translating into the log domain, and allowing for strict positivity, this becomes:

$$\log(e^{\tilde{S}_{ij} + \tilde{S}_{kl}} + 1) \leq \epsilon \quad \forall_{i,j,k,l: (k \in \{i,j\} \vee l \in \{i,k\}) \wedge (i,j) \neq (k,l)} \quad (4.21)$$

Adding constraint (4.21) to the program Convex-Constraint-CLAP gives Pseudo-Integral Convex-Constraint-CLAP. This program is convex while yielding  $\epsilon$ -integral solutions. Additionally, it is equivalent to Convex-Constraint-CLAP within a small margin when  $S \approx 0$  or 1. The AMPL representations of Convex-Constraint-CLAP and Pseudo-Integral Convex-Constraint-CLAP are given in Listing D.3 on page 252. The problem CCLAP as written is the pseudo-integral variant. Removing the constraints “CCLAP\_DUPLEX\_PAIRWISE...” gives Convex-Constraint-CLAP.

Note that the effect of adding strong duplex constraints on the problem’s decomposability is not considered here; indeed, decomposition of Pseudo-Integral Convex-Constraint-CLAP is not addressed in this dissertation.

Functional	Degree	Variables			
		$S$	$V$	$D$	$B$
Objective: (4.4)	1	✓			
(4.5)	1	✓			
(4.7)	1	✓	✓		
Constraint (4.6)	3	✓	✓	✓	
(4.8)	1			✓	✓
(4.9)	1				✓

Table 4.1: Constraint matrix structure of CLAP

### 4.3 Second Decomposition: Lagrangian Relaxation on CLAP

Consider CLAP, given on page 76. This system of equations presents a near-block structure, shown in table 4.1. Note that (4.7) is shown above (4.6). Without inequality (4.6) (the SINR constraint), the problem decomposes into two sub-problems: One involving the variables  $S$  and  $V$  (link and node activation), and the other involving  $D$  and  $B$  (antenna gain and pattern). Note also that the primary computational difficulty in CLAP comes from constraint (4.6) which is still order 3 and mixed-integer.

Let us define vector-valued convenience function  $d^s(\cdot)$ , entry  $ij$  of which is given by:

$$d^s(S, D, V)_{ij} \triangleq - \left( \frac{P_i D_{ij} D_{ji}}{Lb(i, j) N_r} S_{ij} + \gamma_1 (1 + M_{ij}) (1 - S_{ij}) - \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_k D_{kj} D_{kj}}{Lb(k, j) N_r} V_k \right) \right) \quad (4.22)$$

Note that constraint (4.6) is then  $d^s(S, D, V) \leq 0$ . Let the Lagrangian function with regard to (4.4) and (4.6) be:

$$\mathcal{L}(S, \lambda) = \bar{\beta}^T S - \lambda^T d^s(S, D, V) \quad (4.23)$$

This gives a dual function:

$$\phi(\lambda) = \max_{S, D, V} \mathcal{L}(S, D, V, \lambda) \quad (4.24)$$

The corresponding Lagrangian dual problem is CLAP-dual-1, below:



[CLAP-dual-11]

$$\begin{aligned}
& \min_{\lambda} && \phi(\lambda) \\
& \text{s.t.} && \sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} S_{ji} \leq 1 && \forall i \\
& && S_{ij} \leq V_i && \forall i \\
& && D_{ik} = \sum_{p \in P} G_{ikp} B_{ip} && \forall i, k \\
& && \sum_{p \in P} B_{jp} = 1 && \forall j \\
& && S_{ij}, B_{jp} \in \{0, 1\} && \forall i, j, p
\end{aligned} \tag{4.25}$$

The resulting Lagrangian relaxed primal problem (RPP) of CLAP is given in 4.26, where  $\bar{\lambda}$  denotes an **estimate** of the optimal multipliers  $\lambda^*$ .

$$\begin{aligned}
& \max_{S, D, V} && \bar{\beta}^T S + \bar{\lambda}^T d^s(S, D, V) \\
& \text{s.t.} && \text{constraints (4.5) – (4.9)} \\
& && \text{except (4.6).}
\end{aligned} \tag{4.26}$$

This RPP is block-structured and separable into two subproblems (given in 4.27) coupled by the Lagrange multipliers  $\lambda$ . I label these the Fixed-antenna Link Activation Problem (**FLAP**) and the Fixed-link Antenna Reconfiguration Problem (**FARP**). FLAP takes estimated Lagrange multipliers and antenna gains  $\bar{\lambda}$ ,  $\bar{D}$  as parameters and computes link activations  $S$ . Conversely, FARP takes  $\bar{\lambda}$  and estimated link activations  $\bar{S}$  as parameters and computes antenna gains  $D$ :

$$\begin{aligned}
& \max_{S, V} && \beta^T S + \bar{\lambda}^T d(S, \bar{D}, V) && \max_D && \beta^T \bar{S} + \bar{\lambda}^T d(\bar{S}, D, \bar{V}) \\
& \text{s.t.} && \text{ineq. (4.5)} && \text{s.t.} && \text{ineq. (4.8)} \\
& && \text{ineq. (4.7)} && && \text{ineq. (4.9)}
\end{aligned} \tag{4.27}$$

Concretely, FLAP is given by (4.28) and FARP is (4.29) on the next page.

$$\begin{aligned}
 & \text{[FLAP]} \\
 & \max_{S, V} \left\{ \begin{aligned} & \bar{\beta}^T S - \sum_{ij} \bar{\lambda}_{ij} \left( \frac{P_i \bar{D}_{ij} \bar{D}_{ji}}{Lb(i, j) N_r} S_{ij} + \right. \\ & \quad \left. \gamma_1 (1 + M_{ij}) (1 - S_{ij}) - \right. \\ & \quad \left. \left. \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_k \bar{D}_{kj} \bar{D}_{jk}}{Lb(k, j) N_r} V_k \right) \right) \right\} \quad (4.28) \\
 & \text{s.t.} \quad \sum_{j: (i, j) \in A} S_{ij} + \sum_{j: (j, i) \in A} S_{ij} \leq 1 \quad \forall i \\
 & \quad S_{ij} \leq V_i \quad \forall i, j \\
 & \quad S_{ij} \in 0, 1 \quad \forall i, j
 \end{aligned} \right.
 \end{aligned}$$

This problem has the integrality property, and so the constraint  $S_{ij} \in 0, 1 \forall i, j$  can be dropped.

**Proposition 4.3.1.** *The continuous relaxation of FLAP is equivalent to FLAP with integer  $S$ .*

*Proof.* The constraint matrix of continuous FLAP is totally unimodular by Ghouila-Houri's Theorem. Therefore every extreme point of the feasible polytope is in  $\mathbb{Z}^n$ . The function to be maximized is concave, implying that no maximum occurs within the feasible polytope, and therefore that the constrained optimum occurs at an extreme point. Therefore the integer optimum and continuous optimum occur at the same point. The inverse of the constraint matrix is also totally unimodular by Cramer's Rule, and so the same argument holds for the dual problem.  $\square$

$$\begin{aligned}
& \text{[FARP]} \\
& \max_{D,B} \left\{ \begin{aligned} & \bar{\beta}^T \bar{S} - \sum_{ij} \bar{\lambda}_{ij} \left( \frac{P_i D_{ij} D_{ji}}{Lb(i,j)N_r} \bar{S}_{ij} + \right. \\ & \quad \left. \gamma_1 (1 + M_{ij})(1 - \bar{S}_{ij}) - \right. \\ & \left. \left. \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i,j\}} \frac{P_k D_{kj} D_{jk}}{Lb(k,j)N_r} \bar{V}_k \right) \right) \right\} \quad (4.29) \\
& \text{s.t. } D_{ik} - \sum_{p \in P} G_{ikp} B_{ip} = 0 \quad \forall_{i,k} \\
& \quad \sum_{p \in P} B_{ip} = 1 \quad \forall_i
\end{aligned}
\right.
\end{aligned}$$

The AMPL statements of the FLAP and FARP programs are given in Listings D.4, D.5, and D.6 on pages 257–278<sup>‡</sup>. Note that a Lagrangian decomposition on a non-convex problem introduces the possibility of a **duality gap**. Techniques for eliminating the duality gap include Augmented Lagrangians [Grothey 01] and Lagrange-type functions with convolution functions [Rubinov 03]. The approach pursued here avoids the need to for partitioning, but see discussion in [Dür 97, Horst 00]. The problem Convex-CLAP avoids the possibility of a duality gap entirely. Problems (4.28) and (4.29) look rather messy arithmetically, but the structure becomes clearer when constants are factored out. Note that the estimates  $f^{\bar{o}}$  are constants with regard to the current iteration of either problem. Let  $K_1, \dots, L_n$  denote the set of constants occurring in either problem. The two problems can then be rewritten as (4.30) and (4.31):

$$\begin{aligned}
& \max_{S,V} K_1 S + \sum_{ij} \bar{\lambda}_{ij} \left( K_2 S_{ij} + K_3 + K_4 S_{ij} - \gamma_1 - \sum_{k \in N \setminus \{i,j\}} K_5 V_k \right) \\
& \text{s.t. } \sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} S_{ij} \leq 1 \quad \forall_i \quad (4.30) \\
& \quad S_{ij} \leq V_i \quad \forall_i
\end{aligned}$$

$$\begin{aligned}
& \max_{D,B} K_7 + \sum_{ij} \bar{\lambda}_{ij} \left( K_7 D_{ij} D_{ji} + K_8 - \gamma_1 - \sum_{k \in N \setminus \{i,j\}} K_9 D_{kj} D_{jk} \right) \\
& \text{s.t. } D_{ij} - \sum_{p \in P} K_{10} B_{jp} = 0 \quad \forall_{i,j} \quad (4.31) \\
& \quad \sum_{p \in P} B_{jp} = 1 \quad \forall_j
\end{aligned}$$

---

<sup>‡</sup> These files contain multiple variants of both problems.

The substitutions are shown in table 4.2. Note that they are constants in the sense of not involving any decision variables, but they do take different values for different  $i, j, k$ .

$$\begin{array}{ll}
K_1 & = \beta^T \\
K_2 & = \frac{P_i \bar{D}_{ij} \bar{D}_{ji}}{Lb(i,j)N_r} \\
K_3 & = \gamma_1 - \gamma_1 M_{ij} \\
K_4 & = -\gamma_1 (1 + M_{ij}) \\
K_5 & = \gamma_1 \frac{P_k \bar{D}_{kj} \bar{D}_{jk}}{Lb(k,j)N_r} \\
K_6 & = \beta^T \bar{S} \\
K_7 & = \frac{P_i \bar{S}_{ij}}{Lb(i,j)N_r} \\
K_8 & = \gamma_1 (1 + M_{ij})(1 - \bar{S}_{ij}) \\
K_9 & = \gamma_1 \frac{P_k \bar{V}_k}{Lb(k,j)N_r} \\
K_{10} & = G_{ikp}
\end{array}$$

Table 4.2: Constant substitutions for LR decomposition.

The decomposition shown in (4.27) - (4.31) has the following properties: The  $S, V$  DPP is a linear program. Its size is fairly large. There are  $O(n^2)$  variables where  $n$  is the number of nodes.  $S$  is a vector of potential size  $n^2$ , and  $V$  is a vector of potential size  $n$ . In practice, potential links  $i, j$  for which there is no link-layer demand need not be considered at all, so  $|S|$  may be substantially smaller than  $n^2$ . In a network which is small enough (relative to the communication technology) that any node can potentially reach any other, up to  $n(n-1)$  links may exist. As a network grows larger (spatially) it is reasonable to assume that any given node can only communicate with some local neighborhood, so the number of possible links per node (the node degree) is limited by the network's density, not its size. In that case, the expected number of possible links is  $O(n)$ . For a spatially-large network, then, both  $|S|$  and  $|V|$  are  $O(n)$ .

#### 4.4 Block Separability of FARP

FARP is given by equation (4.29) on the previous page. Note that it is a program with linear constraints and a quadratic objective function. Unlike linear functions, which are by nature convex (or concave), the convexity of a quadratic function depends on the coefficients of the quadratic terms. The structure of FARP is in general not necessarily convex, but it has a number of nice properties which make its computation tractable.

**Proposition 4.4.1.** *FARP has convex-anticonvex structure, as defined by [Hiriart-Urruty 98, §3.1].*

*Proof.* Let  $x$  refer to  $D, B$  and  $f(x)$  denote the objective function of FARP from (4.29), repeated here:

$$\beta^T \bar{S} - \sum_{ij} \bar{\lambda}_{ij} \left( \frac{P_i D_{ij} D_{ji}}{Lb(i, j) N_r} \bar{S}_{ij} + \gamma_1 (1 + M_{ij}) (1 - \bar{S}_{ij}) - \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_k D_{kj} D_{jk}}{Lb(k, j) N_r} \bar{V}_k \right) \right)$$

Let  $k$  denote the (constant) contribution of all terms in  $f(x)$  not dependent on  $x$ . Let:

$$g(x) \triangleq \sum_{ij} \left( \bar{\lambda}_{ij} \bar{S}_{ij} \frac{P_i}{Lb(i, j) N_r} D_{ij} D_{ji} \right) + k \quad (4.32)$$

$$h(x) \triangleq \sum_{ij} \sum_{k \in N \setminus \{i, j\}} \left( \gamma_1 \bar{V}_k \bar{\lambda}_{ij} \frac{P_k}{Lb(k, j) N_r} D_{kj} D_{jk} \right) \quad (4.33)$$

$$(4.34)$$

Let  $f'(x) = -f(x)$ . Note that FARP is stated as  $\max f(x)$ , which is equivalent to  $\min f'(x)$ . Now,

$$f'(x) = g(x) - h(x)$$

The coefficients  $\bar{\lambda}_{ij} \bar{S}_{ij} \frac{P_i}{Lb(i, j) N_r}$  and  $\gamma_1 \bar{V}_k \bar{\lambda}_{ij} \frac{P_k}{Lb(k, j) N_r}$  are always  $\geq 0$ . Consequently, for any given  $i, j, k$ ,  $\bar{\lambda}_{ij} \bar{S}_{ij} \frac{P_i}{Lb(i, j) N_r} D_{ij} D_{ji}$  and  $\gamma_1 \bar{V}_k \bar{\lambda}_{ij} \frac{P_k}{Lb(k, j) N_r} D_{kj} D_{jk}$  are concave functionals. The sum combination of concave (convex) functionals is itself concave (convex). Being constant,  $k$  is convex and concave.

Therefore,  $g(x)$  and  $h(x)$  are both concave. Consequently  $f'(x)$  is convex-anticonvex.  $\max f(x)$  is equivalent to  $\min f'(x)$ , and therefore the objective function of FARP is convex-anticonvex.  $\square$

Note that **convex-anticonvex** structure is a special case of **difference of convex** or simply **D.C.** structure. A review of D.C. programming is beyond the scope of this dissertation, however excellent discussion is provided in [Shor 98, Horst 00, Hiriart-Urruty 98, Grothey 01].

Furthermore, FARP is almost separable. Consider the following algebraic transformation of  $h(x)$ . This groups the penalties for interference with the interferer as well as the interferee.

$$h'(x) \triangleq \sum_{i \in N} \left( \sum_{j \in N} \sum_{k, l \in N \setminus \{i, j\}} \left( \frac{1}{2} \gamma_1 \bar{V}_i \bar{\lambda}_{kl} \frac{P_i}{Lb(i, l) N_r} D_{il} D_{li} \right) \right) \quad (4.35)$$

$$f''(x) \triangleq g(x) - h'(x) \quad (4.36)$$

Observe that  $h' * (x)$  is equivalent to  $h(x)$ , and therefore  $f''(x)$  is equivalent to  $f'(x)$ . Therefore, FARP can be restated as equation (4.37).

$$\begin{aligned} \max_{D,B} \quad & 1 - f''(D, B) \\ \text{s.t.} \quad & D_{ij} - \sum_{p \in P} P_{ijp} B_{jp} = 0 \quad \forall_{i,j} \\ & \sum_{p \in P} B_{jp} = 1 \quad \forall_j \end{aligned} \quad (4.37)$$

Note that in equation (4.37) all of the constraints pertaining to a given node  $i$  pertain **only** to node  $i$ . Observe also that  $f''(x)$  can be read as a sum  $\sum_i f''_i(x_i)$ , where  $x_i \triangleq \cup_{k \neq i} \{D_{ik}, D_{ki}\}$ . Note that the variables  $x_i$  form a partition of  $x$ . Unfortunately, the  $x_i$  appear in the constraints of nodes other than  $i$ , and so equation (4.37) is not separable by that definition. If, rather,  $x$  is divided into  $x_i \triangleq \cup_{k \neq i} D_{ik}$ , this restores the separability of the constraints, but each  $f''_i(x)$  involves more than  $x_i$ .

This makes intuitive sense: The CLAP SINR constraint  $d(\cdot)$ , and consequently the function  $f''(x)$ , pertains to absolute power received at each destination. The effect, in linear units of power at  $k$ , of changing the gain  $D_{ik}$  depends on  $D_{ki}$ .

Using the division  $x_i \triangleq \cup_{k \neq i} D_{ik}$  produces the following formulations. Note that  $\bar{\beta}^T \bar{S}$  is a constant and is dropped for simplicity in subsequent formulations:

$$g_i(x) \triangleq \begin{cases} \sum_j \left( \frac{1}{2} \bar{\lambda}_{ij} \bar{S}_{ij} \frac{P_i}{Lb(i,j)N_r} D_{ij} \bar{D}_{ji} \right) + \frac{k}{|N|} & \text{if } i \text{ is a transmitter} \\ \sum_j \left( \frac{1}{2} \bar{\lambda}_{ji} \bar{S}_{ji} \frac{P_j}{Lb(j,i)N_r} \bar{D}_{ji} D_{ij} \right) + \frac{k}{|N|} & \text{if } i \text{ is a receiver} \end{cases} \quad (4.38)$$

$$h'_i(x) \triangleq \begin{cases} \sum_j \left( \sum_{k,l \in N \setminus \{i,j\}} \left( \frac{1}{2} \gamma_1 \bar{S}_{ij} \bar{\lambda}_{kl} \frac{P_i}{Lb(i,l)N_r} D_{il} \bar{D}_{li} \right) \right) & \text{if } i \text{ is a transmitter} \\ \sum_j \left( \sum_{k,l \in N \setminus \{i,j\}} \left( \frac{1}{2} \gamma_1 \bar{S}_{ji} \bar{\lambda}_{ji} \frac{P_k}{Lb(k,i)N_r} \bar{D}_{ki} D_{ik} \right) \right) & \text{if } i \text{ is a receiver} \end{cases} \quad (4.39)$$

$$f''_i(x) = g_i(x_i) - h'_i(x) \quad (4.40)$$

$$f''(x) = \sum_i f''_i(x) \quad (4.41)$$

Note that the transmitter and receiver cases of equation (4.38) on the previous page together describe both ends of an intended link, while equation (4.39) on the preceding page describes both ends of an unwanted transmission. Additionally, note that the division into cases is not actually necessary:  $\bar{S}_{ij} \neq 0$  if and only if  $i$  is a transmitter, and  $\bar{S}_{ji} \neq 0$  if and only if  $i$  is a receiver, so the cases can simply be added together. Therefore, a Lagrangian decomposition of FARP is possible as follows below. **The Single-Node Antenna Reconfiguration Problem (SNARP)** is given in problem 4.42.

[SNARP<sub>*i*</sub>]

$$\max_{D,B} \quad 1 - f'_i(D, B) \quad (4.42a)$$

$$\text{s.t.} \quad D_{ik} - \sum_{p \in P} G_{ikp} B_{ip} = 0 \quad \forall k \quad (4.42b)$$

$$\sum_{p \in P} B_{ip} = 1 \quad (4.42c)$$

$$B_{ip} \leq 1 \quad \forall p \in P \quad (4.42d)$$

$$B_{ip} \geq 0 \quad \forall p \in P \quad (4.42e)$$

The SNARP<sub>*i*</sub> subproblems are given in AMPL form in Listing D.5; see especially line 213 on page 274.

**Proposition 4.4.2.** *SNARP<sub>*i*</sub> with continuous variables has an optimal solution equal to that with boolean  $B_{ip}$ .*

*Proof.* SNARP<sub>*i*</sub> is a linear program in  $D, B$ , but can be re-written purely in  $B$  by substituting  $\sum_{p \in P} G_{ikp} B_{ip}$  for  $D_{ik}$  in the objective function. So written, it is a linear program with  $|P|$  variables and  $2|P| + 1$  constraints. By the fundamental theorem of linear programming [Chvátal 80, Theorem 3.4]  $\exists$  a basic solution in which  $|P|$  constraints are satisfied with equality. Constraint (4.42c) must be one of them. This forces  $|P| - 1$  out of (4.42d), (4.42e) to be satisfied with equality, which means

that  $|P| - 1$  of the variables must be either 0 or 1. Those variables must then sum to either 0 or 1, based on (4.42c). Those options force the remaining variable to be 1 or 0, respectively, in order to satisfy (4.42c).  $\square$

## 4.5 Second Lagrangian Decomposition on CLAP

In the interest of scalability, it would be desirable to similarly separate FLAP. Unfortunately the duplex constraint prevents this, and is not easily massaged away algebraically. To address this, extend the Lagrangian relaxation of CLAP to the constraint  $\sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} S_{ji} \leq 1 \forall i$ . Paralleling equation (4.22) on page 88, let  $d^d(S)$  be the function having the  $i$ -th element given by equation (4.43):

$$d^d(S)_i \triangleq \sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} S_{ji} - 1 \quad (4.43)$$

Let  $d^s(S, V)$  be  $d^s(S, D, V)$  where the antenna gain variables  $D$  are replaced with fixed estimates  $\bar{D}$ , where element  $ij$  is given by:

$$d^s(S, V)_{ij} \triangleq - \left( \frac{P_i \bar{D}_{ij} \bar{D}_{ji}}{Lb(i, j) N_r} S_{ij} + \gamma_1 (1 + M_{ij}) (1 - S_{ij}) - \gamma_1 \left( 1 + \sum_{k \in N \setminus \{i, j\}} \frac{P_k \bar{D}_{kj} \bar{D}_{jk}}{Lb(k, j) N_r} V_k \right) \right) \quad (4.44)$$

Let us define a new Lagrangian function  $\mathcal{L}'(\cdot)$  as follows:

$$\mathcal{L}'(S, D, V, \lambda, \mu) = \bar{\beta}^T S - \lambda^T d^s(S, D, V) - \mu^T d^d(S) \quad (4.45)$$

This gives a new dual function  $\phi'(\lambda, \mu)$  and corresponding problem dual problem CLAP-dual-2, both given below:

$$\phi'(\lambda, \mu) = \max_{S, D, V} \mathcal{L}'(S, D, V, \lambda, \mu)$$



[CLAP-dual-2]

$$\begin{aligned}
\min_{\lambda, \mu} \quad & \phi'(\lambda, \mu) \\
\text{s.t.} \quad & S_{ij} \leq V_i \quad \forall_i \\
& D_{ik} = \sum_{p \in P} G_{ikp} B_{ip} \quad \forall_{i,k} \\
& \sum_{p \in P} B_{jp} = 1 \quad \forall_j
\end{aligned} \tag{4.46}$$

This produces a new relaxed primal version of FLAP, RP-FLAP. FARP remains unchanged.

[RP-FLAP]

$$\begin{aligned}
\max_{S, V} \quad & \bar{\beta}^T S + \bar{\lambda}^T d'_s(S, V) - \bar{\mu}^T d^d(S) \\
\text{s.t.} \quad & S_{ij} \leq V_i \quad \forall_{ij}
\end{aligned} \tag{4.47}$$

RP-FLAP is separable along the index  $i$ . Let us group the link  $ij$  with node  $i$ , defining  $d^d$  in equation (4.48):

$$d^d(S)_i = \sum_{j:(i,j) \in A} S_{ij} + \sum_{j:(j,i) \in A} \bar{S}_{ji} - 1 \tag{4.48}$$

Using the preceding definition, we define the following:

$$\begin{aligned}
\bar{\beta}_w &\triangleq \{\bar{\beta}_{ij} | i = w\} \\
\bar{\lambda}_w &\triangleq \{\bar{\lambda}_{ij} | i = w\} \\
\bar{\mu}_w &\triangleq \{\bar{\mu}_i | i = w\} \\
d_w^d(S) &\triangleq \{d^d(S)_i | i = w\} \\
S_w &\triangleq \{S_{ij} | i = w\} \\
d_w^{s'}(S_w, V) &\triangleq \{d_w^{s'}(S, V)_{ij} | i = w\}
\end{aligned}$$

The partitioned form of RP-FLAP is the Single Node Relaxed Primal FLAP (SNRP-FLAP).

[SNRP-FLAP<sub>w</sub>]

$$\max_S \quad \bar{\beta}_w^T S_w + \bar{\lambda}_w^T d_w^s(S_w, V_w) - \bar{\mu}_w^T d_w^d(S_w) \quad (4.49a)$$

$$\text{s.t.} \quad S_{wj} \leq V_w \quad \forall_j \quad (4.49b)$$

The preceding series of decompositions replace the relaxed primal problem (RPP) with  $2N$  easy subproblems which can be solved in parallel. Each instance of SNARP<sub>*i*</sub> is a linear program with  $|P|$  variables and 1 general constraint. By Proposition (4.4.2), it can be solved by simply enumerating the objective value for each  $p \in P$ , of which there are a small constant number, and choosing the pattern with the highest value. Therefore the overhead of a general-purpose solver can be avoided. Each instance of SNRP-FLAP is a linear problem with  $O(N)$  variables and constraints, although it will be further re-formulated. The final form (SDQ-FLAP) is given in AMPL specification appears in Listing D.5 – the problem definition is given in line 197 on page 273.

## 4.6 Economic Interpretation

This formulation lends itself to the following semantic interpretation: In the coupling between the restricted master problem (RMP) and CLAP, the dual values  $\bar{\beta}_{ij}$  represent the estimated value in terms of improvement to the overall schedule of accommodating more traffic on link  $ij$ . In the coupling between Lagrangian subproblems,  $\bar{\lambda}_{ij}$  represents the value of improving the SINR on link  $ij$ , and  $\bar{\mu}_i$  represents the value of decreasing the usage of node  $i$ .

In SNRP-FLAP, each node chooses to activate links to maximize its utility, where  $\bar{\beta} \geq 0$  is the reward for activating each link,  $\bar{\lambda} \geq 0$  is the penalty for any SINR reduction on each link, and  $\bar{\mu} \geq 0$  is the penalty for using each node.

Correspondingly, in SNARP, each node chooses antenna gains to maximize a different utility, defined solely in terms of  $\bar{\lambda}$ . When all the constants have their values substituted in, the objective

function of  $\text{SNARP}_i$  is of the form in equation (4.50), where the actual value of constant  $k_{ij}$  is determined by  $\bar{\lambda}$ , node  $j$ 's antenna configuration, and RF parameters  $Lb$ ,  $P$ , and  $Nr$ .

$$\max_{D_{ij}} \sum_{j \neq i} D_{ij} k_{ij} \quad (4.50)$$

$$k_{ij} \begin{cases} \geq 0 & \text{if } ij \text{ or } ji \text{ is an active link} \\ \leq 0 & \text{if } ij \text{ or } ji \text{ is an "interference link"} \\ = 0 & \text{otherwise} \end{cases}$$

It is worth noting that **this interpretation generalizes** beyond the specific formulation used in the rest of this dissertation: In choosing to start from Björklund's formulation, I've committed to a particular way of representing demand, and a to particular (and unintuitive) expression of the SINR constraint. The overall decomposition approach developed does not depend on either, especially at this semantic level.

## 4.7 Lagrange Multiplier Updates

The combined problems  $\text{SNRP-FLAP}_i$  and  $\text{SNARP}_i$  for all nodes  $i$  implement the relaxed primal problem. Solution proceeds by iteratively solving the RPP and updating the Lagrange multipliers  $\lambda$  and  $\mu$  so that they converge to an optimal solution of the dual problem. This work uses a **subgradient** method because it lends itself to distributed implementation and because subgradient methods tend to scale well with the problem size. At time  $t$ , let  $s^t$  denote the degree of constraint violation,  $\alpha^t$  the step size, and  $[\cdot]_+$  projection onto the nonnegative orthant. The subscripts  $\lambda$  and  $\mu$  are used to distinguish the values pertaining to each set of Lagrange multipliers.

$$s_{\lambda}^t = d^s(S^t, D^t, V^t)$$

$$s_{\mu}^t = d^d(S^t)$$

$$\bar{\lambda}^{t+1} \leftarrow [\bar{\lambda}^t + \alpha_{\lambda}^t s_{\lambda}^t]_+$$

$$\bar{\mu}^{t+1} \leftarrow [\bar{\mu}^t + \alpha_{\mu}^t s_{\mu}^t]_+$$

We define step size rule  $\alpha^t = \frac{a}{(t+b)^2}$ ,  $a > 0$ ,  $b \geq 0$ . The  $a$  and  $b$  are tunable parameters and are not related to Guan's  $a$  and  $b$  in SDQ-FLAP and [Guan 95]. The AMPL specification of the multiplier update process is given in Listing D.13, especially lines 138 to 153 on page 310, and the underlying logic is given in Listing D.8 on page 280.

#### 4.7.1 Convergence Properties

The subgradient method described above will produce optimal values of the Lagrange multipliers for CLAP-dual-2.

**Proposition 4.7.1.** *The sequences  $\{\bar{\lambda}^t\}$  and  $\{\bar{\mu}^t\}$  converge to  $\lambda^* \in \boldsymbol{\lambda}^*$  and  $\mu^* \in \boldsymbol{\mu}^*$ , where  $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$  are the optimal sets of CLAP-dual-2.*

*Proof.* For convenience, let  $X$  refer to the set of all decision variables,  $x$  refer to a vector value in  $X$ , and  $x_0$  refer to some specific value of  $x$ , **not** a scalar component of  $x$ . Let  $G_s$  be any subgradient of  $d^s$  and  $G_d$  be any subgradient of  $d^d$ . Then  $\lambda G_s(x_0) + \mu G_d(x_0)$  is a subgradient of  $-\bar{\beta}^T S + \lambda^T d^s(S, D, V) + \mu^T d^d(S)$ , by Shor's [Theorem 15][Shor 98]. This equals equation (4.45) on page 96.  $\therefore S_\lambda^t + S_\mu^t$  is a subgradient of (4.45). The sum over all  $i$  of the objectives and constraints of SNRP-FLAP $_i$  and SNARP $_i$  equal the objectives and constraints of CLAP-dual-2. Assume that the Slater condition holds, otherwise the problem and JBSS-MP are infeasible.

$\int \frac{a}{(t+b)^n} d(t) = \frac{a(t+b)^{1-n}}{1-n}$ ,  $n > 1$ , which diverges as  $t \rightarrow \infty$ .  $\therefore \sum_{t=0}^{\infty} \alpha^t = +\infty$ .  $\lim_{t \rightarrow \infty} \alpha^t = 0$  for  $n, a > 0$ . Therefore,  $\{x\}$  converges to optimal  $x^*$  by [Shor 98, Thm 31].  $\square$

It does not follow from the above that the sequence of **primal** values produced will converge to optimal  $S^*, D^*$  even though the problems exhibit strong duality. To address this, we define the following sequence:

$$\hat{S}^t = (1 - \alpha^t)\hat{S}^{t-1} + \alpha^t S^t \quad (4.51)$$

**Proposition 4.7.2.**  *$\{\hat{S}^t\}$  converges to  $S^*$ , and the analogous  $\{\hat{D}^t\}$  converges to  $D^*$*

*Proof.* We appeal to a result by Larsson *et al.* [Larsson 99].  $\{S^t\}$  is generated by a dual subgradient process satisfying his criteria (9)-(11).  $\{\hat{S}^t\}$  is an ergodic sequence satisfying (7), (13). It follows from [Larsson 99, Theorem 1] that  $\{S^t\}$  converges to the solution set. The same applies to  $\{\hat{D}^t\}$ .  $\square$

## 4.8 Complexity Results

This section provides a crude evaluation of the computational benefit of decomposing the CLAP problem. It is difficult to directly compare the two approaches, as the algorithms and software tools differ substantially. Consequently, I use raw CPU time as a common metric. Figure 4.4 on the next page shows a box plot comparison of the time required to solve random problem instances of different sizes using the two approaches. The panel labelled “Centralized” shows a direct solution of CLAP using the KNITRO 6.0.1 solver; “Distributed” shows the relaxed primal problem / subgradient solution described above, using IPOPT [Wächter 06] and CPLEX for the quadratic and linear component problems, respectively. In order to easily gather CPU usage information, in the “distributed” case, each node’s computations were executed in sequence on the same CPU (see Listings D.15 and D.16.) Note that for the centralized case, the final three data points (for 18, 24, and 48 nodes) are lower bounds only – they show the elapsed time so far on experiments which have not yet terminated. The values for the distributed case are actual time to termination. Figure 4.5 on page 103 shows the same data but scaled to reflect the CPU time **per process** for the distributed case.

## 4.9 Summary

This dissertation will evaluate several strategies for integrating beam selection with link scheduling. The introduction presents the baseline case of no integration or minimal integration, which represents the current state of the art in wireless packet relay networking. The preceding section, 3, describes a completely-integrated mathematical program and a decomposition-based algorithm for its tractable solution.

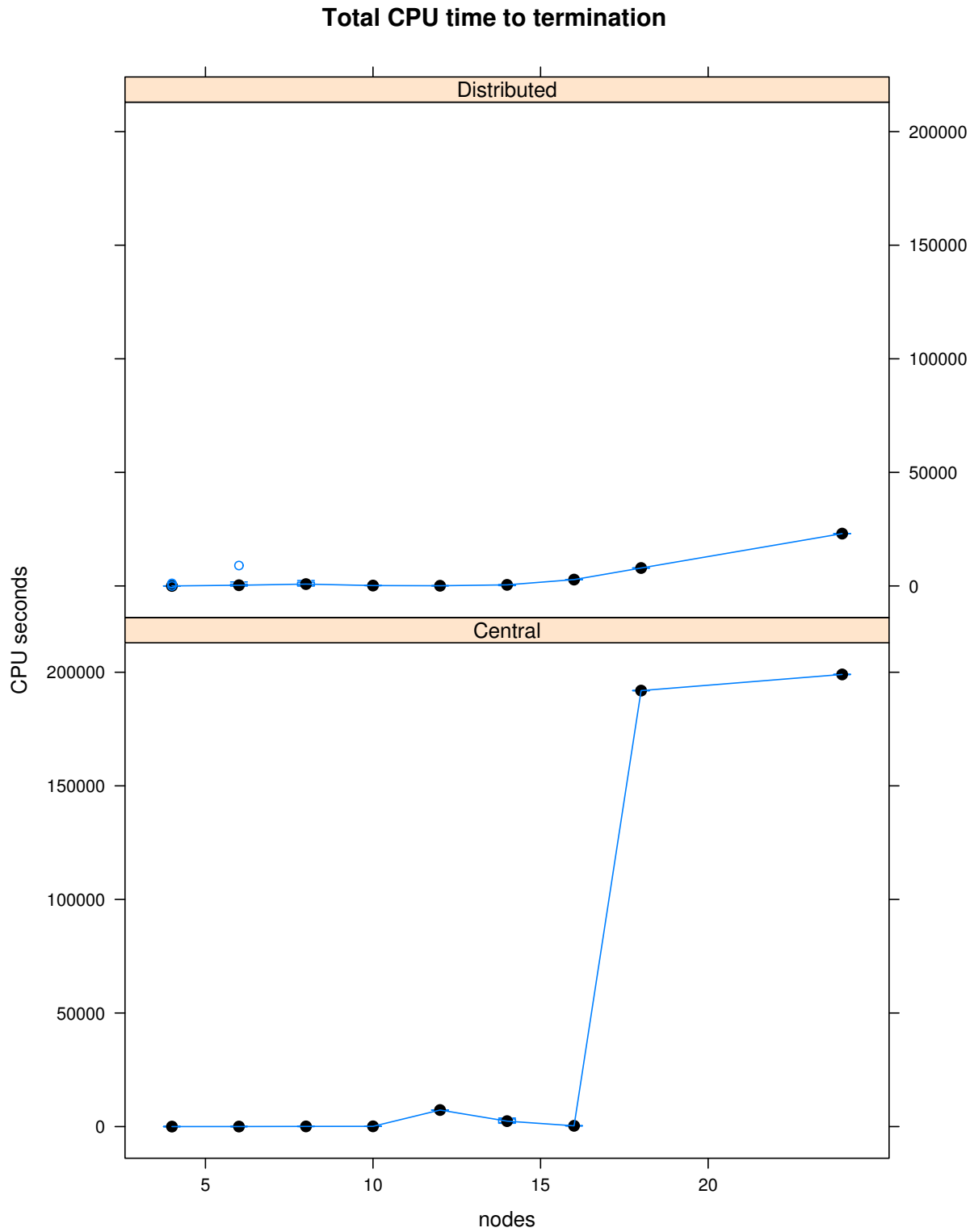


Figure 4.4: Scaling comparison of centralized direct solution and distributed decomposed solution. Figure shows total user CPU time consumed.

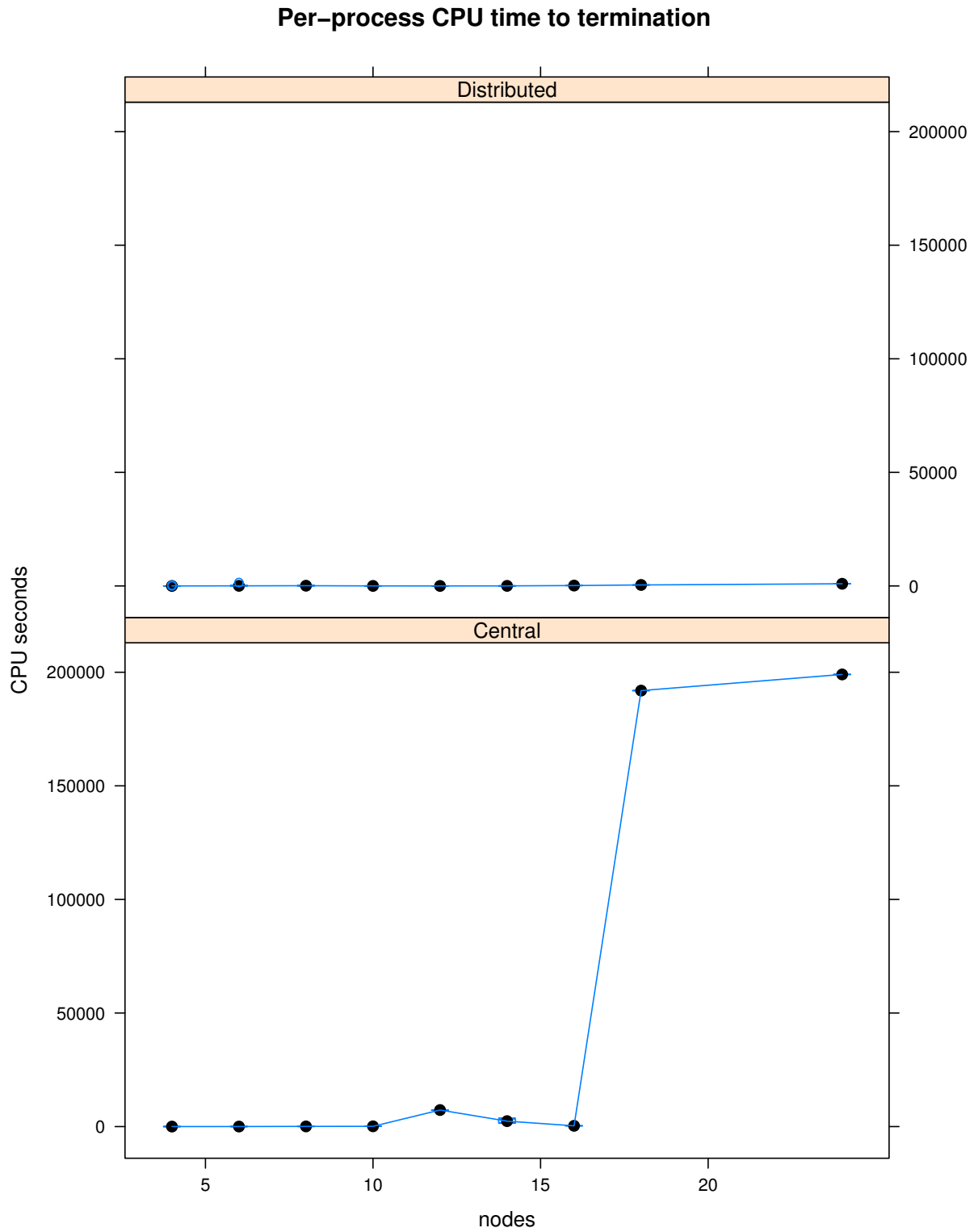


Figure 4.5: Scaling comparison of centralized direct solution and distributed decomposed solution. Figure shows user CPU time consumed **per logical process**.

## Chapter 5

### Mathematical Issues in System Implementation

This chapter discusses the mathematical issues involved in bringing the preceding formulation into a real-world system. In addition to the design and implementation problems inherent in all systems work, certain issues with the formulation itself present themselves in the context of implementation and deployment.

#### 5.1 Solution Oscillation

The problem formulation from Chapter 4 exhibits a well-known issue with subgradient methods: Small changes in the Lagrange multipliers produce large changes in primal solutions, causing oscillation around the ideal search trajectory. This can considerably slow the solution process. The linear objective function and previously-mentioned integrality property contribute to this behavior in FLAP and its derived problems.

In the case of FLAP specifically, the (dualized) SINR constraint is deliberately constructed so that it is always satisfied for unused ( $S = 0$ ) links. This causes the SINR constraint to oscillate\* between being slightly violated and grossly satisfied, which in turn leads to the Lagrange multiplier bouncing between its  $S = 0$  state (0) and its  $S = 1$  state. The linear-scale figure figure 5.1 on the next page illustrates the effect.

Additionally, when the relevant SINR prices  $\lambda$  are 0 and the  $\bar{\beta}$  values are the same, links which share a node exhibit the **homogeneous subproblem** property where any given dual price

---

\* One would normally refer to this as “flapping,” but I wish to avoid overloading the term.



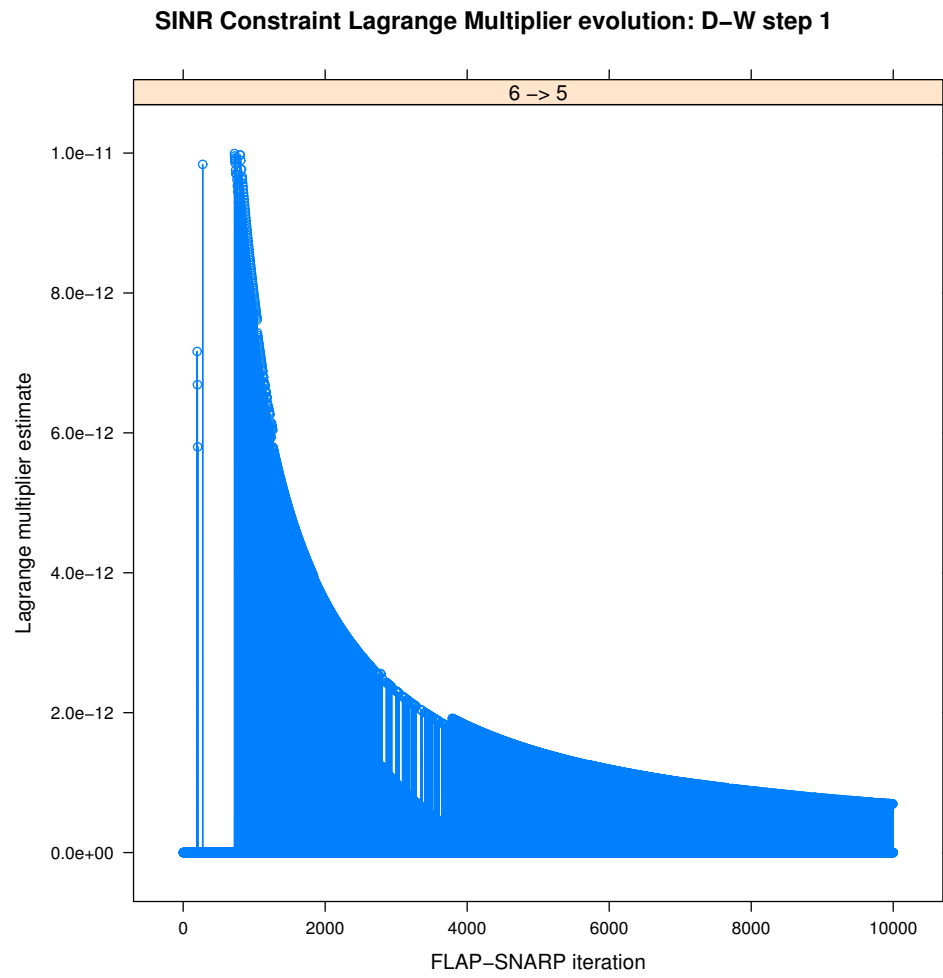


Figure 5.1: Lagrange multiplier oscillation example (random scenario snapshot SVN r.1088)

$\mu$  will result in the same primal outcome for all links. This issue arises in the context of the hydrothermal **unit commitment** problem, and is commonly addressed by the use of surrogate subgradient methods. To simplify decentralized implementation, however, we avoid that approach and instead use a nonlinear approximation method of the form presented in [Guan 95]. This is conceptually very similar to an augmented Lagrangian, but the additional quadratic parameter is computed locally for each subproblem, maintaining the separable structure of the original linear program. Based on this transformation, we introduce the Single-node Dual Quadratic FLAP, or SDQ-FLAP, where  $a_{ij}$  and  $b_{ij}$  are defined as in [Guan 95].

[SDQ-FLAP<sub>*i*</sub>]

$$\max_S \sum_{j:i_j \in A} -a_{ij} S_{ij}^2 + (b_{ij} - \bar{\lambda}_{ij} S_{ij}) - \frac{\bar{\mu}_i + \bar{\mu}_j}{2} S_{ij} \quad (5.1a)$$

$$\text{s.t.} \quad S_{ij} \leq V_i \quad \forall j \quad (5.1b)$$

**Proposition 5.1.1.** *Any stable solution to SDQ-FLAP<sub>*i*</sub> is also a solution to SNRP-FLAP<sub>*i*</sub>.*

*Proof.* The constraints of SDQ-FLAP<sub>*i*</sub> are identical to those of FLAP<sub>*i*</sub>. At any point  $x_0$ , the nonlinear approximation  $f'$  generated at  $x_0$  is parallel to  $f$  [Guan 95]. For both SDQ-FLAP and SNRP-FLAP, the constraints are all differentiable and convex, and the objective is convex (when stated as minimization). Therefore the Karush-Kuhn-Tucker conditions are sufficient for global optimality.

Let  $x^*$  be an optimal solution of SDQ-FLAP<sub>*i*</sub> as constructed at  $x_0$ . Let The KKT conditions therefore hold. Assume  $x^*$  is stable, therefore  $x^* = x_0$ . Suppose that  $x_0$  is not an optimal solution of SNRP-FLAP<sub>*i*</sub>. The KKT conditions other than stationarity are the same in both cases, so they must hold for SNRP-FLAP<sub>*i*</sub>. Therefore the stationarity condition must hold for SDQ-FLAP<sub>*i*</sub> but not for SNRP-FLAP<sub>*i*</sub>. That requires, for the same constraints, that  $\nabla f'(x^*) \neq \nabla f(x^*)$ , which is a contradiction.  $\square$

The preceding formulation significantly reduces oscillation relative to FLAP or its decomposed analogue, SNRP-FLAP. The constraint (5.1b) can be ignored: The variable  $V_i$  does not appear in the objective function and is otherwise free, so the problem can be solved for  $S$  and  $V_i$  chosen to be  $\max(S_{ij})$ . SDQ-FLAP $_i$  is therefore an **unconstrained** (or bound-constrained) quadratic program with at most  $2N$  variables.

## 5.2 Partial Pricing

Recall that the objective of the column generation subproblem is to find improving feasible points for inclusion in the restricted master problem. The optimality of the overall result does not require that the subproblem finds the **most** improving point, only that it finds **an** improving point if one exists. We exploit this by using the well-established technique of “partial pricing” and returning the first improving primal feasible result  $(\hat{S}^t, \hat{D}^t)$  – which may or may not be the best possible – without waiting for the subgradient process to converge [Desrosiers 05]. It is only necessary to allow the subproblem to fully converge to prove that there is no as-yet-undiscovered feasible improving point.

Every solution to an iteration of the restricted master problem is a valid schedule. Each such schedule can be put into place in the network immediately if it is superior to the current schedule, regardless of whether or not it is the final, best schedule. Consequently, terminating the subproblem early and re-solving the RMP yields a useful result sooner than solving the subproblem to optimality, even though it may or may not improve the overall running time.

## 5.3 Distributed Consensus

The preceding sections decompose the original problem into a form where  $2N$  small problems are solved in parallel for each subgradient update iteration. Going from a parallel algorithm to a distributed one requires some consideration of the communication processes. We make use of a very simple and robust model due to [Tsitsiklis 86]: Every node maintains its own version of every variable, and nodes announce their variable values to other nodes occasionally. A node may actively

compute locally-generated values for some, all, or no variables. Upon computing a new value or receiving other nodes' variable values, a node updates its own values according to a weighted averaging scheme. Under surprisingly light requirements on the weights and communication frequencies, it is shown that this scheme has the same convergence properties as its centralized counterpart. The results in [Tsitsiklis 86] are only shown for objective functions which are continuously differentiable and Lipschitz, while the Lagrangian dual function is generally non-differentiable. Similar results are proven for the non-differentiable case in [Nedić 01].

## Chapter 6

### Performance Evaluation

This chapter presents some minimal performance evaluation results for the algorithms developed in the preceding chapters. Section 6.1 presents numerical experiments, showing that optimal solutions are both achieved quickly and offer substantial speedup over (non-spatial-reuse) TDMA schedules. Section 6.2 discusses a testbed proof-of-concept implementation of our approach, and finally a summary is given in section 6.4.

#### 6.1 Numerical Experiments

This section considers the performance of the optimization process taken in isolation. These experiments emulate a distributed algorithm in that each node's computations are performed separately. Experiments were conducted by running the algorithm over a large number of scenarios constructed with varying initial values. In total, 1396 experiments were run. The following major parameters were varied: Number of nodes (between 0 and 81), the number of links (between  $\frac{1}{2}$  and 3 links per node), and the size of the simulated region (between 1 and 16 square km). For each set of parameters, nodes were randomly placed within the simulated area with uniform probabilities, and pairwise path losses were estimated using the Green-Obaidat model [Green 02]. All possible links were identified based on a hypothetical transmission power of 14.7 dBm, a required signal strength of -80 dBm, and the best-case antenna gains given a measured phase array antenna beam pattern. The requested number of links were chosen randomly from the pool of possible links; if enough possible links did not exist, a new layout was generated. The results presented here are

aggregates across all of these scenarios — a full factorial analysis is planned for future empirical studies of these algorithms and associated STDMA MAC.

### 6.1.1 Running time

A well-known limitation of subgradient methods for updating Lagrange multipliers is that they are very slow to reach a provably converged state. This means in practice that such algorithms may find optimal values relatively quickly, but then require a longer period to verify that no better values exist. As alluded to in section 5.2 on page 107, **termination** may not be the best criterion for an on-line system. It is expected that schedule optimization will be a continuous process, converging and diverging as system parameters change. Consequently, we find it useful to examine the time required to find optimal and near-optimal solutions as well as the time to termination. In our experiments, we find that by either measure, execution time is **at worst** linear in the size of the input.

To quantify the behavior, see Figure 6.1 which plots the distribution of the number of iterations required to first reach the optimal solution across all of our simulation runs. Figure 6.2 shows the upper-left portion of the curves in more detail. We can see that in more than 90% of the cases, the optimal solution is found within 500 iterations (the mean is 61 iterations and 92% are solved to optimality within 200), yet some scenarios may require as many as 3250 iterations to settle on the optimal solution. On average, we are able to get within 10% of optimal within 59 iterations and within 20% of optimal within only 26 iterations.

### 6.1.2 Schedule Efficiency

In addition to convergence properties, our numerical experiments provide a window into the ability of the algorithm to produce efficient (high-reuse) schedules across a vast number of randomly generated scenarios. Figure 6.3 plots a speedup metric which is the ratio of the time required by a TDMA MAC to transfer its workload as compared to the time required by our optimized system. In our experiments, we see speedup values ranging from 1 (no speedup) to 6 with an average speedup

## Iterations to Specified Fraction of Optimality

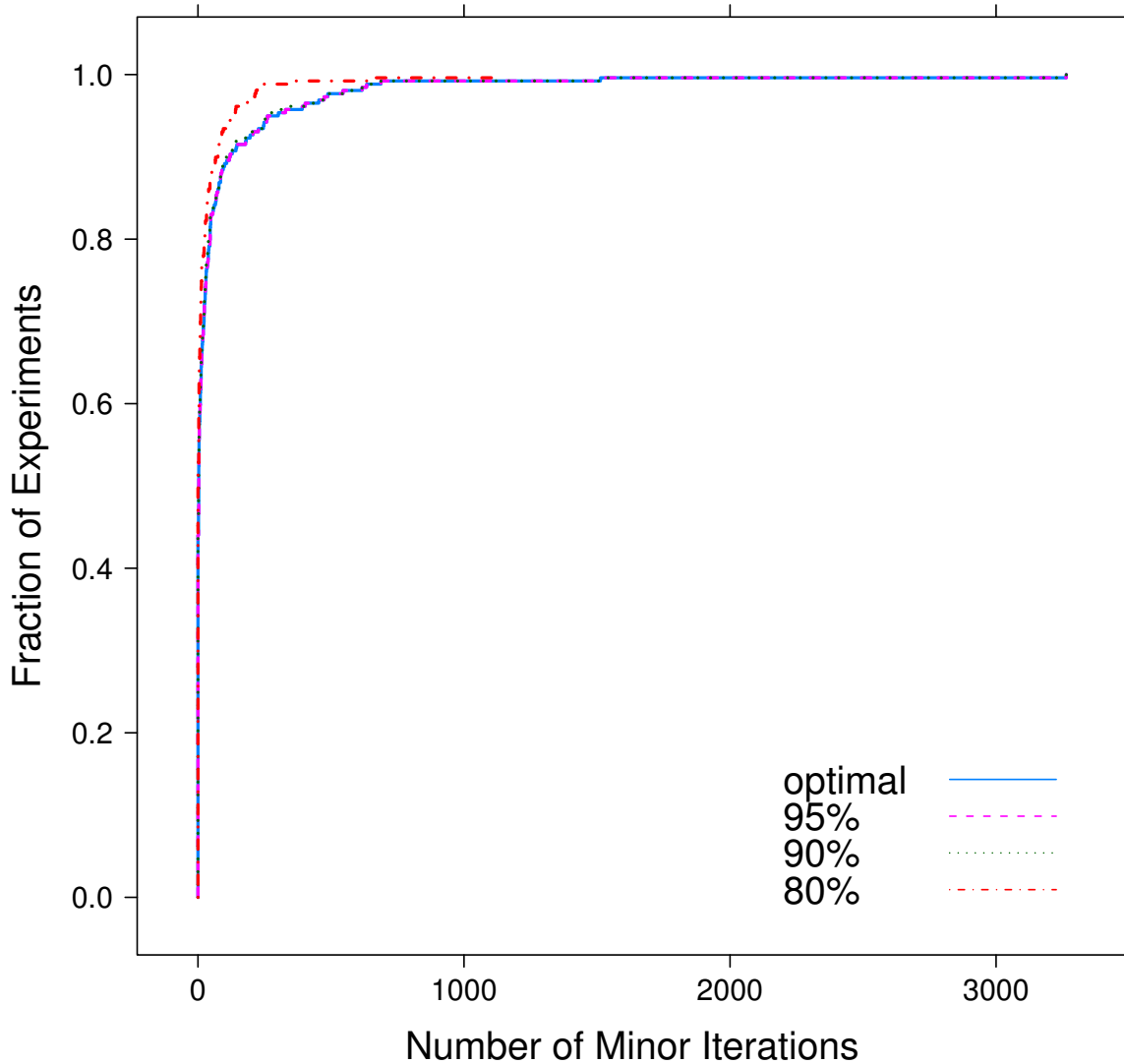


Figure 6.1: Distribution of number of (minor) iterations necessary in simulations to first reach and optimal solution.

## Iterations to Specified Fraction of Optimality (Detail)

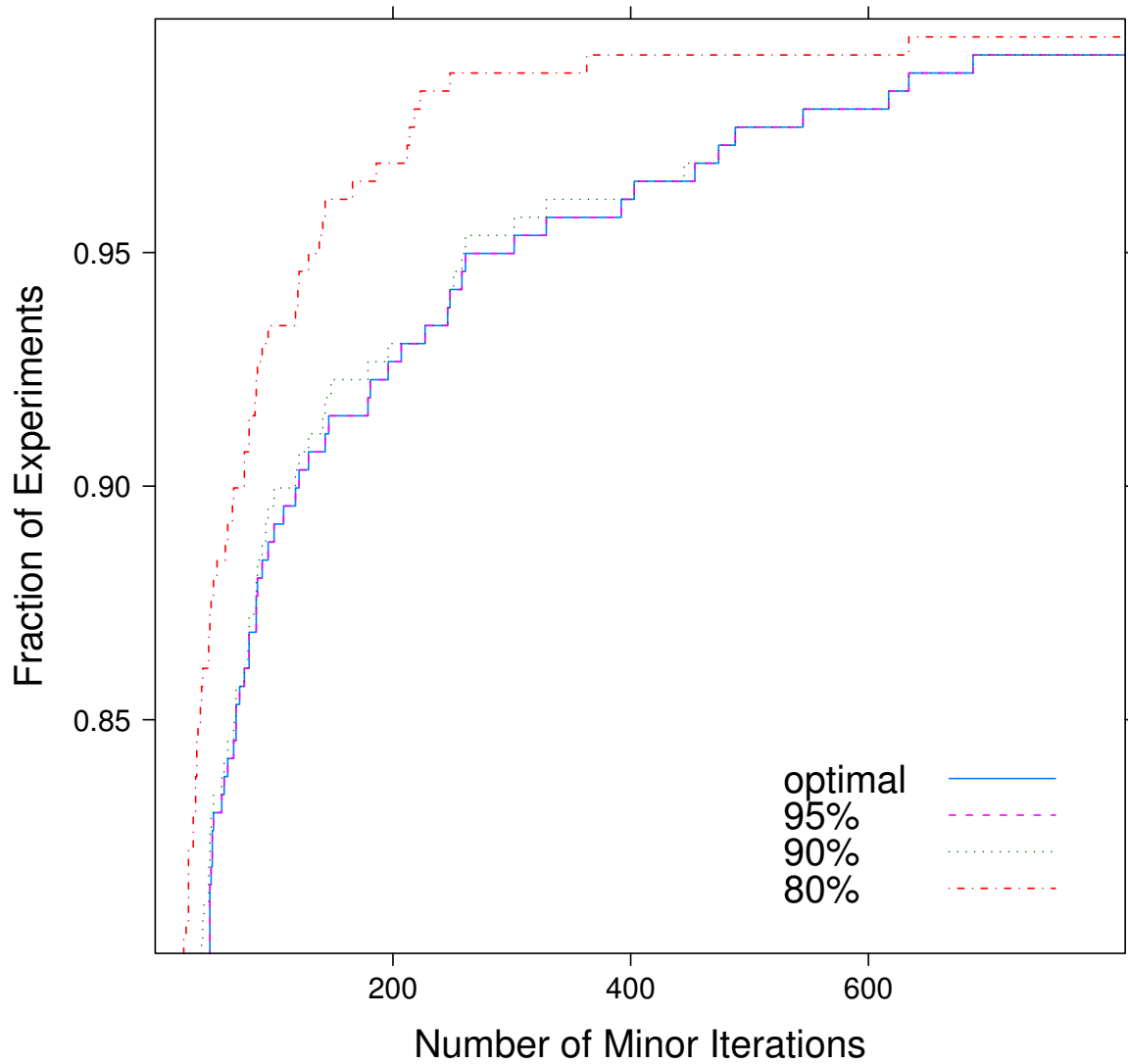


Figure 6.2: Distribution of number of (minor) iterations necessary in simulations to first reach and optimal solution – detail.



of 1.74 across all scenarios ( $\sigma = 0.96$ ). This number is biased downward by the inclusion of a significant number of simulations for which 1 was the greatest possible speedup; this also biases the running time estimates. For more detail, speedup is shown relative to the number of links in Figure 6.4 on page 115.

### 6.1.3 Alternate Cases

As mentioned before, optimal link scheduling has been shown to be NP-hard. Since the data discussed so far suggest distinctly sub-exponential computational complexity, it makes sense to consider input data which might cause the algorithms to exhibit exponential complexity. We consider two such cases, one which fails to challenge the algorithms, and one which succeeds.

#### 6.1.3.1 Grid Scenario

For the `grid` scenario, nodes are placed on a grid at 250m intervals, and the same propagation and antenna settings described above are used. Traffic flows from every node to its two neighbors in the positive  $x$  and  $y$  directions, as illustrated in Figure 6.5 on page 116. It had been our observation that discrete constraints contributed to cycling and other degenerate algorithmic behavior, suggesting that a highly-connected scenario might perform poorly.

In actuality, however, this scenario exhibited quite reasonable scaling behavior. Figure 6.6 on page 117 shows the scaling over the range of 4 to 81 nodes. (The  $x$  and  $y$  scales are chosen for consistency with the `clique` scenario below.) It gives the appearance of being nearly constant, though it must be remembered that the amount of computation involved in each iteration increases linearly with the number of nodes.

#### 6.1.3.2 Clique Scenario

The `clique` scenario represents the most difficult decision scenario, and one with a clear reduction to a hard SAT problem. The scenario was created by suspending realistic radio models: It is fully connected in all senses. Every node has traffic for every other, but also every node

## Achieved Speedup in Numerical Simulations

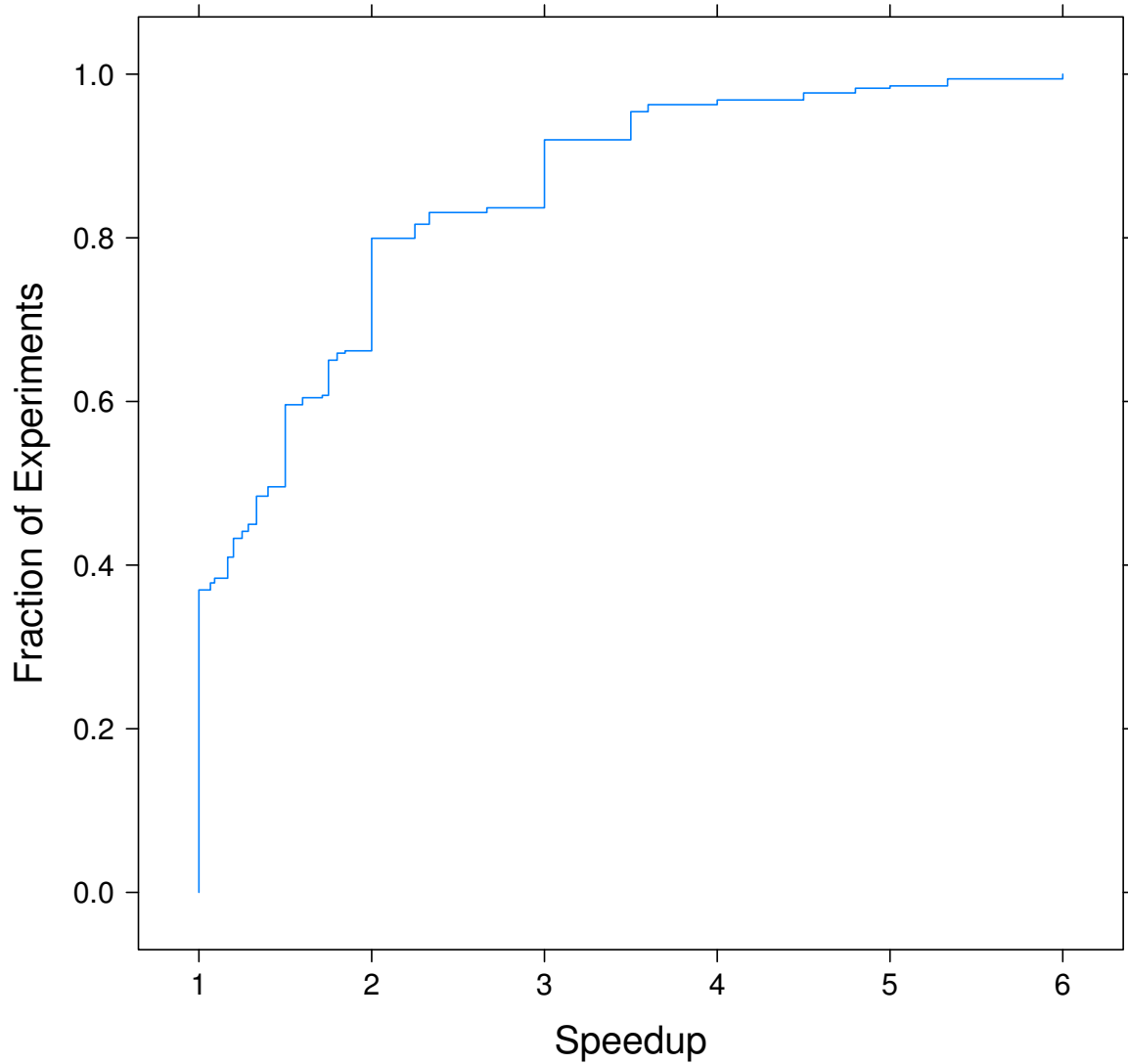


Figure 6.3: Empirical cumulative distribution of achieved speedup (ratio of optimal to TDMA performance) across all simulations.

## Speedup Relative to Number of Links

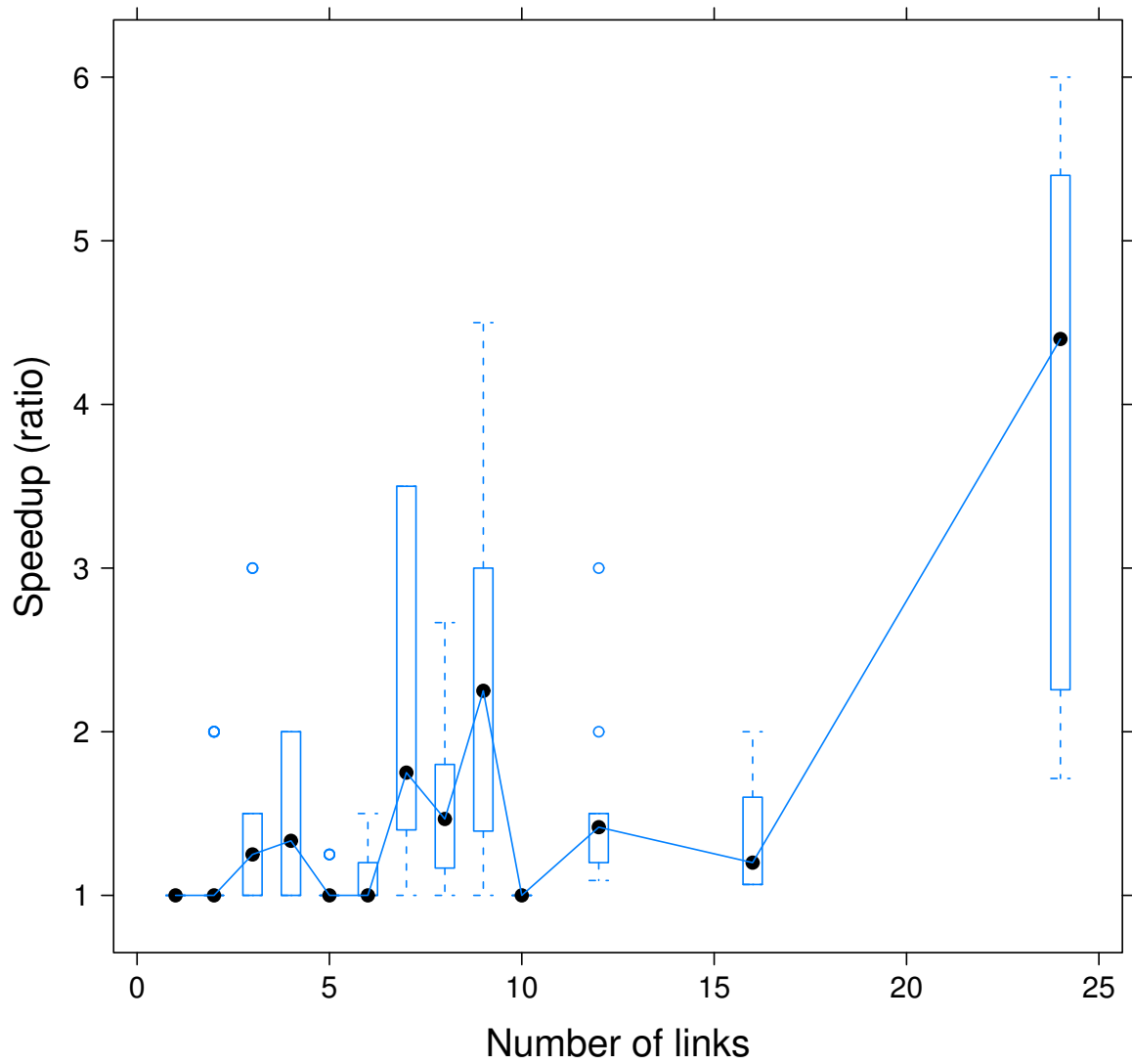


Figure 6.4: Achieved speedup by number of links.

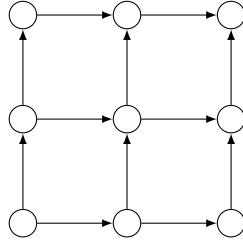


Figure 6.5: Grid scenario, 9 nodes

the same propagation characteristics to every other, and every combination of antenna patterns produces the same effect. Interestingly, this admits a very easy solution by inspection or by group-theoretic argument – there is no possibility for spatial reuse – but it is a pathological case for the algorithms presented here. The scaling properties are shown in Figure 6.7 on page 118. All scenarios with greater than 14 nodes failed to terminate within a 100,000 iteration limit. Note also that the constant time to an optimal value is potentially misleading – the algorithm always starts with a conservative feasible solution (simple TDMA), and this scenario allows no possible improvement. It is consequently a rather vacuous optimality, although it is also an indication that the algorithms we present here do “something useful” even in the worst cases.

It would be nice to conclude that an artificial scenario is required in order to produce exponential scaling behavior. To do so would be overreaching the data, but I at least have not succeeded in producing “naturally plausible” scenario with that effect.

## 6.2 Deployed System

Thus far, this chapter has described and evaluated our **mathematical** design; this subsection addresses its concrete implementation. The system we present here operates in a **fully distributed, asynchronous** manner. Nodes maintain and exchange variables as described in section 5.3.

In addition to the subproblem solver processes, there is a separate process for the early termination check and restricted master problem (RMP). When this process detects that its current estimates  $(\bar{S}^t, \bar{D}^t)$  constitute a primal feasible solution with negative reduced cost, a corresponding new column is added to the RMP, which is re-solved. The resulting new schedule, updated dual

## Running Time vs. Problem Size: grid

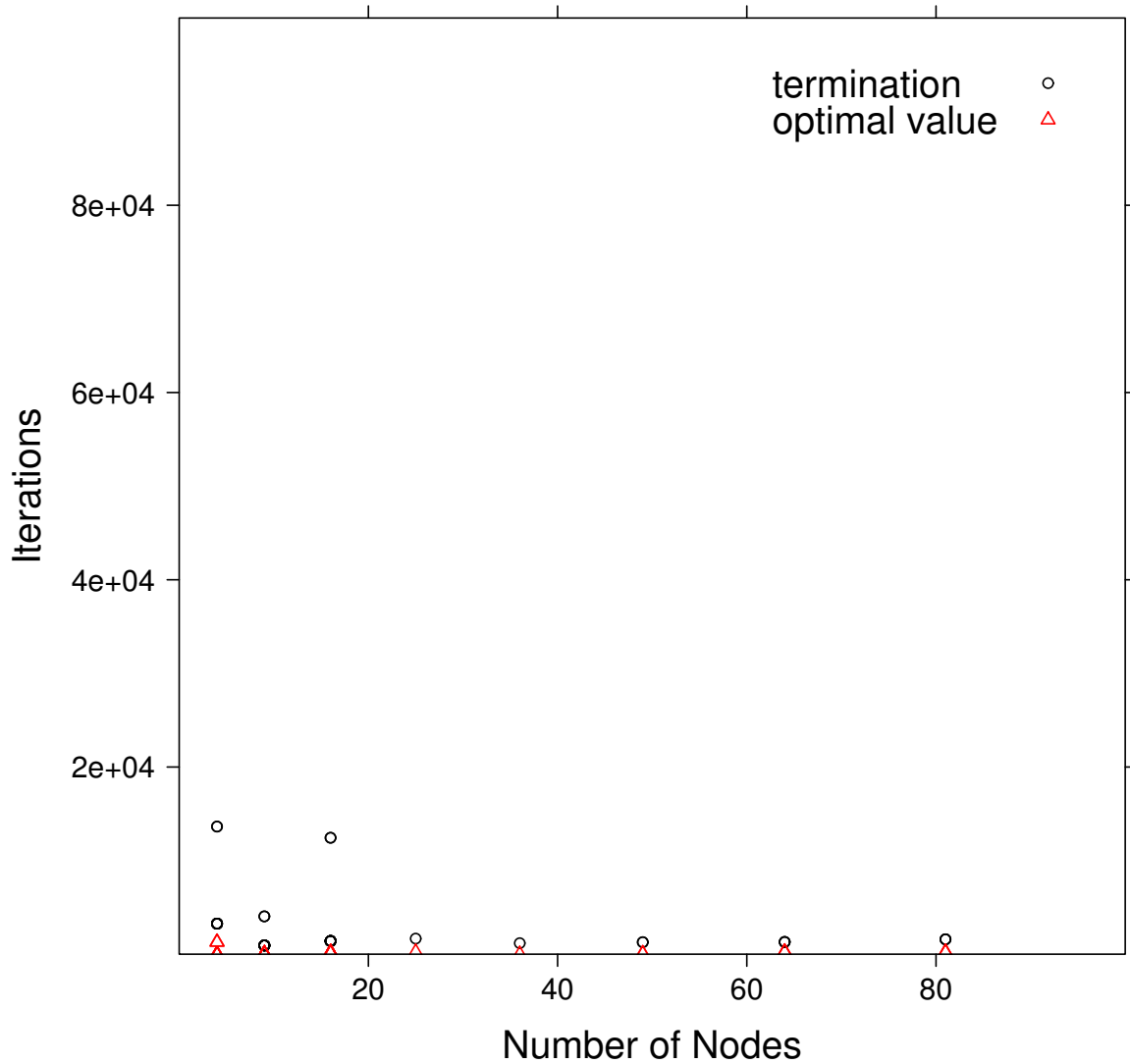


Figure 6.6: Iterations to optimality and to termination for the grid scenario

## Running Time vs. Problem Size: clique

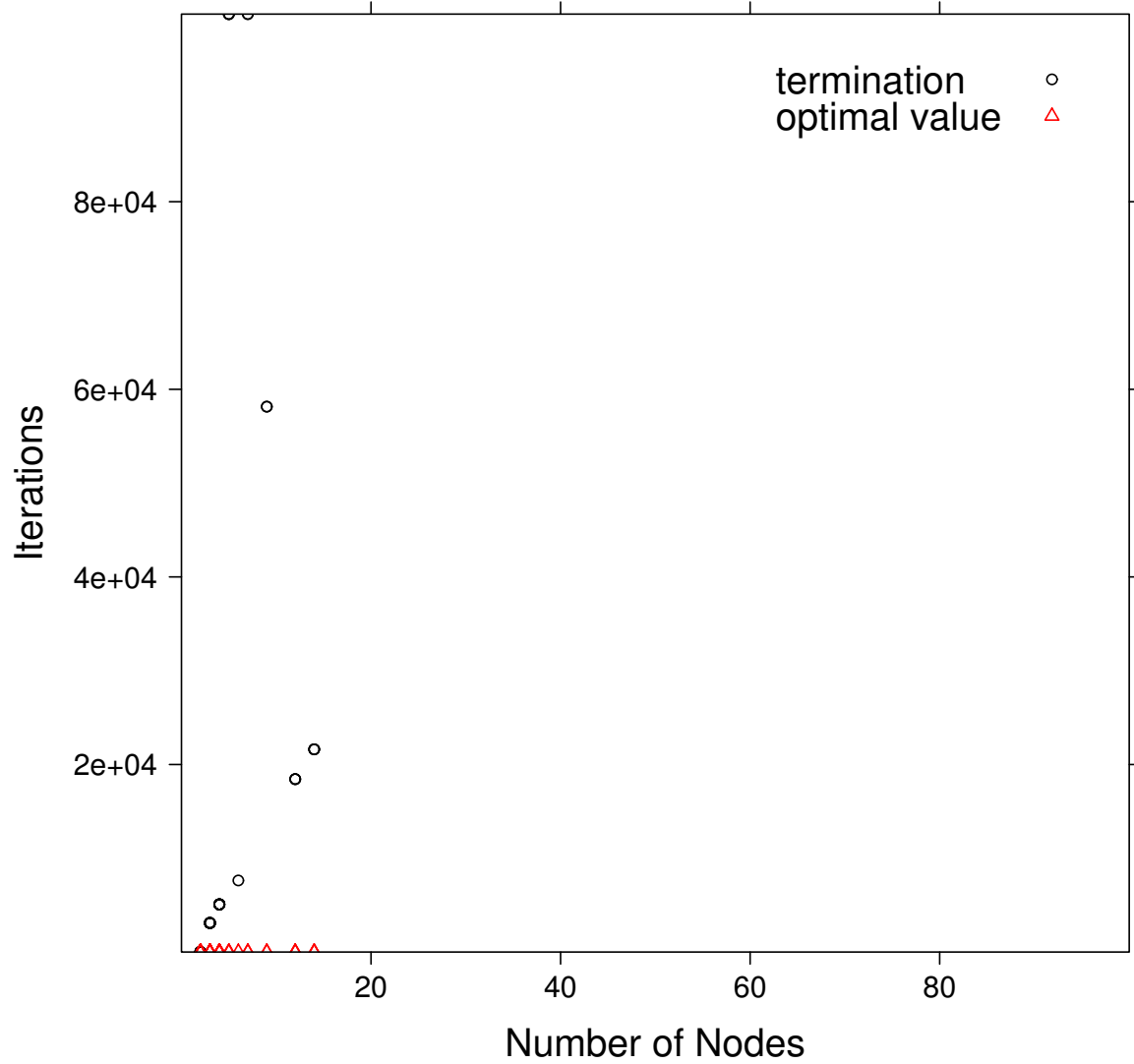


Figure 6.7: Iterations to optimality and to termination for the clique scenario

prices  $\bar{\beta}$ , and step size reference time are sent to all nodes by a flooding protocol.

Figure 6.8 on the next page shows the execution of the algorithm on a wide-area phase array antenna testbed, as observed by a single node (C). In this scenario, node B is sending traffic (a stream of UDP packets with 1024 byte payloads) to node A while node C sends a similar volley to node D. With naïve beam-steering, these links are bad neighbors—node B will cause substantial interference at node C (62.5 dBm on average). Hence, these links can be activated simultaneously, but only carefully.

The top strip of this figure shows the evolution of the SINR Lagrange multiplier estimates  $\bar{\lambda}$ , the second strip shows the consensus estimated link activations  $\hat{S}$ , and the third strip shows the combined antenna gains for the signals and interference. Times of interest are marked with a vertical bar and labelled (1, 2, ...) on all strips. Note that no change to actual system state occurs until a new execution of the RMP; the link activation and antenna configurations referred to are variable values. Qualitatively, the execution of the algorithm can be understood in the following stages:

Prior to time 1, node C's estimate  $\bar{\lambda}_{CD}$  is 0 and does not appear on the log-scale. This drives the link activation  $\hat{S}_{CD}$  toward 1, while the SNARP objective is undefined and the resulting gains are low.

At time 1, the combination of high activation and low gain causes the SINR constraint for link C  $\rightarrow$  D to be violated. The price  $\bar{\lambda}_{CD}$  takes a large step to  $> 10^{-20}$ . The increased price drives  $\hat{S}_{CD}$  back toward 0, and causes  $\hat{D}_{CD}\hat{D}_{DC}$  to start trending up. The price  $\bar{\lambda}_{CD}$  decreases steadily as the low activation and higher gain stay within the constraints.

At time 2, the combination of low  $\lambda_{CD}$  and higher gain allows  $\hat{S}_{CD}$  to increase to near 1. At time 3,  $\hat{S}_{CD}$  gets close enough to 1 to violate the SINR constraint again and drive up  $\bar{\lambda}_{CD}$ . The dramatic increase in  $\bar{\lambda}_{CD}$  relative to  $\bar{\lambda}_{BA}$  drives the antennas to favor  $\hat{D}_{CD}\hat{D}_{DC}$ . Note that this antenna configuration at node C has high gain toward A, raising the unwanted gain  $\hat{D}_{CA}\hat{D}_{AC}$ .

Between times 3 and 4,  $\bar{\lambda}_{CD}$  and  $\bar{\lambda}_{BA}$  trend down, but changes in their relative magnitude cause the overall antenna state to switch back and forth. Immediately before time 4,  $\hat{S}_{BA}$  increases

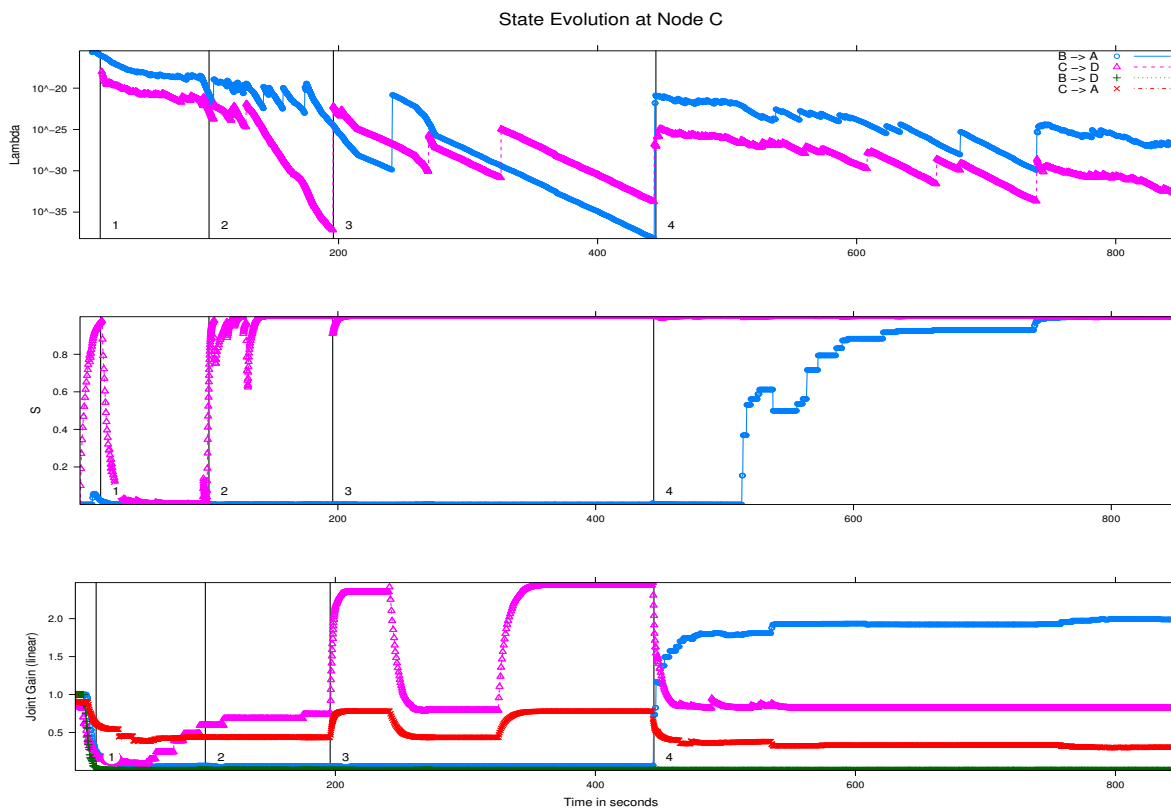


Figure 6.8: Trace of algorithm scheduling links  $B \rightarrow A$  and  $C \rightarrow D$  concurrently, as seen locally at node C. The top strip shows  $\bar{\lambda}$ , middle strip shows  $\hat{S}$ , and the bottom strip shows the **combined** gain  $\hat{D}_{ij}\hat{D}_{ji}$ . Note that  $B \rightarrow D$  and  $D \rightarrow A$  are interference if both data links are active. The aligned  $x$  axis is time in seconds.



almost invisibly. Recall that node C is **not computing**  $S_{BA}$ , so a change in  $\hat{S}_{BA}$  reflects the incorporation of a value broadcast by node B. This change is sufficient to cause an SINR constraint violation, driving  $\bar{\lambda}_{BA}$  up at time 4.

Note two changes with regard to the gains: First  $\hat{D}_{AB}\hat{D}_{BA}$  increases dramatically, reflecting a change in antenna configuration by node B. Second, the change in  $\bar{\lambda}_{BA}$  causes node C to change its antenna configuration to diminish  $\bar{D}_{CA}$ , at the cost of also reducing  $\bar{D}_{CD}$ . This new system-wide antenna configuration can accommodate both links, and  $\hat{S}_{BA}$  tends toward 1 as node C receives updates from other nodes. At this point the RMP can schedule the two links concurrently with the configuration given. Note that  $\bar{\lambda}$  continues to vary but this variation does not effect the primal estimates.

### 6.3 Performance Problems

This section discusses some of the weaknesses of the proposed approach and its implementation. The first subsection describes problems specific to the current implementation, which can likely be resolved without changing the basic algorithms.

#### 6.3.1 Implementation Issues

The simplest and least fundamental performance issues are those stemming directly from the implementation. The current implementation which I have referring to as “Prototype 3” is best regarded as a proof of concept: It is a working system, but not one designed for performance. Every subproblem solution involves writing several files, synchronizing processes, and running a stand-alone NLP solver. Especially, the STDMA driver schedules very large time slots, so freshly computed values wait as much as a second to be sent. We expect that a performance-conscious implementation could reduce time per iteration by at least an order of magnitude – possibly several – without any changes to the fundamental algorithms.

The following procedures describe the current mode of operation:

Ignore for the moment the asynchronous nature of this algorithm, and imagine that all pro-

**Algorithm 6.1:** Subproblem Iteration – Wireless Node

```

repeat
  if ASYNCHRONOUS RECV (packet, from=wireless) then
    | ASYNCHRONOUS XMIT (packet, to=solver host)
  if ASYNCHRONOUS RECV (packet, from=solver host) then
    | update local cache
until my transmit slot
if time since last broadcast  $\geq$  timeout or SIZEOF (cache)  $\geq$  full packet then
  | SYNCHRONOUS XMIT (cache, to=wireless)

```

**Algorithm 6.2:** Subproblem Process – Solver Thread

```

repeat
  acquire_lock ()
  local_copy  $\leftarrow$  node_variables
  release_lock ()
  write_fifo (local_copy); /* blocking IPC */
  /* Wait for AMPL cycle */
  new_vars  $\leftarrow$  read_fifo (); /* blocking IPC */
  acquire_lock ()
  node_variables  $\leftarrow$  update_estimates (local_copy, new_vars, node_variables)
  release_lock ()
until forever

```

**Algorithm 6.3:** Subproblem Process – Solver Host Main Thread

```

repeat
  acquire_lock
  if  $\exists$  new schedule state then
    | SCHEDULE PROTOCOL
  release_lock
  foreach  $v \in$  node_variables do
    | if random ()  $\geq$  thresholdv then /* depends on source of value */
      | ASYNCHRONOUS XMIT (v, to=wireless node)
  if ASYNCHRONOUS RECV (packet, from=wireless node) then
    enqueue (packet)
    if length (queue) > threshold then
      acquire_lock
      node_variables  $\leftarrow$  update_estimates (queue, node_variables)
      release_lock
until forever

```

cesses' cycles are aligned. Suppose that the estimate of some variable has changed value at some node in round  $n$  – the following events will have to occur before that change can be reflected in round  $n + 1$  at some other node.

- (1) The source node's cache freshness timeout expires.
- (2) A protocol data transmit slot occurs in the schedule.
- (3) The packet is transmitted and received by user-space daemons written in Python.
- (4) The packet is re-transmitted to a solver host.
- (5) Another timeout expires (or batch size is reached) on the solver host.
- (6) Thread synchronization occurs (within a single Python process).
- (7) The solver interaction thread blocks on named pipe IPC with the AMPL process.
- (8) AMPL reads the named pipe; AMPL goes to sleep and the solver interaction thread unblocks.
- (9) The solver interaction thread unblocks and writes.
- (10) AMPL unblocks and reads.
- (11) AMPL performs some computation and writes a problem description file.
- (12) AMPL `fork()`s and `exec()`s a solver executable; the original process blocks.
- (13) The solver reads the problem description file, performs some computations, writes a result file, and terminates.
- (14) AMPL unblocks and reads the result file.
- (15) Steps 11 – 14 are repeated (the process occurs for SDQ-FLAP and SNARP).
- (16) AMPL opens named pipe; AMPL goes to sleep and the solver interaction thread unblocks.

- (17) The solver interaction thread unblocks and reads.
- (18) AMPL unblocks and writes.
- (19) Thread synchronization occurs within Python.
- (20) An updated value is sent from the solver host to the wireless node.

Even without measuring the delay associated with every component, several salient points stand out: There is a great deal of interprocess communication, inter-thread synchronization, file I/O and process creation involved for an ultimately very small amount of computation. Profiling the solver-host processes shows that in each of its cycles (steps 4 – 20) the time spent in the solvers doing the actual optimization is  $\approx 10^{-3}$  of the total elapsed time.

Another major source of latency is waiting for a protocol data slot (or another assigned slot). In the Prototype 3 driver design, the shortest available time slot is 40 ms, but the protocol data generated in a new iteration typically requires  $\approx 1$  ms to send. This makes for an unnecessarily long schedule directly, but also has a second-order effect: In order to keep the fraction of time available for actual data transmission reasonably high, it is often appropriate to assign multiple data slots **to each link** for each slot assigned to protocol data, meaning that a long protocol time slot also implies long data slot(s). The combined effect of this and related scheduling limitations is that the elapsed time between protocol data slots is frequently 10 – 50 times longer than necessary.

There are reasonable solutions to all of these problems: The inefficient solution process results from using AMPL directly to implement the prototype: I use an external “solver host” because AMPL is not available for the ARM architecture used on the wireless nodes; the complex IPC and threading structure exists to create a synchronization and IPC capability where AMPL natively has none. The scheduling challenges reflect a hard-wired kernel timer resolution of 100 Hz in the Linux kernel version in use on our wireless nodes. I have subsequently made kernel modifications to support both a higher rate fixed clock in the original kernel version and a newer hard real-time, high-resolution kernel on the Phocus array boards\* .

---

\* I am grateful to Xi Liu and Alan Schmitz for their help with these modification

### 6.3.2 Algorithmic Issues

Beyond the implementation issues described above, there are also more fundamental performance questions. I will briefly discuss two of those here.

#### 6.3.2.1 Discrete Algorithms

Discrete, graph-based algorithms offer substantially faster solutions to problems close but not exactly equal to the joint beam selection and scheduling (JBSS) problem discussed in this work. In particular, if there is either no interference constraint, or only pairwise interference, then graph coloring algorithms are sufficient. For a significant fraction of input problems, those constraints fully determine the outcome, and it is likely that they could significantly reduce the search space in others. The possibility of combining algorithmic approaches has so far not been explored.

#### 6.3.2.2 Centralized vs. Distributed Solution

The scaling results in Section 4.8 on page 101 argue for a decomposed – and possibly parallel – solution. They do not, however, mean that a distributed algorithm is naturally superior to a centralized one. The messaging complexity of a the distributed implementation is higher:  $O(ni)$ , where  $i$  is the number of iterations required to converge, vs  $O(n)$  for a collect-process-disseminate scheme. The number of rounds  $i$ , as well as the amount of data sent per node per round, depends on the specific problem in a non-trivial way. If  $i$  is  $O(n)$  then the messaging complexity for a distributed solution is  $O(n^2)$ .

Distributed processing also opens the door to a class of security risks which are absent in the centralized scheme. Assume that the standard cryptographic goals of confidentiality, authenticity, and integrity are satisfied. Nodes can still report false information of several sorts. In the centralized collect-process-disseminate model, the remote nodes are likely involved in collecting and reporting what I will call **primary** data: the channel measurements, antenna effects, load (depending on how that is defined), etc. Distributed computation also allows them to tamper with **secondary** data – the results of computation on the primary data.

A malicious attacker could disrupt or mislead the algorithms in any number of ways: Changing the primary data could lead to slow schedules, to load being ignored, or to interference constraints being violated. Changing the secondary data could prevent the algorithms from converging, or could produce potentially arbitrary solutions. It is not immediately clear what primary data a selfish node would benefit from changing. In terms of secondary data, the variables which convey the importance of a node's traffic or signal ( $\beta$  and  $\lambda$ , respectively) are promising targets.

There is no obvious cryptographic solution to either primary or secondary data misreporting, although trusted hardware, encrypted computation, and other approaches might be applicable. Excluding those options, various forms of consistency checking suggest themselves as defenses. The options for checking primary data will depend very much on the specific measurement; convincingly falsifying a path loss value might require collusion by the nodes on both ends, for example. The secondary data is all produced by deterministic processes. In principle, any node's calculations can be checked – on line or after the fact – by any other node that has access to the input data. To take advantage of this in practice would probably require auditing support to be designed into the protocols, with some associated storage, communication, and computational overhead.

## 6.4 Conclusion

This chapter has presented quantitative performance results for the algorithms and system developed in the preceding chapters. The data presented here show that this approach is feasible, and that it yields significant gains with moderate computational overhead. Most importantly, this data show that the algorithms' execution time scales well, admitting the solution of large problems. Direct comparison with lighter-weight scheduling algorithms is planned for future work.

## Chapter 7

### Conclusions and Future Directions

#### 7.1 Current Work

In ongoing work, I am performing a quantitative experimental evaluation of this system in comparison with other MAC protocols and scheduling algorithms. I am additionally evaluating the extensions described in Section 3.1.2 on page 68, especially as applied to cognitive radio networking. More generally, I will be examining applying the same decomposition scheme to a broader range of optimization formulations – including the model introduced in Section 4.2 – so as to support other models of demand and link quality, and cross-layer integration with routing and congestion control schemes. Subsequent work will consider the dynamic behavior of this system, and on-line algorithms for more robust and responsive scheduling.

#### 7.2 Conclusion

This dissertation presents and solves the joint beam steering and scheduling problem. Optimal spatial reuse TDMA scheduling is known to be NP-hard, and the addition of antenna configuration increases the state space **exponentially** in the number of nodes. This research develops and explores algorithms for optimally scheduling wireless links with configurable antennas. Beyond being directly useful, optimal solutions provide a reference point for the design and evaluation of other systems. This is the **first implementation of STDMA scheduling based on dual decomposition**. These algorithms are computationally efficient—they find the optimal solutions within hundreds of iterations, each of which requires only minimal computation. Moreover, despite the

NP-hard nature of the underlying problem, in practice the running time appears linear in the problem size. The algorithms make no assumptions about the patterns of the antennas' directionality beyond the existence of some finite number of antenna states with known gains toward other nodes in each state. Lastly, this work identifies a price-coupled decomposition structure which can form a basis for other algorithms and protocols, including ones making different optimality-overhead trade-offs. I firmly believe that optimization decomposition is a paradigm that will drive next-generation wireless networks, and offer this work here as an important step towards realizing the theoretical gains of these algorithms on a real system.



## Bibliography

- [Abhayawardhana 05] V.S. Abhayawardhana, I.J. Wassell, D. Crosby, M.P. Sellars & M.G. Brown. Comparison of empirical propagation path loss models for fixed wireless access systems. In Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st, volume 1, pages 73–77. IEEE, June 2005.
- [Aein 65] J. M. Aein & J. W. Schwartz. Multiple access to a communication satellite with a hard-limiting repeater. Defense Technical Information Center, INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA., Ft. Belvoir, 1965. WorldCat ID. 227338744.
- [Aguayo 04a] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd & Robert Morris. Link-level Measurements from an 802.11b Mesh Network. In Proc. Sigcomm 2004. ACM, August 2004.
- [Aguayo 04b] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd & Robert Morris. Link-level measurements from an 802.11b mesh network. In SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pages 121–132, New York, NY, USA, 2004. ACM.
- [Alicherry 05] Mansoor Alicherry, Randeep Bhatia & Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking, pages 58–72, New York, NY, USA, 2005. ACM Press.
- [Andersen 95] Jørgen Bach Andersen, Theodore S. Rappaport & Susumu Yoshida. Propagation Measurements and Models for Wireless Communications Channels. IEEE Communications Magazine, vol. 33, no. 1, pages 42 – 49, Jan 1995.
- [Anderson 08a] Eric W. Anderson, Caleb T. Phillips, Kevin S. Bauer, Dirk C. Grunwald & Douglas C. Sicker. Modeling Directionality in

- Wireless Networks. In SIGMETRICS 2008, June 2008. Extended Abstract.
- [Anderson 08b] Eric W. Anderson, Caleb T. Phillips, Dirk Grunwald & Douglas Sicker. Modeling Environmental Effects on Directionality in Wireless Networks. Rapport technique CU-CS-1044-08, Department of Computer Science, University of Colorado at Boulder, July 2008.
- [Anderson 09a] Eric Anderson, Caleb Phillips, Douglas Sicker & Dirk Grunwald. Modeling Environmental Effects on Directionality in Wireless Networks. In 5th International workshop on Wireless Network Measurements (WiNMee), June 2009.
- [Anderson 09b] Eric Anderson, Caleb Phillips, Gary Yee, Douglas Sicker & Dirk Grunwald. Challenges in Deploying Steerable Wireless Testbeds. Rapport technique CU-CS-1068-09, Department of Computer Science, University of Colorado at Boulder, December 2009.
- [Anderson 09c] Eric Anderson, Gary Yee, Caleb Phillips, Michael Buettner, Douglas Sicker & Dirk Grunwald. Challenges in Deploying Steerable Wireless Testbeds. In submission to Mobile Networks and Applications, 2009.
- [Anderson 09d] Eric Anderson, Gary Yee, Caleb Phillips, Dirk Grunwald & Douglas Sicker. The Impact of Directional Antenna Models on Simulation Accuracy. In 7th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), June 2009.
- [Anderson 10a] Eric Anderson, Caleb Phillips, Gary Yee, Douglas Sicker & Dirk Grunwald. Challenges in Deploying Steerable Wireless Testbeds. In Proc. 6th International conference on testbeds and research infrastructures for the development of networks and communities (TridentCom), 2010.
- [Anderson 10b] Eric Anderson, Caleb Phillips, Douglas Sicker & Dirk Grunwald. Good fences make good neighbors: Optimal scheduling and beam switching for spatial reuse. In In Submission, 2010.
- [Arikan 83] Erdal Arikan. Some complexity results about packet radio networks. Rapport technique LIDS-P ; 1284, Massachusetts Institute of Technology. Laboratory for Information and Decision Systems., 1983.
- [Arikan 84] Erdal Arikan. Some Complexity Results about Packet Radio Networks. IEEE Transactions on Information Theory, vol. 30, no. 4, pages 681 – 685, July 1984.

- [Asakura 99] H. Asakura & T. Matsumoto. Cooperative signal reception and down-link beam forming in cellular mobile communications. *Vehicular Technology, IEEE Transactions on*, vol. 48, no. 2, pages 333–341, 1999.
- [Atheros Communications, Inc. 04] Atheros Communications, Inc., Paul J. Husted, Huanchun Ye & Aman Singla. Adaptive Interference Immunity Control. United States patent. World Intellectual Property Organization (WIPO) publication number WO/2005/048473, 2004.
- [auf der Heide 02] Friedhelm Meyer auf der Heide, Christan Schindelhauer, Klaus Volbert & Matthias Grünewald. Energy, congestion and dilation in radio networks. In *SPAA '02: Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, pages 230–237, New York, NY, USA, 2002. ACM Press.
- [Awerbuch 04] Baruch Awerbuch, David Holmer & Herbert Rubens. High Throughput Route Selection in Multi-Rate Ad Hoc Wireless Networks. In *Wireless On-demand Network Systems (WONS 2004)*, Madonna di Campiglio, Italy, January 2004.
- [Babich 06] F. Babich, M. Comisso, M. D’Orlando & L. Manià. Interference Mitigation on WLANs Using Smart Antennas. *Wirel. Pers. Commun.*, vol. 36, no. 4, pages 387–401, 2006.
- [Balasundaram 06] Balabhaskar Balasundaram & Sergiy Butenko. Graph domination, coloring, and cliques in telecommunications. In Mauricio G.C. Resende & Panos M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, chapter Graph domination, coloring, and cliques in telecommunications, pages 865 – 890. Springer, 1 edition, 2006.
- [Baldo 07] Nicola Baldo, Federico Maguolo, Marco Miozzo, Michele Rossi & Michele Zorzi. ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2. In *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [Baldo 08] N. Baldo & M. Zorzi. Fuzzy logic for cross-layer optimization in cognitive radio networks. *Communications Magazine, IEEE*, vol. 46, no. 4, pages 64–71, April 2008.
- [Bandyopadhyay 01] S. Bandyopadhyay, K. Hasuike, S. Horisawa & S. Tawara. An adaptive MAC and directional routing protocol for ad hoc wireless network using ESPAR antenna. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile*

ad hoc networking & computing, pages 243–246, New York, NY, USA, 2001. ACM.

- [Bao 01] Lichun Bao & J. J. Garcia-Luna-Aceves. Channel access scheduling in Ad Hoc networks with unidirectional links. In DIALM '01: Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications, pages 9–18, New York, NY, USA, 2001. ACM Press.
- [Bao 02a] L. Bao & J.J. Garcia-Luna-Aceves. Distributed Transmission Scheduling Using Code-Division Channelization. In Proc. Networking 2002, number 2345 in Lecture Notes in Computer Science (LNCS), Pisa, Italy,, May 2002.
- [Bao 02b] Lichun Bao & J. J. Garcia-Luna-Aceves. Transmission Scheduling in Ad Hoc Networks with Directional Antennas. In MOBICOM '02, Proceedings of, pages 48 – 58. ACM, 2002.
- [Bao 03] Lichun Bao & J. J. Garcia-Luna-Aceves. Distributed dynamic channel access scheduling for ad hoc networks. J. Parallel Distrib. Comput., vol. 63, no. 1, pages 3–14, 2003.
- [Behzad 03] A. Behzad & I. Rubin. On the performance of graph-based scheduling algorithms for packet radio networks. In Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, volume 6, pages 3432–3436 vol.6, Dec. 2003.
- [Behzad 07] Arash Behzad & Izhak Rubin. Optimum Integrated Link Scheduling and Power Control for Multihop Wireless Networks. In IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, volume 56, pages 194 – 205. IEEE, January 2007.
- [Bellare 95] Mihir Bellare & Phillip Rogaway. The Complexity of Approximating a Nonlinear Program. Journal of Mathematical Programming, vol. 69, no. 3, pages 429 – 441, September 1995.
- [Bellofiore 02a] S. Bellofiore, C.A. Balanis, J. Foutz & A.S. Spanias. Smart-antenna systems for mobile communication networks. Part 1. Overview and antenna design. Antennas and Propagation Magazine, IEEE, vol. 44, no. 3, pages 145–154, Jun 2002.
- [Bellofiore 02b] S. Bellofiore, J. Foutz, C.A. Balanis & A.S. Spanias. Smart-antenna system for mobile communication networks .Part 2. Beamforming and network throughput. Antennas and Propagation Magazine, IEEE, vol. 44, no. 4, pages 106–114, Aug 2002.

- [Bellofiore 02c] S. Bellofiore, J. Foutz, R. Govindarajula, I. Bahceci, C.A. Balanis, A.S. Spanias, J.M. Capone & T.M. Duman. Smart antenna system analysis, integration and performance for mobile ad-hoc networks (MANETs). *Antennas and Propagation, IEEE Transactions on*, vol. 50, no. 5, pages 571–581, May 2002.
- [Ben Abdesslem 06] Fehmi Ben Abdesslem, Luigi Iannone, Marcelo Dias de Amorim, Konstantin Kabassanov & Serge Fdida. On the feasibility of power control in current IEEE 802.11 devices. In *PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops*, page 468, Washington, DC, USA, 2006. IEEE Computer Society.
- [Bertsimas 97] Dimitris Bertsimas & Xiaodong Luo. On the worst case complexity of potential reduction algorithms for linear programming. *Mathematical Programming*, vol. 77, pages 321 – 333, 1997.
- [Bicket 05] John Bicket, Daniel Aguayo, Sanjit Biswas & Robert Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *MobiCom '05*. ACM, August 2005.
- [Björklund 03] Patrick Björklund, Peter Varbrand & Di Yuan. Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 818– 824. IEEE, March 2003.
- [Björklund 06] Patrik Björklund. Applications of Resource Optimization in Wireless Networks. PhD thesis, Linköping Universitet, Linköping, Sweden, 2006.
- [Boche 05] H. Boche, M. Wiczanowski & S. Stanczak. Unifying view on min-max fairness and utility optimization in cellular networks. In *Proc. Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1280– 1285. IEEE, March 2005.
- [Boche 07] Holger Boche, Marcin Wiczanowski & Slawomir Stanczak. Unifying view on min-max fairness, max-min fairness, and utility optimization in cellular networks. *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 1, pages 5–5, 2007.
- [Bonuccelli 04] Maurizio A. Bonuccelli, Francesca Martelli & Susanna Pelagatti. An optimal packet scheduling and load balancing algorithm for LEO/MEO satellite networks. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 47–51, New York, NY, USA, 2004. ACM Press.

- [Boyd 04a] Stephen Boyd & Lieven Vandenberghe. Convex optimization. Cambridge University Press, 2004.
- [Boyd 04b] Stephen P. Boyd, Seung Jean Kim, Arash Hassibi & Lieven Vandenberghe. A Tutorial on Geometric Programming. Rapport technique, Stanford University, 2004.
- [Boyd 07] Steven Boyd, Seung-Jean Kim, Lieven Vandenberghe & Arash Hasibi. A tutorial on geometric programming. Optimization Engineering, vol. 8, pages 67 – 127, Apr 2007.
- [Briscoe 07] Bob Briscoe. Flow rate fairness: dismantling a religion. SIGCOMM Comput. Commun. Rev., vol. 37, no. 2, pages 63–74, 2007.
- [Broustis 07] Ioannis Broustis, Jakob Eriksson, Shrikanth V. Krishnamurthy & Michalis Faloutsos. A Blueprint for a Manageable and Affordable Wireless Testbed: Design, Pitfalls, and Lessons Learned. In TridentCom, 2007.
- [Brzezinski 08] A. Brzezinski, G. Zussman & Eytan Modiano. Distributed Throughput Maximization in Wireless Mesh Networks via Pre-Partitioning. IEEE/ACM Trans. on Networking, vol. , page , 2008. to appear.
- [Buettner 07] Michael Buettner, Eric W. Anderson, Gary Yee, Dola Saha, Doug Sicker & Dirk Grunwald. A Phased Array Antenna Testbed for Evaluating Directionality in Wireless Networks. In MobiEval '07, San Juan, Puerto Rico, USA, June 2007. ACM.
- [Cai 03] Zhijun Cai, Mi Lu & Costas N. Georghiadis. Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks. Vehicular Technology, IEEE Transactions on, vol. 52s, no. 4, pages 970–984, July 2003.
- [Cain 03] J. Bibb Cain, Tom Billhartz, Larry Foore, Edwin Althouse & John Schlorff. A link scheduling and ad hoc networking approach using directional antennas. In Military Communications Conference, 2003, volume 1, pages 643– 648. IEEE, Oct 2003.
- [Camp 06] Joseph Camp, Joshua Robinson, Christopher Steger & Edward Knightly. Measurement driven deployment of a two-tier urban mesh access network. In MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services, pages 96–109, New York, NY, USA, 2006. ACM.
- [Chakraborty 04] Goutam Chakraborty. Genetic algorithm to solve optimum TDMA transmission schedule in broadcast packet radio

- networks. Communications, IEEE Transactions on, vol. 52, no. 5, pages 765–777, May 2004.
- [Chawla 99] K. Chawla & Xiaoxin Qiu. Quasi-static resource allocation with interference avoidance for fixed wireless systems. Selected Areas in Communications, IEEE Journal on, vol. 17, no. 3, pages 493–504, 1999.
- [Chen 05] Lijun Chen, Steven H. Low & John C. Doyle. Joint congestion control and media access control design for ad hoc wireless networks. In IEEE INFOCOM, volume 3, pages 2212 – 2222. IEEE, March 2005.
- [Chen 06] Lijun Chen, Steven H. Low, Mung Chiang & John C. Doyle. Optimal Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks. In Proceedings of IEEE INFOCOM 2006, 2006.
- [Cherreddi 06] Chandrakanth Cherreddi, Pradeep Kyasanur & Nitin H. Vaidya. Design and implementation of a multi-channel multi-interface network. In REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, pages 23–30, New York, NY, USA, 2006. ACM.
- [Chiang 05a] Mung Chiang. Balancing transport and physical Layers in wireless multihop networks: jointly optimal congestion control and power control. Selected Areas in Communications, IEEE Journal on, vol. 23, no. 1, pages 104–116, 2005.
- [Chiang 05b] Mung Chiang. Geometric programming for communication systems. now Publishers Inc. Hannover, MA, 2005. ISBN 1-933019-09-3.
- [Chiang 06a] Mung Chiang, S.H. Low, A.R. Calderbank & J.C. Doyle. Layering As Optimization Decomposition: Current Status and Open Issues. In Proc. 40th Annual Conference on Information Sciences and Systems, pages 355–362, 2006.
- [Chiang 06b] Mung Chiang, S.H. Low, A.R. Calderbank & J.C. Doyle. Layering as Optimization Decomposition: Questions and Answers. In Proc. Military Communications Conference MILCOM 2006, pages 1–10, 2006.
- [Chiang 07] Mung Chiang, S.H. Low, A.R. Calderbank & J.C. Doyle. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. Proceedings of the IEEE, vol. 95, no. 1, pages 255–312, 2007.
- [Chlamtac 87] Imrich Chlamtac & Shlomit S. Pinter. Distributed nodes organization algorithm for channel access in a multihop

- dynamic radio network. IEEE Trans. Comput., vol. 36, no. 6, pages 728–737, 1987.
- [Chlamtac 94] Imrich Chlamtac & Andr’as Faragó. Making transmission schedules immune to topology changes in multi-hop packet radio networks. Networking, IEEE/ACM Transactions on, vol. 2, no. 1, pages 23–29, Feb 1994.
- [Choudhury 02] Romit Roy Choudhury, Xue Yang, Ram Ramanathan & Nitin H. Vaidya. Using directional antennas for medium access control in ad hoc networks. In MobiCom ’02: Proceedings of the 8th annual international conference on Mobile computing and networking, pages 59–70, New York, NY, USA, 2002. ACM.
- [Choudhury 05] Romit Roy Choudhury & Nitin H. Vaidya. Performance of ad hoc routing using directional antennas. Ad Hoc Networks, vol. 3, no. 2, pages 157–173, March 2005.
- [Choudhury 06] Romit Roy Choudhury, Xue Yang, Ram Ramanathan & Nitin H. Vaidya. On Designing MAC Protocols for Wireless Networks using Directional Antennas. IEEE Transactions on Mobile Computing, vol. 5, no. 5, pages 477–491, 2006.
- [Chvátal 80] Vašek Chvátal. Linear programming. Freeman Press, 1980.
- [Clark 90] David D. Clark & David L. Tennenhouse. Architectural considerations for a new generation of protocols. SIGCOMM Comput. Commun. Rev., vol. 20, no. 4, pages 200–208, 1990.
- [Conejo 02] A. J. Conejo, F. J. Nogales & F. J. Prieto. A decomposition procedure based on approximate Newton directions. Mathematical Programming, vol. 93, no. 3, pages 495 – 515, 2002.
- [Conejo 06] Antonio J. Conejo, Enrique Castillo, Robert Mínguez & Raquel García-Bertrand. Decomposition techniques in mathematical programming: Engineering and science applications. Springer-Verlag Berlin / Heidelberg, 2006.
- [Coupechoux 05] Marceau Coupechoux, Bruno Baynat, Christian Bonnet & Vinod Kumar. CROMA: an enhanced slotted MAC protocol for MANETs. Mob. Netw. Appl., vol. 10, no. 1-2, pages 183–197, 2005.
- [Cover 91] Thomas M. Cover & Joy A. Thomas. Elements of information theory. Wiley-Interscience, New York, NY, USA, 1991.
- [Crawdad 08] Crawdad. CRAWDAD. <http://crawdad.org/>, Feb 2008.
- [Cruz 02] Rene L. Cruz & Arvind V. Santhanam. Optimal link scheduling and power control in cdma multihop wireless networks. In Globecom, pages 52–56, November 2002.



- [Dantzig 60] George B. Dantzig & Philip Wolfe. Decomposition Principle for Linear Programs. Operations Research, vol. 8, no. 1, pages 101–111, 1960.
- [Dantzig 63] George B. Dantzig. Linear programming and extensions. Princeton University Press, 1963.
- [Das 06] Saumitra M. Das, Himabindu Pucha, Dimitrios Koutsonikolas, Y. Charlie Hu & Dimitrios Peroulis. DMesh: Incorporating Practical Directional Antennas in Multi-Channel Wireless Mesh Networks. Journal on Selected Areas in Communications, vol. 24, no. 11, special issue on Multi-Hop Wireless Mesh Networks, November 2006.
- [Das 07] Saumitra Das, Dimitrios Koutsonikolas & Y. Charlie Hu. Practical Service Provisioning for Wireless Meshes. In In Proceedings of the 3rd ACM SIGCOMM International Conference on emerging Networking EXperiments and Technologies (CoNEXT 07). ACM, December 2007.
- [de Leon 04] C.A. Gutierrez Diaz de Leon, M.C. Bean & J.S. Garcia. On the generation of correlated Rayleigh envelopes for representing the variant behavior of the indoor radio propagation channel. In Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on, volume 4, pages 2757 – 2761, Sept 2004.
- [De 05] P. De, A. Raniwala, S. Sharma & T. Chiueh. MiNT: a miniaturized network testbed for mobile wireless research. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4, pages 2731–2742 vol. 4, March 2005.
- [De 06] Pradipta De, Ashish Raniwala, Rupa Krishnan, Krishna Tatavarthi, Jatan Modi, Nadeem Ahmed Syed, Srikant Sharma & Tzi cker Chiueh. MiNT-m: an autonomous mobile wireless experimentation platform. In MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services, pages 124–137, New York, NY, USA, 2006. ACM.
- [Deb 08] Supratim Deb, Vivek Mhatre & Venkatesh Ramaiyan. WiMAX relay networks: opportunistic scheduling to exploit multiuser diversity and frequency selectivity. In MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking, pages 163–174, New York, NY, USA, 2008. ACM.
- [Deopura 07] Ashish Deopura & Aura Ganz. Provisioning link layer proportional service differentiation in wireless networks with

- smart antennas. *Wirel. Netw.*, vol. 13, no. 3, pages 371–378, 2007.
- [Desautel 02] E.G. Desautel, Dongsu Kim, J.S. Kenney & D. Kiesling. Interference mitigation in WLAN networks using client-based smart antennas. In *Radio and Wireless Conference, 2002. RAWCON 2002*. IEEE, 2002.
- [Desrosiers 05] Jacques Desrosiers & Marco E. Lübbecke. A Primer in Column Generation. In Guy Desaulniers, Jacques Desrosiers & Marius M. Solomon, editors, *Column Generation*, Groupe d'études et de recherche en analyse des décisions (GERAD) 25th anniversary series, pages 1 – 33. Springer, 2005.
- [Djukic 07a] P. Djukic & S. Valaee. Distributed Link Scheduling for TDMA Mesh Networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3823–3828, June 2007.
- [Djukic 07b] P. Djukic & S. Valaee. Link Scheduling for Minimum Delay in Spatial Re-Use TDMA. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, pages 28–36, May 2007.
- [Djukic 08] Petar Djukic. Scheduling Algorithms for TDMA Wireless Multihop Networks. PhD thesis, Department of Electrical and Computer Engineering; University of Toronto, 2008.
- [Duffin 67] Richard J. Duffin, Elmore L. Peterson & Clarence Zener. *Geometric programming – theory and application*. Wiley, 1967.
- [Dür 97] M. Dür & Reiner Horst. Lagrange Duality and Partitioning Techniques in Nonconvex Global Optimization. *Journal of Optimization Theory and Applications*, 1997.
- [Dyberg 02a] K. Dyberg, F. Farman L.and Eklof, J. Grönkvist, U. Sterner & J. Rantakokko. On the performance of antenna arrays in spatial reuse TDMA ad hoc networks. In *MILCOM 2002*, volume 1, pages 270– 275, Oct 2002.
- [Dyberg 02b] Karin Dyberg & Linda Farman. Antenna Arrays in Spatial reuse TDMA Ad Hoc Networks. Methodology Report FOI-R-0444-SE, FOI Swedish Defense Research Agency, Command and Control Systems, P.O. Box 1165, SE-581 11 Linköping, Sweden, March 2002. ISSN 1650-1942.
- [Ecker 80] J. G. Ecker. Geometric Programming: Methods, Computations and Applications. *SIAM Review*, vol. 22, no. 3, pages 338 – 362, July 1980.
- [Efrimidis 08] Pavlos S. Efrimidis. The complexity of linear programming in  $([\gamma],[\kappa])$ -form. *Information Processing Letters*, vol. 105, no. 5, pages 199 – 201, 2008.

- [ElBatt 02a] T. ElBatt & A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In IEEE Proceedings of INFOCOM, 2002.
- [Elbatt 02b] T. Elbatt & Bo Ryu. On the channel reservation schemes for ad-hoc networks utilizing directional antennas. In Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on, volume 2, pages 766–770 vol.2, Oct. 2002.
- [Ephremides 90] Anthony Ephremides & Thuan V. Truong. Scheduling broadcasts in multihop radio networks. Communications, IEEE Transactions on, vol. 38, no. 4, pages 456–460, Apr 1990.
- [Ertin 06] Emre Ertin, Anish Arora, Rajiv Ramnath, Mikhail Nesterenko, Vinayak Naik, Sandip Bapat, Vinod Kulathumani, Mukundan Sridharan, Hongwei Zhang & Hui Cao. Kansei: A Testbed for Sensing at Scale. In IPSN/SPOTS, 2006.
- [Eryilmaz 06] Atila Eryilmaz & R. Srikant. Joint Congestion Control, Routing, and MAC for Stability and Fairness in Wireless Networks. Selected Areas in Communications, IEEE Journal on, vol. 24, no. 8, pages 1514–1524, 2006.
- [Evans 99] Joseph B. Evans, Gary J Minden, K.S. Shanmugan, Glen Prescott, Victor S. Frost, Ben Ewy, Ricardo Sanchez, Craig Sparks, K. Malinmohan, James Roberts, Richard Plumb & Dave Petr. The Rapidly Deployable Radio Network. IEEE Journal on Selected Areas in Communication, vol. 17, no. 4, pages 689 – 703, April 1999.
- [Fahmy 02] N.S. Fahmy, T.D. Todd & V. Kezys. Ad hoc networks with smart antennas using IEEE 802.11-based protocols. In Communications, 2002. ICC 2002. IEEE International Conference on, volume 5, pages 3144–3148 vol.5, 2002.
- [Fattah 02] H. Fattah & C. Leung. An overview of scheduling algorithms in wireless multimedia networks. Wireless Communications, IEEE, vol. 9, no. 5, pages 76–83, Oct. 2002.
- [Foschini 93] Gerard J. Foschini & Zoran Miljanic. A Simple Distributed Autonomous Power Control Algorithm and its Convergence. In IEEE Transactions on Vehicular Technology (VTC), volume 42, pages 641 – 646. IEEE, Nov 1993.
- [Freeman 97] Roger L. Freeman. Radio system design for telecommunications. Wiley Series In Telecommunications And Signal Processing. Wiley InterScience, 2nd edition, 1997. ISBN:0-471-16260-4.

- [Galvan-Tejada 01] G.M. Galvan-Tejada & J.G. Gardiner. Theoretical model to determine the blocking probability for SDMA systems. *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 5, pages 1279–1288, 2001.
- [Garache 08] Marvin Sánchez Garache. Multihop Wireless Networks with Advanced Antenna Systems - An Alternative for Rural Communication. PhD thesis, Royal Institute of Technology (KTH), May 2008.
- [Garetto 05] Michele Garetto, Jingpu Shi & Edward W. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 200–214, New York, NY, USA, 2005. ACM Press.
- [Ghaderi 09] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley & R. Cruz. TCP-Aware Channel Allocation in CDMA Networks. *Mobile Computing, IEEE Transactions on*, vol. 8, no. 1, pages 14–28, Jan. 2009.
- [Gilmore 61] P. C. Gilmore & R. E. Gomory. A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, vol. 9, no. 6, pages 849–859, 1961.
- [Gilmore 63] P. C. Gilmore & R. E. Gomory. A Linear Programming Approach to the Cutting Stock Problem-Part II. *Operations Research*, vol. 11, no. 6, pages 863–888, 1963.
- [Godara 97a] Lal C. Godara. Application of Antenna Arrays to Mobile Communications, Part I: Performance Improvement, Feasibility and System Considerations. *Proceedings of the IEEE*, vol. 85, no. 7, pages 1031–1060, July 1997.
- [Godara 97b] Lal C. Godara. Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Considerations. *Proceedings of the IEEE*, vol. 85, no. 8, pages 1195 – 1245, August 1997.
- [Godara 04] Lal Chand Godara. *Smart antennas*. CRC Press, 2004. ISBN: 0-8493-1206-X.
- [Gondzio 06] Jacek Gondzio & Andreas Grothey. Direct Solution of Linear Systems of Size  $10^9$  Arising in Optimization with Interior Point Methods. *Parallel Processing and Applied Mathematics*, no. 2911, pages 513–525, 2006.
- [Gore 07] Ashutosh Deepak Gore, Srikanth Jagabathula & Abhay Karandikar. On High Spatial Reuse Link Scheduling in

- STDMA Wireless Ad Hoc Networks. ArXiv Computer Science e-prints, vol. , Jan 2007. Also presented at GLOBECOM 2007.
- [Green 02] D.B Green & A.S. Obaidat. An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices. In Communications, 2002. ICC 2002. IEEE International Conference on, volume 5, pages 3424 – 3428, 2002.
- [Grönkvist 01] Jimmi Grönkvist & Anders Hansson. Comparison between graph-based and interference-based STDMA scheduling. In MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pages 255–258, New York, NY, USA, 2001. ACM Press.
- [Grothey 01] Andreas Grothey. Decomposition Methods for Nonlinear Nonconvex Optimization Problems. PhD thesis, Department of Mathematics and Statistics University of Edinburgh, June, 2001.
- [Guan 95] Xiaohong Guan, Peter B. Luh & Lan Zhang. Nonlinear Approximation Method in Lagrangian Relaxation-Based Algorithms for Hydrothermal Scheduling. IEEE Transactions on Power Systems, vol. 10, no. 2, pages 772 – 778, May 1995.
- [Guo 03] Xingang Guo, Sumit Roy & W. Steven Conner. Spatial reuse in wireless ad-hoc networks. In Vehicular Technology Conference, 58th, volume 3, pages 1437– 1442. IEEE, Oct 2003.
- [Gupta 00] Piyush Gupta & P. R. Kumar. The Capacity of Wireless Networks. IEEE Transactions on Information Theory, vol. IT-46, no. 2, pages 388–404, March 2000.
- [Hajek 88] Bruce Hajek & Galen Sasaki. Link scheduling in polynomial time. Information Theory, IEEE Transactions on, vol. 34, no. 5, pages 910–917, Sep 1988.
- [Hiriart-Urruty 98] Jean-Baptiste Hiriart-Urruty. Conditions for Global Optimality 2. J. of Global Optimization, vol. 13, no. 4, pages 349–367, 1998.
- [Ho 98] Ming-Ju Ho, G.L. Stuber & M.D. Austin. Performance of switched-beam smart antennas for cellular radiosystems. Vehicular Technology, IEEE Transactions on, vol. 47, no. 1, pages 10–19, Feb 1998.
- [Horst 00] Reiner Horst, Panos Pardalos & Nguyen V. Thoai. Introduction to global optimization, volume 48 of Nonconvex Optimization and Its Applications. Kluwer Academic Press, Inc., 2nd edition, 2000.

- [Hottinen 06] Ari Hottinen & Tiina Heikkinen. Subcarrier Allocation in a Multiuser MIMO Channel using Linear Programming. In EU-SIPCO 2006 Special Session on ?Multi-user MIMO Communications?, 2006.
- [Huang 01] Xiao Long Huang & Brahim Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pages 221–231, New York, NY, USA, 2001. ACM.
- [Huang 02] Zhuochuan Huang, Chien-Chung Shen, Chavalit Srisathapornphat & Chaiporn Jaikaeo. Topology Control for Ad Hoc Networks with Directional Antennas. In Proc. IEEE Int. Conference on Computer Communications and Networks, pages 16 – 21, 2002.
- [Hultberg 65] Richard M. Hultberg, Floyd H. Jean & Milton E. Jones. Time Division Access for Military Communications Satellites. Aerospace and Electronic Systems, IEEE Transactions on, vol. AES-1, no. 3, pages 272–282, Dec. 1965.
- [IEEE 99] IEEE. IEEE Std. 802.11a-1999 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. High Speed Physical Layer in the 5 GHz Band. Rapport technique ISO/IEC 8802-11:1999/Amd 1:2000(E), LAN/MAN Standards Committee, IEEE Computer Society, 1999.
- [Iskander 02] Magdy F. Iskander & Zhengqing Yun. Propagation Prediction Models for Wireless Communication Systems. IEEE Transactions on microwave theory and techniques, vol. 50, no. 3, pages 662 – 673, March 2002.
- [Jaikaeo 03] C. Jaikaeo & C.-C. Shen. Multicast communication in ad hoc networks with directional antennas. In Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on, pages 385–390, Oct. 2003.
- [Jarquín 02] Oscar Somarriba Jarquín. On Constrained Power Control for Spatial TDMA in Multi-hop Ad Hoc Radio Networks. In Proceedings Swedish Workshop on Wireless Ad-hoc Networks, Mar 2002.
- [Johansson 06] M. Johansson & L. Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. Wireless Communications, IEEE Transactions on, vol. 5, no. 2, pages 435–445, 2006.

- [Johnson 06] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci & J. Lepreau. Mobile Emulab: A Robotic Wireless and Sensor Network Testbed. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1–12, April 2006.
- [Jorswieck 07] Eduard Jorswieck, Patrick Svedman & Björn Ottersten. On the Performance of TDMA and SDMA based Opportunistic Beamforming. IEEE Transactions on Wireless Communications, vol. unk., Mar 2007. Submitted.
- [Ju 98] Ji-Her Ju & Victor O.K. Li. An optimal topology-transparent scheduling method in multihop packet radio networks. Networking, IEEE/ACM Transactions on, vol. 6, no. 3, pages 298–306, Jun 1998.
- [Ju 99] Ji-Her Ju & Victor O.K. Li. TDMA scheduling design of multihop packet radio networks based on latin squares. Selected Areas in Communications, IEEE Journal on, vol. 17, no. 8, pages 1345–1352, Aug 1999.
- [Julian 02] D. Julian, Mung Chiang, D. O’Neill & S. Boyd. QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 477 – 486, jun. 2002.
- [Kandukuri 02] S. Kandukuri & S. Boyd. Optimal power control in interference-limited fading wireless channels with outage-probability specifications. Wireless Communications, IEEE Transactions on, vol. 1, no. 1, pages 46 –55, jan. 2002.
- [Kelly 98] Frank. P. Kelly, A. K. Maulloo & D. K. H.Xi Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. The Journal of the Operational Research Society, vol. 49, no. 3, pages 237–252, 1998.
- [Kim 05] Il-Min Kim, R. Yim & H. Chaskar. Optimum scheduling for smart antenna systems in Rayleigh fading channels. Communications, IEEE Transactions on, vol. 53, no. 7, pages 1210–1219, July 2005.
- [Ko 00] Young-Bae Ko, Vinaychandra Shankarkumar & Nitin H. Vaidya. Medium Access Control Protocols using Directional Antennas in Ad Hoc Networks. In INFOCOM (1), pages 13–21, 2000.

- [Kodialam 03] Murali Kodialam & Thyaga Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 42–54, New York, NY, USA, 2003. ACM Press.
- [Kodialam 05] Murali Kodialam & Thyaga Nandagopal. Characterizing the capacity region in multi-radio multichannel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM Press.
- [Kohmura 08] Naoya Kohmura, Hikaru Mitsuhashi, Masahiro Watanabe, Masaki Bandai, Sadao Obana & Takashi Watanabe. UNAGI: a protocol testbed with practical smart antennas for ad hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 1, pages 59–61, 2008.
- [Koutsonikolas 07] Dimitrios Koutsonikolas, Saumitra M. Das & Y. Charlie Hu. An interference-aware fair scheduling for multicast in wireless mesh networks. *Journal of Parallel and Distributed Computing*, vol. , 2007.
- [Koutsopoulos 03] Iordanis Koutsopoulos, Tianmin Ren & Leandros Tassiulas. The Impact of Space Division Multiplexing on Resource Allocation: A Unified Approach. In *INFOCOM*, San Francisco, April 2003. IEEE.
- [Kozat 04] Ulaş Kozat, Iordanis Koutsopoulos & Leandros Tassiulas. A Framework for Cross-layer Design of Energy-efficient Communication with QoS Provisioning in Multi-hop Wireless Networks. In *Infocom 2004*, volume 1, Hong Kong, March 2004.
- [Kuehner 01] R. Kuehner, T.D. Todd, F. Shad & V. Kezys. Forward-link capacity in smart antenna base stations with dynamic slot allocation. *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 4, pages 1024–1038, Jul 2001.
- [Kyasanur 05a] Pradeep Kyasanur. Mesh networking protocols to exploit physical layer capabilities. In *First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.
- [Kyasanur 05b] Pradeep Kyasanur & Nitin H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 43–57, New York, NY, USA, 2005. ACM Press.



- [Kyasaur 06a] Pradeep Kyasanur. Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad hoc Wireless Networks. SIG-MOBILE Mobile Computing and Communications Review, vol. 10, no. 1, pages 31–43, January 2006.
- [Kyasaur 06b] Pradeep Narayanaswamy Kyasanur. Multichannel Wireless Networks: Capacity and Protocols. PhD thesis, University of Illinois at Urbana-Champaign, 2006.
- [La 02] Richard J. La & Venkat Anantharam. Utility-based rate control in the Internet for elastic traffic. IEEE/ACM Trans. Netw., vol. 10, no. 2, pages 272–286, 2002.
- [Lakshmanan 08] Sriram Lakshmanan, Cheng-Lin Tsao, Raghupathy Sivakumar & Karthikeyan Sundaresan. Securing Wireless Data Networks against Eavesdropping using Smart Antennas. ICDCS, vol. 0, pages 19–27, 2008.
- [Lal 04a] D. Lal, T. Joshi & D.P. Agrawal. Localized transmission scheduling for spatial multiplexing using smart antennas in wireless adhoc networks. In Proc. 13th IEEE Workshop on Local and Metropolitan Area Networks LANMAN 2004, pages 175–180, 2004.
- [Lal 04b] Dhananjay Lal, Vivek Jain, Qing-An Zeng & Dharma P. Agrawal. Performance Evaluation of Medium Access Control for Multiple-Beam Antenna Nodes in a Wireless LAN. IEEE Trans. Parallel Distrib. Syst., vol. 15, no. 12, pages 1117–1129, 2004. Student Member-Dhananjay Lal and Student Member-Vivek Jain and Member-Qing-An Zeng and Fellow-Dharma P. Agrawal.
- [Larsson 99] Torbjörn Larsson, Michael Patriksson & Ann-Brith Strömberg. Ergodic, primal convergence in dual subgradient schemes for convex programming. Mathematical Programming, vol. 86, pages 283 – 312, Nov 1999.
- [Lee 05] Ian W.C. Lee & Abraham O. Fapojuwo. Stochastic processes for computer network traffic modeling. Computer Communications, vol. 29, no. 1, pages 1–23, December 2005.
- [Li 97] Yingjie Li, N.J. Feuerstein & D.O. Reudink. Performance evaluation of a cellular base station multibeam antenna. In Vehicular Technology, IEEE Transactions on, 1997.
- [Li 05a] Guoqing Li, Lily Yang, W. Steven Conner & Bahar Sadeghi. Opportunities and Challenges in Mesh Networks Using Directional Antennas. In WiMesh Workshop, SECON. IEEE, September 2005.

- [Li 05b] Xiang-Yang Li, Wen-Zhan Song & Weizhao Wang. A unified energy-efficient topology for unicast and broadcast. In *Mobi-Com '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 1–15, New York, NY, USA, 2005. ACM Press.
- [Liao 02] Wen-Hwa Liao, Yu-Chee Tseng & Kuei-Ping Shih. A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Network. In *IEEE International Conference on Communications (ICC 2002)*, New York, NY, April 2002.
- [Lin 04] Tzu-Ming Lin & Juin-Jia Dai. A Collision Free MAC Protocol using Smart Antenna in Ad Hoc Networks. In *CCNC 2004*, Jan 2004.
- [Lin 06a] X. Lin, N.B. Shroff & R. Srikant. A Tutorial on Cross-Layer Optimization in Wireless Networks. *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pages 1452–1463, 2006.
- [Lin 06b] Xiaojun Lin & Ness B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pages 302–315, 2006.
- [Lin 07] Xiaojun Lin & Shahzada Rasool. A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks. In *26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1118–1126. IEEE, May 2007.
- [Liu 98] Rui Liu & Errol L. Lloyd. A Distributed Protocol For Adaptive Link Scheduling in Ad-hoc Networks. In *2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1998.
- [Liu 01] Xin Liu, Edwin K. P. Chong & Ness B. Shroff. Opportunistic transmission scheduling with resource-sharing constraints in wireless networks. *JSAC*, vol. 19, no. 10, pages 2053–2064, Oct 2001.
- [Liu 07] H. Liu, H. Yu, X. Liu, C-N. Chuah & P. Mohapatra. Scheduling Multiple Partially Overlapped Channels in Wireless Mesh Networks. In *IEEE Proc. International Conference on Communications (ICC)*, June 2007.
- [Lloyd 02] Errol L. Lloyd. Broadcast scheduling for TDMA in wireless multihop networks. In *Handbook of wireless networks and mobile computing*, pages 347–370. John Wiley & Sons, Inc., New York, NY, USA, 2002.

- [Lundgren 02] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom & C. Tschudin. A large-scale testbed for reproducible ad hoc protocol evaluations. *Wireless Communications and Networking Conference, 2002. WCNC2002*. 2002 IEEE, vol. 1, pages 412–418 vol.1, Mar 2002.
- [Luo 00] Haiyun Luo, Songwu Lu & Vaduvur Bharghavan. A new model for packet scheduling in multihop wireless networks. In *Mobile Computing and Networking (MOBICOM)*, pages 76–86, 2000.
- [Luo 04] Haiyun Luo, Songwu Lu, Vaduvur Bharghavan, Jerry Cheng & Gary Zhong. A packet scheduling approach to QoS support in multihop wireless networks. *Mob. Netw. Appl.*, vol. 9, no. 3, pages 193–206, 2004.
- [Ma 98] X. Ma & E.L. Lloyd. A distributed protocol for adaptive broadcast scheduling in packet radio networks. In *Workshop record of the 2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M for Mobility)*, 1998.
- [Macedo 98] A.S. Macedo & E.S. Sousa. Antenna-sector time-division multiple access for broadband indoor wireless systems. *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 6, pages 937–952, Aug 1998.
- [Madan 05] Ritesh Madan, Shuguang Cui, Sanjay Lall & Andrea J. Goldsmith. Mixed Integer-Linear Programming for Link Scheduling in Interference-Limited Networks. In *Proceeding: workshop on Resource Allocation in Wireless Networks*, page 18, April 2005.
- [Madan 06] R. Madan, Shuguang Cui, S. Lall & N.A. Goldsmith. Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks. *Wireless Communications, IEEE Transactions on*, vol. 5, no. 11, pages 3142–3152, 2006.
- [Mandke 07] K. Mandke, Soon-Hyeok Choi, Gibeom Kim, R. Grant, R.C. Daniels, Wonsoo Kim, R.W. Heath & S.M. Nettles. Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed. *Vehicular Technology Conference, 2007. VTC2007-Spring*. IEEE 65th, pages 1896–1900, April 2007.
- [Massoulie 02] L. Massoulie & J. Roberts. Bandwidth sharing: objectives and algorithms. *Networking, IEEE/ACM Transactions on*, vol. 10, no. 3, pages 320–328, Jun 2002.
- [Matsumoto 97] T. Matsumoto, S. Nishioka & D.J. Hodder. Beam-selection performance analysis of a switched multibeam antenna system in mobile communications environments. *Vehicular Technology, IEEE Transactions on*, vol. 46, no. 1, pages 10–20, 1997.

- [Megiddo 87] Nimrod Megiddo. Advances in economic theory, fifth world congress, chapter On the complexity of linear programming, pages 225 – 268. Cambridge University Press, 1987.
- [Miao 06] Lei Miao & Christos G. Cassandras. Optimal Transmission Scheduling for Energy-Efficient Wireless Networks. In Proceedings of IEEE INFOCOM 2006, 2006.
- [Mishra 05] Arunesh Mishra, Suman Banerjee & William Arbaugh. Weighted coloring based channel assignment for WLANs. SIG-MOBILE Mob. Comput. Commun. Rev., vol. 9, no. 3, pages 19–31, 2005.
- [Mishra 06a] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan & W. Arbaugh. A Client-Driven Approach for Channel Management in Wireless LANs. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1–12, April 2006.
- [Mishra 06b] Arunesh Mishra, Vivek Shrivastava, Dheeraj Agrawal, Suman Banerjee & Samrat Ganguly. Distributed channel management in uncoordinated wireless environments. In MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking, pages 170–181, New York, NY, USA, 2006. ACM Press.
- [Mitsuhashi 07] Hikaru Mitsuhashi, Masahiro Watanabe, Sadao Obana, Masaki Bandai & Takashi Watanabe. A Testbed with a Practical Smart Antenna for Directional MAC Protocols in Ad hoc Networks. In AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, pages 731–736, Washington, DC, USA, 2007. IEEE Computer Society.
- [Molina 79] Francisco Walter Molina. A Survey of Resource Directive Decomposition in Mathematical Programming. ACM Comput. Surv., vol. 11, no. 2, pages 95–104, 1979.
- [Moscibroda 06] Thomas Moscibroda, Roger Wattenhofer & Aaron Zollinger. Topology control meets SINR: the scheduling complexity of arbitrary topologies. In MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pages 310–321, New York, NY, USA, 2006. ACM.
- [Naguib 94] A.F. Naguib, A. Paulraj & T. Kailath. Capacity improvement with base-station antenna arrays in cellular CDMA. Vehicular Technology, IEEE Transactions on, vol. 43, no. 3, pages 691–698, Aug 1994.

- [Navda 07] Vishnu Navda, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel & Samir Das. MobiSteer: using steerable beam directional antenna for vehicular network access. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 192–205, New York, NY, USA, 2007. ACM.
- [Nedić 01] Angelina Nedić, Dimitri Bertsekas & V.S. Borkar. Distributed Asynchronous Incremental Subgradient Methods. Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, pages 381 – 407, 2001.
- [Nelson 85] Randolph Nelson & Leonard Kleinrock. Spatial TDMA: A Collision-Free Multihop Channel Access Protocol. *Communications, IEEE Transactions on*, vol. 33, no. 9, pages 934– 944, Sep 1985.
- [Neskovic 00] Aleksandar Neskovic, Natasa Neskovic & George Paunovic. Modern Approaches in Modeling of Mobile Radio Systems Propagation Environment. *IEEE Communications Surveys and Tutorials*, vol. 3, no. 3, 2000.
- [Neufeld 05] Michael Neufeld, Jeff Fifield, Christian Doerr, Anmol Sheth & Dirk Grunwald. SoftMAC - Flexible Wireless Research Platform. In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.
- [Nordstrom 05] E. Nordstrom, P. Gunningberg & H. Lundgren. A testbed and methodology for experimental evaluation of wireless mobile ad hoc networks. *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pages 100–109, Feb. 2005.
- [Oestges 04] Claude Oestges & Arogyaswami J. Paulraj. Propagation into buildings for broad-band wireless access. *IEEE Transactions on Vehicular Technology*, vol. 53, no. 2, pages 521 – 526, March 2004.
- [Oikonomou 04] Konstantinos Oikonomou & Ioannis Stavrakakis. Analysis of a probabilistic topology-unaware TDMA MAC policy for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 7, pages 1286– 1300, Sept 2004.
- [Ott 05] Maximilian Ott, Ivan Seskar, Robert Siracusa & Manpreet Singh. ORBIT Testbed Software Architecture: Supporting Experiments as a Service. In *Proceedings of IEEE Tridentcom 2005, Trento, Italy, Feb 2005*.

- [Otyakmaz 04] A. Otyakmaz, U. Fornefeld, J. Mirkovic, D.C. Schultz & E. Weiss. Performance evaluation for IEEE 802.11G hot spot coverage using sectorised antennas. In *Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 2, pages 1460–1462, September 2004.
- [Peraki 03] Christina Peraki & Sergio D. Servetto. On the maximum stable throughput problem in random networks with directional antennas. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 76–87, New York, NY, USA, 2003. ACM Press.
- [Petrus 98] P. Petrus, R.B. Ertel & J.H. Reed. Capacity enhancement using adaptive arrays in an AMPS system. *Vehicular Technology, IEEE Transactions on*, vol. 47, no. 3, pages 717–727, 1998.
- [Pond 89] L.C. Pond & V. O. K. Li. A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks. In *Military Communications Conference (MILCOM)*, volume 1, pages 70–74, Oct 1989.
- [Radhakrishnan 02] R. Radhakrishnan, Dhananjay Lai, J.Jr. Caffery & D.P. Agrawal. Performance comparison of smart antenna techniques for spatial multiplexing in wireless ad hoc networks. *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, vol. 1, pages 168–171, Oct 2002.
- [Radunović 03] B. Radunović & J. Le Boudec. Joint Scheduling, Power Control and Routing in Symmetric, One-dimensional, Multi-hop Wireless Networks. In *WiOpt*, Nice, France, March 2003.
- [Radunović 04a] B. Radunović & J.-Y. Le Boudec. Optimal power control, scheduling, and routing in UWB networks. *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 7, pages 1252–1270, Sept. 2004.
- [Radunović 04b] B. Radunović & J.Y. Le Boudec. Rate performance objectives of multihop wireless networks. *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pages 334–349, Oct.-Dec. 2004.
- [Ramachandran 06] Krishna Ramachandran, Elizabeth Belding-Royer, Kevin Almeroth & Milind Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Proceedings of IEEE INFOCOM 2006*, 2006.
- [Raman 07] Bhaskaran Raman & Kameswari Chebrolu. Experiences in using WiFi for Rural Internet in India. *IEEE Communications Magazine*, Jan 2007. Special Issue on New Directions In Networking Technologies In Emerging Economies.

- [Ramanathan 97] Ram Ramanathan. A Unified Framework and Algorithm for (T/F/C)DMA Channel Assignment in Wireless Networks. In IEEE INFOCOM, 1997.
- [Ramanathan 99] Ram Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. Wireless Networks, vol. 5, no. 2, pages 81–94, 1999.
- [Ramanathan 00] Ram Ramanathan & Regina Rosales-Hain. Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment. In INFOCOM (2), pages 404–413, 2000.
- [Ramanathan 01] Ram Ramanathan. On the performance of ad hoc networks with beamforming antennas. In Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing (MobiHoc), pages 95–105, Long Beach, CA, USA, 2001. ACM Press.
- [Ramanathan 05] Ram Ramanathan, Jason Redi, Cesar Santivanez, David Wiggins & Stephen Polit. Ad hoc networking with directional antennas: a complete system solution. Selected Areas in Communications, IEEE Journal on, vol. 23, no. 3, pages 496– 506, March 2005.
- [Rangnekar 06] Aniruddha Rangnekar & Krishna M. Sivalingam. QoS aware multi-channel scheduling for IEEE 802.15.3 networks. Mob. Netw. Appl., vol. 11, no. 1, pages 47–62, 2006.
- [Rappaport 01] Theodore Rappaport. Wireless communications: Principles & practice. Prentice Hall, 2nd edition, 2001. ISBN: 0130422320.
- [Rashid-Farrokhi 98a] F. Rashid-Farrokhi, L. Tassiulas & K.J.R. Liu. Joint optimal power control and beamforming in wireless networks using antenna arrays. Communications, IEEE Transactions on, vol. 46, no. 10, pages 1313–1324, Oct 1998.
- [Rashid-Farrokhi 98b] Farrokh Rashid-Farrokhi, K. J. Ray Liu and F. Liu & Leandro Tassiulas. Transmit beamforming and power control for cellular wireless systems. Selected Areas in Communications, IEEE Journal on, vol. 16, no. 8, pages 1437–1450, Oct 1998.
- [Raychaudhuri 05a] D. Raychaudhuri, M. Ott & I. Secker. ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In Proc. First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities Tridentcom 2005, pages 308–309, 2005.
- [Raychaudhuri 05b] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu & M. Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation

- wireless network protocols. In Proc. IEEE Wireless Communications and Networking Conference, volume 3, pages 1664–1669 Vol. 3, 2005.
- [Razavilar 00] J. Razavilar, F. Rashid-Farrokhi & K.J.R. Liu. Traffic improvements in wireless communication networks using antenna arrays. Selected Areas in Communications, IEEE Journal on, vol. 18, no. 3, pages 458–471, 2000.
- [Rhee 06] Injong Rhee, Ajit Warrier, Jeongki Min & Lisong Xu. DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. In MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, pages 190–201, New York, NY, USA, 2006. ACM Press.
- [Rhee 09] Injong Rhee, Ajit Warrier, Jeongki Min & Lisong Xu. DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. IEEE Transactions on Mobile Computing, 2009.
- [Rockafellar 74] R. Tyrrell Rockafellar. Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming. SIAM Journal on Control, vol. 12, no. 2, pages 268–285, 1974.
- [Rockafellar 93] R. Tyrrell Rockafellar. Lagrange Multipliers and Optimality. SIAM Review, vol. 35, no. 2, pages 183 – 238, Jun 1993.
- [Roy 03] Siuli Roy, Dola Saha, S. Bandyopadhyay, Tetsuro Ueda & Shin-suke Tanaka. A Network-Aware MAC and Routing Protocol for Effective Load Balancing in Ad Hoc Wireless Networks with Directional Antenna. In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pages 88 – 97, Annapolis, Maryland, USA, 2003. ACM.
- [Rubinov 03] Alexander Rubinov & Xiaoqi Yang. Lagrange-type functions in constrained non-convex optimization. Kluwer Academic Publishers, 2003.
- [Saha 03] Amit Kumar Saha & David B. Johnson. Routing Improvement using Directional Antennas in Mobile Ad Hoc Networks. Report technique TR03-420, Rice University Dept. of Computer Science, July 2003. See GLOBECOM04.
- [Saha 04] Amit Kumar Saha & David B. Johnson. Routing Improvement using Directional Antennas in Mobile Ad Hoc Networks. In IEEE Global Telecommunications Conference (GLOBECOM), Dallas, TX, Nov. 2004.
- [Sakr 00] Charbel Sakr & Terence D. Todd. Carrier-Sense Protocols for Packet-Switched Smart Antenna Basestations. International Journal of Wireless Information Networks, vol. 7, no. 3, pages 133 – 147, 2000.



- [Salem 05] N.B. Salem & J.-P. Hubaux. A fair scheduling for wireless mesh networks. In Proc. of WiMesh, 2005.
- [Salonidis 04] Theodoros Salonidis & Ros Tassiulas. Distributed on-line schedule adaptation for balanced slot allocation in Bluetooth scatternets and other ad hoc network architectures. In Proc. IEEE International Workshop on Quality of Service (IWQoS), volume 16, 2004.
- [Salonidis 05] Theodoros Salonidis & Leandros Tassiulas. Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks. In MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pages 145–156, New York, NY, USA, 2005. ACM.
- [Sánchez 99] Marvin Sánchez & Jens Zander. Adaptive Antennas in Spatial TDMA Multihop Packet Radio Networks. In RadioVetenskap och Kommunikation (RVK), Karlskrona, Sweden, June 1999.
- [Sánchez 02a] Marvin Sánchez. Multiple access protocols with smart antennas in multihop ad hoc rural-area networks. Licentiate thesis, Department of Signals, Sensors and Systems, Kungl Tekniska Högskolan (KTH), June 2002.
- [Sánchez 02b] Marvin Sánchez, Jens Zander & Tim Giles. Combined Routing & Scheduling for Spatial TDMA in Ad hoc Networks. In Proceedings Swedish Workshop on Wireless Ad-hoc Networks, March 2002.
- [Sánchez 02c] Marvin Sánchez, Jens Zander & Tim Giles. Combined Routing & Scheduling for Spatial TDMA in Multihop Ad hoc Networks. In Proceedings International Symposium on Wireless Personal Multimedia Communications, October 2002.
- [Sánchez 07] Marvin Sánchez, Bo Hagerman & Jens Zander. Radio Resource Allocation in Spatial TDMA Multihop Networks with Advanced Antennas. In Wireless Rural and Emergency Communications WRECOM 2007, October 2007.
- [Sanghani 03] S. Sanghani, T.X. Brown, S. Bhandare & S. Doshi. EWANT: the emulated wireless ad hoc network testbed. Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, vol. 3, pages 1844–1849 vol.3, March 2003.
- [Schwartz 66] J.W. Schwartz, J.M. Aein & J. Kaiser. Modulation techniques for multiple access to a hard-limiting satellite repeater. Proceedings of the IEEE, vol. 54, no. 5, pages 763–777, May 1966.
- [Shad 01] Faisal Shad, Terence D. Todd, Vytas Kezys & John Litva. Dynamic slot allocation (DSA) in indoor SDMA/TDMA using

- smart antenna basestation. IEEE/ACM Trans. Netw., vol. 9, no. 1, pages 69–81, 2001.
- [Sharma 06] Gaurav Sharma, Ravi R. Mazumdar & Ness B. Shroff. On the complexity of scheduling in wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 227–238, New York, NY, USA, 2006. ACM Press.
- [Sharma 07] G. Sharma, N.B. Shroff & R.R. Mazumdar. Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks. In *Proc. INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, pages 2072–2080, 2007.
- [Shor 93] Julie Shor & Thomas G. Robertazzi. Traffic Sensitive Algorithms and Performance Measures for the generation of Self-Organizing Radio Network Schedule. In *Communications, IEEE Transactions on*, volume 41, pages 16–21. IEEE, January 1993.
- [Shor 98] Naum Z. Shor. *Nondifferentiable optimization and polynomial problems. Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Boston / Dordrecht / London, 1998.
- [Shrivastava 07] Vivek Shrivastava, Dheeraj Agrawal, Arunesh Mishra, Suman Banerjee & Tamer Nadeem. Understanding the limitations of transmit power control for indoor wlans. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 351–364, New York, NY, USA, 2007. ACM.
- [Singh 05] Harkirat Singh & Suresh Singh. Smart-aloah for multi-hop wireless networks. *Mob. Netw. Appl.*, vol. 10, no. 5, pages 651–662, 2005.
- [Soldati 04] Pablo Soldati. *Distributed cross-layer optimized flow control for STDMA wireless networks*. Master's thesis, Royal Institute of Technology, November 2004.
- [Soldati 06] Pablo Soldati, Björn Johansson & Mikael Johansson. Proportionally Fair Allocation of End-to-End Bandwidth in STDMA Wireless Networks. In *Proceedings of the the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'06*, May 2006.
- [Soldati 08] Pablo Soldati, Björn Johansson & Mikael Johansson. Distributed cross-layer coordination of congestion control and resource allocation in S-TDMA wireless networks. *Wirel. Netw.*, vol. 14, no. 6, pages 949–965, 2008.

- [Somarriba 03] Oscar Somarriba & Tim Giles. Scheduling for Variable Power and Rate Control for Spatial TDMA in Wireless Ad Hoc Networks. In Proceedings of the Wireless Networking and Communications Group (WNCG), Wireless Networking Symposium, Oct 2003.
- [Somarriba 07] Oscar Somarriba & Jens Zander. Evaluation of Heuristic Strategies for Scheduling, and Power Allocation in STDMA Wireless Networks. In International Symposium on Signals, Systems, and Electronics, pages 427–430, August 2007.
- [Sommariba 04] Oscar Sommariba. Heuristic Algorithms for Combined Scheduling and Routing in Spatial TDMA Wireless Ad hoc Networks. In Proceedings Swedish Workshop on Wireless Ad-hoc Networks, May 2004.
- [Subramanian 08] Anand Prabhu Subramanian, Pralhad Deshpande, Jie Gao & Samir R. Das. Drive-by Localization of Roadside WiFi Networks. In 27th Annual IEEE Conference on Computer Communications (INFOCOM 2008), Phoenix, Arizona, April 2008.
- [Sundaresan 04] Karthikeyan Sundaresan & Raghupathy Sivakumar. A unified MAC layer framework for ad-hoc networks with smart antennas. In MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, pages 244–255, New York, NY, USA, 2004. ACM.
- [Sundaresan 06] Karthikeyan Sundaresan, Weizhao Wang & Stephan Eidenbenz. Algorithmic aspects of communication in ad-hoc networks with smart antennas. In MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pages 298–309, New York, NY, USA, 2006. ACM.
- [Sundaresan 07] Karthikeyan Sundaresan & Raghupathy Sivakumar. A unified MAC layer framework for ad-hoc networks with smart antennas. IEEE/ACM Trans. Netw., vol. 15, no. 3, pages 546–559, 2007.
- [Sussman 80] Steven M. Sussman. A Survivable Network of Ground Relays for Tactical Data Communications. Communications, IEEE Transactions on, vol. 28, no. 9, pages 1616 – 1624, September 1980.
- [Swales 90] S.C. Swales, M.A. Beach, D.J. Edwards & J.P. McGeehan. The performance enhancement of multibeam adaptive base-station antennas for cellular land mobile radio systems. Vehicular Technology, IEEE Transactions on, vol. 39, no. 1, pages 56–67, Feb 1990.

- [Takagi 84] Hideaki Takagi & Leonard Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, vol. 32, no. 3, pages 246–257, March 1984.
- [Takai 01] Mineo Takai, Jany Martin & Rajive Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 87–94, New York, NY, USA, 2001. ACM.
- [Takai 02] Mineo Takai, Jay Martin, Rajive Bagrodia & Aifeng Ren. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 183–193, New York, NY, USA, 2002. ACM.
- [Tan 05] Liansheng Tan, Xiaomei Zhang, L.L.H. Andrew & M. Zukerman. Price-based Max-Min Fair Rate Allocation in Wireless Multi-hop Networks. *TENCON 2005 2005 IEEE Region 10*, pages 1–6, Nov. 2005.
- [Tan 07] Chee Wei Tan, D.P. Palomar & Mung Chiang. Exploiting Hidden Convexity For Flexible And Robust Resource Allocation In Cellular Networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, pages 964–972, May 2007.
- [Tan 09] Chee Wei Tan, Daniel P. Palomar & Mung Chiang. Energy-Robustness Tradeoff in Cellular Network Power Control. *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pages 912 – 025, June 2009.
- [Tangmunarunkit 02] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker & Walter Willinger. Network topology generators: degree-based vs. structural. In *SIGCOMM '02*, pages 147–159, New York, NY, USA, 2002. ACM.
- [Tassiulas 92] L. Tassiulas & A. Ephremides. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, vol. 37, no. 12, Dec 1992.
- [Tingley 01] R.D. Tingley & K. Pahlavan. Space-time measurement of indoor radio propagation. *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 1, pages 22 – 31, Feb 2001.
- [Todd 02] Michael J. Todd. The many facets of linear programming. *Mathematical Programming*, vol. 91, no. 3, pages 417 – 436, Feb 2002.

- [Toumpis 00] Stavros Toumpis & Andrea Goldsmith. Some Capacity Results for Ad Hoc Networks. In Proceedings: Allerton Conference on Communications, Control, and Computing, Oct 2000.
- [Toumpis 03] Stavros Toumpis & Andrea J. Goldsmith. Capacity regions for wireless ad hoc networks. IEEE Trans. on Wireless Communications, vol. 2, no. 4, pages 736–748,, July 2003.
- [Tsitsiklis 86] John N. Tsitsiklis, Dimitri P. Bertsekas & Michael Athans. Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms. In IEEE Transactions on Automatic Control, volume AC-31, pages 803 – 812, September 1986.
- [Veronesi 04] R. Veronesi, J. Zander, V. Tralli, M. Zorzi & F. Berggren. Distributed dynamic resource allocation for multicell SDMA packet access networks. In Communications, 2004 IEEE International Conference on, volume 1, pages 202–207, June 2004.
- [Veronesi 06] R. Veronesi, V. Tralli, J. Zander & M. Zorzi. Distributed dynamic resource allocation for multicell SDMA packet access net. Wireless Communications, IEEE Transactions on, vol. 5, no. 10, pages 2772–2783, Oct. 2006.
- [Villegas 05] E. Garcia Villegas, R. Vidal Ferré & J. Paradells Aspas. New Algorithm for Distributed Frequency Assignments in IEEE 802.11 Wireless Networks. In 11th European Wireless Conference, volume 1, pages 211–217, April 2005.
- [Vilzmann 05] Robert Vilzmann, Christian Bettstetter & Christian Hartmann. On the Impact of Beamforming on Interference in Wireless Mesh Networks. In WIMESH05, 2005.
- [Wächter 06] A. Wächter & L. T. Biegler. On the Implementation of a Primal-Dual Interior Point Line Search Algorithm for Large-Scale Nonlinear Programming. Mathematical Programming, vol. 106, no. 1, pages 25 – 57, 2006.
- [Walker 08] Brenton D. Walker, Ian D. Vo, Matthew Beecher & Matthew Seligman. A Demonstration of the MeshTest Wireless Testbed for Delay-Tolerant Network Research. In Proc. CHANTS, pages 105 – 107. ACM, September 2008. ACM 978-1-60558-186-6/08/09.
- [Wallin 03] Erik Wallin. Access point/qos tradeoff in multihop cellular networks using spatial reuse tdma. Master’s thesis, Kungl Tekniska Högskolan, Radio Communication Systems Laboratory, Department of Signals, Sensors and Systems, September 2003.

- [Wan 01] Peng-Jun Wan, Gruia Calinescu, Xiangyang Li & Ophir Frieder. Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks. In Proc. IEEE Infocom 2001, pages 1162–1171, Anchorage, AK, April 2001.
- [Wang 07] Jianfeng Wang, Yuguang Fang & Dapeng Wu. Enhancing the performance of medium access control for WLANs with multi-beam access point. *Wireless Communications, IEEE Transactions on*, vol. 6, no. 2, pages 556–565, 2007.
- [Wang 08] Xin Wang & J. J. Garcia-Luna-Aceves. Distributed joint channel assignment, routing and scheduling for wireless mesh networks. *Comput. Commun.*, vol. 31, no. 7, pages 1436–1446, 2008.
- [Ward 92] J. Ward & Jr. Compton R.T. Improving the performance of a slotted ALOHA packet radio network with an adaptive array. *Communications, IEEE Transactions on*, vol. 40, no. 2, pages 292–300, Feb 1992.
- [Warrier 08] Ajit Warrier, Sangtae Ha, Prashant Wason, Injong Rhee & J.H. Kim. DiffQ: Differential Backlog Congestion Control for Wireless Multi-hop Networks. In Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Francisco, US, 2008.
- [Warrier 09] Ajit Warrier, Sankararaman Janakiraman, Sangtae Ha & Injong Rhee. DiffQ: Practical Differential Backlog Congestion Control for Wireless Networks. In INFOCOM, Rio de Janeiro, Brazil, 2009.
- [Wattenhofer 03] Roger Wattenhofer & Aaron Zollinger. XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks. Rapport technique 407, Department of Computer Science, ETH Zurich, 2003.
- [White 02] Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb & Abhijeet Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In Proc. of the Fifth Symposium on Operating Systems Design and Implementation, pages 255–270, Boston, MA, December 2002. USENIX Association.
- [Winters 94] J.H. Winters & M.J. Gans. The range increase of adaptive versus phased arrays in mobile radio systems. In M.J. Gans, editor, Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, volume 1, pages 109–115 vol.1, 1994.

- [Winters 99] J.H. Winters & M.J. Gans. The range increase of adaptive versus phased arrays in mobile radio systems. *Vehicular Technology, IEEE Transactions on*, vol. 48, no. 2, pages 353–362, 1999.
- [Wirth 01] Wulf-Dieter Wirth. *Radar techniques using array antennas*. Number 10 in *Radar, sonar, navigation and avionics series*. The Institution of Electrical Engineers, 2001.
- [Wittman 67] J. Wittman. Categorization of Multiple-Access/Random-Access Modulation Techniques. *Communication Technology, IEEE Transactions on*, vol. 15, no. 5, pages 724–725, October 1967.
- [Wolffe 05] G. Wolffe, R. Wahl, P. Wertz, P. Wildbolz & F. Landstorfer. Deterministic Propagation Model for the Planning of Hybrid Urban and Indoor Scenarios. In *Personal, Indoor and Mobile Radio Communications, IEEE 16th International Symposium on (PIMRC)*, volume 1, pages 659 – 663, Sept. 2005.
- [Wu 07] Xinzhou Wu, R. Srikant & James R. Perkins. Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks. *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pages 595–605, June 2007.
- [Xiao 04] Lin Xiao, M. Johansson & S.P. Boyd. Simultaneous routing and resource allocation via dual decomposition. *Communications, IEEE Transactions on*, vol. 52, no. 7, pages 1136–1144, July 2004.
- [Yi 03] Su Yi, Yong Pei & Shivkumar Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 108–116. ACM, ACM, June 2003.
- [Yu 04] Shao Yu. *Integrated routing, scheduling and power control in STDMA wireless ad-hoc networks*. Master's thesis, Kungl Tekniska Högskolan, Institutionen för Signaler, Sensorer & System Reglerteknik, 100 44 Stockholm, Sweden, January 2004.
- [Yuan 05] J. Yuan, Z. Li, W. Yu & B. Li. A cross-layer optimization framework for multicast in multi-hop wireless networks. In *Proc. First International Conference on Wireless Internet*, pages 47–54, 2005.
- [Zander 90] J. Zander. Slotted ALOHA multihop packet radio networks with directional antennas. *Electronics Letters*, vol. 26, no. 25, pages 2098–2100, Dec. 1990.

- [Zander 92] Jens Zander & K.-A. Ahl. Capacity of time-space switch cellular radio link systems for metropolitan area networks. In *Communications, Speech and Vision, IEE Proceedings I*, volume 139, pages 533 – 538, October 1992. Uses scanning steerable antenna on the BS to hit multiple clients in a TDM way.
- [Zegura 96] Ellen W. Zegura, Ken Calvert & S. Bhattacharjee. How to Model an Internetwork. In *Infocom. IEEE*, 1996.
- [Zhang 01] Ruifeng Zhang. Optimal space-time packet scheduling for reservation ALOHA networks. *Signals, Systems and Computers*, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on, vol. 2, pages 1205–1209, 2001.
- [Zhang 08] Yan Zhang, Honglin Hu & Hsiao-Hwa Chen. QoS differentiation for IEEE 802.16 WiMAX mesh networking. *Mob. Netw. Appl.*, vol. 13, no. 1-2, pages 19–37, 2008.
- [Zheng 07] Dong Zheng, Weiyan Ge & Junshan Zhang. Distributed opportunistic scheduling for ad-hoc communications: an optimal stopping approach. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 1–10, New York, NY, USA, 2007. ACM.
- [Zhu 98] C. Zhu & M.S. Corson. An Evolutionary-TDMA Scheduling Protocol (E-TDMA) for Mobile Ad Hoc Networks. Rapport technique CSHCN TR 98-14 (ISR TR 98-32), University of Maryland, Center for Satellite and Hybrid Communication Networks, 1998.
- [Zhu 01a] C. Zhu & M. S. Corson. A new protocol for scheduling TDMA transmissions in mobile ad hoc networks. Rapport technique CSHCN TR 2001-19, Center for Satellite and Hybrid Networks, University of Maryland, 2001.
- [Zhu 01b] Chenxi Zhu & M. S. Corson. A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks. *Wireless Networks*, vol. 7, no. 4, pages 371–384, Aug 2001.
- [Zhu 05] W. Zhu, D. Browne & M. Fitz. An open access wideband multiantenna wireless testbed with remote control capability. *Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2005. Tridentcom 2005. First International Conference on, pages 72–81, Feb. 2005.
- [Zou 06a] Jun Zou & Dongmei Zhao. Bottleneck-first scheduling for real-time traffic in IEEE 802.11 infrastructure-based mesh networks. In *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 593–598, New York, NY, USA, 2006. ACM.



- [Zou 06b] Jun Zou & Dongmei Zhao. Real-time voice traffic scheduling and its optimization in IEEE 802.11 infrastructure-based wireless mesh networks. In QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks, page 48, New York, NY, USA, 2006. ACM.
- [Zukerman 08] Moshe Zukerman, Musa Mammadov, Liansheng Tan, Iradj Ouyeyi & Lachlan L. H. Andrew. To be fair or efficient or a bit of both. *Comput. Oper. Res.*, vol. 35, no. 12, pages 3787–3806, 2008.

## Appendix A

### Modeling Effects of Directional Antennas

#### A.1 Introduction

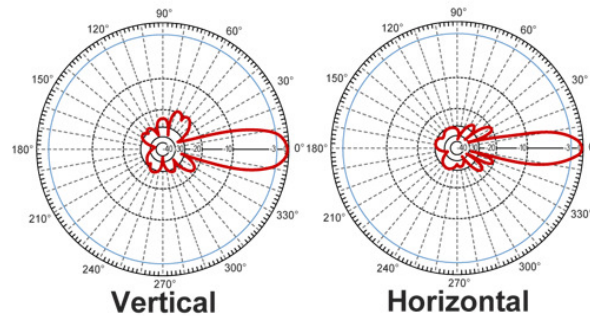


Figure A.1: Sample directional antenna gain pattern displayed on a polar graph

Increasingly, wireless networks are using **directional** antennas to improve the throughput, reach of networks [Ramanathan 01], or to reduce interference between adjacent networks and other noise sources. A more recent development is the use of electronically steerable directional or phase array antennas [Navda 07, Babich 06, Subramanian 08]. These antennas provide better network performance by dynamically controlling the radiation pattern of the antenna. Networks that utilize these antennas can reap substantial improvements in efficiency at all layers of the networking stack.

Figure A.1 shows a common visualization used to understand the antenna gain pattern for a particular highly directional antenna.\* The pattern shows a predominant main lobe along with a number of “side lobes” interspersed with “nulls” or regions of strongly reduced gain. Fixed

---

\* This particular example is the 2.4 GHz 19 dBi Die Cast Directional Reflector Grid Wireless LAN Antenna Model: HG2419G by HyperLink Technologies.

or steerable directional antennas provide better network performance by controlling the radiation pattern of the antenna, increasing the gain or alternatively reducing interference by “steering a null” at a radio on the same channel.

Different network simulators model such antennas with different degrees of fidelity. In this chapter, I argue that the models in the most common network simulators make such simplifying assumptions that it is often difficult to draw strong conclusions from the simulations derived using those models. This is demonstrated using a series of measurements with several different and widely used directional antenna configurations. A more accurate model is presented based on measurements and intuitions about radio propagation.<sup>†</sup> This model captures more about the uncertainty of the environment than the specifics of the antenna and that our results should be generally applicable to many different directional antenna patterns with similar gain characteristics.

The initial measurement study uses sophisticated measurement equipment, including a vector signal analyzer (VSA) and signal generator (VSG). Since the costs of such equipment can be prohibitive, we also develop a method that uses inexpensive equipment (such as standard networking cards) to produce the data needed for the derived models.

The remainder of this chapter is organized as follows: Section A.2, discusses the basics of existing radio propagation models, their limitations, and how our proposal fits in. Section A.3 describes the data collection method and section A.4 describes the set of measurements that were used to derive the model. Section A.5 contains a description of the model and derivation of its parameters. Section A.6 describes how this model can be used in simulations. Finally, section A.7 concludes.

## A.2 Background And Related Work

This section describes the propagation models used by current network simulators, gives an overview of related work, and discusses how the proposed model addresses problems with those

---

<sup>†</sup> All of our measurements are available publicly at [Crawdad 08] and an implementation of our model for the Qualnet 4.5.1 simulator is available at <http://systems.cs.colorado.edu/wiki/EDAM>.

approaches.

### A.2.1 Path Loss Models

Wireless network simulators use a **path loss model** to model the degradation of a transmitted signal as a function of distance; when a signal is too degraded, it cannot be received reliably. Assuming a simplified (i.e., naïve) model, energy is propagated in all directions and the energy that actually strikes the receiver would seem to be proportional to the square of the distance between the transmitter and receiver—the signal is attenuated  $\propto r^2$ . This simple path loss model ignores the significant reflection, scattering, refraction, and absorption effects as radio-frequency (RF) energy interacts with the earth, the atmosphere and other smaller features. One of the major effects is **multipath interference**, where the RF waves bounce off objects in the environment and converge at the receiver after having traversed different distances.

The two-ray model uses a reflection from the earth and the heights of the transmitter and receiver to indicate the likely signal strength at a given distance. This model is specific to the radio frequency used; Figure A.2 is an example of a two-ray calculation from a survey tutorial on antenna propagation models for a 900MHz signal for an 8.7m high transmitter and a 1.6m high receiver; the horizontal axis is a logarithmic scale [Neskovic 00].

This diagram shows that the signal strength decreases roughly at  $r^k$ ,  $2 \leq k \leq 4$ , but that there is considerable variation over short distances. Other models for such effects are based on fitting empirical measurements rather than *a-priori* analysis. There are general purpose models such as the Hata / COST231 model and the Longley-Rice model [Abhayawardhana 05, Oestges 04], and several specific to the wavelength and operating characteristics of wireless LAN cards [Green 02]. Additionally, indoor environments are sufficiently different from outdoor environments that they justify their own approach (see [Andersen 95], [Neskovic 00] and [Iskander 02] for excellent surveys).

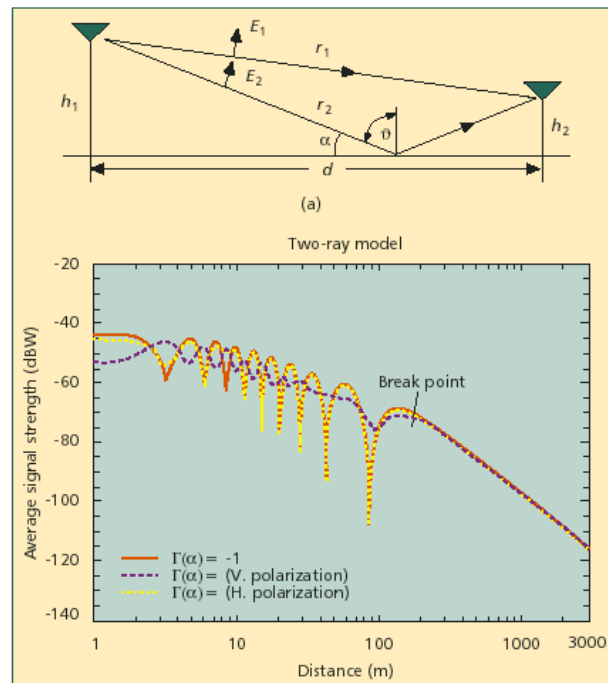


Figure A.2: Example of two-ray model attenuation, from [Neskovic 00].

### A.2.2 Fading Models

The preceding work describes relatively large scale phenomena. In addition to whatever long range attenuation there may be, there is also small-scale fading that is the result of multipath interference and occurs at the scale of single wavelengths. Though such interference can theoretically be predicted analytically, it requires that the environment be known with a level of detail that is generally impractical [Wolfe 05, Tingley 01].

A common way to address such situations is through **statistical** fading models. Rather than determine the signal strength at any exact place or time, it is modeled as a random variable with a known distribution. In general, the distributions are fairly well established, but the parameters are very environment specific (e.g., [de Leon 04]). There are several common models, among them **Rayleigh fading**, which assumes that there are many comparable multipath signals, and **Rician fading**, which assumes a less “cluttered” environment in which line of sight (LOS) paths are more important.

Our model for directional antennas adopts a similar approach to the empirical models and the Rayleigh fading model—we use empirical measurements to identify the characteristics of the random or stochastic process. Where we differ is that our model is primarily concerned with effects on directionality.

### A.2.3 Directional Models

The simulators commonly used in networking research do not consider antenna directionality and radio propagation as interacting variables. This chapter considers three widely used simulators, *OpNet*, *QualNet*, and *NS-2*. Each one supports several models of radio propagation, but they all follow the same general model with regard to antenna gain: For any two stations  $i$  and  $j$ , the received signal strength is computed according to the general form of equation A.1:

$$\text{Received Power} = P_{tx} * G_{tx} * |PL(i, j)| * G_{rx} \tag{A.1}$$

The received power  $P_{rx}$  is the product of the transmitted power  $P_{tx}$ , the transmitter's gain  $G_{tx}$ , the magnitude of path loss between the two stations  $|PL(i, j)|$ , and the receiver's gain  $G_{rx}$ .

The transmitter and receiver gains are treated as constants in the case of omnidirectional (effectively isotropic in the azimuth plane) antennas. For directional antennas, however, gain is an antenna-specific function of the direction of interest. We model the orientation of an antenna in terms of its zenith ( $\phi$ ) and azimuth ( $\theta$ ). Then, for a given antenna  $a$ , we can define a characterization function  $f_a(\phi, \theta)$ :

$$\text{Gain in direction } (\phi, \theta) = f_a(\phi, \theta) \quad (\text{A.2})$$

$$\text{Combined gain} = f_a(\phi, \theta) * f_b(\phi', \theta') \quad (\text{A.3})$$

Correspondingly, the receiver gain is modeled by a (potentially different) function of the direction from which the signal is received. Besides being a source of interference for a dominant signal, the energy traveling along secondary paths also carries signal. If one of the weaker signals for a transmitter happens to be aligned with a high gain direction of a receiving antenna, the received power from that path can be greater than that of the primary path. Thus, in environments with significant multipath, the gain cannot be determined based solely on a single direction. This is easier to understand using Figure A.3, which combines a transmitter (on the left) and a receiver (on the right).

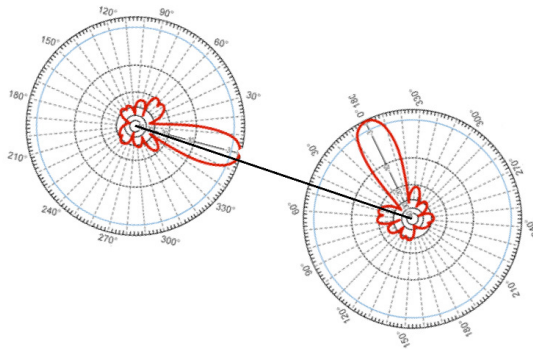


Figure A.3: Illustration of the common path loss model for directional antennas

In this figure, the transmitter gain is indicated by the (large) gain of the antenna pattern; the

receiver gain is indicated by the (much smaller) gain from the “side lobe” that is linearly located between transmitter and receiver. The complex-valued path loss,  $PL(i, j)$ , is related to the length of the dark line separating the transmitter and receiver.

The above models describe the power emitted in, or received from, a single direction. In reality, the transmitter’s power is radiated in all directions, and the receiver aggregates power (be it signal or noise) from all directions. Although the simulators we are considering assume that the single direction of interest for each station is precisely toward the other station, we can generalize equations A.1 and A.3 to the case where there are multiple significant signal paths:

$$P_{rx} = \sum_{l \in \text{paths}} P_{tx} * f_a(\phi_l, \theta_l) * PL_l(i, j) * f_b(\phi'_l, \theta'_l) \quad (\text{A.4})$$

In Equation A.4, note that that  $P_{rx}$  is not necessarily all “signal”. It may be the case that only one signal is decodable and the others destructively interfere. In this case equation A.5 is a better model:

$$P_{rx} = \max_{l \in \text{paths}} P_{tx} * f_a(\phi_l, \theta_l) * PL_l(i, j) * f_b(\phi'_l, \theta'_l) \quad (\text{A.5})$$

Both of these models assume that there is some way to describe available paths that a signal may take. As with the Rayleigh and Rician fading models, it may be possible to build a parameterized model of those paths for “cluttered” and “uncluttered” environments. This is the approach we take, using measured data to determine the model.

With any of the three simulators we consider, the user has the freedom to provide any type of mapping between gain and angle. This means that the user could conceivably make measurements with their desired hardware in their desired environment, much as we have done, and then install this as the pattern. However, even though the antenna can conceivably be modeled arbitrarily well, we will show that the **directionality of the signal** is an effect of the interaction between antenna and environment and that modeling both in isolation, however well, misses significant effects. We propose a combined empirical model that attempts to account for both the pattern of the antenna



and the deviation from this pattern due to environmental effects.

### A.3 Method

In this section we will describe the method we devised for deriving empirical models for antenna patterns using commodity hardware and address any reservations about their accuracy by providing a means for equipment calibration.

#### A.3.1 Data Collection Procedure

Two laptops are used, one configured as a receiver and the other as a transmitter. Each is equipped with an Atheros-based MiniPCI-Express radio that is connected to an external antenna using a U.FI to N pigtail adapter and a length of LMR-400 low loss antenna cable. The receiver laptop is connected to a 7 dBi omnidirectional antenna on a tripod approximately two meters off the ground. The transmitter laptop is connected to the antenna we intend to model on a tripod 30.5 m from the receiver, also two meters off the ground. The transmitter tripod features a geared triaxial head, which allows precise rotation.

The transmitter radio is put in 802.11x ad hoc mode on the least congested channel. The transmitter's ARP table is manually hacked to allow it to send UDP packets to a nonexistent receiver. The receiver is put in monitor mode on the same channel and logs packets with tcpdump. Finally, both the receiver and transmitter must have antenna diversity disabled. With the equipment in place, the procedure is as follows: For each 5 degree position about the azimuth, send 500 unacknowledged UDP packets. Without intervention otherwise, due to MAC-layer retransmits, each will be retried 8 times, resulting in 4000 distinct measurements.

During the experiment, the researchers themselves must be careful to stay well out of the nearfield of the antennas and to move to the same location during runs (so that they, in effect, become a static part of the environment). If additional data is desired for a given location, multiple receivers can be used, provided the data from them is treated separately (as each unique path describes a unique environment).

### A.3.2 Commodity Hardware Should Suffice

To ensure that it is safe to use commodity 802.11x-based hardware to measure antenna patterns, we calibrate the sensitivity of our radios and analyze losses in the packet-based measurement platform.

In the process of collection, some packets will be dropped due to interference or poor signal. In our experience, the percentage of dropped frames *per angle* is very small: the maximum lost frames per angle in our datasets is on the order of 5%, with less than 1% lost being more common (the mean is 0.01675%). Moreover, the correlation coefficient between angle and loss percentage is -0.0451, suggesting that losses are uniformly distributed across angles. Given that we have taken 4000 samples in each direction, noise in our measurements due to packet loss is negligible.

To get an idea of how accurate our commodity radios are in measuring received signal strength (RSS), we directly connected each of four radio cards (all Atheros-based Lenovo-rebranded Mini-PCI Express) to an Agilent E4438C VSG. The VSG was configured to generate 802.11 frames and the laptop to receive them. For each of the four cards we collected many samples while varying the transmit power of the VSG between -20 dBm and -95 dBm (lower than the receive sensitivity threshold of just about any commodity 802.11 radio) on 5 dB increments. We performed a linear least squares fit, finding a slope of 0.9602 and adjusted R-squared value of 0.9894 (indicating a strong fit to the data). The commodity radios perform remarkably well in terms of RSS measurement. To correct for the minor error they do exhibit, we use the slope of this fit to adjust our measurements, dividing each measurement by the slope value.

## A.4 Measurements

In this section we will explain the datasets we collected, discuss our normalization procedure, and give some high level statistical characterization of the data.

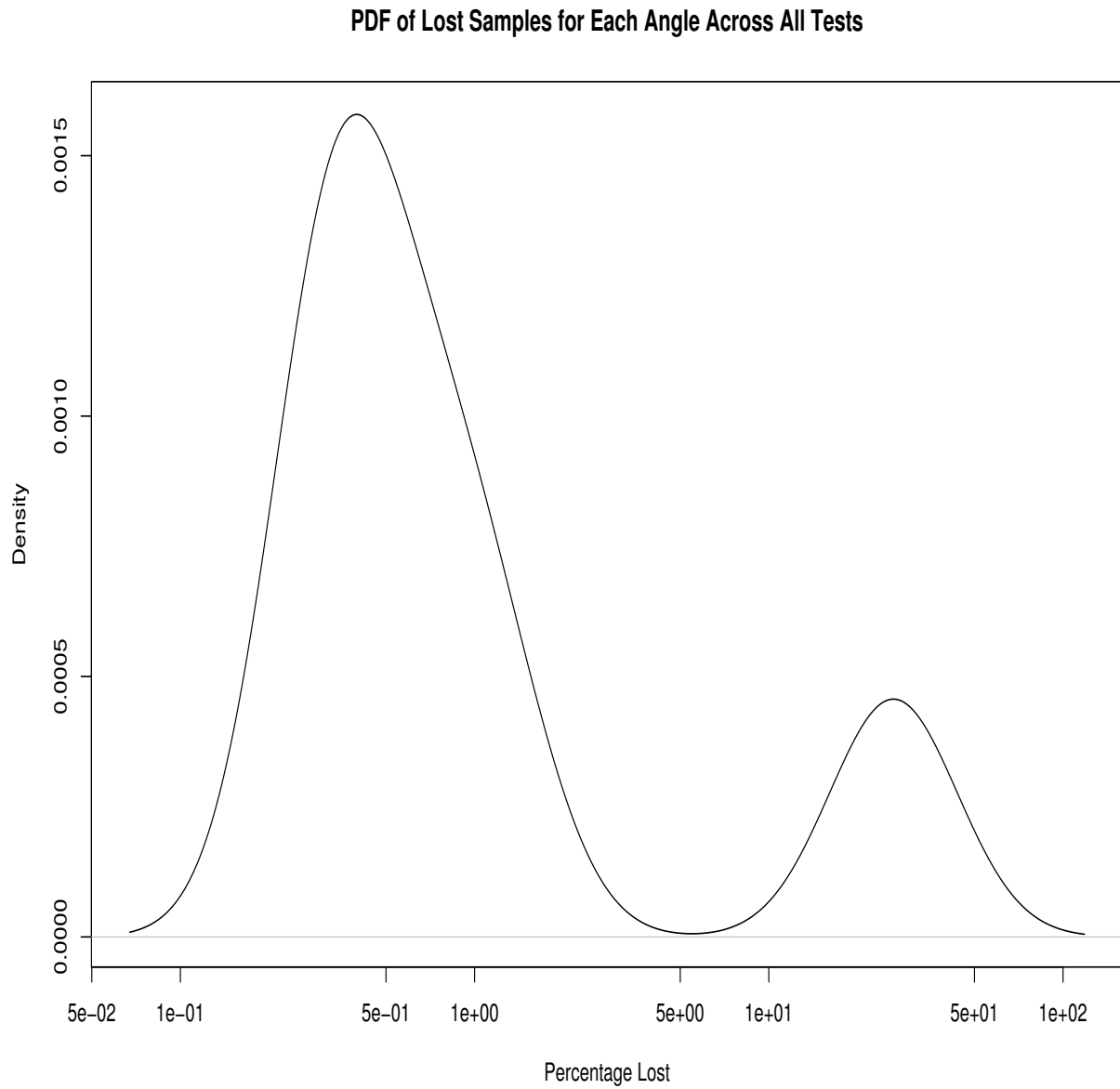


Figure A.4: Probability Density Function of percentage of dropped measurement packets in a given angle for all angles and all data sets.

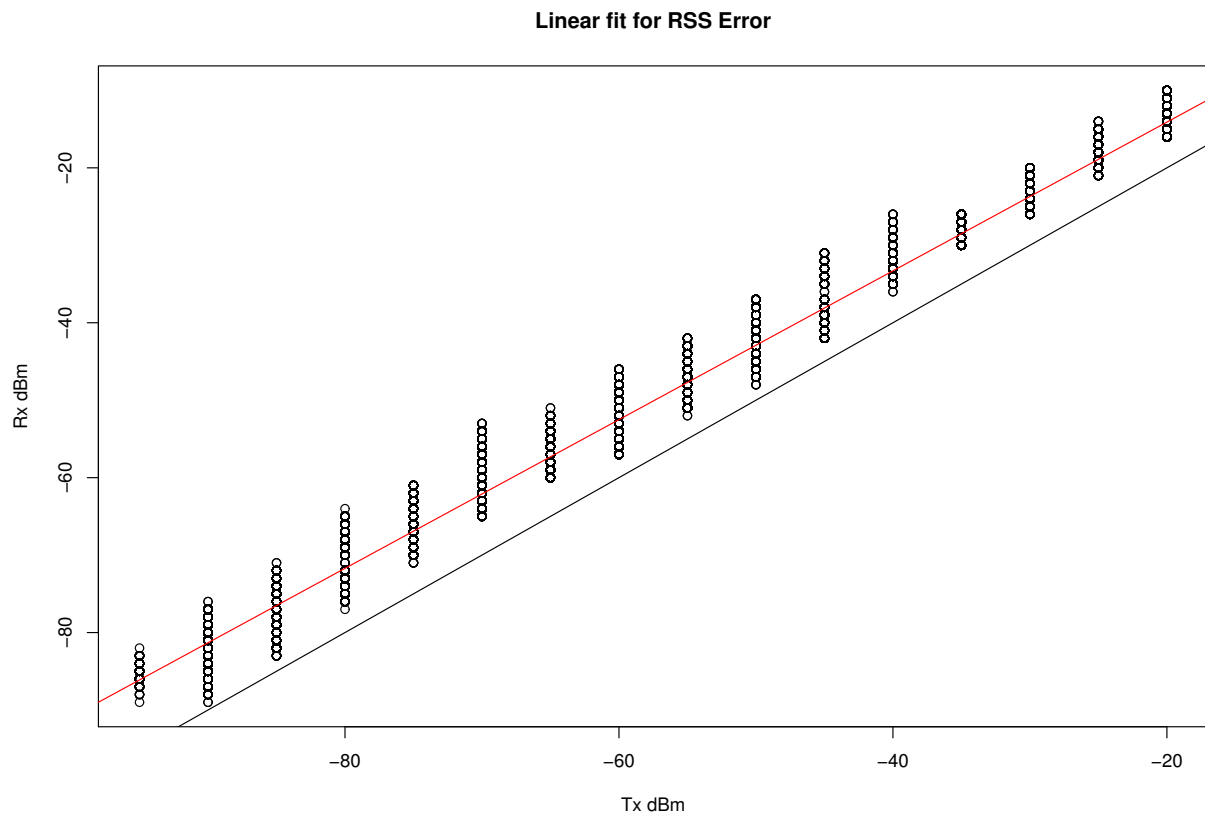


Figure A.5: Linear fit to RSS error observed from commodity cards during calibration. The red (upper) line indicates the regression fit and the black (lower) line is perfect equality.

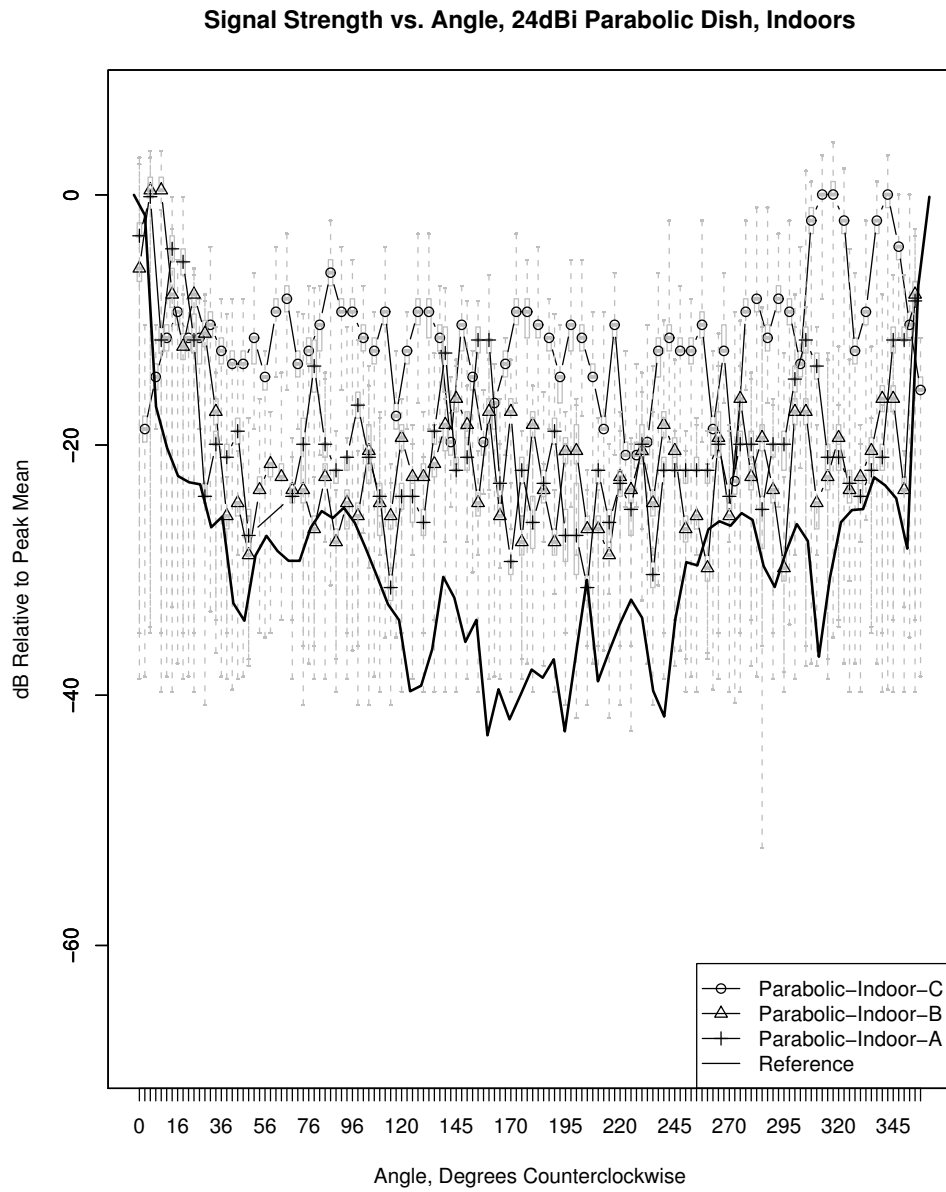


Figure A.6: Comparison of signal strength patterns across different environments and antennas: Parabolic dish indoor environments.

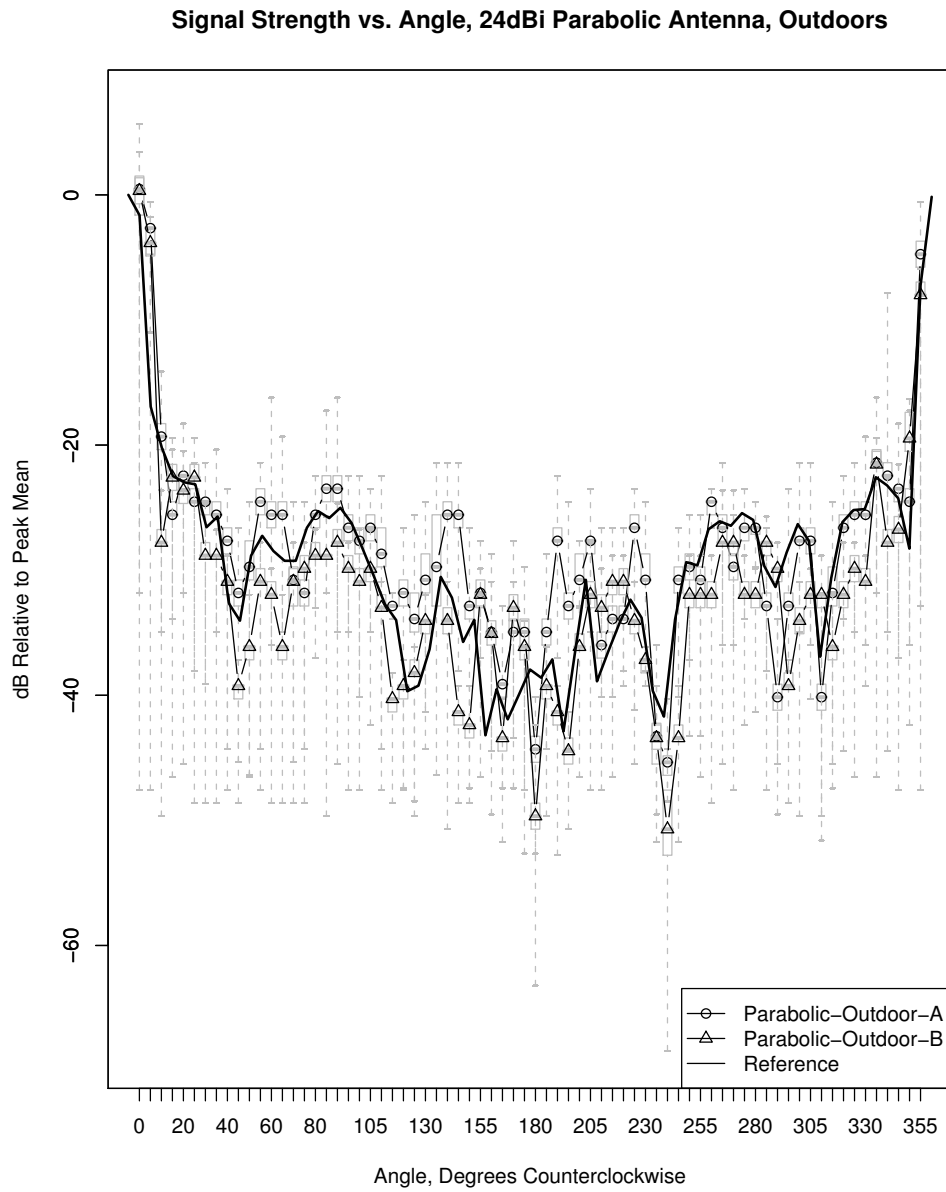


Figure A.7: Comparison of signal strength patterns across different environments and antennas: Parabolic dish outdoor environments.

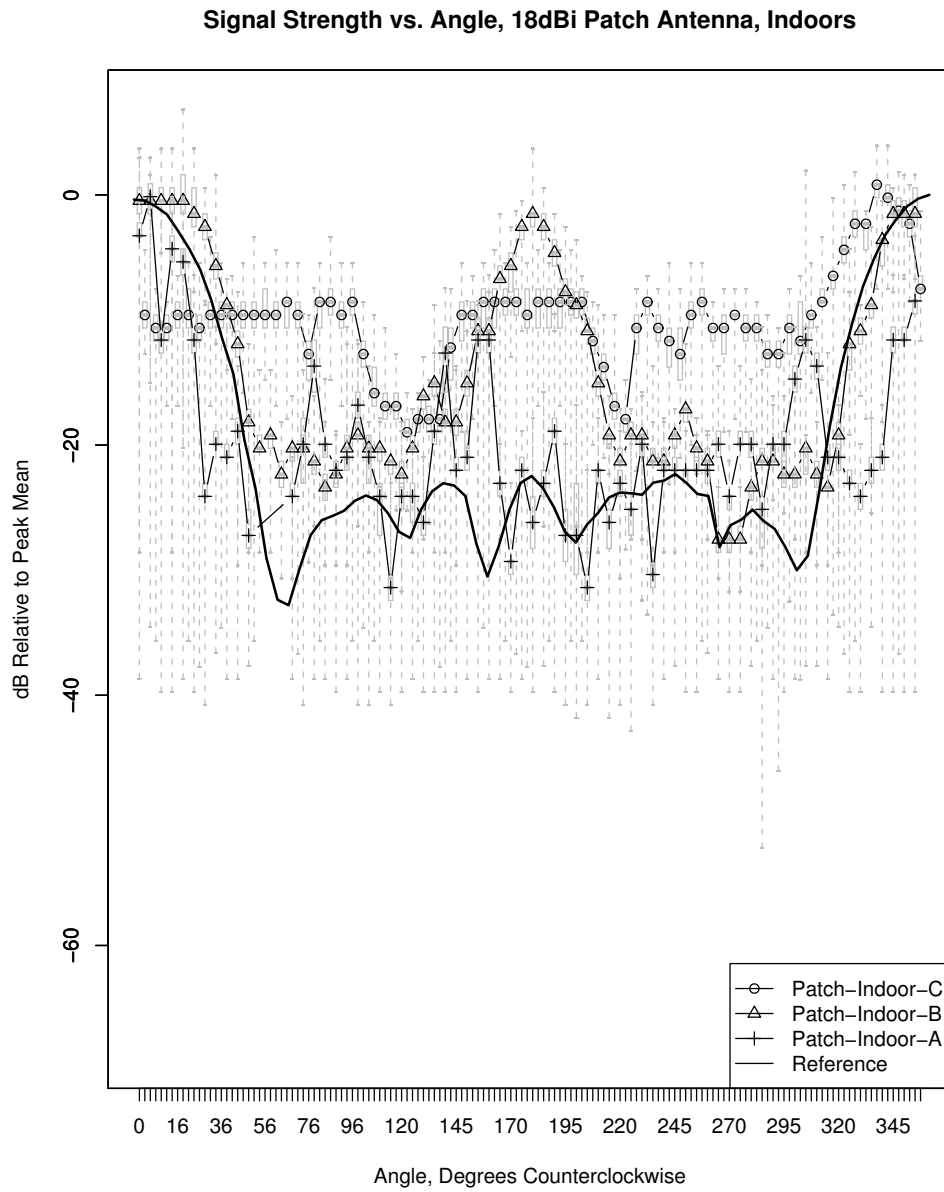


Figure A.8: Comparison of signal strength patterns across different environments and antennas: Patch panel indoor environments.

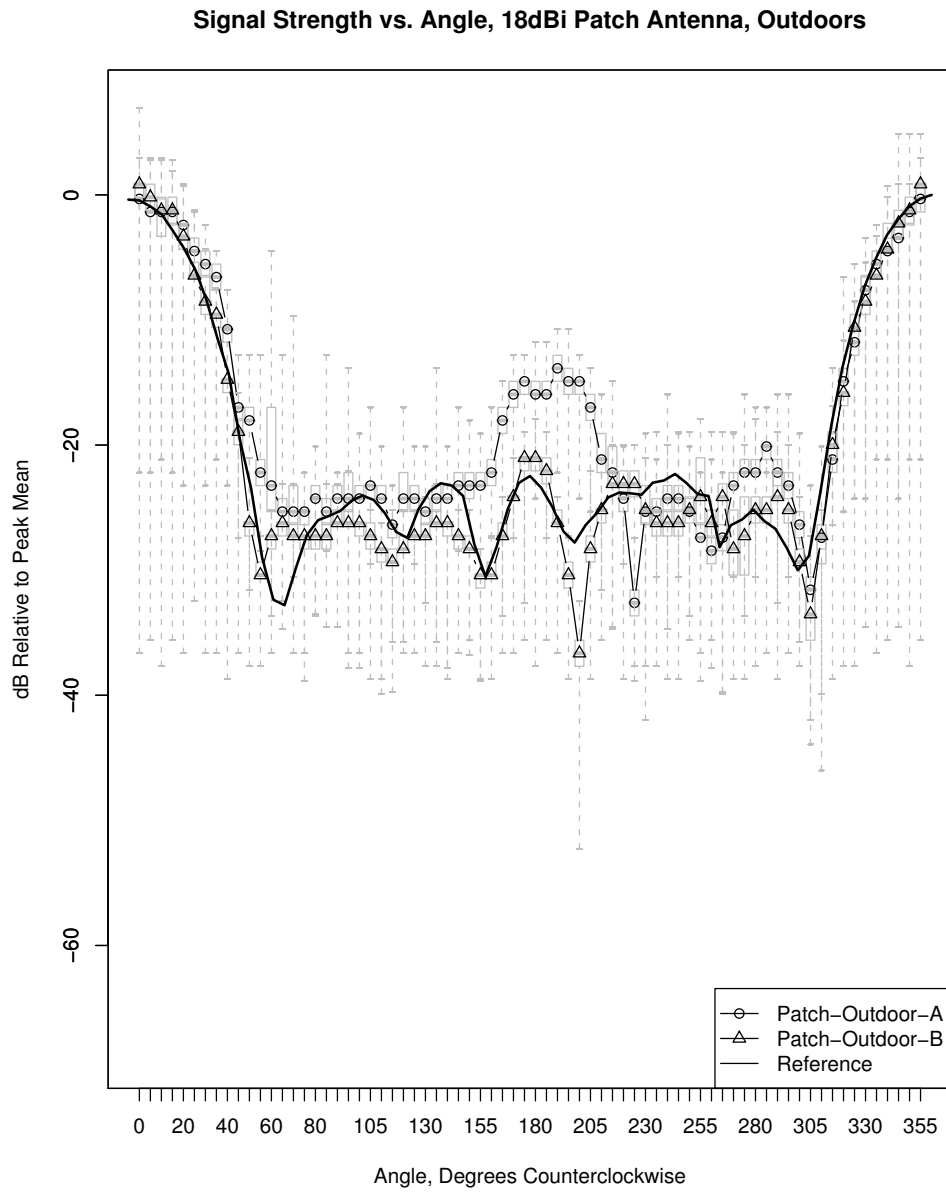


Figure A.9: Comparison of signal strength patterns across different environments and antennas: Patch panel outdoor environments.



### A.4.1 Experiments Performed

In order to derive an empirical model that better fits real world behavior, we collected data in several disparate environments with three different antennas. A summary of these datasets is provided in table A.1. With the exception of the reference patterns, all of the measurements were made with commodity hardware by sending many measurement packets between two antennas and logging received signal strength (RSS) at the receiver. The three antenna configurations used include: (1) a HyperLink 24dBi parabolic dish with an 8 degree horizontal beamwidth, (2) a HyperLink 14dBi patch with a 30 degree horizontal beamwidth, and (3) a Fidelity Comtech Phocus 3000 8-element uniform circular phased array with a main lobe beamwidth of approximately 52 degrees. This phased array functions as a switched beam antenna and can form this beam in one of 16 directions (on 22.5 degree increments around the azimuth). For the HyperLink antennas, we used the same antenna in all experiments of a particular type to avoid intra-antenna variation due to manufacturing differences.

In addition to the **in situ** experiments, we have a “reference” data set for each configuration. The Array-Reference data set was provided to us by the antenna manufacturer. Because HyperLink could not provide us with data on their antennas, Parabolic-Reference and Patch-Reference were derived using an Agilent 89600S VSA and an Agilent E4438C VSG in a remote floodplain<sup>‡</sup>.

Following is a brief description of each of the experiments:

**Parabolic-Outdoor-A, Patch-Outdoor-A:** A large open field on the University of Colorado campus was used for these experiments. The field is roughly 150m on each side and is surrounded by brick buildings on two of the four sides. Although there is line of sight and little obstruction, the surrounding structures make this location most representative of an urban outdoor deployment.

**Parabolic-Outdoor-B, Patch-Outdoor-B:** A large University-owned floodplain on the edge

---

<sup>‡</sup> We were unable to acquire access to an anechoic chamber in time for this study, but would like to make use of one in future work, for even cleaner reference measurements.



Figure A.10: Receiver side of measurement setup in floodplain



Figure A.11: Floor plan of office building used in Array-Indoor-A, Array-Indoor-B, Patch-Indoor-B, Patch-Indoor-C, Parabolic-Indoor-B, and Parabolic-Indoor-C.

of town was used for our most isolated data sets. The floodplain is flat, recessed, and free from obstruction for nearly a quarter mile in all directions. This location is most representative of a rural backhaul link.

**Array-Outdoor-A:** The same open field is used as in the Parabolic-Outdoor-A and Patch-Outdoor-A data sets. The collection method here differs from that described in section A.3. A single phased array antenna is placed approximately 30 meters away from an omnidirectional transmitter. The transmitter sends a volley of packets from its fixed position as the phased array antenna electronically steers its antenna across each of its 16 states, spending 20 ms in each state. Several packets are received in each directional state. The phased array antenna is then manually rotated in 10 degree increments while the omnidirectional transmitter remains fixed. The same procedure is repeated for each of 36 increments. Moving the transmitter changes not only the angle relative to the antenna but also the nodes' positions relative to their environment. To address this confound, each physical position is treated as a separate experiment. This means that the number of angles *relative to the steered antenna pattern* is limited to the number of distinct antenna states (16). The transmission power of the radio attached to the directional antenna was turned down to 10dBm to produce more tractable noise effects (the default EIRP is much too high to model small-scale behavior).

**Parabolic-Indoor-A and Patch-Indoor-A:** For this data set we used the University of Colorado Computer Science Systems Laboratory. The directional transmitter was positioned approximately 6 meters from the receiver in a walkway between cubicles and desks. This is our most cluttered environment.

**Parabolic-Indoor-B, Parabolic-Indoor-C, Patch-Indoor-B, and Patch-Indoor-C:** An indoor office space was used for this set of tests. Two receivers were used here: one with line of sight and one without line of sight, placed amidst desks and offices.

**Array-Indoor-A and Array-Indoor-B:** Seven phased array antennas are deployed in the same 25x30m indoor office space used for Parabolic-Indoor-B, Parabolic-Indoor-C, Patch-Indoor-B and Patch-Indoor-C. Data from two of the seven antennas are analyzed here. Each antenna electronically steers through its 16 directional states, spending 20 ms at each state. Two mobile omnidirectional transmitters move through the space and transmit 500 packets at 44 distinct positions. For each packet received by a phased array, the packet's transmission location and orientation is recorded (i.e., which of the four cardinal directions the transmitter was facing) along with the directional state in which the packet arrived and the RSSI value.

**Parabolic-Reference and Patch-Reference:** The large floodplain is used here. An Agilent VSA is connected to the omnidirectional receiver and makes a 10 second running average of power samples on a specific frequency (2.412 GHz is used). Three consecutive averages of both peak and band power are recorded for each direction. The directional transmitter is rotated in five degree increments and is connected to a VSG outputting a constant sinusoidal tone at 25 dBm on a specific frequency. Before, after, and between experiments, we make noise floor measurements, and as a post-processing step, we subtract the mean of this value (-59.62 dBm or 1.1 nW) from the measurements.

#### A.4.2 Normalization

Our first task in comparing data sets is to come up with a scheme for normalization so that they can be compared to one another directly. For each data set, we find the mean peak value, which is the maximum of the mean of samples for each discrete angle. This value is then subtracted from every value in the data set. The net effect is that the peak of the measurements in each data set will be shifted to zero, which allows us to compare measurements from diverse RF environments directly.

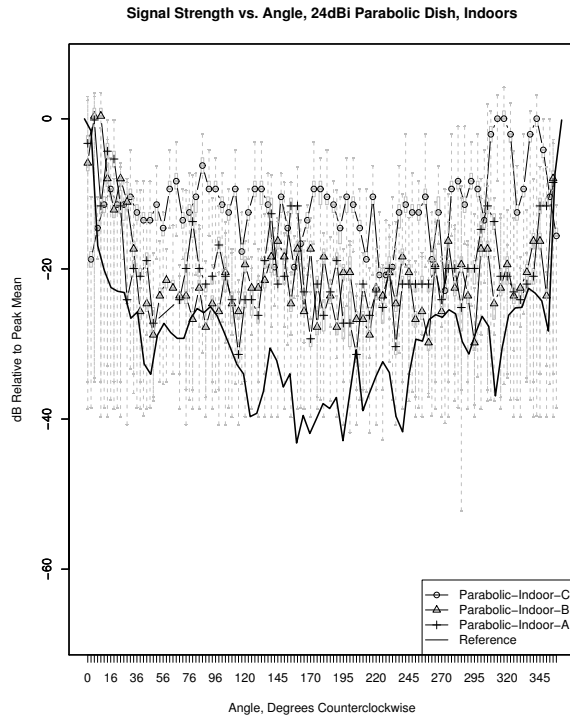
Label	Environment	LOS?	Dist. (m)	Samples	Loss (%)
Parabolic-Outdoor-A	Open Field on Campus	Yes	30.5	214471	24.81
Parabolic-Outdoor-B	Empty Floodplain	Yes	30.5	258876	7.05
Parabolic-Indoor-A	Laboratory	Yes	30.5	267092	2.21
Parabolic-Indoor-B	Office Building	Yes	$\approx 60$	268935	10.41
Parabolic-Indoor-C	Office Building	No	$\approx 15$	283104	5.12
Parabolic-Reference	Empty Floodplain	Yes	30.5	219	N/A
Patch-Outdoor-A	Open Field on Campus	Yes	30.5	455952	12.44
Patch-Outdoor-B	Empty Floodplain	Yes	30.5	278239	4.99
Patch-Indoor-A	Laboratory	Yes	30.5	290030	2.21
Patch-Indoor-B	Office Building	Yes	$\approx 60$	265593	7.40
Patch-Indoor-C	Office Building	No	$\approx 15$	278205	2.65
Patch-Reference	Empty Floodplain	Yes	30.5	219	N/A
Array-Outdoor-A	Open Field on Campus	Yes	$\approx 30$	475178	N/A
Array-Indoor-A	Office Building	Mixed	Varies	2672050	N/A
Array-Indoor-B	Office Building	Mixed	Varies	2708160	N/A
Array-Reference	Open Urban Area	Yes	$\approx 5$	360	N/A

Table A.1: Summary of data sets.

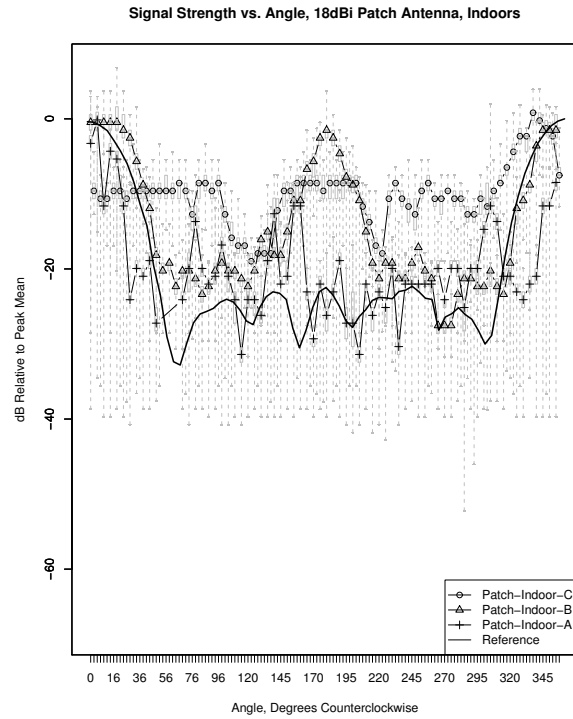
#### A.4.3 Error relative to the reference

Figure A.12 on the next page shows the normalized measured **in situ** patterns and their corresponding (also normalized) reference patterns. Recall that the reference pattern is generated and recorded by calibrated signal processing equipment and the measured data is collected using commodity 802.11 cards. There is great variation in the measured patterns and consequently in how significantly they differ from the reference (which we would typically classify as error). As we would expect, the measurements in outdoor environments exhibit less noise due to less clutter, but still deviate from the reference on occasion. As further confirmation that our measurement process works well, notice that Parabolic-Outdoor-B and Patch-Outdoor-B (figures A.7 and A.9) are highly correlated with the reference pattern. (Recall that these experiments were done in the same floodplain as the reference, indicating that the commodity hardware can compete with the expensive specialized equipment in a similar environment).

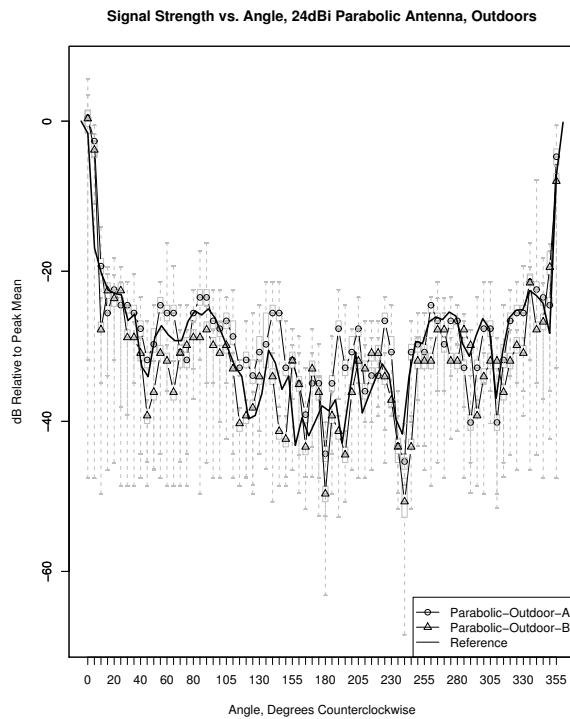
On inspection of this data, our first question is whether there is a straight-forward explanation for error in the measured patterns. Figure A.13 provides a CDF of all error for each antenna. The three antennas provide similar error distributions, although offset in the mean. The array data is the most offset from the others (presumably because its reference pattern is theoretical rather



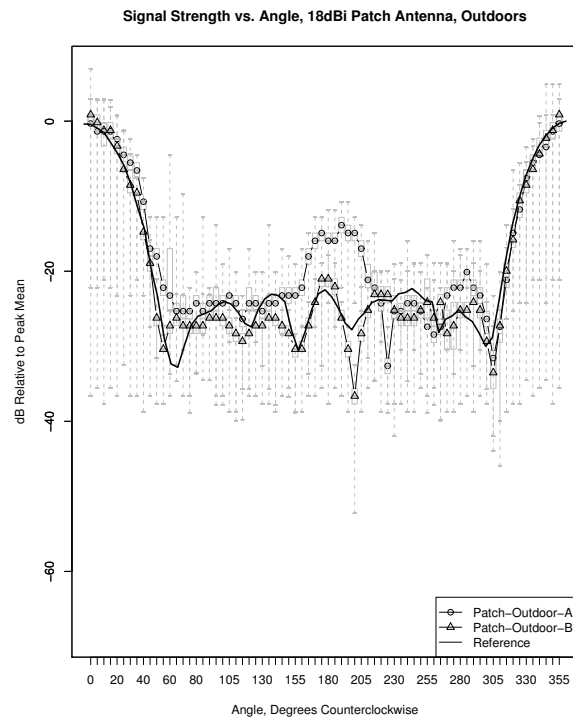
(a) Parabolic dish indoor environments (A.6)



(c) Patch panel indoor environments (A.8)



(b) Parabolic dish outdoor environments (A.7)



(d) Patch panel outdoor environments (A.9)

Figure A.12: Comparison of signal strength patterns across different environments and antennas. Repeated from Figures A.6 to A.9 on pages 173–176 for ease of comparison.

than measured) and exhibits some bimodal behavior. The patch measurements are closest to the reference, showing a large kurtosis about zero.

Clearly, the antennas have different error characteristics. However, for each antenna, and for each data set, it might be that the error in a given direction is correlated with that in other directions—if this were true, we could use a single or small set of probability distributions to describe the error process in a given environment with a given antenna.

We used a Shapiro-Wilkes test on the per angle error for each data set. The resulting p-values are well under the  $\alpha = 0.05$  threshold, and in all cases we can reject the null hypothesis that the error is normally distributed; this means that standard statistical tests (and regression models) that assume normality cannot be used. A pairwise Mann-Whitney U-test can be used to determine which pairs of samples grouped on some criterion (in our case angle) are drawn from the same distribution. For each dataset, we generate a “heat map” where each cell corresponds to a pair of angles. The cell is colored by the p-value produced by the U-test when run pairwise, comparing the error for the reference pattern and the **in situ** pattern for those angles. Remarkably, all of our traces produce similar heat maps: In the majority of pairs we reject the null hypothesis that their error process is drawn from the same distribution. However, for angles near zero, we are unable to reject this hypothesis. This observation – that **measurements where the main lobe of the directional antenna is pointed at the receiver may exhibit correlated error processes** – motivated another series of tests.

To further explore “possibly well behaved” error processes about the main lobe, we applied a Kruskal-Wallis rank-sum test to two scenarios: (1) For angles near zero, are batches with the same antenna (but different environments) equivalent? (2) For angles near zero, are batches with the same environment (but different antennas) equivalent?

For (1), the null hypothesis is soundly rejected for all combinations (p-value  $\ll 0.05$ ) For (2), the results still point strongly toward rejection (mean p-value = 0.0082), however there is one outlier: In the case of 355 degrees in the laboratory environment, we achieve a p-value of 0.2097. One outlier, however, is not sufficient to overcome the evidence that neither antenna configuration



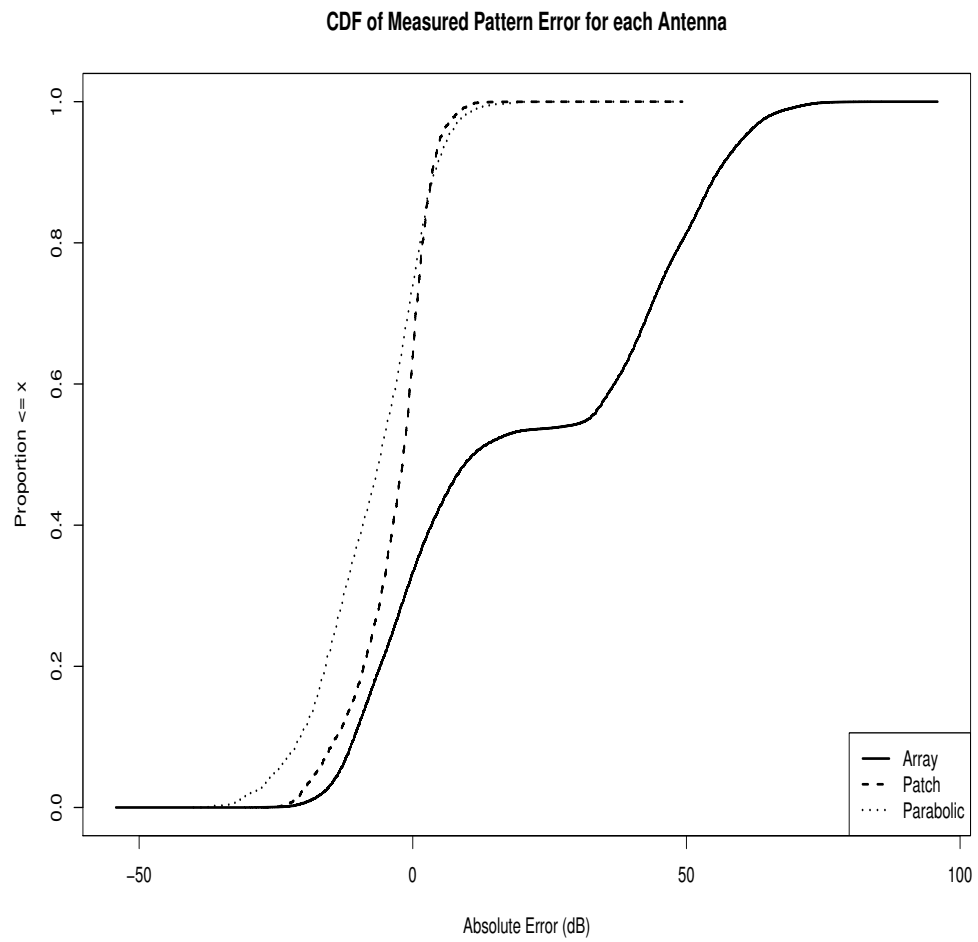


Figure A.13: Cumulative Density Functions for the error process (combined across multiple traces) for each antenna type.

nor environment alone is sufficient to account for intra-angle variation in error—even in the more seemingly well-behaved cone of the antenna main lobe.

#### A.4.4 Observations

There are several qualitative points that are worth bringing out of this data: (1) In the indoor environments, none of the measurements track the reference signal closely; (2) In all environments, there is significant variation between data sets; and (3) The maximum signal strength is generally realized in **approximately** the direction of maximum antenna gain, but directions of low antenna gain often do not have correspondingly low signal strength. This means that **no system for interference mitigation can safely rely on predetermined antenna patterns.**

### A.5 A New Model of Directionality

We began this chapter with the observation that path loss and antenna gain are typically regarded as orthogonal components of the power loss between transmission and reception (equations A.1 – A.3). In this section, we evaluate the **best case** accuracy of this approach, and suggest a new model based on the limitations identified.

#### A.5.1 Limitations of Orthogonal Models

If transmit power and path loss do not vary with antenna angle, the received power relative to antenna angle can be modeled as:

$$\widehat{P}_{rx} = \beta_0 * f(\phi, \theta) \tag{A.6}$$

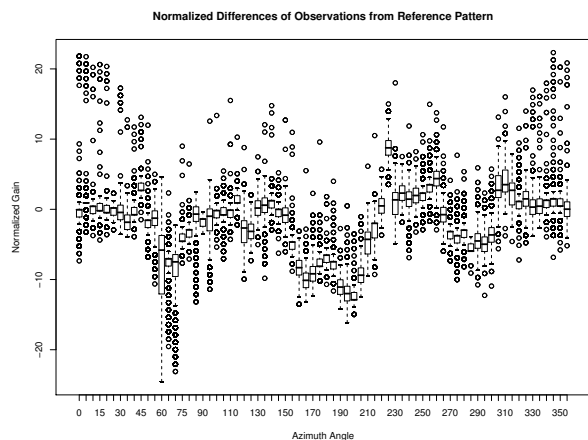
$\beta_0$  is a constant combining the path loss—however calculated—and the gain of the non-rotating antenna.  $f(\phi, \theta)$  is a function describing the gain of the other antenna relative to the signal azimuth  $\theta$  and zenith  $\phi$ . Without loss of generality, we will assume that the antenna being varied is the receiver, and that the zenith,  $\phi$ , is fixed.

To evaluate the accuracy of this model, we start by finding the estimate  $b_0$  for  $\beta_0$  that minimizes the sum of squared error (SSE). In effect, this is assuming the *best possible path loss estimate*, without specifying how it is determined. If the function  $f(\phi, \theta)$  correctly describes the antenna, and if path loss and antenna gain are in fact orthogonal components of the received signal strength, then the remaining error should be **randomly** distributed about 0.

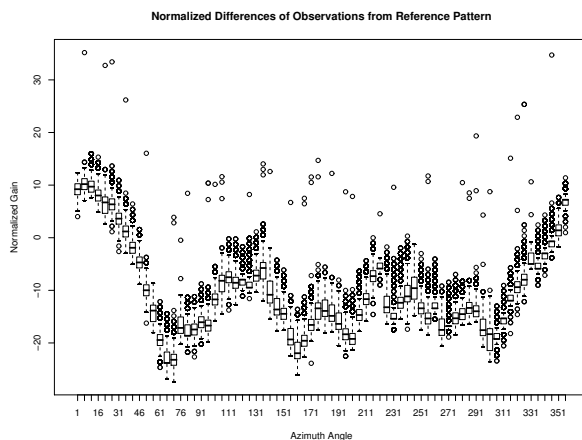
Figures A.14a on the following page through A.14f on the next page depict the error of this *orthogonal model* for several data sets. There are several qualitative observations to be made: First and most importantly, *the error value is not uniformly random, but rather correlated with direction*. The variability within any given direction is less than that for the data set as a whole. Second, the error is significant. In the worst states, the *mean error* is between 8 and 10 dB, in either direction. Third, the model overestimates signal strength in the directions where the gain is highest and underestimates in the directions where the gain is lowest. That is, *the difference in actual signal strength between peaks and nulls is less than the antenna in isolation would produce*. This has significant implications for systems that use null steering to manage interference.

The data in figures A.14e on the following page and A.14f on the next page is aggregated from 36 distinct physical configurations. In each configuration, the directional receiver was (electronically) rotated in 22.5 degree increments, and between configurations, the omnidirectional transmitter was physically moved around the receiver by ten degrees. A consequence of this method is that these 10 degree changes represent not only a change of the angle between the transmitter and the antenna, but also a change of location with the attendant possibility of fading effects. To account for this, we consider each of the 36 configurations individually. This gives less angular resolution, but also fewer confounds. Figure A.15 on page 189 displays each configuration as a separate line. The model accuracy is fairly consistent: The residual standard error of the aggregate is 8 dB, and the individual cases range from 5.74 dB to 11.4 dB with a mean of 7.6 dB.

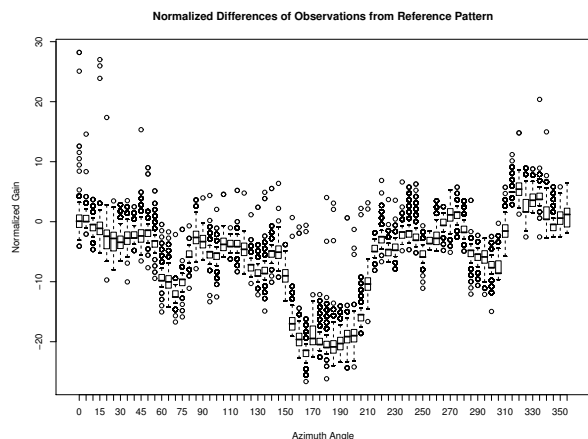
The path loss value used for each data set was the lowest error fit for that specific data and the antenna patterns ( $f(\theta)$ ) for the patch and parabolic antennas were measured using the specific individual antenna in question. Note also that error patterns differ from environment to



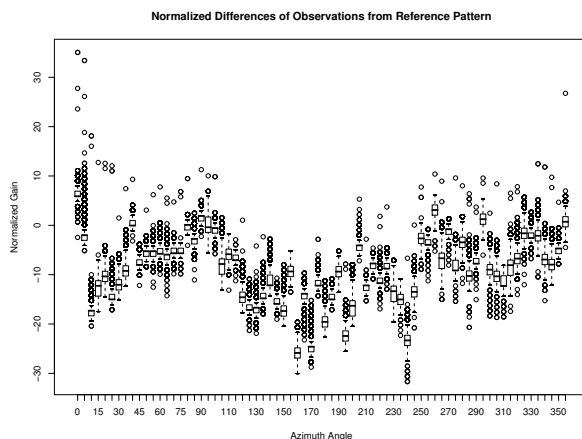
(a) Patch-Outdoor-A



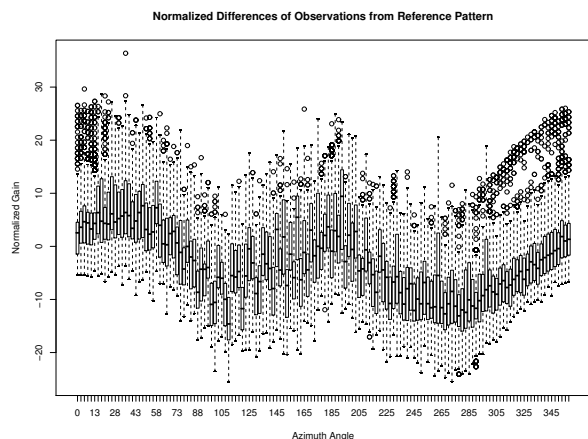
(b) Patch-Indoor-B



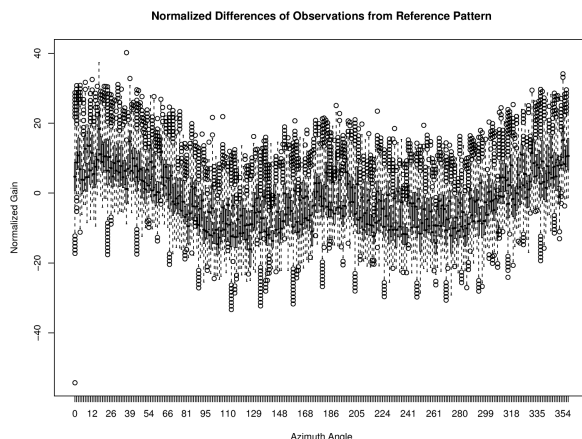
(c) Patch-Indoor-C



(d) Parabolic-Indoor-C



(e) Array-Outdoor-A



(f) Array-Indoor-A

Figure A.14: Differences between the orthogonal model and observed data in dB:  $\hat{P}_{rx} - P_{rx}$ .

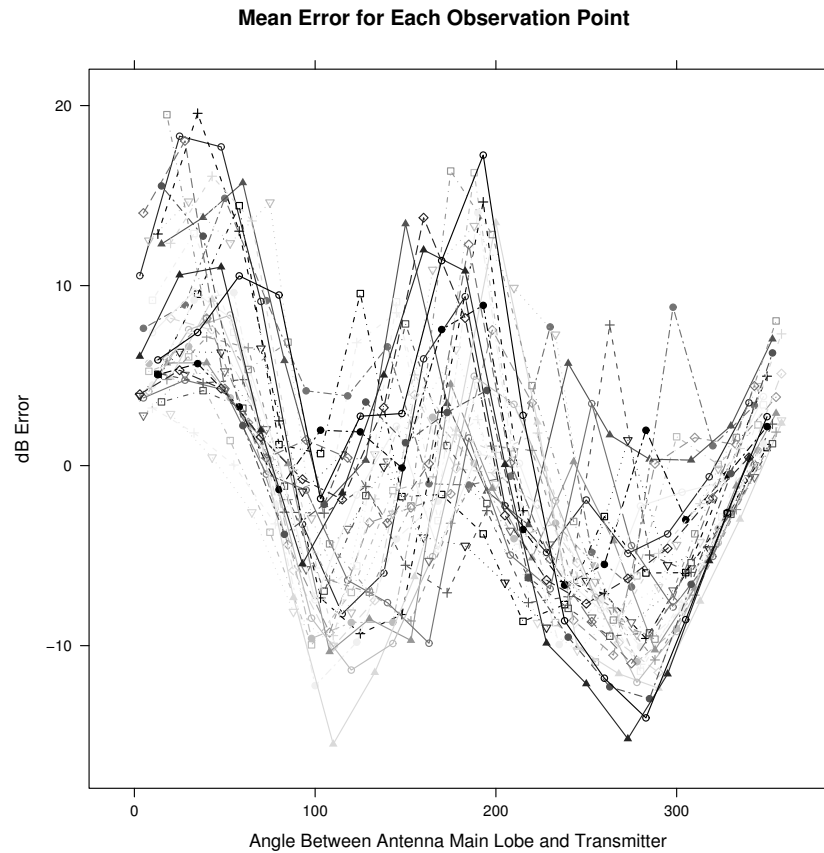


Figure A.15: Mean error of orthogonal model for each observation point in the Array-Outdoor-A data set. The format is the same as in figures A.14a on the previous page through A.14f on the preceding page.

environment: One could derive an **ex post facto**  $f(\theta)$  to eliminate the error in a single data set, but it would not be applicable to any other.

The magnitude and **systematic nature** of the error suggest that the orthogonal model has inherent limitations that cannot be alleviated by improving either the antenna model or path loss model separately.

### A.5.2 An Integrated Model

To address these limitations, we propose an integrated model that addresses the systematic errors discussed above, while remaining simple enough to use in analysis and simulations.

We address the **environment-specific, direction-specific** error shown in figures A.14a through A.14f with the following environment aware model, given in equation A.7. The expected received power is given by a constant  $\beta_0$ , the antenna gain function  $f(\phi, \theta)$ , and a yet to be determined environmental offset function  $x(\phi, \theta)$ :

$$\widehat{P}_{rx} = \beta_0 * f(\phi, \theta) * x(\phi, \theta) \quad (\text{A.7})$$

As with the orthogonal model, we assume a constant zenith and consider  $f(\phi, \theta)$  and  $x(\phi, \theta)$  with regard to the azimuth  $\theta$ . Equation A.7 can be converted to a form that lends itself to least squares (linear regression) analysis in the following way: First, we rewrite equation A.7 as addition in a logarithmic domain, and second we substitute a discrete version for the general  $x(\theta)$ . In the discrete  $x(\theta)$ , the range of angles is partitioned into  $n$  bins such that bin  $i$  spans the range  $[B_i, T_i)$ . Each bin has associated with it a boxcar function  $d_i(\theta)$  to be 1 if and only if the angle  $\theta$  falls within bin  $i$  (equation A.8) and an unknown constant **offset value**  $\beta_i$ . These transformations yield the model given in equation A.10.

$$d_i(\theta) = \begin{cases} 1, & B_i \leq \theta < T_i \\ 0, & \text{otherwise} \end{cases} \quad (\text{A.8})$$

$$x(\theta) = \sum_{i=1}^n d_i(\theta) \beta_i \quad (\text{A.9})$$

$$f(\theta) - \widehat{P}_{rx} = \beta_0 + \beta_1 d_1(\theta) + \beta_2 d_2(\theta) + \cdots + \beta_n d_n(\theta) \quad (\text{A.10})$$

If  $x(\theta)$  is discretized into  $n$  bins, the model has  $n + 1$  degrees of freedom: One for each bin and one for  $\beta_0$ , the signal strength without antenna gain. For any given signal direction, exactly one of the  $d_i(\theta)$  functions will be 1, so each prediction is an interaction of two coefficients:  $\beta_0$  and  $\beta_i$ . Consequently,  $\beta_0$  could be eliminated and an equivalent model achieved by adding  $\beta_0$ 's value to each  $\beta_i$ . Mathematically, this means that there are only  $n$  independent variables in the SSE fitting, and the full set is collinear. In practice, we drop the constant  $\beta_n$ , but this does not mean that packets arriving in that bin are any less well-modeled. Rather, one can think of bin  $n$  as being the “default” case.

The azimuth can be divided into arbitrarily many bins. The more finely it is divided, the more degrees of freedom the model offers, and thus the more closely it can be fitted to the environment. To investigate the effect of bin number, we modeled every data set using from two to twenty bins. Figure A.16 shows the residual standard error as a function of bin count. The grey box plot depicts the mean and interquartile range for all of the data collectively, and the foreground lines show values for links individually. In general, there appears to be a diminishing return as the number of bins increases, with the mean remaining nearly constant above 16 bins.

In discussing parameters for this model, we will use the 16-bin case specifically. We find the same patterns across other numbers, though the actual coefficients are bin count specific. One result of note with regard to bin count is this: Several environments exhibit a “sawtooth” pattern in which the odd bin counts do better than the even ones, or vice versa. This appears to be an effect of the **alignment** of the bins relative to environmental features, rather than the **number** of bins as such.

Our model has significantly less error than the orthogonal model: Across all data sets, the mean residual standard error is 4.0 dB (**4.4dB indoors**), compared to 6.15 dB (**7.312 dB indoors**) for the orthogonal model. More importantly, the error remaining in the discrete offset model is largely noise: The mean error is almost exactly zero for several ways of grouping the data. Figure A.17 depicts the error (predicted value minus observed value). While the outliers reveal some direction-correlated effect that is not accounted for, this model is much better for the bulk of

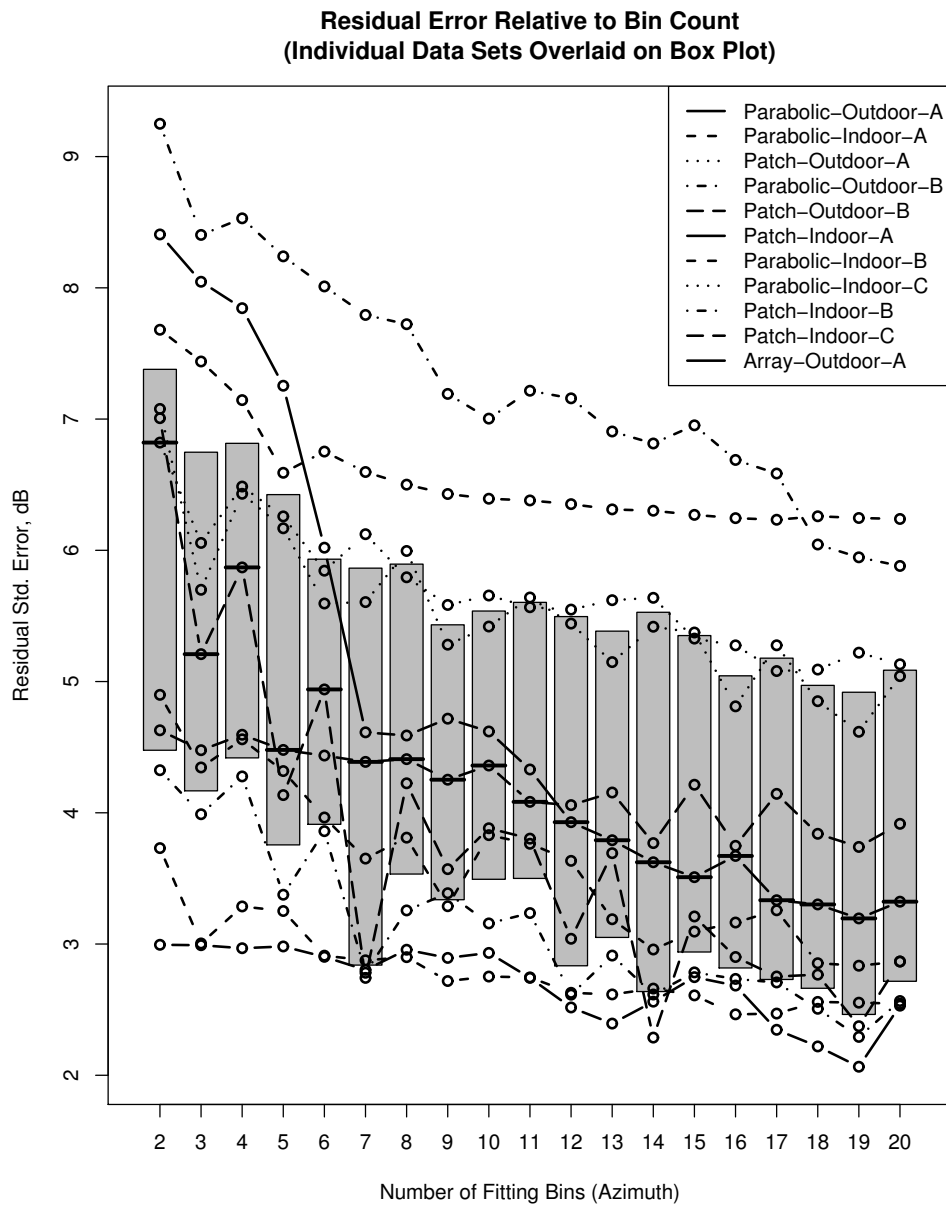


Figure A.16: Effect of increasing bin count (decreasing bin size) on modeling precision.



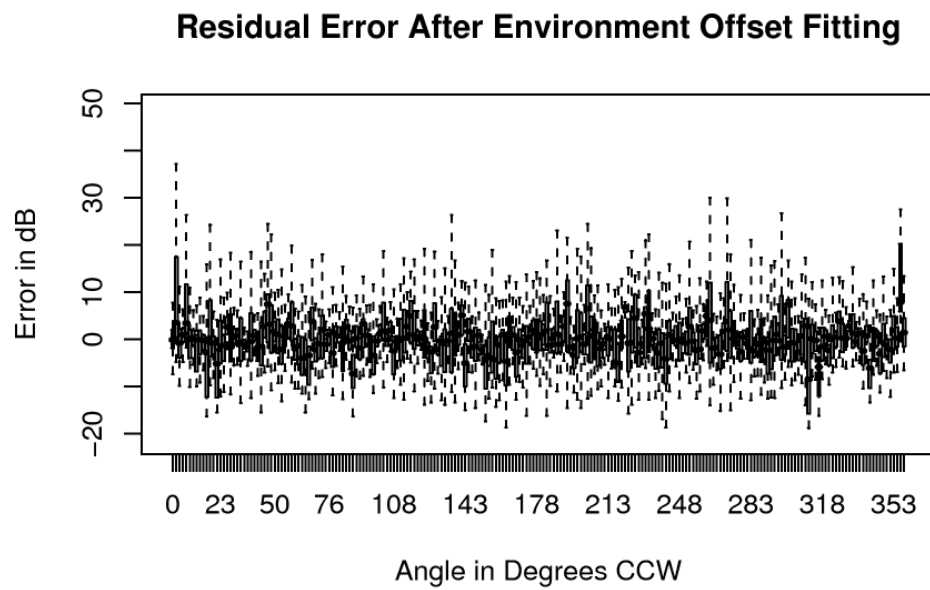


Figure A.17: Residual error of the discrete offset model with 16 bins.

the traffic. Over 99.9% of the traffic **at every angle** falls within the whisker interval.

### A.5.3 Describing and Predicting Environments

The environmental offset function  $x(\phi, \theta)$ , or its bin offset counterpart, models the impact of a particular environment combined with a particular antenna. This can serve as an **ex post facto** description of the environment encountered, but it also has predictive value: If one knows the offset function for a given environment, it is possible to more accurately model wireless systems in that environment. We are not aware of any practical way to know the exact spatial RF characteristics of an environment—and thus its offsets—without actually measuring it. However, our results suggest that it is possible to identify parameters generating the **distribution** from which the offset values for a **class of environments** are drawn.

We analyzed a range of possible determining factors for the fitted offsets across all traces and a range of bin counts. A linear regression fit and ANOVA test found significant correlation with two factors: The nominal antenna gain  $f(\theta)$  and the observation point. None of the other factors examined were consistently significant. Table A.2 shows the regression coefficients and P-values for both factors for a variety of traces. The observation angle was always statistically significant, but the coefficient is constantly near zero. For each factor, the regression coefficient describes the correlation between the fitted offset and the factor. That is, the coefficient shows how much the actual signal strength can be expected to differ from the orthogonal model for any value of that factor. For example, the antenna gain coefficients of .668 and .703 for Parabolic-Indoor-C and Patch-Indoor-C mean that in those data sets for every dB difference in antenna gain between two angles, **the best fit difference in actual signal strength is only  $\approx 0.3$  dB**.

There are two key results pertaining to the antenna gain regression coefficient: First, the coefficients for different antennas in the same environment are very close. Second, the coefficients for distinct but similar environments are fairly close. This suggests that classes of environments can reasonably be characterized by their associated coefficients, which provides a compact representation of environment classes that lends itself easily to simulation. In this way, the task of the

Data Set	Factor	Coefficient	P-value
Parabolic-Outdoor-A	Antenna Gain	0.185	1.02e-87
	Obs. Angle	0.00301	5.1e-06
Patch-Outdoor-A	Antenna Gain	0.146	6.4e-50
	Obs. Angle	0.00744	1.14e-17
Array-Outdoor-A	Antenna Gain	0.41	2.03e-206
	Obs. Angle	-0.0271	5.36e-188
Parabolic-Outdoor-B	Antenna Gain	0.0377	8.68e-05
	Obs. Angle	-0.00323	5.95e-05
Patch-Outdoor-B	Antenna Gain	0.00919	0.0492
	Obs. Angle	-0.00198	3.08e-06
Parabolic-Indoor-A	Antenna Gain	0.33	4.6e-102
	Obs. Angle	0.00463	1.91e-05
Patch-Indoor-A	Antenna Gain	0.258	1.22e-122
	Obs. Angle	0.00894	3.09e-24
Parabolic-Indoor-B	Antenna Gain	0.378	2.2e-134
	Obs. Angle	0.00971	1.97e-16
Patch-Indoor-B	Antenna Gain	0.372	1.1e-81
	Obs. Angle	0.014	3.87e-18
Parabolic-Indoor-C	Antenna Gain	0.668	1.39e-234
	Obs. Angle	-0.0146	4.15e-36
Patch-Indoor-C	Antenna Gain	0.703	0
	Obs. Angle	-0.0154	2.63e-48

Table A.2: Factors influencing fitted offset values, 16-bin case.

researcher is reduced to choosing amongst several **representative environment classes** when designing their experiment.

## A.6 Simulation Process

The statistical model laid out above can be used as the basis for more realistic simulations. It has long been recognized that radio propagation involves very environment-specific effects. We identify three major ways of addressing such effects in modeling and simulation: The first is to simply ignore the variability and use a single representative value in all cases. The second, which goes to the opposite extreme, is to model specific environments in great detail. A third approach is to randomly generate values according to a representative process and perform repeated experiments.

Each approach has its benefits, but we are advocating the repeated sample approach. Precisely modeling a specific environment probably has the greatest fidelity, but it provides no information as to how well results achieved in a single environment will generalize to others. Stochastic models have the advantage of being able to produce arbitrarily many “similar” instances, and parametric models make it possible to study the impact of varying a given attribute of the environment. Such approaches are frequently used to model channel conditions [Neskovic 00], network topology [Zegura 96, Tangmunarunkit 02], and traffic load [Lee 05].

The following algorithms produce signal strength values consistent with our statistical findings. The key parameters are the gain offset correlation coefficient  $K_{gain}$ , the offset residual error  $S_{off}$ , and the per packet signal strength residual error  $S_{ss}$ . We computed these values across many links for two types of environments in sections A.5.3 and A.5.2. Table A.3 summarizes these results.

<b>Environment</b>	$K_{gain}$	$S_{off}$	$S_{ss}$
Open Outdoor	0.01 - 0.04	1.326 - 2.675	2.68 - 3.75
Urban Outdoor	0.15 - 0.19	2.244 - 3.023	2.46 - 2.75
LOS Indoor	0.25 - 0.38	2.837 - 5.242	2.9 - 5.28
NLOS Indoor	0.67 - 0.70	3.17 - 3.566	3.67 - 6.69

Table A.3: Summary of Data Derived Simulation Parameters: Gain-offset regression coefficient ( $K_{gain}$ ), offset residual std. error ( $S_{off}$ ), and signal strength residual std. error ( $S_{ss}$ ).

Algorithm A.1 is a one-time initialization procedure which computes the offsets between the

antenna gain in any direction and the expected actual signal gain.

**Algorithm A.1:** Compute Directional Gain

```

 $K_{gain} \leftarrow$  gain offset correlation coefficient
 $S_{off} \leftarrow$  offset residual std. error
begin Direct-Gain
  forall Node  $n \in$  all nodes do
     $P \leftarrow$  partition of azimuth range  $[-\pi, \pi)$ 
    forall  $p_i \in P$  do
       $\theta_i \leftarrow$  center angle of  $p_i$ 
       $X \leftarrow$  random value ( $\mu = 0, \sigma^2 = S_{off}$ )
       $o_{n,p_i} \leftarrow K_{gain} * f_n(\theta_i) + X$ 
    end
  end

```

Algorithm A.2 computes the expected end-to-end gain for a given packet, *not including fixed path loss*. Thus, the simulated signal strength would be determined by the transmit power, path loss, receiver gain, fading model (if any), and the directional gain from algorithm A.2. Note that a fading model that accounts for interpacket variation for stationary nodes might make the random error  $\epsilon$  in line 0 redundant.

## A.7 Conclusion

In this chapter, we have presented an empirical study of the way different environments and antennas interact to affect the directionality of signal propagation. The three primary contributions of this work are:

- (1) A well-validated method for surveying propagation environments with inexpensive commodity hardware.
- (2) A characterization of several specific environments ranging from the very cluttered to the very open.
- (3) New, more accurate, techniques for modeling and simulating directional wireless networking.

**Algorithm A.2:** Compute Packet Gain

```

 $S_{pss} \leftarrow$  residual error of packet signal strengths
begin Directional-Packet-Gain
   $\theta_{src} \leftarrow$  direction from  $src$  toward  $dst$ 
   $\theta_{dst} \leftarrow$  direction from  $dst$  toward  $src$ 
   $p_{src} \leftarrow$  partition at  $src$  containing  $\theta_{src}$ 
   $p_{dst} \leftarrow$  partition at  $dst$  containing  $\theta_{dst}$ 
   $G_{src} \leftarrow f_{src}(\theta_{src}) - o_{dst,p_{src}}$ 
   $G_{dst} \leftarrow f_{dst}(\theta_{dst}) - o_{src,p_{dst}}$ 
o  $\epsilon \leftarrow$  random value ( $\mu = 0, \sigma^2 = S_{pss}$ )
  return( $G_{src} + G_{dst} + \epsilon$ )
end

```

Wireless signal—and interference—propagation is more complicated than widely-used previous models have acknowledged. Because models of the physical layer guide the development and evaluation of higher layer systems, it is important that these models describe reality with sufficient accuracy. Indeed, in [Anderson 09d] we show that application layer results reported by simulators can be affected dramatically by the way directional antenna models are simulated, producing results that deviate significantly from reality. Our measurements, and the resulting model, bring to light several important aspects of the physical environment that previous models have failed to capture. The **effective** directionality of a system depends not only on the antenna, but is influenced by the environment to such a large extent that many decisions cannot be made without **in situ** measurements.

## Appendix B

### Simulation Practices

Increasingly, directional antennas are being used in wireless networks. Such antennas can improve the quality of individual links and decrease overall interference. However, the interaction of environmental effects with signal directionality is not well understood. We observe that state of the art simulators make simplifying assumptions which are often unrealistic and can give a misleading picture of application layer performance. Because simulators are often used for prototyping and validating new ideas, their realism and accuracy are of primary importance. In this chapter, we apply a new empirical simulation method for directional antennas and study how well it models reality. We show that not only is our model easy to implement, but is also more accurate and thus better able to predict the performance of propagation-sensitive applications.

#### B.1 Introduction

Using directional antennas is currently one of the main techniques for improving link quality by increasing signal strength in some directions while lowering interference in others. Many directional networking protocols and applications, however, are studied using wireless simulation models that assume directional antennas experience environmental effects in the same way that omnidirectional antennas do. This, in turn, influences the expected behavior of the entire network stack, potentially producing significant discrepancies between simulations and empirical results.

This work makes the following contributions to improve the fidelity of wireless network simulators:

- We show that current state of the art techniques do not accurately capture the effects of the environment on directional signal propagation and can thus produce misleading results at higher layers of the network stack.
- We introduce an empirically-derived model for **signal** directionality, the **Effective Directivity Antenna Model** (EDAM), that incorporates the environment’s effects on directional antennas as a stochastic process.
- We verify EDAM’s accuracy as a simulation technique by using it to model a data-stripping application where the physical boundaries of successful packet reception are critical to overall success.
- We perform real-world indoor and outdoor experiments and compare the results with those obtained by various simulation techniques. We find that simulation based on EDAM can improve fidelity by about 60%.

In the next section, we discuss related work. In section A.3, we discuss our proposed simulation approach and the model on which it is based. Section B.4 presents a security-oriented smart-antenna application as a case study. We discuss its implementation, simulation, and an analysis of the accuracy of the various simulation approaches. Finally, in section B.5, we conclude.

## B.2 Background and Related Work

In this section, we discuss the state of the art with respect to the way network simulators model the physical layer. Figure B.1 shows the simulation framework we conceptualize in this work. We argue that while path loss models and fading models capture some of the vagaries of the medium, they insufficiently model the effects of the environment on signal directionality. Additionally, prior work [Takai 01] has shown that the way the physical layer is simulated can have substantial effects on higher layer results. This motivates our work into building an empirically derived model for the environmental effects on antenna directionality, which we call a “directivity model” and can be



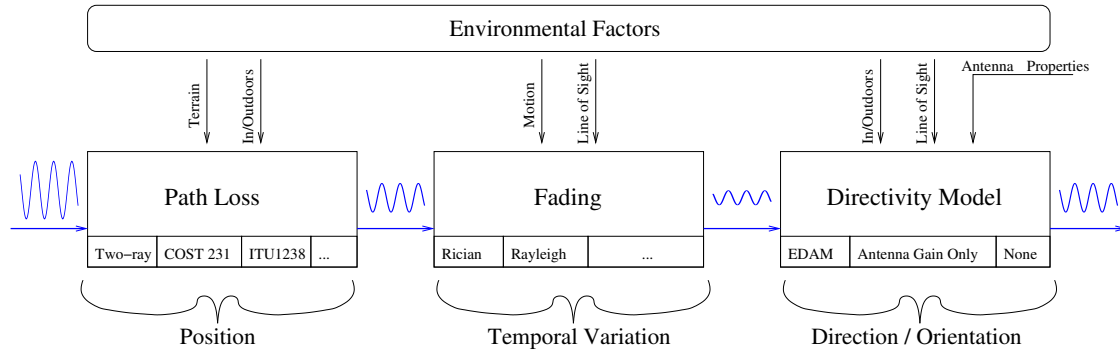


Figure B.1: Physical-layer simulation framework.

used in combination with fading and path loss models to produce a more realistic simulation of the physical layer effects in systems where antenna directionality plays a role.

Wireless network simulators use a **path loss model** to model the degradation of a transmitted signal. In free space, energy is propagated in all directions and the energy that actually strikes the receiver is proportional to the square of the distance between the transmitter and receiver – the signal is attenuated  $\propto r^2$ . This, however, ignores significant effects found in terrestrial environments. Of particular concern are absorption and refraction by obstacles and **multipath interference**, where the radio frequency (RF) waves bounce off objects in the environment and converge at the receiver after traversing different distances (and thus potentially out of phase.)

The commonly-used “two-ray” path-loss model uses a reflection from the earth and the heights of the transmitter and receiver to indicate the likely signal strength at a given distance. Other models for such effects are based on fitting empirical measurements rather than *a-priori* analysis. There are general-purpose models such as the Hata / COST231 model and the Longley-Rice model [Abhayawardhana 05, Oestges 04], and several specific to the wavelength and operating characteristics of wireless LAN cards [Green 02]. Additionally, the propagation characteristics of indoor environments are sufficiently different from outdoor environments that there are a number of measurement studies and models (see [Andersen 95], [Neskovic 00] and [Iskander 02] for excellent surveys).

The preceding models describe relatively large-scale phenomena. In addition to whatever

long-range attenuation there may be, there is also **small-scale fading**, which is the result of multipath interference and occurs at the scale of single wavelengths. While this can theoretically be predicted analytically, it would require that the environment be known with a level of detail that is generally impractical [Wolffe 05, Tingley 01]. A common way of addressing such situations is through **statistical** fading models. Rather than determine the signal strength at any exact place or time, it is modeled as a random variable with a known distribution. In general, the distributions are fairly well-established, but the parameters are environment-specific (see e.g. [de Leon 04]). There are several common models, among them **Rayleigh fading**, which assumes that there are many comparable multipath signals, and **Rician fading**, which assumes a less “cluttered” environment in which line-of-site paths are more important.

The concern addressed by this chapter is that those models do not consider an environmental component to directivity. Our model for directional antennas adopts a similar approach to the empirical path-loss and statistical fading models – we use empirical measurements to identify the characteristics of the random or stochastic process. Where we differ is that our model is primarily concerned with effects on directionality.

The most commonly used simulators in networking research (OPNET, QualNet, and NS-2) do not consider antenna directionality and radio propagation as interacting variables. Each one supports several models of path loss, and possibly fading, but they all follow the same general model with regard to antenna gain: For any two stations  $i$  and  $j$ , the received signal strength is computed according to the general form of equation A.1, repeated below:

$$\text{Received Power} = P_{tx} * G_{tx} * L(i, j) * G_{rx}$$

The received power  $P_{rx}$  is the product of the transmitted power  $P_{tx}$ , the transmitter’s gain  $G_{tx}$ , the path gain (loss)  $L(i, j)$  between the two stations, and the receiver’s gain  $G_{rx}$ . The transmitter and receiver gains are essentially constants in the case of omnidirectional (effectively isotropic in the azimuth plane) antennas. For directional antennas, gain is an antenna-specific function of the direction of interest. For some given zenith  $\phi$ , azimuth  $\theta$ , and an antenna-specific characterization

function  $f_a()$ , the power transmitted in that direction is given by equation A.2, repeated here:

$$\begin{aligned}\text{Gain in direction } (\phi, \theta) &= f_a(\phi, \theta) \\ \text{Combined gain} &= f_a(\phi, \theta) * f_b(\phi', \theta')\end{aligned}$$

Correspondingly, the receiver gain is modeled by a (potentially different) function  $f_b()$  of the direction from which the signal is received.

The above models describe the power emitted in or received from a single direction (see Figure B.2). In reality, **the transmitter’s power is radiated in all directions, and the receiver aggregates power (be it signal or noise) from all directions.** Besides being a source of interference for a dominant signal, the energy traveling along secondary paths (due to side lobes) also carries signal. Network simulators model the antenna gain and path loss using the angles and straight-line distance between the transmitter and receiver. However, if one of the “secondary” reflected or refracted signal paths is aligned with a high-gain direction of a transmitting or receiving antenna, the received power from that path can be greater than that of the “primary” path. Thus in environments with significant multipath, the gain cannot be determined based solely on a single direction. It makes intuitive sense that if a narrow beam is directed into a scattering environment, the resulting signal is probably not narrowly focussed.

Although the simulators we are considering assume that the single direction of interest for each station is precisely toward the other station, one can generalize equations A.1 and A.3 to the case where there are multiple significant signal paths. In this case, it is crucial to note that  $f_a$ ,  $f_b$ , and  $L_l$  are complex, so summation does not automatically imply an **increase** in magnitude.

$$P_{rx} = \sum_{l \in \text{paths}} P_{tx} * f_a(\phi_l, \theta_l) * L_l(i, j) * f_b(\phi'_l, \theta'_l)$$

This assumes that there is some way to identify the available paths that a signal may take. As with the Rayleigh and Rician fading models, it may be possible to build a parameterized model of the cumulative effect of those paths for “cluttered” and “uncluttered” environments.

<b>Environment</b>	$K_{gain}$	$S_{off}$ (dB)	$S_{ss}$ (dB)
Open Outdoor	0.01 - 0.04	1.326 - 2.675	2.68 - 3.75
Urban Outdoor	0.15 - 0.19	2.244 - 3.023	2.46 - 2.75
LOS Indoor	0.25 - 0.38	2.837 - 5.242	2.9 - 5.28
NLOS Indoor	0.67 - 0.70	3.17 - 3.566	3.67 - 6.69

Table B.1: Summary of Data-Derived Simulation Parameters, repeated from Table A.3 on page 196.

### B.3 A New Simulation Approach

In [Anderson 09a], we present a statistical model for the environmental effects on antenna directionality. This statistical model can be used as the basis for more realistic simulations. It has long been recognized that radio propagation involves very environment-specific effects. We identify three major ways of addressing such effects in modeling and simulation: The first is to simply ignore the variability and use a single representative value in all cases. The second, which goes to the opposite extreme, is to model specific environments in great detail. A third approach is to randomly generate values according to a representative process and perform repeated experiments.

Each approach has its benefits, but we are advocating the repeated-sample approach. Precisely modeling a specific environment probably has the greatest fidelity, but it provides no information as to how well results achieved in a single environment will generalize to others. Stochastic models have the advantage of being able to produce arbitrarily many “similar” instances, and parametric models make it possible to study the impact of varying a given attribute of the environment. Such approaches are frequently used to model channel conditions [Neskovic 00], network topology [Zegura 96, Tangmunarunkit 02], and traffic load [Lee 05].

The key parameters to our method are the gain offset correlation coefficient  $K_{gain}$ , the offset residual error  $S_{off}$ , and the per-packet signal strength residual error  $S_{ss}$ . These values were computed across many links for multiple environments. Table A.3 summarizes these results. Importantly, similar environments produced similar values, even with different antennas. Because of this, it is possible for a researcher to select representative values based on a qualitative understanding of the environment of interest.

EDAM’s principle of operation is that it generates randomized environmental effects based on the fitted distributions of effects measured in real environments. This has two main components: Algorithm A.1 on page 197 is a one-time initialization procedure which computes offsets between the antenna gain in any direction and the expected actual signal gain. Algorithm A.2 on page 198 computes the expected end-to-end gain for a given packet, *not including fixed path loss*. Thus, the simulated signal strength would be determined by the transmit power, path loss, receiver gain, fading model (if any), and directional gain from Algorithm A.1 on page 197. Note that a fading model that accounts for inter-packet variation for stationary nodes may make the random error  $\epsilon$  in line 0 redundant. In our simulation configurations below we refer to this error term as “implicit Gaussian” fading and consider scenarios where it is replaced with Rician and Log-normal fading distributions.

#### B.4 Case Study: Physical Space Security using Smart Antennas

In this section, we use the work of Lakshmanan et al. as a case study for the way antenna simulation strategy effects application layer performance. In [Lakshmanan 08], the authors propose “Data Striping” as a way of achieving physical space security by steering antennas. Downstream packets are encrypted and split into multiple parts so that all parts must be received in order to decode any portion of the packet. Several access points, which are presumed to have smart antennas, then transmit the packet parts so that the only point at which all the required information is available is at the intended receiver. In this scenario (see Figure B.3), an eavesdropper who is outside the coverage area of **any** of the access points will only receive a subset of the packet parts and therefore be unable to reconstruct the message. The measure of the effectiveness is the size of the region in which an attacker can successfully receive and reconstruct packets for any given probability of success.

The authors verify their work using a custom simulator that implements the International Telecommunication Union’s (ITU) indoor attenuation model combined with log-normal fading. This channel model fits well with our discussion in Section B.2: While the path loss and fading

models are nontrivial, there is no interaction between the environment and directionality. Because directionality is crucial to the proper function of this application, it is important to understand how the environment may affect performance.

#### **B.4.1 Implementation**

In order to understand the effects of the environment on the application and get a baseline for further analysis, we built a custom measurement testbed and ran tests in multiple environments – both indoors and outdoors. Figure B.3 shows the conceptual layout of the experiments. Five nodes were used – three APs, one client, and one eavesdropper. The eavesdropper was positioned in many locations on a grid, at each of which the access points sent a volley of broadcast packets (approximately 500) to the client while the eavesdropper attempted to overhear them from its location. The indoor experiment was carried out in a cluttered office with 83 unique measurement points. The outdoor tests required 437 measurement points and were carried out in a large field on the University of Colorado campus.

All five nodes in the experiment were laptops running Linux, with Atheros radios. The access points used 24 dBi parabolic dish antennas, mounted on tripods and manually aimed at the client according to signal strength values. The client and eavesdropper used external omnidirectional antennas with approximately 5 dBi gain. For the indoor experiments, we reduced the power on the access points so that the received power at the (stationary) client was between -70 and -75 dBm. This was motivated by prior observations on the large amount of uncorrelated noise produced by high-power antennas in highly reflective indoor environments [Anderson 08a]. The outdoor experiments were carried out without any power reduction.

#### **B.4.2 Simulation**

For simulation, we used the popular network simulator QualNet 4.5 with physical layer simulations of varied complexity. Each configuration has some combination of the simulation layers listed in Table B.2. We conducted a factorial experimental design, trying each unique combination

Path Loss	Fading	Directivity Model
Two-ray ITU 1238	Log-normal Rician Implicit Gaussian None	EDAM “Pure” “Omni”

Table B.2: Physical-layer simulation options

of path loss model, directivity model, and fading model. While there are a variety of established path loss and fading models, we are not aware of any existing directivity models analogous to what we propose. The alternatives considered are essentially two null hypotheses: The first is that there is no significant environmental effect, and the antenna gain pattern sufficiently describes the signal directionality. This is the “**pure**” directivity model. The second is that environmental effects completely dominate the antenna effects, and so the signal is effectively isotropic. This is the “**omni**” directivity model. One might expect difficulty rejecting the first null hypothesis in an open outdoor environment and the second in a highly-cluttered indoor environment.

The simulated experiments were modeled directly after the implementation discussed in section B.4.1. Five nodes were simulated, placed in the same relative positions as in the actual experiments. The transmitters and intended receiver were stationary, while the eavesdropper moved to the same points as in the implementation. Both indoor and outdoor experiments were run. The simulation processes were identical except for the EDAM parameters: The indoor simulation used the “non-line-of-sight (NLOS) indoor” values, while the outdoor simulation used the “urban outdoor” values. To deal with power calibration, we calibrated each simulation configuration manually so that the RSS values were comparable to those we observed in the actual implementation for only the intended receiver. We made ten unique runs per simulation, each with a different random seed.

### B.4.3 Analysis

In alignment with the literature [Takai 01], our results show that system performance varies tremendously between simulation models. Table B.3 shows the number of locations at which an

Directivity Model	Vulnerability region (points)
Measured	38
Pure antenna	3 - 5
EDAM	54 - 79
Omni (no directionality)	83 (100%)

Table B.3: Size of 50% vulnerability region, indoor scenario.

eavesdropper can successfully decode  $\geq 50\%$  of all packets. **The actual vulnerability region is 10 times what a current simulator would predict.**

By plotting the probability of an eavesdropper receiving a decodable packet at each position, we can observe that the simulations with the EDAM model are closer to reality than those without it. To quantify this effect, see Figure B.4 where a cumulative density function (CDF) of the probability of decoding a packet is plotted for each of the ten seeds against the measured data from the implementation. Looking at Figures B.4a and B.4b, we can see that EDAM performs well. On the other hand, consider Figures B.4c and B.4d, where state of the art models (such as those used here) without a directivity model grossly overestimate the effect of the antenna pattern on actual signal strength, and thus the performance of this application.

Figures B.4e and B.4f give plots of outdoor results. Figure B.4e is a pathological case, with the “pure” directivity model and no fading model. In this case, the boundaries are stark – nearly 60% of locations are protected, while the remaining 40% are not. Although this performs poorly, it is worth considering as not all simulation software uses a fading model by default. For instance, the popular simulation package NS-2 has none unless it is paired with an extension such as [Baldo 07]. Finally, figure B.4f shows the best performing outdoor simulation strategy. We can see that the benefits of EDAM are more pronounced in indoor simulations where multipath reflections are more prevalent.

To determine which simulation approach produces application-layer results that are most consistent with the measured data, we compare the distribution of simulated application-layer performance with the distribution of actual performance. We use a two sample Kolmogorov-Smirnov



(KS) test, which is effectively the maximal distance between the CDFs of the two samples. We then perform an analysis of variance (ANOVA) on the KS test results to determine how much the various factors (directional model, path loss model, and fading model) contribute to the overall accuracy.

Figure B.5 provides a box and whiskers plot of the KS test-statistic for each configuration. In this diagram lower values indicate better performance, meaning that the distribution of simulated performance closely follows the measured real-world performance. Alternately, high values indicate that the simulated performance deviates wildly from the measured performance. We can see that the configurations utilizing EDAM perform very well – producing application layer results which are much closer to reality than any other configuration. EDAM performs best in the indoor simulations, claiming the top three positions with this metric – EDAM with ITU 1238 and Rician fading performs best, with less than 0.3 difference from the empirical data at maximum. The other two top positions are taken by EDAM with other fading or path loss models.

In the outdoor simulations, the conclusions are less strong – EDAM with two-ray and log-normal fading performs best, but it is closely followed by EDAM with ITU 1238 and Rician fading, and two-ray with Rician fading and the “pure” directivity model. While the strength of EDAM is greatest in cluttered environments such as our indoor environment, it is important to note that it still offers a significant improvement in the outdoor environment.

Table B.4 shows the results of a factorial analysis of variance (ANOVA), using the KS statistic and the various physical layer simulation models as the factors. Note that the “omni” directivity model is not included because it is so inaccurate that its inclusion obscures the other effects. The test results show that the choice of the directivity model is by far the dominant factor indoors, and a substantial factor outdoors.

In the indoor environment, the effect of directivity model is **4.9 times greater than any other factor**. Outdoors, the path loss model is the dominant factor, followed by the fading model and then the directivity model. Both indoors and outdoors, a Rician fading model performed better than log-normal or implicit Gaussian models. Somewhat predictably, the ITU 1238 indoor path

Factor	Indoor			Outdoor		
	Df	Mean Sq.	F-Value	Mean Sq.	F-Value	F-Value
Dir. Model	1	332.2	159911.55	27.14	34476	
P.L. Model	1	68.0	32735.13	127.82	162367	
Fad. Model	3	25.6	12315.35	73.46	93313	
Dir. Model * P.L. Model	2	57.8	27841.84	31.49	39998	
Dir. Model * Fad. Model	3	7.42	3572.82	27.70	35182	
P.L. Model * Fad. Model	3	0.1	26.81	17.90	22733	
Dir. Model * P.L. Model * Fad. Model	3	0.4	196.02	32.38	41135	
Error		0.0021		0.001		

Table B.4: Summary of results for factorial ANOVA on KS-test statistic across all simulation configurations except for the “omni” directivity model. P-values are omitted because all treatments are statistically significant at a level of  $p < 2.2 * 10^{-16}$ . Error / residual Df are 9948 indoor, 52428 outdoor.

loss model did better than the two-ray model indoors, but the two-ray model did better outdoors.

## B.5 Conclusion

In this chapter we have presented EDAM, a novel empirical method for improving the modeling of directional antennas in simulators. EDAM is both easy to implement and generalizable to a wide variety of directional antennas. We have shown that state of the art techniques for modeling physical-layer behavior for directional wireless networks can be misleading. Moreover, the addition of a directivity model to the conventional simulation stack provides a critical contribution to the ability of a simulator to produce realistic application layer results. *Not only do simulations using EDAM produce application layer results that are significantly more consistent with reality than traditional models in the application we study, but the choice of directivity model is the most influential factor in realistic simulation of indoor environments.* We have verified this with a factorial experimental design and a test-bed implementation in representative indoor and outdoor locations. EDAM is easily incorporated into wireless networking simulations\* , and is consistently more accurate than the state of the art for systems involving directional antennas.

---

\* Our implementation of EDAM is available as a patch to version 4.5 of the QualNet Simulator at <http://systems.cs.colorado.edu/wiki/EDAM>.

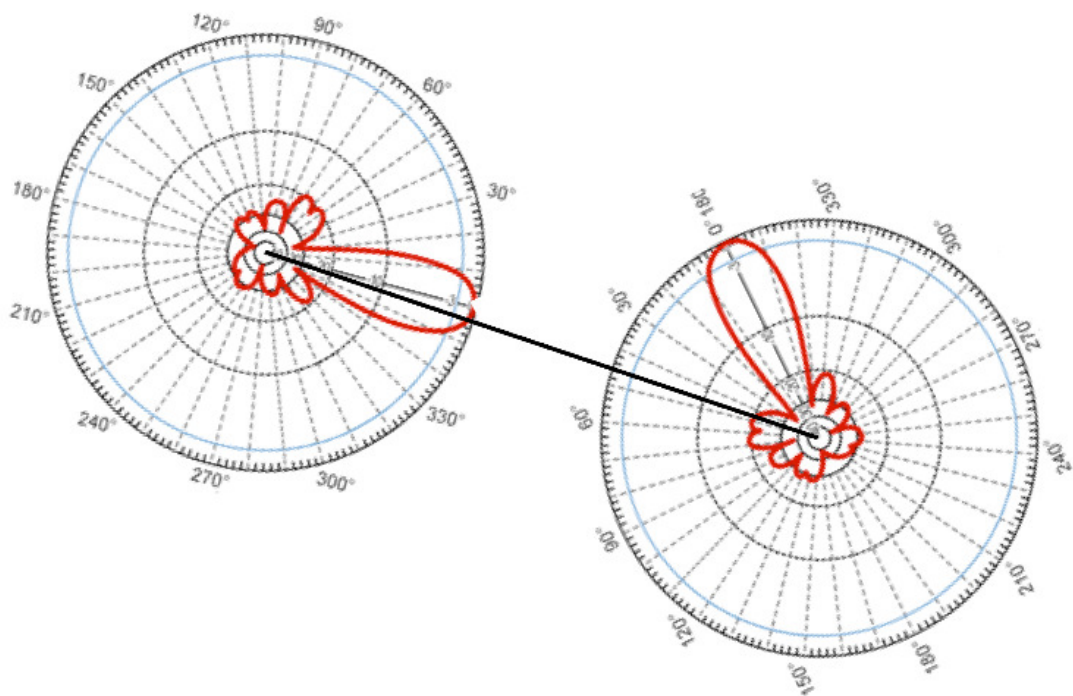


Figure B.2: Standard simulation model of directional antennas assumes all signals are emitted along a single path.

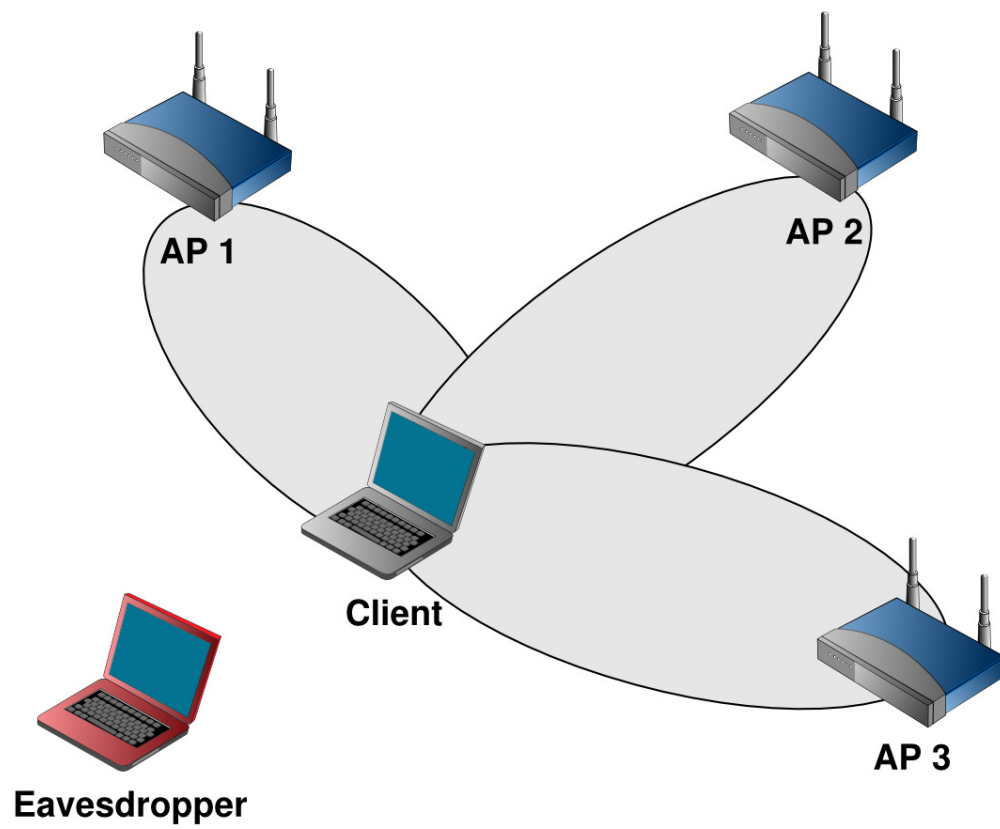
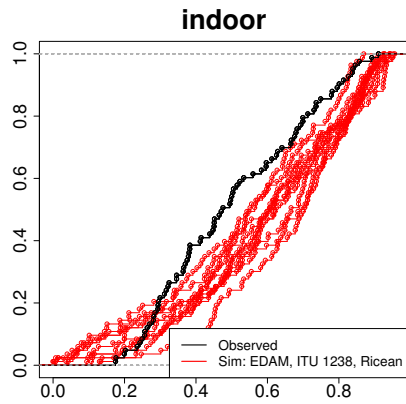
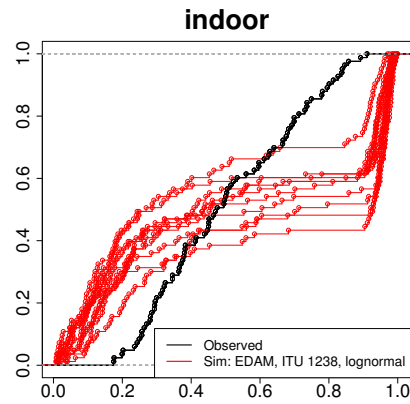


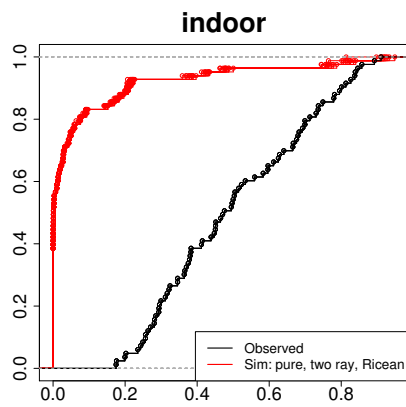
Figure B.3: Example of data striping application



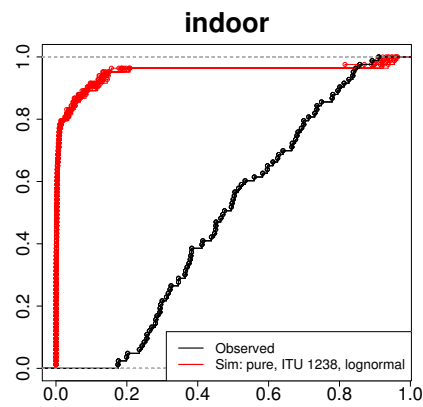
(a) EDAM, ITU 1238, Rician fading, indoor



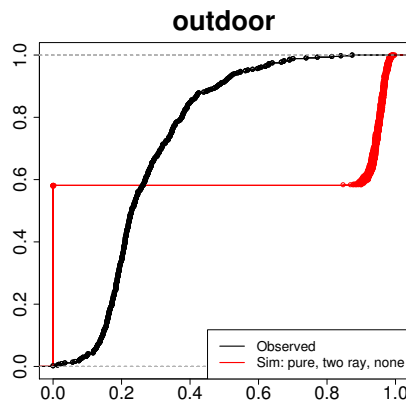
(b) EDAM, ITU 1238, Log-Normal fading, indoor



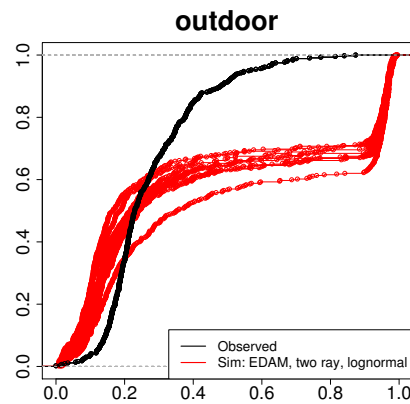
(c) Pure antenna model, Two-ray, Rician fading, indoor



(d) Pure antenna model, ITU 1238, Log-normal fading, indoor



(e) Pure antenna model, Two-ray, no fading, outdoor



(f) EDAM, Two-ray, Log-normal fading, outdoor

Figure B.4: CDF plots of application layer performance for simulation configurations: The black line is the **observed data**, and the red (or grey) lines are the results of repeated simulation runs. The X axis is the proportion of decodable packets, and the Y axis is the cumulative fraction.

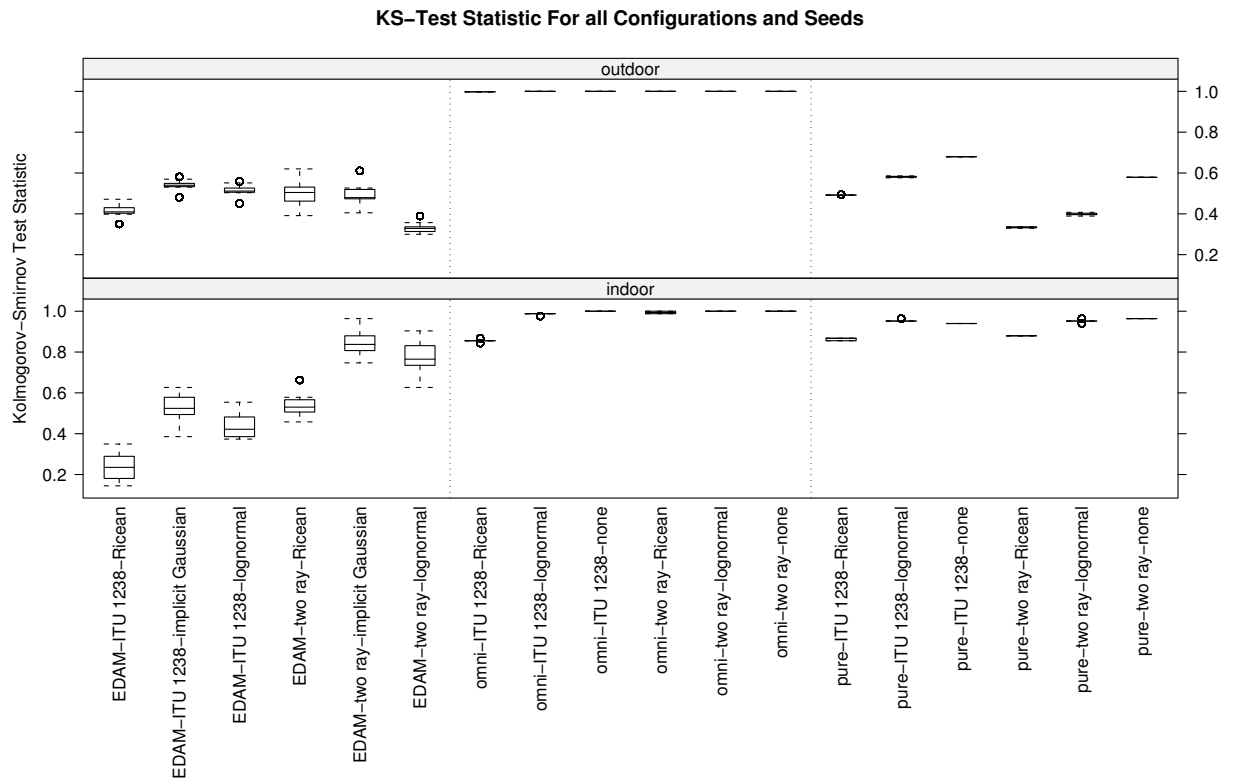


Figure B.5: Test-statistic of a two sample Kolmogorov-Smirnov test, run for each simulation configuration against the measured data (smaller values are better).

## Appendix C

### The Wide-Area Radio Testbed

#### C.1 Introduction

Directional antennas, both fixed and steerable, are proving to be important in the next generation of wireless networking protocols. These antennas give nodes further control over both signal strength and interference, allowing optimization techniques which can yield greater network throughput with fewer errors. While protocols incorporating directional or “smart” antennas have been proposed, their evaluation has been limited. Those researchers who have attempted real-world evaluation of their ideas have often used one-off testbeds assembled to perform a small number of experiments [Ramanathan 05, Mitsuhashi 07, Kohmura 08]. Most proposals, however, rely solely on simulation or theoretical analysis (for instance, [Takai 02, Singh 05]).

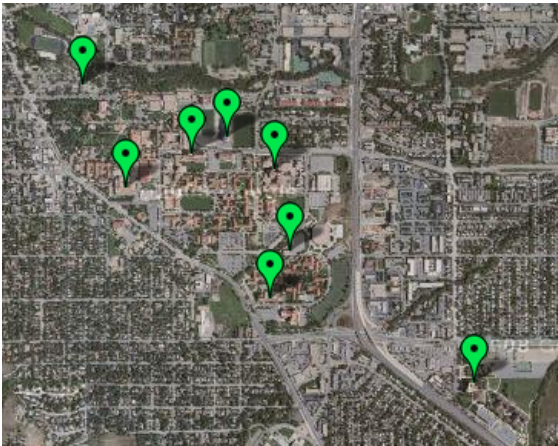
In this chapter we introduce the University of Colorado Wide-Area Radio Testbed (WART) as a platform for studying uses of directional, steerable, and smart antennas in wireless networking. Given the widely-recognized difficulty of accurately simulating radio environments, real-world experiments are essential for fully understanding wireless networking. The effects of antenna configuration are especially dependent on the vagaries of radio propagation, so physical fidelity is particularly important for this area of research (see Chapter A on page 162).

WART is currently the only permanent facility for studying smart antennas in a large and diverse urban environment. The system consists of eight phased array antenna nodes mounted to the rooftops of university buildings, spanning an area of 1.8 x 1.4 kilometers. The entire testbed is linked together via wired Ethernet and can be controlled from a single administration point.



This architecture ensures that WART can both offer the geographic scale and realism of large-scale distributed testbeds [Aguayo 04a] and also give its users the degree of control and ease of management only seen in dense indoor testbeds such as ORBIT and Emulab [Raychaudhuri 05b, Johnson 06].

The production and deployment of such a testbed, however, is itself an engineering problem. In addition to the capabilities of WART, this chapter describes some of the logistical challenges encountered in planning, installing, and maintaining a centrally-controlled wide area rooftop network.



(a) Campus Testbed (1.8 x 1.4 km)



(b) Installed Antenna Node

### C.1.1 Design Goals

WART is intended to be a dedicated experimental testbed for studying the impact of omnidirectionality, directionality, null-steering and beam-forming throughout the network stack. Given this objective, there were three design goals for WART:

- (1) The testbed must be able to perform outdoor omni-directional, fixed directional, and beam-forming experiments.
- (2) The testbed must be able to test a diverse set of link distances of varying link qualities.
- (3) WART nodes must be simple to reconfigure for varying experiments and provide an easy recovery mechanism in case of failure.

The environment chosen was the rooftops of several tall buildings at the University of Colorado at Boulder. These sites were chosen to provide a variety of link lengths and line-of-sight between most, but not all, pairs of nodes. It was important to get a number of long links in order to study links with lower signal strengths at varying transmit powers. Note that this is in contrast to producing weak links by decreasing transmit power, which is only an approximation of long links. An indoor setting or an environment with a large number of reflections would not have been as appropriate for our directional studies due to the significant effects such an environment would have had on beam patterns[Anderson 08b].

The remainder of this chapter describes the hardware, software, and centralized architecture of WART which helps fulfill the design goals of easy maintenance and administration.

### **C.1.2 Smart Antenna System**

In this section we describe the hardware and software that comprise WART. These components give it the unique ability to perform smart antenna research at all network stack levels and address challenges with its administration and experimental setup.

#### **C.1.2.1 Hardware**

Each smart antenna node consists of two major components: the phased array antennas and the embedded computer. The phased array antennas used in our study were designed and constructed by Fidelity Comtech. The antennas operate in the 2.4GHz ISM band and use an 8 element uniform circular array of dipole antennas that support a minimum  $42^\circ$  primary lobe when configured for a unidirectional pattern, as shown in Figure C.1. The tight unidirectional pattern has a primary lobe gain of 18dBi. Additionally, the ratio of the lowest null to the highest peak is  $\approx 40$  dB, which allows for selectively “nulling out” interfering signals.

Each dipole is controlled by a vector modulator which in turn is controlled by a distinct embedded processor. Intrinsic antenna reconfiguration time is  $\approx 10\mu$ seconds, although the interface with the transceiver boards limits the effective reconfiguration time to  $\approx 100\mu$ seconds. The

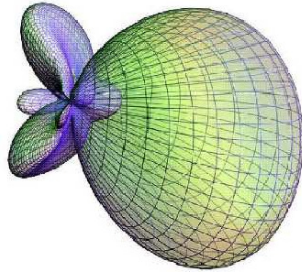


Figure C.1: Unidirectional Pattern

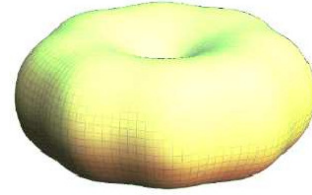


Figure C.2: Omnidirectional Pattern

transceiver boards are controlled by a series of phase-amplitude settings stored in flash memory, which allows fast reconfiguration between set patterns. For example, the antenna can quickly change the direction of the pattern shown in Figure C.1, or switch to the omnidirectional pattern in Figure C.2, by indicating the pre-computed configuration to be used.

The embedded computer is a single-board computer (SBC) based on the Intel XScale IXP425 processor. The entire system runs off 128 MB of memory and thus relies on the wired network connection for reading/writing to a long-term storage device. The wireless interface card used is a Senao 5345MP MiniPCI adapter, which uses an Atheros chipset. The combined antenna and embedded computer is roughly 26x23x23 cm in size and can be mounted on vehicles, light poles, and buildings.

#### C.1.2.2 Software

The default image used by each WART node is a standard OpenWRT Kamikaze distribution with some modifications to the default wireless drivers and startup scripts. This Linux distribution was selected because of its maturity, support for the embedded IXP425 processor, and standard tools such as python and tcpdump. The wireless driver is based on the Multi-band Atheros Driver (MADWiFi) version 0.9.4.5 and is modified to control the loading and selection of antenna patterns. Lastly, NFS is used to transmit data from the smart antenna node to long term storage.

## C.2 Commodity Hardware as a Research Platform

In this section, we discuss limitations of commodity hardware with respect to research applications and the solutions we have developed to mitigate them. Principally, we want to:

- Be confident in the fidelity of physical-layer (PHY) measurements and settings.
- Implement and study experimental medium access control (MAC) protocols.
- Have precise control of timing and strict clock synchronization.

### C.2.1 Received Signal Strength Accuracy

To ensure that it is safe to use commodity IEEE 802.11x-based hardware to measure signal and interference levels, we calibrated the sensitivity of our radios against known signal sources.

To get an idea of how accurate our commodity radios are in measuring received signal strength (RSS), we directly connected each of our radio cards to an Agilent E4438C vector signal generator (VSG). The VSG was configured to generate IEEE 802.11 frames and the laptop to receive them. For each of the cards we collected many samples while varying the transmit power of the VSG between -20 dBm and -95 dBm (lower than the receive sensitivity threshold of just about any commodity 802.11 radio) by 5 dBm increments. The resulting data is plotted in Figure C.3 along with a linear fit with a slope of 0.9602 and adjusted R-squared value of 0.9894 (indicating a strong fit to the data). The commodity radios perform remarkably well in terms of RSS measurement. To correct for the error they do exhibit, we use the slope and intercept of this fit to adjust our measurements.

After calibration, the residual error has nearly zero mean (-0.05 dBm) and a standard error of 1.7 dBm. The standard error of the sample mean varies as  $SE_{\bar{x}} = \frac{s}{\sqrt{N}}$ . This implies that any reasonable confidence level can be achieved by taking a practical number of samples. For example, 12 samples give a 95% confidence interval of  $\pm 1$ dBm, 45 samples gives  $\pm 0.5$ dBm, and so on.

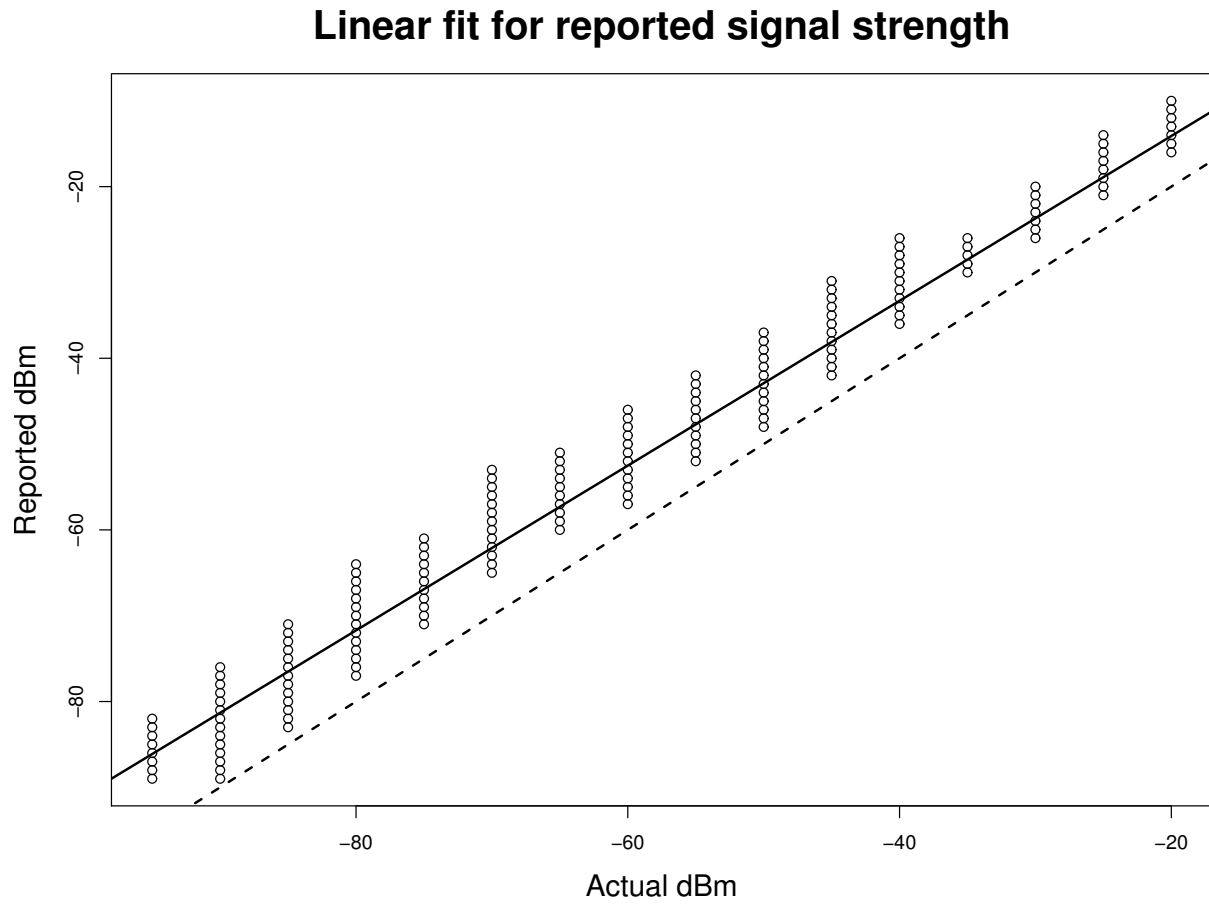


Figure C.3: Linear fit of reported versus actual signal strength on commodity cards during calibration.

### C.2.2 Transmit Power Precision

At least two studies have analyzed the fidelity of transmit power control in commodity wireless network interface cards (NICs) [Shrivastava 07, Ben Abdesslem 06]. Neither provides an exact calibration for our specific hardware, but they provide sufficient guidance for the type of experiments we have been performing. The devices studied offer a software API for setting transmit power, accepting settings in 1 dBm increments. These setting requests are implemented at a much coarser granularity by all of the hardware considered, including Atheros chipsets.\* It is therefore not safe to assume that the requested power level matches the actual power level without first identifying the specific power levels supported by the hardware in use.

Because the phased array antenna provides additional – and relatively fine-grained – amplification, we are not particularly concerned with the absolute power level produced by the wireless NICs. Of more concern is the relative consistency. Shrivastava et. al. provide a conservative estimate: Their paper analyzes the combined variability of the transmitter, the channel, and the receiver. In the situation with the least expected exogenous variability (**LOS-light**), the apparent standard deviation of signal strength is less than 2 dBm. Additionally, their stationarity analysis shows a very low Allan deviation over both short (tens of packets) and long (thousands of packets) intervals [Shrivastava 07].

This suggests that the sample sizes discussed for mitigating receiver measurement error are also reasonable for transmitter variability, and that samples separated by significant periods of time ought to be comparable.

### C.2.3 MAC-Layer Flexibility

A challenge associated with using COTS wireless cards for research purposes is that the driver-card combination functions as a “black box.” The exposed functionality is generally not sufficient for physical and MAC-layer experimentation.

---

\* The Linux Wireless Extensions API allows device drivers to specify the set of supported power levels, but does not define the proper behavior for a device if an unsupported power level is requested. All of the hardware-driver combinations of which I am aware round to a supported level without returning an error code.

One of the most basic requirements for a platform for experimental MAC design is the ability to send data frames exactly when and how the user wishes. There are several ways in which normal driver/hardware setups fall short:

- Not exposing information needed by experimental MAC protocols.
- Not offering a sufficient control interface for the physical parameters of interest.
- Imposing unwanted aspects of an existing protocol (e.g. IEEE 802.11).

We addressed the first two with modest driver modifications. The chipset in the WiFi cards offers control over all the **IEEE 802.11 a, b, and g** PHY parameters on a per-frame basis, although channel changes cannot be made that quickly. The phased array antenna driver was originally coupled to the IEEE 802.11 protocol, but the two were fairly easy to separate. Harder than controlling **how** frames are sent is controlling **when**. Sections C.2.5 and C.2.6 discuss our approach to the timing problems in more detail.

There are several important aspects of the IEEE 802.11 protocol which tend to be implemented in hardware, making it challenging to use that hardware to explore significantly different protocols. In our WiFi chipset, these include MAC-layer retries and acknowledgements, carrier sense multiple access collision avoidance (CSMA/CA) back-off, and frame checksums. The rationale for implementing these functions in hardware is presumably speed: The turn-around time for raising an interrupt, sending information from an expansion card to the processor, waiting for the kernel to handle the interrupt and so on can be significant. One study found that doing acknowledgements in software took over 150 microseconds while the hardware implementation took less than 10 microseconds [Neufeld 05]. Such hardware-implemented features need to be either disabled or tolerated. Retries turn out to be easily disabled: There is a flag in the frame descriptor (`HAL_TXDESC_NOACK`) that causes the hardware not to wait for an acknowledgement after transmitting a frame. The frame checksums, and a few other mandatory header bits, we just accept. They are at worst overhead: The receiver can be configured to pass frames up the stack even if they are

not addressed to that device or fail the hardware checksum test, so experimental protocols are not constrained to obey the semantics of those mandatory fields, only to fill them with values that the hardware will accept.

#### C.2.4 Implementing Non-CSMA MACs

Suppressing CSMA/CA is critical for exploring non-contention-based MACs. In a few scenarios, such as a time division multiplexing (TDM) MAC with no outside noise sources, the medium should always be free whenever any node senses it and so CSMA/CA is harmless. In others, especially any system with intentional spatial reuse, multiple nodes may legitimately be active at the same time.

We developed a series of driver modifications to control CSMA/CA-related functions in the Atheros AR5212 chip set. Unlike retry-less transmission, which is already used for various broadcast frames in IEEE 802.11, CSMA-less operation is not an intended function of WiFi hardware. Consequently, this behavior has to be specified indirectly, and the necessary steps are not part of the documented public interface to the hardware<sup>†</sup>. Our group, with help from the broader Free Software community, reverse-engineered a procedure for practically disabling (and re-enabling) clear-channel assessment (CCA) in the cards we are using. Credit for analyzing closed-source driver behavior to identify registers touched during normal operation is due to the members of the `madwifi-devel` and `ath5k-devel` mailing lists.

Our patch to the MADWiFi driver changes three main parameters in the AR5212 chip. They seem somewhat redundant, but empirically the desired behavior is not always achieved without all three:

- **Diagnostic/Debugging Mode:** Set ignore bits for the Network Allocation Vector (NAV) in overheard packets, and physical carrier sensing.

---

<sup>†</sup> As of 29 November 2008, Atheros Corporation has released the source code to their Hardware Abstraction Layer and announced that the free Linux drivers will be their public reference platform. This is likely to increase the publicly-available documentation significantly.



- **Inter-Frame Spacing:** Configure the card to use the smallest possible durations for the gaps between frame transmission used in IEEE 802.11. If carrier-sensing is not being performed, these introduce pointless delay.
- **Disable Queue Backoff:** Prevent the card from backing off after draining a single hardware queue if there are other hardware queues with packets.

A patch adding this CCA control to MADWiFi is publicly available as part of our Commodity Atheros Research Platform (CARP) project<sup>‡</sup>.

#### C.2.4.1 Evaluation

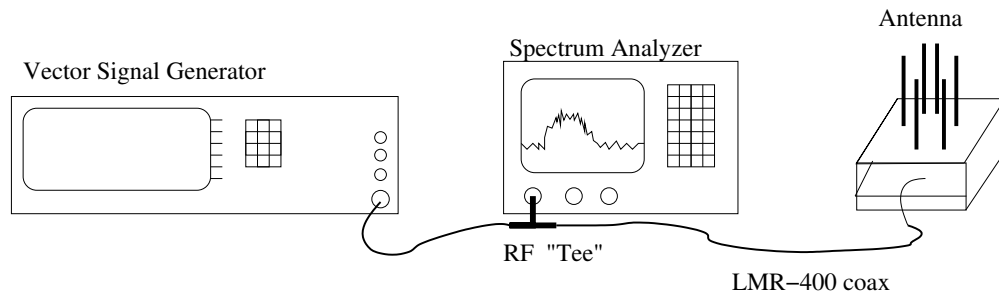


Figure C.4: CSMA/CA Evaluation Apparatus

To verify that CCA has effectively been disabled, the WiFi card in the phased array antenna node is disconnected from the antenna and connected to the test equipment shown in Figure C.4. The embedded computer is configured to produce a continuous stream of packets, and the vector signal generator (VSG) is used to create a competing signal on the same channel. The spectrum analyzer is used to determine whether the expected packet transmissions from the computer are occurring.

The testing procedure is shown in Algorithm C.1. For each type of VSG signal, the experimenter verifies that packets are sent despite the interfering signal **only when CCA is disabled**. There is reason to believe that different mechanisms and thresholds are used for detect-

<sup>‡</sup> Available at <https://systems.cs.colorado.edu/projects/carp/>. Based on personal correspondence, we know this is being used by researchers at IIT Delhi, the Dublin Institute of Technology, Communications Research Centre Canada, the University of Wisconsin, the University of Pittsburgh, WINLAB at Rutgers, and Stony Brook University.

**Algorithm C.1:** CSMA/CA (CCA) testing procedure

```

forall VSG signal types do
  Configure vector signal generator
  Turn VSG RF output off
  for CCA in {on, off} do
    Set system CCA  $\leftarrow$  CCA
    Start computer sending packets
    forall power in -100 dBm to +10dBm do
      VSG RF output power  $\leftarrow$  power
      Check for IEEE 802.11-like signal and VSG signal on spectrum analyzer

```

ing different types of signals. In particular, IEEE standards define different power thresholds for deferring to signals recognized as valid PLCP headers and other “generic” signals. Further, a patent issued to Atheros describes their apparent approach to interference mitigation in more detail [Atheros Communications, Inc. 04]. The mechanism employs a general power measurement component and specific detectors for OFDM and CCK modulations. Additionally, signal detections which correlate with successful packet reception are treated differently than those which do not. To address all of these cases, we tested with the following signal types:

- Sine wave (carrier only)
- FM-modulated carrier
- Continuous (“unframed”) DSSS/CCK/DQPSK modulated carrier
- Continuous (“unframed”) OFDM/QAM-16 modulated carrier
- Framed complete packets: IEEE 802.11b 11 Mbps DSSS/CCK/DQPSK beacon frames
- Framed complete packets: IEEE 802.11g 54 Mbps OFDM/QAM-64 beacon frames

The last four were produced using Agilent Signal Studio and then replayed on the VSG.

In all cases, the system performs as expected. With CCA suppression activated, the test computer produces a steady stream of packets regardless of the background signal from the VSG. Without CCA suppression, two different effects are seen: the valid packet streams cause the test

computer to back off indefinitely and simple wave forms produce a more complex behavior. At low power levels, when the signal is initiated, the test computer stops sending packets for several seconds and then resumes. At high enough power levels, however, the test computer stops producing packets and does not resume. This behavior likely represents the “adaptive interference immunity control” described in the patent, whereby signal measurements which do not correlate with actual packet reception are identified as “false positives” and the threshold required to induce back off is adjusted. We did not identify the specific power thresholds or delay periods associated with this function.

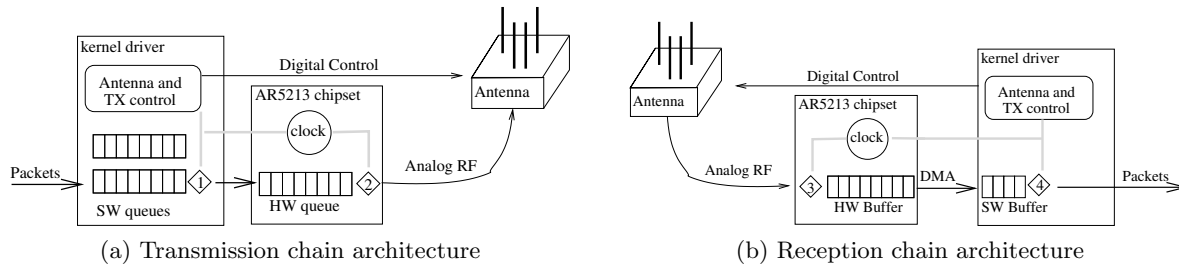
### C.2.5 Precise Timing Control

Precise timing is important for both efficient experimentation and a variety of MAC protocols. We are interested in both when packets are sent and when experimental antenna equipment changes state. We have developed infrastructure for quickly switching states in a coordinated manner across the entire system. There are two main challenges: (1) To interpret the results, it must be possible to match each packet sent or received to the antenna configuration in effect at the time. (2) To conduct experiments involving multiple nodes, it must be possible to synchronously change states so that the system state remains consistent.

We address both of these challenges by clocking our system off the high-resolution clock included in the adapter’s chipset. Most of the difficulty in connecting packets to antenna states comes from non-deterministic timing: On the sending side, the host can know when a packet is passed to the hardware (diamond 1 in Figure C.5a), but it cannot know exactly when the packet will leave the antenna, especially if the card performs CCA and CSMA/CA backoff. Similarly, there is a variable delay between when the packet passes through the receiving antenna and when the host’s interrupt handler is called to service the packet (diamond 4 in Figure C.5b).

While there is a large margin of error associated with the **system time** when the packet was actually sent, the **MAC time** at reception can be known much more precisely. The MAC time, used for calculating retransmission timeouts and back-offs, is maintained by a high-resolution

clock on the interface card. Packets are stamped by the hardware with the MAC time upon arrival (diamond 3), so there is almost no non-deterministic delay between the actual reception and the time-stamp. Since the AR5212 chipset also makes this time available, antenna transitions are scheduled relative to the MAC time.



### C.2.6 Time Synchronization

Using the on-chip timer helps with clock synchronization between nodes. MAC time synchronization is already required by the IEEE 802.11 protocol and is done in the interface hardware. In both BSS and IBSS (ad-hoc) modes, stations include their MAC time in beacon packets. Listening stations then set their own clocks off the beacons. Since this is done in the chipset (diamonds 2 and 3), the variability in delay is much lower – and thus the synchronization is much tighter – than what can be achieved using software on the end hosts.

## C.3 Administration and Maintenance Infrastructure

The previous sections have discussed challenges related to using commodity equipment as a research platform. This section focuses on generic challenges likely to face any distributed wireless testbed.

Operational and maintenance issues become increasingly important as the number of nodes, their geographic distribution, physical inaccessibility, and heterogeneity of network connections all increase. The next several sections will describe the design decisions and support infrastructure developed to make the testbed as useful as possible. In 2004, our experimental procedure consisted of an operator with a laptop controlling each physical node, and human-layer signaling with cell

phones or FRS radios. Experimental equipment was pre-configured in the laboratory before being transported to the test sites. Experiments were controlled and monitored by the operators, and results were downloaded onto the local laptops for later analysis. The subsequent testbed design has been driven by the need to address problems with that approach.

### C.3.1 Centralization

The simple approach described above might be sufficient for small experiments if everything worked as intended. However, experimental hardware and software is almost inevitably flawed, and faults which escape notice during testing regularly cause problems during live experiments. When problems do occur, equipment needs to be rebooted, experiments need to be re-started, scripts need to be edited, and sometimes new software needs to be installed.

The (human) communication overhead of trying to identify and correct problems across all test locations quickly becomes prohibitive, even when the necessary fixes are small. In early tests we found that even when nothing went wrong, coordinating a four node experiment required at least a half-hour of overhead for setup, configuration checks, synchronization, starting the experiment, downloading the data afterwards, and running basic sanity checks on the data. Overall, the ratio of time expended to successful experiment time was very high.

Our primary requirement for the testbed infrastructure was that it enable centralized management. In particular, it is necessary at a minimum to be able to perform the following tasks, for all of the experimental nodes, from a single location:

- Configure, start, and stop experiments
- Gather and analyze data
- Replace experimental software
- Reboot crashed equipment

Additionally, it is not strictly necessary but very useful to be able to:

- Monitor the progress of experiments
- Actively identify crashed or mis-configured nodes
- Replace all system software

Our testbed infrastructure is designed to provide these capabilities. At its core, this infrastructure consists of a control plane network, a “management box” connected to each experimental antenna unit, and a collection of software tools. All of these will be described in detail in upcoming sections.

### C.3.2 Management System

Every experimental antenna unit is directly connected to a management box, depicted in Figure C.5. Each box contains a flexible single-board computer (SBC) along with hardware required for remote power control. These serve multiple purposes, the most basic of which is connecting the research equipment into the control plane network. The phased array antenna systems have built-in Ethernet, but the management boxes provide a number of critical services which are not possible without them.

The management boxes contain flexible general-purpose computers, and can be installed indoors at a significant distance from the antenna unit. All of the currently-deployed boxes have Ethernet connections (though not to the same Ethernet), but they can accommodate other data connections with minimal configuration changes. WiFi has been verified – as long as the box is sufficiently separated in frequency or space from experiments – and there are no apparent barriers to using 900 MHz radio modems, GPRS/EVDO cellular devices, or more esoteric connections.

Besides providing network connectivity, the management boxes also provide network booting to the antenna units. This approach greatly simplifies reconfiguration: Any software change, from one configuration file to a new operating system, can be made by uploading a new image to the management system and rebooting the experimental equipment. The equipment could boot from a remote network server, but only if the network to which they were attached had both the

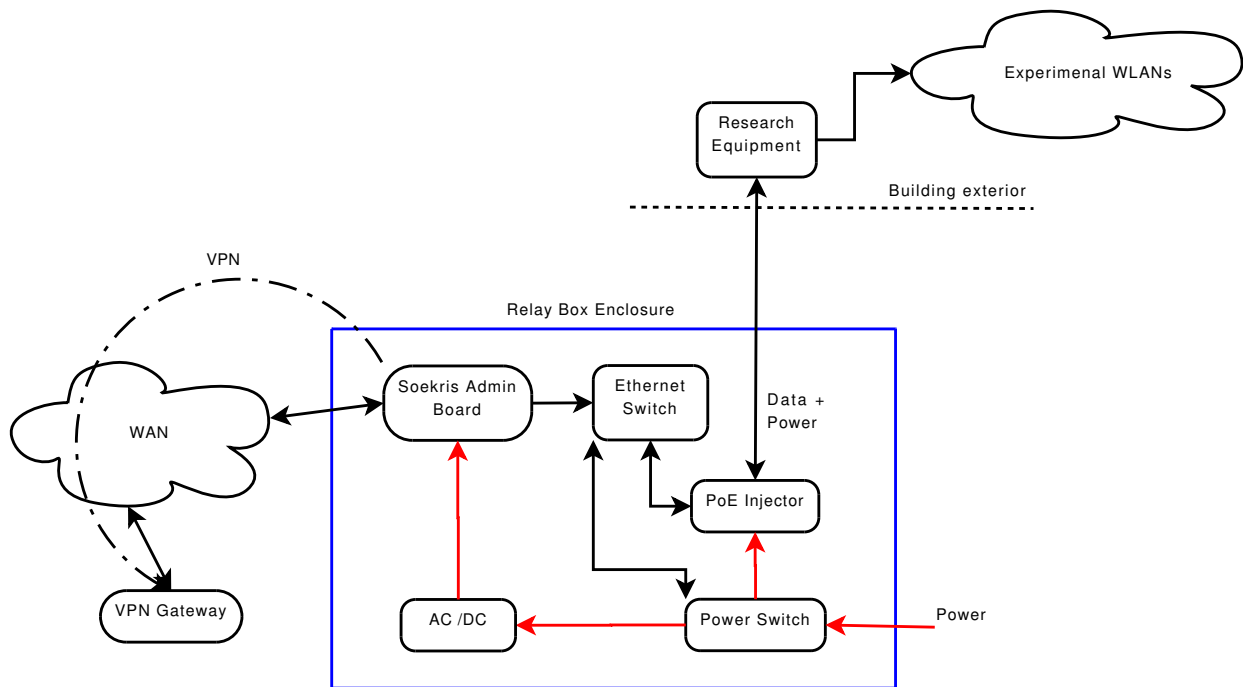


Figure C.5: Management box configuration

configuration and performance to support it, which would limit options substantially.

### C.3.3 Infrastructure Configuration

We designed the infrastructure with the goal of having as few “moving parts” as possible, because configuration errors are easy to introduce and can be very difficult to remedy once equipment has been deployed. To minimize opportunities for error, much of the system configuration is fixed, both between nodes and over time on any given node. We tried to identify the unavoidable sources of variability and isolate them so that as little of the overall system as possible has the potential to handle it incorrectly. The **unavoidable variability** comes from the address and configuration available on the outside (Internet) network link, the need to distinguish between units, new software images for the experimental equipment, and the passage of time.

The computer in each management box has one inward-facing network interface, a range of software processes, and one or more outward-facing interfaces. Except for time, which is rather pervasive, and the boot image, which is limited to one file served up by the TFTP daemon, the variability can be localized to the software directly interacting with the outward-facing network interface. The network configuration for the inward-facing interface and the devices on the internal network (the network-controlled power switch and the research equipment boot loader) is hard-coded and identical between units.

The organization of the control plane network relies heavily on the use of a virtual private network (VPN) and network address translation (NAT). On each management computer, the external IP address, DNS servers, and default routes are automatically configured by DHCP. Those are the only aspects that need to “know” anything about the network to which the box is attached. DNS is used to locate our VPN server, although the current IP address is also configured as a fall-back. Every management system is loaded with a different private key and X.509 certificate for connecting to the VPN, and this is the **only** hard state difference between boxes. The VPN daemon on the management board attempts to connect to the server on boot, or if it becomes disconnected for any reason.



### C.3.4 Reliability and Availability

A key characteristic of a large testbed is that physical access to the equipment is likely to be difficult and time-consuming. In our testbed, the experimental equipment is mounted on rooftops and in several cases requires a ladder or safety equipment for access. At night or in inclement weather, on-location maintenance is effectively limited to swapping out the entire unit. The management boxes are indoors, but access is often difficult for administrative reasons, and is inconvenient under the best circumstances.

Availability is generally defined as  $A = \frac{MTBF}{MTBF+MTTR}$ , where  $MTBF$  and  $MTTR$  are mean time between failures and mean time to repair, respectively. Many of the design and configuration decisions described in sections C.3.2 and C.3.3 are intended to avoid failures, but the primary goal is to minimize the set of failures which require on-site physical intervention to repair, should they occur. A secondary goal is to make such intervention as quick and simple as possible.

### C.3.5 Remote Repair

The most common significant failure in our testbed is a kernel hang in one of the phased array antenna units. A large portion of our experimental code has to run in kernel space, either for performance reasons or because it is an integral part of a device driver. The IXP425 platform includes a watchdog timer, and it is enabled, but some errors (especially acquiring locks and failing to release them) render the kernel effectively useless while still allowing the watchdog process to keep resetting the timer. Additionally, this platform has a limitation that the soft reset instruction resets the CPU but does not always reset the peripherals correctly, meaning that the device can reboot directly into a bad state.

We address this by including a network-controlled power switch in the management box. The experimental equipment and management computer are on separate switched circuits, and either can be turned off or power-cycled remotely using this switch. A limitation of this design is that the switch is only reachable if the computer is forwarding packets, so it cannot be used to address

a hung management system.<sup>§</sup>

Another possible failure is corruption of the operating system on the experimental systems. This could easily result from either a kernel error, an intentional upgrade that proved to be faulty, an interrupted upgrade, or other circumstances. We considered several possibilities involving fail-safe operating system images and similar approaches, but always booting from the network sidesteps the entire issue: Nothing important is installed or stored on the experimental system except for the boot loader. As long as that remains intact, it is always possible to restore or replace the system software by simply rebooting.

### C.3.6 Interchangeable Parts

On-site repairs, besides being time consuming, take place in less-than-ideal environments. It can be loud, windy, cold, hot, vibrating, high off the ground, or otherwise physically awkward. The person making the repair has far fewer resources than would be available in the lab. Consequently, it is beneficial to make the repair process as simple as possible, and especially to avoid the need for on-site configuration and testing as part of the repair process.

This was a significant reason for the fixed-and-uniform configuration approach described in section C.3.3. Every phase array antenna unit or network power switch has exactly the same hardware and configuration as every other. Every management computer is the same as every other except for the contents of a removable compact flash card. This makes it easier to develop testing processes for each component and means that a faulty or suspect component can be replaced with no thinking or configuration required. In fact, it is often easiest to replace the entire management unit as a whole – except for the flash card – and then diagnose the faulty one in the comfort of the lab.

---

<sup>§</sup> A previous version of the management box design used a power switch which was **itself** prone to hanging, a situation with little hope for remote repair.

### C.3.7 Security

Since WART nodes are connected to untrusted networks, they are potentially susceptible to the same attacks that many other machines on the University of Colorado network experience on a day-to-day basis. Several steps have been taken to ensure that only authorized access is given to both the phased array antenna node and management board.

First, communication to the WART management nodes is restricted to nodes that are part of the same VPN. This requires having a certificate signed by the certificate authority, a process which is performed off-line. Once this trust has been established, we utilize SSH keys to allow remote logins directly to the phased array antenna nodes.

It is important to note that this last security stage is not without its weaknesses. This is due to the fact that the phased array antenna nodes run off a ramdisk and are thus without any real permanent storage. This forces each node to regenerate their SSH keys upon every reboot. This makes the nodes susceptible to man-in-the-middle attacks should an attacker obtain access to the VPN via a trusted certificate. One possible remedy to this challenge could be to embed the SSH keys directly into the OS image, which would allow anyone with an OS image to impersonate any antenna node, but would still be an improvement.

Another possible attack could stem from the wireless interface side. Should an attacker associate with a node, the node could potentially begin routing packets from unauthorized users. For now, we have disabled all routing services, but this remains a risk for future multi-hop experiments.

## C.4 Deployment Logistics

Deploying a physically large testbed, especially with outdoor equipment, involves a number of challenges outside the traditional realm of computer science. There is a modest inherent engineering component that is significantly compounded by the need for approval and cooperation from various outside parties. All of the WART nodes are located on University of Colorado property, meaning that we only had to interact with a single owner, but it is a very large and bureaucratic one.

We suspect that broadly similar issues would be likely to arise in working with another large organization, and possibly with multiple smaller ones.

In practice, deploying and operating equipment indoors in laboratory and office spaces has required only the informal approval of the research groups using that space. There may be relevant building codes or university policies, but there is no enforced approval process. However, equipment installed on the outside of buildings, or visible from the outside, requires the involvement of the campus-wide organizations responsible for all construction projects. Fundamentally, there seems to be no administrative category for a project which spans a large area but with very minimal requirements. Building an outdoor testbed therefore becomes a university construction project with all of the overhead that entails.

Some of the more prominent logistical challenges encountered were:

- **Architectural Approval:** The aesthetic impact on campus buildings had to be approved by the campus architect.
- **Antenna Siting and RF Interference Approval:** A separate antenna committee had to be convinced that the proposed sites would not interfere with existing radio equipment.
- **Electrical Design and Installation:** The electrical requirements of the testbed equipment are extremely low; each node uses less power than a desk lamp. However, all construction projects involving new electrical connections are subject to the same approval process, regardless of the actual load. This means that an electrical design for each node had to be completed and signed off by a certified electrical engineer, and installation of the electrical components had to be performed by licensed electricians. Both had to be done by outside contractors hired through the office of facilities management, requiring an additional round of financial approvals before work could begin. Additionally, the waterproof plastic enclosures we had designed and fabricated for the management boxes had to be scrapped and replaced with metal enclosures specifically rated for containing electrical equipment.
- **Environmental Health and Safety:** All construction projects have to be audited for

safety risks to both the workers and the campus in general. The primary concern was pre-existing asbestos building materials, although we also had to vouch for the microwave radiation levels.

- **Roof Integrity:** Because the equipment was to be mounted on the outside of buildings, both the attachment methods and cable connections had to be evaluated for waterproofing, fire sealing, and structural impact. In the cases where new holes had to be made through the roof, the penetration and waterproofing had to be installed by campus roofing services.
- **Antenna Structure:** Local building codes and campus design rules establish standards for wind, snow, and ice tolerance. The university requirements were the more stringent in this case, requiring that equipment be designed for 120 mile per hour (53.6 m/s) wind load. Antenna mounting equipment, especially in the WiFi market, seldom meets those requirements. While commercial options do exist, we found it more cost-effective to design and construct our own.
- **Financial Approvals:** After our research group and department decided to allocate funds for the testbed, there were still a significant number of delays waiting for work orders and payments to be approved by other university entities. In particular, payments from the computer science department to facilities management, and from facilities management to outside contractors all required administrative approval before the payee could begin work.

#### C.4.1 Timeline

The testbed deployment process has required a total of two years. Most of that time has consisted of waiting for some necessary action by parties outside our department. Within that waiting, most of the time has been for administrative approvals, with actual design and construction requiring relatively little. Table C.1 shows our actual timeline; with more foresight it probably could have been compressed.

The architectural and RF approval steps are an unavoidable bottleneck, as they determine

whether and where equipment can be installed. In our case, it required approximately nine months from the first informal proposals to a preliminary approval of the sites chosen. Once those decisions had been made, several of the remaining steps could likely have proceeded at once.

The obvious deployment tasks, namely physically installing the antenna node and management box, and running conduit and Ethernet cable between them, required on the order of one week per node.

<b>Date</b>	<b>Task</b>
12/2006	Initial talks with campus architect, campus network admin., and facilities management
01/2007	Initial proposal to campus architect Preliminary approval from campus network admin.
05/2007	Preliminary approval from campus architect
08/2007	Preliminary approval from facilities management
09/2007	Environmental health and safety approval
04/2008	Electrical plans completed Begin wired control plane install
05/2008	First WART node installation
06/2008	Electrical installation done
08/2008	Wired control plane done
11/2008	All WART nodes operational

Table C.1: Deployment Timeline

#### C.4.2 Costs

Table C.2 presents an approximate breakdown of the expense incurred **per node** in building this testbed. The dominant cost is not the research equipment itself but rather labor required for regulatory and university policy compliance. This includes both the electrical work mentioned earlier and the time spent by university employees on evaluation and project oversight.

### C.5 Proof-of-Concept Experiments

As a proof-of-concept experiment for WART, we performed a full pairwise link quality test. In this test, each WART node takes a turn transmitting while the other nodes listen. During each turn, the transmitter and all receivers cycle through 17 pre-configured antenna patterns,

<b>Description</b>	<b>Cost</b>
Phased Array Antenna Node	\$3,000
Management Box and Other Control Plane Equipment	\$1,200
Installment Materials	\$300
External Labor and Fees	\$5,780

Table C.2: Cost of labor and parts per WART node. The labor of research group members is not considered.

so that every combination of transmitter and receiver antenna patterns is tested. The patterns chosen point the main lobe in one of 16 directions about the azimuth plane (the 17th pattern is omnidirectional). Using the measured signal strength of received packets, we are able to determine (a) which links are possible between which nodes and (b) what the optimal “greedy” patterns are for each link. The results of this experiment are provided visually in Figure C.6, which we believe makes a compelling case for the power of steerable directional antennas. When configured with omnidirectional patterns, which are comparable to the antennas used in many single-radio mesh networks, only a few links are even possible, and of those only a small number offer decent signal quality. With steering, however, we see a vast improvement: not only are all link-pairs able to pass traffic, but these links are typically of high quality (greater than -70 dBm).

Our present and future research utilizes WART to evaluate directional medium access control (MAC) protocols, with a particular emphasis on optimization for spatial re-use. We believe that the unique opportunity that WART provides for real-world evaluation of these protocols will lead to important results in this direction, and new insights into methods for improving wireless systems in general.

## C.6 Related Work

In this section we will give a high level overview of other wireless testbeds, both indoor and outdoor, and discuss how they compare to CU-WART.

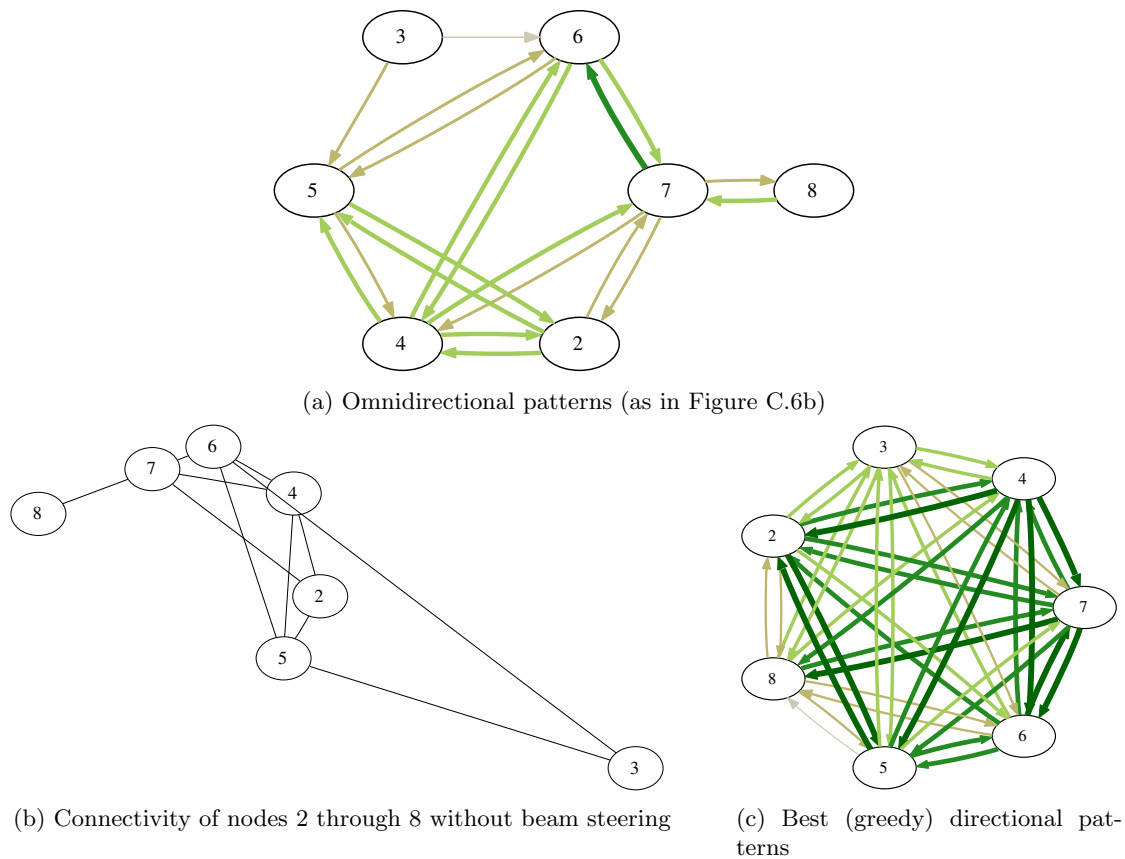


Figure C.6: Comparison of available links and link quality between seven testbed nodes using best-steered directional patterns and omnidirectional patterns. Stronger links are indicated with a wider arrow of a darker color. The best links are those with a link of greater than  $-60$  dBm. The worst links plotted are barely above the noise-floor with greater than  $-95$  dBm achieved RSS.



### C.6.1 Outdoor Wireless Testbeds

The existing outdoor testbeds generally have more operational emphasis and less experimental control and management support than WART or the indoor testbeds. Most use stock IEEE 802.11 at the MAC and physical layers, although additional low-layer information is gathered to inform higher-layer research. This may in part reflect their designers' research interests, and may also reflect limitations resulting from the lack of a stable separate control network.

**Roofnet:** Roofnet is probably the first distributed testbed for IEEE 802.11 mesh networking [Aguayo 04b, Bicket 05]. It consists of 20-40 nodes mounted on the rooftops of mostly residential buildings in Cambridge, MA. The entire network spans over an area of 1.5 x 1.5 kilometers. Unlike WART, Roofnet is unable to experiment using IEEE 802.11g modulation schemes, and is restricted to experiments involving omni-directional beam patterns. Roofnet is also a dual-purpose network; in addition to being a research testbed it also acts as a multi-hop backbone that provides Internet access. In contrast, WART is a dedicated experimental platform.

**Rice/TFA Mesh:** In terms of practical challenges, the RICE/TFA mesh is the most similar to our testbed. The physical size is similar: 2.12 km diameter for TFA, 2.36 km for WART. TFA has 14 nodes<sup>¶</sup>, WART has 7. The TFA-Rice mesh appears to involve equipment located on property with a variety of owners, suggesting similar access difficulties. There is little published information about the design and operation of the network, but it seems likely that their project and ours face similar issues. The deployment approach – in terms of choosing sites, not the logistics – is described in [Camp 06]. There are two primary differences: First, WART is focussed on experimental techniques and equipment at the physical layer, while the TFA mesh is not designed for experimentation at this layer. Second, the TFA mesh has a large operational component, while WART is purely experimental.

---

<sup>¶</sup> Based on TFA public wiki as of 13 December, 2008.

**Mesh at Purdue (MAP):** The MAP network is a primarily indoor research network which uses several fixed directional antennas for point-to-point links between adjacent buildings [Das 06]. There are two approximately 20 meter links and two approximately 60 meter links.

**RuralNet / Digital Gangetic Plains (DGP):** The RuralNet deployment is an experiment in using IEEE 802.11 equipment for very long range point-to-point communication [Raman 07]. The operators use fixed directional antennas on traditional radio towers and buildings to form multi-kilometer links.

**Ad-hoc Protocol Evaluation testbed (APE):** The basic design of the APE project is for humans to carry laptops that are pre-loaded with scripts to control the experiments. Node placement and mobility are controlled by “monkey walks” – human operators following directions displayed on the laptops. The APE software packages include modifications to their wireless network interface cards to collect signal strength information for all received packets [Lundgren 02, Nordstrom 05]. (This information is available as part of the Prism or Radiotap headers reported by many wireless NIC drivers).

### C.6.2 Indoor Wireless Testbeds

There are a large number of indoor wireless testbeds, emphasizing a variety of technologies and design objectives. In general, the indoor testbeds are more compact (dense) than the outdoor testbeds. They also benefit from a much more controlled environment: the problems of remote repair and establishing and maintaining a reliable communication infrastructure, which have been at the forefront of our design challenges, are largely non-issues.

Many of the indoor testbeds have at least an order of magnitude more nodes than any of the outdoor ones: There are 400 nodes in the ORBIT testbed, over 400 (both wired and wireless) in Emulab, and 210 in Kansei [Raychaudhuri 05b, White 02, Johnson 06, Ertin 06]. Much of the infrastructure developed for the indoor testbeds is oriented toward automating the process of con-

figuring, controlling, and aggregating data from such a large collection of devices.

**ORBIT:** The ORBIT testbed consists of a “main grid” of 400 nodes arranged in a 20 by 20 grid, and several smaller “sandboxes” for development and testing [Raychaudhuri 05b]. The nodes consist of single-board computers with IEEE 802.11 NICs and omni-directional antennas. The experimenter can install arbitrary software on the nodes, but there are standard operating system images which include a specialized measurement and control framework [Raychaudhuri 05a, Ott 05].

**Emulab wireless extensions:** Emulab is a well-known testbed for emulating arbitrary wired network topologies. It uses a variety of resource allocation and virtualization mechanisms to support many concurrent – but isolated – experiments [White 02]. Emulab has recently been extended to include several classes of nodes with wireless networking capabilities: Rack-mounted PCs with WiFi radios, PCs with GNU Software Radio and Ettus Research USRP hardware, Mica2 sensor motes, and mobile robots [Johnson 06]. The non-mobile nodes operate very similarly to the wired-only Emulab nodes, with a dedicated Ethernet control plane, while the robots have significant mobility-specific support infrastructure. The mobile-node tracking and control infrastructure is conceptually similar to that described in [De 06]. Most Emulab nodes allow the user to install arbitrary code, down to the OS level. Because the mobile nodes do not have an out-of-band control and reprogramming mechanism, users are significantly constrained to avoid breaking the necessary on-board infrastructure. The mobile nodes do have attached Mica2 motes, over which the user has complete control.

**UCR Testbed:** The UCR testbed consists of single-board computer with stock IEEE 802.11 NICs spread throughout several floors of a single office building. The devices are powered via power-over-Ethernet from a set of PoE-enabled switches, providing a simple interface for power-cycling nodes [Broustis 07]. Although not mentioned in the paper, the project web site indicates that they have added several PCs with USRP hardware to the testbed.

**Hydra:** Hydra is an indoor testbed for SDR experimentation. The physical layer is implemented with GNU software radio and USRP hardware, while higher layers are implemented with Click. The design work seems to be focussed on the prototyping platform, not the testbed aspect facility [Mandke 07]. No information is given about the size or infrastructure of the testbed.

**TRNC/ESPAR:** TRNC/ESPAR is a hardware platform for evaluating directional MAC protocols using Electronically Steerable Parasitic Array Radiator (ESPAR) antennas [Mitsuhashi 07, Kohmura 08]. The authors refer to the system as a testbed, but it is in the sense of prototyping equipment, not a specific facility.

**UCLA UnWiReD Laboratory:** The UnWiReD testbed is a two-node facility for physical-layer experimentation with MIMO systems. The testbed is distinctive in that it provides a very flexible SDR platform for four-way MIMO at both the transmitter and receiver, and includes remotely-controlled mechanical actuators to adjust the antenna positions [Zhu 05].

**Miniaturized Network Testbed: MiNT:** MiNT is an effort to simulate wireless networks with mobility using as little space as possible. It is conceptually very similar to the mobile nodes in Emulab, although developed independently [Johnson 06]. Nodes have multiple wireless interfaces for various purposes; the ones used for the protocol under test are highly attenuated to simulate the loss of much larger areas. Additionally, the MiNT platform integrates with ns-2 to provide a hybrid simulation/emulation environment [De 05]. In the initial version, the mobile nodes were simple antenna platforms connected by RF cables to PCs where the actual processing took place. The MiNT-m paper describes improvements to dispense with the stationary PCs, along with additional management tools. The testbed infrastructure consists mainly of mechanisms for node tracking, positioning, control, state logging, and state rollback [De 06].

**Kansei:** Kansei is another testbed aimed at using high density to emulate a large system in a small area [Ertin 06]. The system consists of 210 Stargate SBCs arranged in a grid with a wired control interface.

**MeshTest:** MeshTest uses standard PCs and an emulated RF environment. Each computer is connected into an RF matrix switch, allowing for programmable attenuation between nodes [Walker 08]. This provides significant flexibility in a very small physical size, although it entails some loss of fidelity. The infrastructure consists of the RF switch, the ORBIT software tools, and a custom-developed application for configuring the switch.

**EWANT: Emulated Wireless Ad-hoc Network Testbed:** EWANT uses standard PCs and a partially-emulated RF environment. Each PC is connected to one or more antennas through a combination of fixed attenuator and an RF multiplexer. The antennas are all positioned within small area, adding giving some measure of propagation realism [Sanghani 03].

## C.7 Conclusion

This chapter has presented WART, a testbed that will facilitate future networking research by providing unique physical layer capabilities not seen in any other outdoor networking testbed. While the testbed covers an entire university campus, it is easy to manage and administer due to its wired control plane, which is remotely accessible from anywhere on the Internet.

The research motivation for building WART was to study the use of directional, steerable, and adaptive antennas. The prominent issues encountered in creating the testbed proved to be only indirectly related to that objective. The direct causes were **using commodity equipment**, **supporting low-level experimentation**, and **spanning a large geographical area**.

**Commodity equipment:** The research equipment (phased array antenna nodes) is comparatively affordable at \$3,000 per node, while specialized test and measurement equipment could easily cost 10 to 20 times more. The consequences of using commodity hardware have been the

need for significant calibration and testing and extensive software hacking to make the hardware operate in unintended ways.

**Low-level experimentation:** Many of the experiments we wish to conduct are low-level both in the sense of being at the physical and MAC layers of the OSI hierarchy, and in the sense of requiring “close to the metal” system implementation. This implies the need for easy reprogramming and crash recovery, high-volume data collection, and a flexible control interface. In practice, these in turn require a control connection that is separate from the experimental wireless system.

**Large geographical area:** It has been amply demonstrated that radio propagation in general, and directionality in particular, are very environmentally dependent [Anderson 09a]. Consequently, it was important that WART encompass a range of node densities and environmental features of interest. However, covering a large area implies **physical distance** and often **administrative diversity**, each of which contribute significant design challenges. Physical distance effectively precludes running dedicated cables from a central location to all of the nodes, which implies that power and network connectivity (if needed) must be supplied using resources available on site. It is this constraint which leads us to the “management box” design, with network support, power conversion, and power switching co-located with every measurement node. It is worth noting that a large testbed without the focus on low-level experiments may be able to dispense with the dedicated control plane and remote-reprogramming capabilities, significantly relaxing these requirements.

Covering a larger area often implies involving more administrative domains. Our sites are all owned by the same university, but building at a campus-wide scale requires the involvement of many departments – administrative and academic – and the approval of several levels of hierarchy. The practical impact of this cannot be overstated. The approval processes – and the cascade of design decisions made in order to secure those approvals – account for at least half of the total time and cost for this project.

This testbed was developed to study particular physical layer technologies, but the design lessons are not specific to that objective. Most of the challenges encountered in designing this

testbed – and the solutions developed – are likely to apply to other outdoor and wide-area testbeds. We have developed an infrastructure for deploying nodes at widely separate, minimally provisioned sites and connecting them into an easily-managed unified research system.

## Appendix D

### Model Code

This chapter contains the AMPL source code for the optimization models presented in this dissertation. I include this because the code is comparatively compact and because any description of the algorithms is ultimately imprecise without it. No effort has been made to clean up this code for public consumption – it contains no-longer-used logic, debugging output, mis-named functions, and otherwise un-helpful or misleading material.

In general, the optimization model can be understood from the model files in Section D.1. Some conceptually important processing, however, does occur in the imperative files in Section D.2. In particular, the subgradient process and loop-termination conditions are largely defined in those files. The off-line execution processes are fully defined in AMPL — *e.g.* `compare-versions.ampl` in Listing D.15 is a top-level script. In the on-line process, the `AMPL` program interacts with other components by passing data through tables. Those components are written in a variety of programming languages, and are too large to include in this chapter.



## Listings

D.1	RMP.mod . . . . .	250
D.2	central-CLAP.mod . . . . .	251
D.3	convex-central-CLAP.mod . . . . .	252
D.4	global-params.mod . . . . .	257
D.5	ea_stdma.mod . . . . .	266
D.6	farp.mod . . . . .	278
D.7	guan-common.mod . . . . .	279
D.8	subgradient-common.mod . . . . .	280
D.9	trial-SINR-logic.mod . . . . .	289
D.10	post-schedule-SINR-opt.mod . . . . .	291
D.11	compute-m.mod . . . . .	294
D.12	master-daemon.ampl . . . . .	295
D.13	solver-daemon.ampl . . . . .	305
D.14	stdma_subproblem.ampl . . . . .	311
D.15	compare-versions.ampl . . . . .	332
D.16	solve_clap.ampl . . . . .	336

### D.1 Model Files

These model files are declarative: They define parameters, variables, constraints, objectives, options, and problems for AMPL.

## Listing D.1: RMP.mod

```

1 #####
2 # Restricted Master Problem:
3 # Assign time (continuous) to link sets
4 # Dual values of capacity constraints are inputs into the
5 # column generation phase (CLAP)
6 #####
7
8 ## Demand for service on each link
9 param q{(i,j) in Links} >= 0;
10
11 ##acg: LinkSets calls num_link_sets
12 param num_link_sets integer default 0;
13 set LinkSets = 1 .. num_link_sets ordered;
14
15 ## Record of link sets (analogous to SijL in STDMA1.py)
16 param link_in_set {1 in LinkSets, (i,j) in Links} default 0;
17
18 ##acg: mark reduced_cost done
19 param reduced_cost = 1-(sum{(i,j) in Links} (beta_t[i,j] * Sbar[i,j]));
20
21 ## x = assignment of time to each link set
22 var x {1 in LinkSets} >= 0 integer;
23
24 ## Minimize total time assignment
25 ##acg: mark RMP_obj done
26 minimize RMP_obj: sum{1 in LinkSets} x[1];
27
28 ## Require that allocation of time to link sets satisfies per-link demand
29 ##acg: mark demand_coverage done

```

```

30 subject to demand_coverage{(i,j) in Links}:
31     sum{l in LinkSets} (x[l] * link_in_set[l,i,j]) >= q[i,j];
32
33 param rc_thresh := -0.9;           #Maximum reduced cost to accept a proposed
        column

```

Listing D.2: central-CLAP.mod

```

1 var SBIN {i in Nodes, j in Nodes} integer >= 0 <= 1;
2 var VBIN {i in Nodes} binary;
3
4 ## Dissertation Eq. 4.4
5 maximize CLAP_OBJ: sum{(i,j) in Links} beta_t[i,j] * SBIN[i,j];
6
7 ## Dissertation Eq 4.5 is in common.mod as "duplex"
8 subject to CLAP_duplex {i in Nodes}:
9     sum{j in Nodes} SBIN[i,j] +
10    sum{j in Nodes} SBIN[j,i] <= 1;
11
12 ## Dissertation Eq. 4.6
13 subject to CLAP_SINR{(i,j) in Links}:
14 #part A
15     ((P[i]*D[i,j]*D[j,i]*SBIN[i,j]) / (Lb[i,j]*Nr) +
16     Gamma1*(1+M[i,j])*(1-SBIN[i,j])
17     -
18     #part B
19     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*D[k,j]*D[j,k]*VBIN
20     [k]) / (Lb[k,j]*Nr))
21     )) >= 0;
22 ## Dissertation Eq. 4.7 is in common.mod as "coupling"

```

```

23 subject to CLAP_coupling {i in Nodes}:
24     sum{j in Nodes: (i,j) in Links} SBIN[i,j] <= VBIN[i];
25
26
27
28 ## Dissertation Eq. 4.8
29 subject to CLAP_antenna_coupling {i in Nodes, k in Nodes}:
30     D[i,k] - sum{p in Pats} (pat_gain[i,k,p] * B[i,p]) = 0;
31
32 ## Dissertation Eq. 4.9 is in common.mod as "one_pat"
33 subject to CLAP_one_pat {j in Nodes}:
34     sum{p in Pats} B[j,p] = 1;
35
36 subject to CLAP_CLARITY {i in Nodes, j in Nodes: !(i,j) in Links}:
37     SBIN[i,j] = 0;
38
39
40 problem CLAP: SBIN, VBIN, D, B, CLAP_OBJ, CLAP_duplex, CLAP_SINR,
    CLAP_coupling, CLAP_antenna_coupling, CLAP_one_pat, CLAP_CLARITY;
41 #option solver knitroampl;
42 option solver ipopt;
43 option halt_on_ampl_error yes;

```

Listing D.3: convex-central-CLAP.mod

```

1 #Converified
2 param epsilon = 1e-30;
3 param duplex_epsilon = 0.5;
4
5 var logS {(i,j) in Links} >= -90 <= log(1);
6 #var logS {(i,j) in Links}== -90;

```

```

7 param inferredS {(i,j) in Links};
8
9 var derivedS {(i,j) in Links} integer;
10
11 # var logT {(i,j) in Links};
12 # param inferredT {(i,j) in Links};
13
14 var logD {i in Nodes, j in Nodes};
15 param inferredD {i in Nodes, j in Nodes};
16
17 #var logV {i in Nodes} >= -90 <= log(1);
18 #var logV {i in Nodes} == -90;
19 #param inferredV {i in Nodes};
20
21 #Nodes which participate in at least 1 link
22 set NodesUsed = {i in Nodes: (exists {j in Nodes} (i,j) in Links) or (exists {
      j in Nodes} (j,i) in Links)};
23
24 ## Equation 4.17
25 maximize CCLAP_OBJ:
26     log(sum{(i,j) in Links} (exp(logS[i,j] + log(beta_t[i,j] + epsilon))
      ));
27
28 param alpha=1.1;
29 subject to CCLAP_SINR_NEW {(i,j) in Links}:
30     log(exp(log(Nr)+log(Gamma1)+log(Lb[i,j])+alpha*logS[i,j]-log(P[i])-
      logD[j,i]-logD[i,j])) +
31     sum{(k_t, k_r) in Links: k_t != i or k_r != j}
32     exp(log(P[k_t])+logS[k_t,k_r]+log(Gamma1)+logD[j,k_t]+logD[k_t,j]+log(
      Lb[i,j]))+

```

```

33      (alpha)*logS[i,j]-log(P[i])-logD[j,i]-logD[i,j]-log(Lb[k_t,j]))
34      <= 0;
35
36 # ## Equation 4.15. The ugly one
37 # subject to CCLAP_SINR1 {(i,j) in Links}:
38 #      log(exp(logT[i,j] + log(Gamma1)) +
39 #      exp(-logD[i,j]-logD[j,i]-logS[i,j]+log(Nr+Gamma1)) +
40 #      sum{k in Nodes: k <> i and k <> j} exp(logV[k]-logD[i,j]-logD[j,i]+logD[
41 #      i,k]+logD[k,i]-logS[i,j]+log(P[k]*Gamma1))))
42      <= 0;
43
44 # ## Equation 4.16. The second half of the transformed SINR constraint
45 # subject to CCLAP_SINR2 {(i,j) in Links}:
46 #      log(exp(-logT[i,j]) * M[i,j]) <= 4;
47
48 ## Equation 4.18. S-V coupling, in the log domain
49 # subject to CCLAP_SV {(i,j) in Links}:
50 #      log(exp(logS[i,j] - logV[i])) <= 0;
51
52
53 ## Equation number MISSING. Transformed form of 4.5 — Duplex
54 subject to CCLAP_DUPLEX {i in NodesUsed}:
55      log(sum{j in Nodes: (i,j) in Links} exp(logS[i,j]) +
56      sum{j in Nodes: (j,i) in Links} exp(logS[j,i])) <= 0;
57
58
59 # Magic duplex mutex conditions: For any i -> j, k -> l conflicts if
60 # exactly one of the endpoints is shared. (If both are shared, it's
61 # the same link)

```

```

62 #
63 #           k-----\
64 #                   l
65 #   k----->l  i----->j  ----->l
66 #           \
67 #           ----->l
68 #
69
70 # subject to CCLAP_DUPLEX_PAIRWISE1 {i in Nodes, j in Nodes, k in Nodes:
71 #     (i,j) in Links and (i, k) in Links and k<>i and k <> j}:
72 #     log(exp(logS[i,j] + logS[i ,k]) + 1) <= duplex_epsilon;
73 # subject to CCLAP_DUPLEX_PAIRWISE2 {i in Nodes, j in Nodes, k in Nodes:
74 #     (i,j) in Links and (k, i) in Links and k<>i and k <> j}:
75 #     log(exp(logS[i,j] + logS[k ,i]) + 1) <= duplex_epsilon;
76 # subject to CCLAP_DUPLEX_PAIRWISE3 {i in Nodes, j in Nodes, k in Nodes:
77 #     (i,j) in Links and (j, k) in Links and k<>i and k <> j}:
78 #     log(exp(logS[i,j] + logS[j ,k]) + 1) <= duplex_epsilon;
79 # subject to CCLAP_DUPLEX_PAIRWISE4 {i in Nodes, j in Nodes, k in Nodes:
80 #     (i,j) in Links and (k, j) in Links and k<>i and k <> j}:
81 #     log(exp(logS[i,j] + logS[k ,j]) + 1) <= duplex_epsilon;
82
83 subject to CCLAP_DUPLEX_PAIRWISE {(i,j) in Links, (k,l) in Links:
84     (k == i or k==j or l==i or l==j) and (i != k or j != l)}:
85     log(exp(logS[i,j] + logS[k,l]) + 1) <= duplex_epsilon;
86
87
88 ## Equation 4.19. Antenna gain reality
89 ## XXX need clarity on whether pat_gain is real or log-scale
90 subject to CCLAP_REAL_GAIN{i in Nodes, k in Nodes}:
91     -logD[i,k] + sum{p in Pats}B[i,p]*pat_gain[i,k,p] == 0;

```

```

92
93 ## Equation 4.9 (That is, unmodified from original CLAP)— Antenna pattern
   choices sum to 1
94 subject to CCLAP_ONEPAT {j in Nodes}:
95     sum{p in Pats} B[j,p] == 1;
96
97 subject to CCLAP_INF_INT {(i,j) in Links}:
98     derivedS[i,j] == exp(logS[i,j]);
99
100 #Debugging
101 subject to DEBUG_FORCE01:
102     logS[0,1] >= 0;
103 subject to DEBUG_FORCE02:
104     logS[0,2] <= -20;
105 subject to DEBUG_FORCE13:
106     logS[1,3] <= -20;
107 subject to DEBUG_FORCE23:
108     logS[2,3] >= 0;
109
110
111 problem CCLAP: logS, logD, B, CCLAP_OBJ, CCLAP_SINR_NEW, CCLAP_DUPLEX,
   CCLAP_REAL_GAIN, CCLAP_ONEPAT, CCLAP_DUPLEX; #CCLAP_DUPLEX_PAIRWISE,
   DEBUG_FORCE01, DEBUG_FORCE02, DEBUG_FORCE13, DEBUG_FORCE23;
112 #CCLAP_DUPLEX_PAIRWISE1, CCLAP_DUPLEX_PAIRWISE2, CCLAP_DUPLEX_PAIRWISE3,
   CCLAP_DUPLEX_PAIRWISE4;
113 option auxfiles 'acfrsu';
114 option solver ipopt;
115 option halt_on_ampl_error yes;
116
117

```



```

118 #Stupid threshold
119 param almost_one = 0.5;           #XXX

```

Listing D.4: global-params.mod

```

1 option display_precision 6;
2 option display_eps 1e-99;
3 #####
4 ## FLAP and common
5 #####
6 set Nodes;
7
8 set Links within (Nodes cross Nodes);
9
10 ## Beta transpose — the duals cost of capacity requirement of each link at
       this MP iteration
11 param beta_t {(i,j) in Links};
12 param local_beta_t {(i,j) in Links};
13
14 ## Power of node i
15 param P {i in Nodes};
16
17 ## Directional gain (estimate) from node i toward node j
18 param Dbar {i in Nodes, j in Nodes};
19
20 ## Path loss from node i to j
21 param Lb {i in Nodes, j in Nodes};
22
23 ## Receiver noise in milliwatts
24 param Nr;
25

```

```

26 ## Gamma 1 — SINR threshold;
27 param Gamma1;
28
29 ## SINR flapping margin, linear: Value above actual threshold to aim for
30 ## in decomposed subproblems. XXX
31 param Gamma1deflap default Gamma1* 2.0; ##3 dB margin
32
33 ## Mij — not easily explained
34 param M {i in Nodes, j in Nodes};
35
36 ## Lbar (lambda) Lagrange multiplier estimate
37 param Lbar {(i,j) in Links};
38
39 ## S — activation of link ij
40 var S {i in Nodes, j in Nodes} >= 0 <= 1;
41
42 ## V — activation of node i
43 var V {i in Nodes} >= 0 <= 1;
44
45 # Objective and constraints are Equation 3.24, Proposal R952, p. 58
46 ##acg: mark flap_obj done
47 maximize flap_obj: sum {(i,j) in Links}(
48     local_beta_t[i,j] * S[i,j] +
49     Lbar[i,j]*(((P[i]*Dbar[i,j]*Dbar[j,i]*S[i,j]) / (Lb[i,j]*Nr) +
50     Gamma1deflap*(1+M[i,j])*(1-S[i,j])) -
51     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
52         Dbar[j,k]*V[k]))/(Lb[k,j]*Nr)))
53 ##acg: mark duplex done
54 subject to duplex {i in Nodes}:

```

```

55         sum{j in Nodes} S[i,j] +
56         sum{j in Nodes} S[j,i] <= 1;
57
58 subject to coupling {i in Nodes}:
59         sum{j in Nodes: (i,j) in Links} S[i,j] <= V[i];
60
61
62 # Estimate of S — activation of link ij
63 param Sbar {i in Nodes, j in Nodes};
64
65 # Estimate of V — activation of node i
66 param Vbar {i in Nodes};
67
68
69
70 #####
71 # QNA-FLAP — Quadratic Nonlinear Approximation FLAP
72 # Evaluating NA as an approach to minimize oscillation
73 # See Guan1995Nonlinear
74 #####
75 param LocalVbar{i in Nodes} default 0.5;
76
77
78 #See "working.mac" in dissertation directory, expression %09.
79 ##acg: mark flap_obj_slope done
80 param flap_obj_slope {(i,j) in Links} =
81     (
82     local_beta_t[i,j] +
83     Lbar[i,j]*(((P[i]*Dbar[i,j]*Dbar[j,i]) / (Lb[i,j]*Nr)) -
84     Gamma1deflap*(1+M[i,j])));

```

```

85
86
87 ##acg: QNA_b calls flap_obj_slope
88 ##acg: mark QNA_b done
89 #See b_{cm} in Guan1995Nonlinear, eq. (4.2) and sec. V.
90 param QNA_b {(i,j) in Links} = 0.85 * flap_obj_slope[i,j];
91
92 # See a_{cm} in Guan1995Nonlinear, eq. (4.2) and sec. V.
93 # See "working.mac", output of "solve(a*S[i,j]^2+b*S[i,j] = flap_obj(S), a);"
94 ##acg: QNA_a calls QNA_b
95 ##acg: mark QNA_a done
96
97 param QNA_a {(i,j) in Links} =
98     -(((Gamma1deflap * Lb[i,j] * Lbar[i,j] *M[i,j]
99     + Gamma1deflap * Lb[i,j] * Lbar[i,j]
100     +(QNA_b[i,j] -local_beta_t[i,j])*Lb[i,j])*Sbar[i,j]
101     -Gamma1deflap * Lb[i,j] * Lbar[i,j]*M[i,j])*Nr
102     + Gamma1deflap * Lb[i,j] * Lbar[i,j] *
103     sum{k in Nodes: k <> i and k <> j} (
104     (Dbar[j,k] * P[k] * LocalVbar[k] *Dbar[k,j])/(Lb[k,j]))
105     );
106 #     - P[i] * Dbar[i,j] * Lbar[i,j] * Sbar[i,j] * Dbar[i,j])/
107 #     (Lb[i,j]*(Sbar[i,j]**2)*Nr);
108
109 #Are the signs right? Who knows!
110 maximize qna_flap_obj: sum{(i,j) in Links}(
111     (-QNA_a[i,j])*(S[i,j]**2) +(QNA_b[i,j]-Lbar[i,j])*S[i,j]
112     );
113
114 param Other_Sbar{i in Nodes, j in Nodes};

```

```

115
116 ## XXX Belated realization: This _requires_ some coordination (read:
117 ## probably dualization) on the duplex constraints.
118
119 param do_single_flap binary default 0;
120 #####
121 # Single-node QNA-FLAP
122 #####
123
124 param flap_node in Nodes;
125 param TmpSbar{i in Nodes, j in Nodes};
126
127 maximize single_qna_flap_obj: sum{(i,j) in Links: i == flap_node}(
128     (-QNA_a[i,j])*(S[i,j]**2) +(QNA_b[i,j]-Lbar[i,j])*S[i,j]
129     );
130
131 #####
132 # Single-node FLAP
133 #####
134 # Objective and constraints are Equation 3.24, Proposal R952, p. 58
135 maximize single_flap_obj: sum {(i,j) in Links: i == flap_node}(
136     local_beta_t[i,j] * S[i,j] +
137     Lbar[i,j]*(((P[i]*Dbar[i,j]*Dbar[j,i]*S[i,j]) / (Lb[i,j]*Nr) +
138     Gamma1deflap*(1+M[i,j])*(1-S[i,j])) -
139     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
140     Dbar[j,k]*V[k]) / (Lb[k,j]*Nr))
141     ));
142
143 # Use Sbar values for every link other than the one(s) selected by flap_node

```

```

144 # No violations AT flap_node
145 subject to single_duplex_A {i in Nodes: i == flap_node}:
146     sum{j in Nodes} S[i,j] +
147     sum{j in Nodes} Sbar[j,i] <= 1;
148
149 # And no violation at target of link FROM flap_node
150 subject to single_duplex_B {(i,j) in Links: i == flap_node}:
151     S[i,j] + sum{k in Nodes} Sbar[k,j] <= 1;
152
153
154 ##acg: mark single_coupling done
155 subject to single_coupling {i in Nodes: i == flap_node}:
156     sum{j in Nodes: (i,j) in Links} S[i,j] <= V[i];
157
158 #####
159 # DUAL FLAP
160 #####
161
162 # Lagrange multiplier of duplex constraint
163 param mu{i in Nodes} default 0.5;
164
165 # Intuition: The sum of beta_t over all links through a node is the
166 # total value of using that node; all but one such link must be denied
167 param mu_initial_est{i in Nodes} := 1;
168 #     (sum{(xi,xj) in Links: xi==i or xj==i} local_beta_t[xi,xj] -
169 #     max{(xi,xj) in Links: xi==i or xj==i} local_beta_t[xi,xj])/2;
170
171
172 maximize dual_flap_obj: sum {(i,j) in Links}(
173     local_beta_t[i,j] * S[i,j] +

```

```

174     Lbar[i,j]*(((P[i]*Dbar[i,j]*Dbar[j,i]*S[i,j]) / (Lb[i,j]*Nr) +
175     Gamma1deflap*(1+M[i,j])*(1-S[i,j])) -
176     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
           Dbar[j,k]*V[k]) / (Lb[k,j]*Nr))
177     ))) -
178     sum {i in Nodes}(mu[i]*sum{(xi,xj) in Links: xi==i} S[xi,xj])
179     ;
180
181 maximize dual_qna_flap_obj: sum{(i,j) in Links}(
182     (-QNA_a[i,j])*(S[i,j]**2) +(QNA_b[i,j]-Lbar[i,j])*S[i,j]
183     ) -
184     sum {i in Nodes}(mu[i]*sum{(xi,xj) in Links: xi==i} S[xi,xj])
185     ;
186
187 ##acg: single_dual_qna_flap_obj calls QNA_a
188 ##acg: single_dual_qna_flap_obj calls QNA_b
189 ##acg: single_dual_qna_flap_obj calls mu
190 ##acg: mark single_dual_qna_flap_obj done
191 maximize single_dual_qna_flap_obj: sum{(i,j) in Links: i == flap_node}(
192     ((-QNA_a[i,j])*(S[i,j]**2) +(QNA_b[i,j]-Lbar[i,j])*S[i,j]) -
193     (0.5*mu[i] + 0.5*mu[j])*S[i,j]);
194
195 ##acg: mark no_self_loop done
196 subject to no_self_loop {i in Nodes}: S[i,i] = 0;
197
198 ##acg: mark no_extraneous_activation done
199 subject to no_extraneous_activation{i in Nodes, j in Nodes: (i,j) not in Links
           }: S[i,j] = 0;
200
201

```

```

202 ## 2nd-order penalty term for duplex constraint. See Guan2002New (I think)
203 ## p is the index of the duplex constraint (there is one per node,) i is the
      index of the subproblem (again, one per node)
204 param guan_v {p in Nodes, i in Nodes} =
205     ## Contribution of  $x[l!=i]$  to constraint  $c^{(p)}$ . 1 is  $d_p$ 
206     (sum{l in Nodes: l  $\diamond$  i} (Sbar[l,p] + Sbar[p,l])) -1
207 ;
208
209 # Uses Sbar, not variable S
210 param guan_wbar{p in Nodes} default 1e-18; ##Just for debugging
211 ##indexing by node relative to which duplex constraint is defined (p) and
212 ##node/subproblem at which the term is evaluated - BUT - there is no S, only
      Sbar, so it's the same everywhere
213 param sbar_guan_penalty_term{i in Nodes, p in Nodes} = (
214     guan_wbar[p] * (max(0,(
215         (Sbar[i,p]+(Sbar[p,i])) + guan_v[p,i] ## $c_i^{(p)} x_i - c_i$ 
          ^{(p)} is just 1
216     )))**2
217     );
218
219 param guan_w {n in Nodes} default 1e-10; ##1e1 ; ## 2nd order penalty term
      weight
220
221 #####
222 ## Only use our own links as variables
223 ## Explicitly use local copies of all variables
224 ## and constraints for distributed implementation
225
226 # Gain at a particular node — linear units
227 var local_D{other in Nodes} >=0;

```



```

228
229 param local_Sbar {i in Nodes, j in Nodes} default 0; #Activation
230 param local_Lbar {(i,j) in Links}; #Lambda
231 param local_Dbar {i in Nodes, j in Nodes}; #Gain
232 param local_mu_bar {n in Nodes} >= 0 default 0;
233
234 #See "working.mac" in dissertation directory, expression %09.
235 param local_flap_obj_slope {(i,j) in Links} =
236     (
237     local_beta_t [i,j] +
238     local_Lbar [i,j]*(((P[i]*local_Dbar [i,j]*local_Dbar [j,i]) / (Lb[i,j]*Nr
239         )) -
240         Gamma1deflap*(1+M[i,j])));
241
242 #See b_{cm} in Guan1995Nonlinear, eq. (4.2) and sec. V.
243 param local_QNA_b {(i,j) in Links} = 0.85 * local_flap_obj_slope [i,j];
244
245 # See a_{cm} in Guan1995Nonlinear, eq. (4.2) and sec. V.
246 # See "working.mac", output of "solve(a*S[i,j]^2+b*S[i,j] = flap_obj(S), a);"
247 param local_QNA_a {(i,j) in Links} =
248     -(((Gamma1deflap * Lb[i,j] * local_Lbar [i,j] *M[i,j]
249     + Gamma1deflap * Lb[i,j] * local_Lbar [i,j]
250     +(local_QNA_b [i,j] -local_beta_t [i,j])*Lb[i,j])*local_Sbar [i,j]
251     -Gamma1deflap * Lb[i,j] * local_Lbar [i,j]*M[i,j])*Nr
252     + Gamma1deflap * Lb[i,j] * local_Lbar [i,j] *
253     sum{k in Nodes: k <> i and k <> j} (
254     (Dbar [j,k] * P[k] * LocalVbar [k] *local_Dbar [k,j]) / (Lb[k,j]))
255     );
256 # - P[i] * Dbar [i,j] * Lbar [i,j] * Sbar [i,j] * Dbar [i,j]) /

```

```

257 #      (Lb[i,j]*(Sbar[i,j]**2)*Nr);
258
259
260
261 var local_S {j in Nodes: (flap_node, j) in Links} >= 0 <=1; ##binary, really

```

## Listing D.5: ea\_stdma.mod

```

1 model global-params.mod;
2 ## XXX double-negative! Is that an accident?
3 maximize single_dual_qna_surr_flap_obj: sum{(i,j) in Links: i == flap_node}{
4     ((-local_QNA_a[i,j])*(local_S[j]**2) +(local_QNA_b[i,j]-local_Lbar[i,j]
5         )*local_S[j]) -
6     (0.5*local_mu_bar[i] + 0.5*local_mu_bar[j])*local_S[j]
7     - sum {p in Nodes: (flap_node, p) in Links}{ ##indexing by node
8         relative to which duplex constraint is defined
9         guan_w[p] * (max(0,(
10            (local_S[p] + local_Sbar[p,i]) + guan_v[p,i]      ##c_i^(p) x_i
11            - c_i^(p) is just 1
12            )))**2
13        )});
14 subject to local_no_self_loop: local_S[flap_node] = 0;
15
16
17 subject to local_no_extraneous_activation{j in Nodes: (flap_node, j) not in
18     Links}: local_S[j] = 0;
19
20
21 model guan-common.mod;
22 model farp.mod;
23 #####

```

```

20 # SNARP — Single-Node Antenna Reconfiguration Problem
21 #####
22
23 # "Working copy" of D for each generation of SNARP
24 param temp_D {i in Nodes, j in Nodes};
25 var local_B {p in Pats} binary;
26
27 # The node for (at) which the subproblem is currently being solved
28 param snarp_node in Nodes;
29 param lbar_epsilon default 0; ###Tiny incentive to care about SINR even when
    the "real" price is zero
30
31 #BIZARRE CONSTANT FACTOR: 1.0e9 so solver knows its head from its ass
32 maximize snarp_obj: 1.0e20 * (sum{(i, j) in Links: i == snarp_node}(
33     ((local_Lbar[i, j] + lbar_epsilon)*local_Sbar[i, j]*(P[i]/(Lb[i, j]*Nr))*
34         local_D[j]*local_Dbar[j, i])
35     - (sum{(k, l) in Links: k <> i and l <> j and l <> i and k <> j}(
36         Gamma1deflap*local_Sbar[i, j]*(local_Lbar[k, l]+lbar_epsilon)*(P[i]/(Lb[
37             i, l]*Nr))
38         *local_D[l]*local_Dbar[l, i]
39     ))) +
40     sum{(i, j) in Links: j == snarp_node}(
41     ((local_Lbar[i, j] + lbar_epsilon)*local_Sbar[i, j]*(P[i]/(Lb[i, j]*Nr))*
42         local_Dbar[i, j]*local_D[i])
43     - (sum{(k, l) in Links: k <> i and l <> j and l <> i and k <> j}(
44         Gamma1deflap*local_Sbar[i, j]*(local_Lbar[i, j] + lbar_epsilon)*(P[k]/(
45             Lb[k, j]*Nr))
46         *local_D[k]*local_Dbar[k, j]
47     ))));

```

```

45 param snarp_component{ref in Nodes, other in Nodes} =
46     sum {(i,j) in Links} ( ## The link involving ref (if any)
47     sum {(k,l) in Links: k  $\diamond$  i and l  $\diamond$  i and k  $\diamond$  j and l  $\diamond$  j}( ##All
         other links
48     (if (ref = i and other = j) ## ref tx to other
49         then ((local_Lbar[i,j] + lbar_epsilon)*local_Sbar[i,j]*(P[i]/(Lb[i,j]
            )*Nr))*local_Dbar[j,i]) else (0)) +
50     (if (ref = j and other = i) ## other tx to ref
51         then ((local_Lbar[i,j] + lbar_epsilon)*local_Sbar[i,j]*(P[i]/(Lb[i,j]
            )*Nr))*local_Dbar[i,j]) else (0)) +
52     (if (ref = i and other = l) ## (ref -> j) interferes with (other ->
         l)
53         then (-(Gamma1deflap*(local_Lbar[k,l]+lbar_epsilon)*local_Sbar[i,j]
            )*(P[i]/(Lb[i,l]*Nr)))) else (0)) +
54     (if (ref = j and other = k) ## (other -> l) interferes with (i-> ref
         )
55         then (-(Gamma1deflap*(local_Lbar[i,j] + lbar_epsilon)*local_Sbar[i,j]
            )*(P[k]/(Lb[k,j]*Nr)))) else (0))
56     ));
57
58 #ref must be given
59 maximize alternate_snarp_obj: sum{other in Nodes} (snarp_component[snarp_node ,
        other] * local_D[other]);
60
61 ##acg: mark snarp_real_pats done
62 subject to snarp_real_pats {j in Nodes}:
63     local_D[j] - sum{p in Pats} (pat_gain[snarp_node ,j ,p] * local_B[p]) =
        0;
64
65 ## Force full power again.

```

```

66 ##acg: mark snarp_one_pat done
67 subject to snarp_one_pat:
68     sum{p in Pats} local_B[p] = 1;
69
70
71 param dbg_D {other in Nodes}; #mirrors local_D
72 param dbg_B {p in Pats};           #mirrors local_B
73
74 param snarp_real_pats_dbg {j in Nodes} =
75     dbg_D[j] - sum{p in Pats} (pat_gain[snarp_node ,j ,p] * dbg_B[p]); #must
76     = 0
77
78 param snarp_one_pat_dbg =
79     sum{p in Pats} dbg_B[p]; #must = 1
80
81 model subgradient-common.mod;
82
83 ##
84 ## Try #3: By confidence (HatS) threshold
85
86 param bs_ss default 0.25;           #binary search step size
87
88 param best_rc default 1;           #No improvement
89
90 param best_thresh default 1;
91
92 param act_threshold default 0.9;
93
94 param trial_S {(i,j) in Links} = if (local_Sbar[i,j] >= act_threshold) then 1
95     else 0;
96
97 model trial-SINR-logic.mod;
98
99 param this_node in Nodes;           #must be set prior to solving.

```

```

94 param tmp_trial_D {i in Nodes, j in Nodes}; #holding param for local_D
95
96 model post-schedule-SINR-opt.mod;
97
98 # Now, the program:
99
100
101 maximize dist_signal_margin: sum{(i,j) in Links} log(
102     ((P[i]*
103         (if (i==this_node) then local_D[j] else trial_D[i,j])*
104         (if (j==this_node) then local_D[i] else trial_D[j,i])*
105         trial_S[i,j]) / (Lb[i,j]*Nr) +
106     Gamma1*(1+M[i,j])*(1-trial_S[i,j])
107     -
108     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*
109     (if (k==this_node) then local_D[j] else trial_D[k,j])*
110     (if (j==this_node) then local_D[k] else trial_D[j,k])*trial_V[k])/(Lb[
111         k,j]*Nr))
112     ))
113 );
114
115 ## Debugging calculation: The bit inside the log() operation for every link
116 param d_s_m_dbg_nolog {(i,j) in Links}= ((P[i]*
117     (trial_D[i,j]) *
118     (trial_D[j,i]) *
119     trial_S[i,j]) / (Lb[i,j]*Nr) +
120     Gamma1*(1+M[i,j])*(1-trial_S[i,j])
121     -
122     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*
123     (trial_D[k,j]) *

```

```

123      (trial_D[j,k])*trial_V[k])/(Lb[k,j]*Nr)
124    ));
125
126 subject to log_not_explode {(i,j) in Links}:
127      ((P[i]*
128      (if (i==this_node) then local_D[j] else trial_D[i,j])*
129      (if (j==this_node) then local_D[i] else trial_D[j,i])*
130      trial_S[i,j]) / (Lb[i,j]*Nr) +
131      Gamma1*(1+M[i,j])*(1-trial_S[i,j])
132      -
133      Gamma1*(1+ sum{k in Nodes: k < i and k < j}((P[k]*
134      (if (k==this_node) then local_D[j] else trial_D[k,j])*
135      (if (j==this_node) then local_D[k] else trial_D[j,k])*trial_V[k])/(Lb[
136      k,j]*Nr))
137      )) >= 0.0001;
138
139 #For RMP:
140 ## Link sets to announce (allows daemon master to set AnnclinkSets to
141 ## null when there's no new data)
142 set AnnclinkSets in LinkSets default {};
143 model compute-m.mod;
144 model RMP.mod;
145 #####
146 #   Parameters for use in separated processes
147 #####
148
149 set updated_links in Links default {};
150 param new_beta_t {(i,j) in updated_links};
151

```

```

152 set inputsteps default {1};
153 param Step {inputsteps} default 1;
154
155 set PatsPlus = Pats union {-1};
156 param master_b {1 in LinkSets, n in Nodes} in PatsPlus default -1; ##
157
158 #####
159 # Problem Defintions
160 #####
161
162 ##acg: other_links calls interference_obj
163 ##acg: other_links calls victim_off
164 ##acg: other_links calls duplex
165 problem other_links: interference_obj, S, victim_off, duplex;
166 option solver ipopt;
167 option solver_msg 1;
168 option times 0;
169 option show_stats 0;
170
171 problem FLAP: S, V, flap_obj, duplex, coupling;
172 option relax_integrality 1;
173 option solver cplexamp;
174
175 problem DUALFLAP: S, V, dual_qna_flap_obj, coupling; #notice no duplex
      constraint
176 option relax_integrality 1;
177 option solver cplexamp;
178
179
180 problem QNA_FLAP: S, V, qna_flap_obj, duplex, coupling;

```



```
181 option relax_integrality 1;
182 option solver cplexamp;
183
184 problem SINGLE_QNA_FLAP: S, V, single_qna_flap_obj, single_duplex_A,
      single_duplex_B, single_coupling;
185 option relax_integrality 1;
186 option solver cplexamp;
187
188 problem SINGLE_FLAP: S, V, single_flap_obj, single_duplex_A, single_duplex_B,
      single_coupling;
189 option relax_integrality 1;
190 option solver cplexamp;
191
192 #problem SDQ_FLAP: S, V, single_dual_qna_flap_obj, single_coupling,
      no_self_loop;
193 ##acg: SDQ_FLAP calls single_dual_qna_flap_obj
194 ##acg: SDQ_FLAP calls single_coupling
195 ##acg: SDQ_FLAP calls no_self_loop
196 ##acg: SDQ_FLAP calls no_extraneous_activation
197 problem SDQ_FLAP: S, V, single_dual_qna_surr_flap_obj, single_coupling,
      no_self_loop, no_extraneous_''activation;
198 option relax_integrality 1;
199 option solver ipopt;
200
201 problem LOCAL_SDQ_FLAP: local_S, single_dual_qna_surr_flap_obj;
202 option relax_integrality 1;
203 option solver ipopt;
204
205 problem FARP: D, B, farp_obj, real_pats, one_pat;
206 option relax_integrality 1;
```

```
207 option solver ipopt;
208
209 ##acg: SNARP calls snarp_obj
210 ##acg: SNARP calls snarp_real_pats
211 ##acg: SNARP calls snarp_one_pat
212
213 problem SNARP: local_D , local_B , snarp_obj , snarp_real_pats , snarp_one_pat;
214 option relax_integrality 1;
215 #option solver ipopt;
216 option solver cplexamp;
217 option cplex_options 'writeprob_foo.mps';
218 #option ipopt_options 'constr_viol_tol 1e-20';
219 #option ipopt_options 'print_options_documentation yes';
220
221
222 ##acg: alt_SNARP calls alternate_snarp_obj
223 ##acg: alt_SNARP calls snarp_real_pats
224 ##acg: alt_SNARP calls snarp_one_pat
225 problem alt_SNARP: local_D , local_B , alternate_snarp_obj , snarp_real_pats ,
        snarp_one_pat;
226 option relax_integrality 1;
227 option solver ipopt;
228
229 ##acg: RMP calls RMP_obj
230 ##acg: RMP calls demand_coverage
231 problem RMP: x, RMP_obj, demand_coverage;
232 option relax_integrality 0;
233 option solver ipopt;
234 #option cplex_options 'sensitivity';
235
```

```

236 problem RECOVER_PRIMAL: PrimalS, PrimalV, closest_primal;
237 #option cplex_options 'prestats=1 presolve=0 prereduce=0';
238 option relax_integrality 0;
239 option solver_msg 1;
240 option times 1;
241 option show_stats 1;
242 #option presolve 0;
243 option solver cplexamp;
244
245 problem improve_antennas: D, B, signal_margin, real_pats, one_pat;
246 option solver ipopt;
247 option halt_on_ampl_error yes;
248
249 problem dist_improve_antennas: local_D, local_B, dist_signal_margin,
      log_not_explode, dist_real_pats, dist_one_pat,
      dist_maintain_signal_margin1, dist_maintain_signal_margin2,
      dist_maintain_signal_margin3;
250 option solver ipopt;
251 option solver_msg 1;
252 option times 0;
253 option show_stats 0;
254 option halt_on_ampl_error yes;
255
256 #####
257 # Debugging Calculations
258 #####
259
260 ## Signal strength (0 if link not active)
261 param SNR {(i,j) in Links} = ((P[i]*Dbar[i,j]*Dbar[j,i]*Sbar[i,j]) / (Lb[i,j]*
      Nr));

```

```

262
263 # Substitute signal strength to make interference constraint satisfied
264 # for unused links.
265 param offOK {(i,j) in Links} = Gamma1deflap*(1+M[i,j])*(1-Sbar[i,j]);
266
267 ## Received interference at each link.
268 param interference {(i,j) in Links} =
269     sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*Dbar[j,k]*Vbar[k])
270         /(Lb[k,j]*Nr));
271
272 ## Main, non-lagrangian, problem (CLAP) objective
273
274 param CLAP_obj = sum{(i,j) in Links} beta_t[i,j] * Sbar[i,j];
275
276 param rssMW{i in Nodes, j in Nodes: i <> j} = P[i]*Vbar[i]*Dbar[i,j]*Dbar[j,i]
277     ]/Lb[i,j];
278
279 param rss_dBm{i in Nodes, j in Nodes: i <> j} = if (rssMW[i,j] > 0) then (10*
280     log10(rssMW[i,j])) else (-Infinity);
281
282 param interfereMW{(i,j) in Links} =
283     sum{k in Nodes: k <> i and k <> j}(rssMW[k,j]);
284
285 param interfere_dBm{(i,j) in Links} = if (interfereMW[i,j] > 0) then (10*log10
286     (interfereMW[i,j])) else (-Infinity);
287
288 param trueSINR{(i,j) in Links} = (rssMW[i,j] / (interfereMW[i,j]+Nr));
289
290 param trueSINR_dB{(i,j) in Links} = if (trueSINR[i,j] > 0) then (10*log10(
291     trueSINR[i,j])) else (-Infinity);

```

```

287
288 param flap_reward_use{(i,j) in Links} = local_beta_t[i,j];
289
290 param flap_reward_SINR{(i,j) in Links}=
291     Lbar[i,j]*((P[i]*Dbar[i,j]*Dbar[j,i]) / (Lb[i,j]*Nr) -
292     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
293         Dbar[j,k]*1)/(Lb[k,j]*Nr))
294     ));
295 param flap_reward_null{(i,j) in Links}=
296     Lbar[i,j]*(Gamma1deflap*(1+M[i,j]) -
297     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
298         Dbar[j,k]*1)/(Lb[k,j]*Nr))
299     ));
300 param flap_lagrange{(i,j) in Links} =
301     Lbar[i,j]*((P[i]*Dbar[i,j]*Dbar[j,i]*Sbar[i,j]) / (Lb[i,j]*Nr) +
302     Gamma1deflap*(1+M[i,j])*(1-Sbar[i,j]) -
303     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*
304         Dbar[j,k]*Vbar[k])/(Lb[k,j]*Nr)))));
305
306 param pat_gain_dB {i in Nodes, j in Nodes, p in Pats} = if (pat_gain[i,j,p] >
307     0) then (10*log10(pat_gain[i,j,p])) else (-Infinity);
308 param Lb_dB {i in Nodes, j in Nodes: i <> j} = if (Lb[i,j] > 0) then (10*log10
309     (Lb[i,j])) else (-Infinity);
310
311

```

```

312 # Time consumed in this iteration of FLAP, FARP (for output purposes).
313 param flap_iter_time default 0;
314 param farp_iter_time default 0;

```

Listing D.6: farp.mod

```

1 #####
2 # FARP
3 #####
4
5 # Set of antenna patterns
6 set Pats;
7
8 ## pat_gain — gain at node i toward node j in pattern p
9 param pat_gain {i in Nodes, j in Nodes, p in Pats};
10
11 #Delta D — hard to explain
12 param delta_d {j in Nodes, i in Nodes, k in Nodes, p in Pats};
13
14 # Directional gain from node i toward node j
15 var D {i in Nodes, j in Nodes} >= 0.0;
16
17
18 var B {i in Nodes, p in Pats} binary;
19
20 param Bbar{i in Nodes, p in Pats} >= 0 <=1;
21
22 # Objective and constraints are Equation 3.25, Proposal R952, p. 58
23 ##acg: mark farp_obj done
24 maximize farp_obj: sum{(i,j) in Links}(
25     local_beta_t[i,j] * Sbar[i,j] +

```

```

26
27     Lbar[i,j]*(((P[i]*D[i,j]*D[j,i]*Sbar[i,j]) / (Lb[i,j]*Nr) +
28     Gamma1deflap*(1+M[i,j])*(1-Sbar[i,j])) -
29     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*D[k,j]*D[j,k
        ]*Vbar[k])/(Lb[k,j]*Nr))
30     )));
31
32 ## For simplicity, write this constraint directly in D, not ddiffs.
33 ## Also, this exceeds the size limits of student edition on AMPL.
34
35 #subject to real_pats {i in Nodes, j in Nodes, k in Nodes}:
36 #      $D[j,i] - D[j,k] - \text{sum}\{p \text{ in Pats}\}(\text{delta\_d}[j,i,k,p] * B[j,p]) = 0;$ 
37
38 ##acg: mark real_pats done
39 subject to real_pats {i in Nodes, j in Nodes}:
40      $D[i,j] - \text{sum}\{p \text{ in Pats}\}(\text{pat\_gain}[i,j,p] * B[i,p]) = 0;$ 
41
42 ##acg: mark one_pat done
43 subject to one_pat {j in Nodes}:
44      $\text{sum}\{p \text{ in Pats}\} B[j,p] = 1;$ 

```

Listing D.7: guan-common.mod

```

1
2 param guan_theta_w_star default 0; ##Estimate of optimal lagrangian.
3     ##This is pretty cheesy, but not a
4     ##terrible estimate in general.
5
6 ## exactly equal level of violation of duplex constraint
7 param guan_g_mu{i in Nodes} =
8      $\text{sum}\{j \text{ in Nodes}\} Sbar[i,j] +$ 

```

```

9      sum{j in Nodes} Sbar[j,i] - 1;
10
11  param guan_g_j = sqrt(sum {i in Nodes}(guan_g_mu[i])**2); ## there is no
      guan_g_lambda
12
13
14  # Augmented lagrangian — what we're supposedly decomposing to get the
      objective
15  param guan_curr_lagrangian = sum{(i,j) in Links}(
16      ((-QNA_a[i,j])*(Sbar[i,j]**2) +(QNA_b[i,j]-Lbar[i,j])*Sbar[i,j]) -
17      (0.5*mu[i] + 0.5*mu[j])*Sbar[i,j]
18      - sum {p in Nodes}( ##indexing by node relative to which duplex
      constraint is defined
19      guan_w[p] * (max(0,(
20      (Sbar[i,p] + Sbar[p,i]) + guan_v[p,i]      ##c_i^(p) x_i — c_i
      ^(p) is just 1
21      )))**2
22      ));
23
24  param guan_max_sj = (guan_theta_w_star - guan_curr_lagrangian)/(guan_g_j**2);

```

Listing D.8: subgradient-common.mod

```

1  ## Includes logic about terminating the RPP-Subgradient iteration process
2  #####
3  # Subgradient Computation
4  #
5  # Note that we use a subgradient update method instead of (e.g.) Newton's
6  # method because the gradient is not necessarily well-defined.
7  #####
8

```



```

9
10 # Note that other termination conditions may kick in before max_iter
11 param max_iter default 100000;
12 set Steps = 1 .. max_iter;
13
14 # This is the constraint (equation 3.16 in [CLAP], proposal R952, p. 54)
15 # dualized to allow the Lagrangian decomposition of CLAP.
16
17
18 # It is written as (part A >= part B) in the paper; This value (part A
19 # - part B) is level of OKness (<0 implies violation)
20
21 ##acg: mark CLAP_SINR_constraint done
22 param CLAP_SINR_constraint {(i, j) in Links} =
23     #part A
24     ((P[i]*Dbar[i, j]*Dbar[j, i]*Sbar[i, j]) / (Lb[i, j]*Nr) +
25     Gamma1deflap*(1+M[i, j])*(1-Sbar[i, j])
26     -
27     #part B
28     Gamma1deflap*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k, j]*
29         Dbar[j, k]*Vbar[k])/(Lb[k, j]*Nr))
30     ));
31 #debugging info from central only
32 param dbg_CSC_a1 {(i, j) in Links} = (P[i]*Dbar[i, j]*Dbar[j, i]*Sbar[i, j]) / (
33     Lb[i, j]*Nr);
34 param dbg_CSC_a2 {(i, j) in Links} = (Gamma1deflap*(1+M[i, j])*(1-Sbar[i, j]));
35 param dbg_CSC_b {(i, j) in Links} = Gamma1deflap*(1+ sum{k in Nodes: k <> i and
36     k <> j} ((P[k]*Dbar[k, j]*Dbar[j, k]*Vbar[k])/(Lb[k, j]*Nr)));

```

```

36 param CLAP_SINR_constraint_real {(i,j) in Links} =
37     #part A
38     ((P[i]*Dbar[i,j]*Dbar[j,i]*Sbar[i,j]) / (Lb[i,j]*Nr) +
39     Gamma1*(1+M[i,j])*(1-Sbar[i,j])
40     -
41     #part B
42     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k,j]*Dbar[j,k]
43         )*Vbar[k]))/(Lb[k,j]*Nr))
44
45
46 # This is a subgradient at point Lbar,Dbar,Sbar,Vbar; There's only one
47 # dualized constraint, so all we're doing is changing the sign.
48 # See equation 2.26, proposal R952, p. 40.
49
50 # # Proper subgradient
51 # param CLAP_subgradient {(i,j) in Links} =
52 #     if (CLAP_SINR_constraint[i,j] < 0)
53 #     then 1-(CLAP_SINR_constraint[i,j])
54 #     else 0.0001 * (1-(CLAP_SINR_constraint[i,j])); #XXX made this up!
55
56 param CLAP_subgradient {(i,j) in Links} =
57     if (Sbar[i,j] > 0)
58     then 0-(CLAP_SINR_constraint[i,j])
59     else (-100000);
60
61 # param CLAP_subgradient {(i,j) in Links} =
62 #     0-(CLAP_SINR_constraint[i,j]);
63

```

```

64 param max_lbar{(i,j) in Links} = max(local_beta_t[i,j] / CLAP_subgradient[i,j
      ], 0);
65
66
67 # Adjusted for nonnegativity of Lagrange Multipliers:
68 # Theoretically equivalent to using CLAP_subgradient and then limiting
69 # the lagrange multiplier values, but much more efficient convergence.
70
71 param CLAP_sg_nn {(i,j) in Links} =
72     if (Lbar[i,j] <= 0 and CLAP_subgradient[i,j] < 0) then 0 else
      CLAP_subgradient[i,j];
73 #     CLAP_subgradient[i,j];
74
75 # L2-norm of tweaked subgradient
76 param CLAP_sg_norm =
77     sqrt(sum{(i,j) in Links}(CLAP_sg_nn[i,j]^2));
78
79 #I just made this up. Total BS!
80 param stepsize{k in Steps} = (1e-13)/((k+10)**2);
81 #param stepsize{k in Steps}=1e-18;
82
83 param ssize default 0;
84
85 # Multiplier update rule: Equation 2.27, propasl R952, p. 40.
86 # Avoidance of divide-by-zero situation is made up.
87 # param Lbar_step_unscaled {(i,j) in Links} =
88 #     if (CLAP_sg_norm==0)
89 #     then (CLAP_subgradient[i,j]/1)
90 #     else (CLAP_subgradient[i,j]/CLAP_sg_norm);
91

```

```

92 # Simple fully local rule (no use of global l2 norm)
93 param Lbar_step_unscaled {(i,j) in Links} =
94     (CLAP_subgradient[i,j]) ;
95
96 param Lbar_prev {(i,j) in Links}; ## Must be assigned in iterative loop
97
98 param lbar_step_upper_limit {(i,j) in Links} = max_lbar[i,j] * 1e-5;
99 param lbar_step_lower_limit {(i,j) in Links} = Lbar[i,j] * -1; ##XXX made up
100
101 param Lbar_step {(i,j) in Links} = if (Lbar_step_unscaled[i,j] > 0) then
102     min(ssize * Lbar_step_unscaled[i,j], lbar_step_upper_limit[i,j])
103     else
104     max(ssize * Lbar_step_unscaled[i,j], lbar_step_lower_limit[i,j]);
105
106 param Lbar_suggest {(i,j) in Links} = Lbar[i,j] + Lbar_step[i,j];
107
108 param Lbar_last_step {(i,j) in Links} = Lbar[i,j] - Lbar_prev[i,j];
109
110 param Lbar_step_l2_norm =
111     sqrt(sum{(i,j) in Links}(Lbar_last_step[i,j]^2));
112
113
114 #####
115 ## Lagrange multipliers (mu) for duplex constraints ##
116 #####
117
118 ##acg: mark duplex_constraint done
119 param duplex_constraint {i in Nodes} =
120     sum{j in Nodes} Sbar[i,j] +
121     sum{j in Nodes} Sbar[j,i] - 1;

```

```

122
123 #param mu_stepsize{k in Steps}=1e-3;
124 #param mu_stepsize{k in Steps}=1e1/(k+100);
125 param mu_stepsize{k in Steps}=1e3/((k+100)**2);
126 #mis-apply Guan surrogate subgradient rule:
127 #param mu_stepsize{k in Steps}= 0.05 * guan_max_sj;
128 param mu_ss;
129
130 param duplex_subgradient {i in Nodes} =
131     duplex_constraint[i];
132
133
134 param duplex_sg_nn {i in Nodes} =
135     if (mu[i] <= 0 and duplex_subgradient[i] < 0) then 0
136     else duplex_subgradient[i];
137
138 # L2-norm of tweaked subgradient
139 param duplex_sg_norm =
140     sqrt(sum{i in Nodes}(duplex_sg_nn[i]^2));
141
142 # param duplex_sg_norm =
143 #     sqrt(sum{i in Nodes}(duplex_subgradient[i]^2));
144
145 ##acg: mu_step calls duplex_sg_nn
146 ##acg: mu_step calls mu_ss
147 param mu_step {i in Nodes} = duplex_sg_nn[i] * mu_ss;
148
149 ##acg: mu_suggest calls mu
150 ##acg: mu_suggest calls mu_step
151 ##acg: mark mu_suggest done

```

```

152 param mu_suggest {i in Nodes} = mu[i] + mu_step[i];
153
154 #####
155 ##      FLAP/{FARP,SNARP} termination conditions.      ##
156 ##              Lots of magic numbers here              ##
157 #####
158
159 param mu_step_l2_norm = sqrt(sum{i in Nodes}(mu_step[i]^2));
160
161 # Step in both Lambda (bar) and mu (bar)
162 param step_l2_norm = sqrt(
163     sum{(i,j) in Links}(Lbar_step[i,j]^2) +
164     sum{i in Nodes}(mu_step[i]^2)
165     );
166
167 # A subgradient solution is demonstrably converged when the step size
diminishes to 0 (XXX - cite)
168 # small_enough is the epsilon for close enough to zero
169 param small_enough := 1e-18;
170 param small_enough_lbar_step := 5e-15;
171 param small_enough_mu_step := 1e-7;
172
173 # Require a certain number of iterations to prevent termination based
# on initial estimate values
174 param abs_min_iter := 5;
175
176
177 # Maximum randomization of local_beta_t values (to prevent ties)
178 #param max_beta_jitter := 0.05;
179 param max_beta_jitter := 0.1;
180

```

```

181 #####
182 ## Early termination conditions: Recall that for column generation,
183 ## each subproblem need not be solved to optimality; it is enough that
184 ## (at least) one improving column be found, if one exists.
185 ## Therefore, we allow an early termination if the reduced cost is < 0
186 ## and an arbitrary number of iterations have occurred
187
188 # Minimum iterations before early termination
189 param early_exit_min_iter := 20;
190
191 param num_sets_per_dw_iter := 1; ##Disable early break to see what happens
      with guan's algorithm
192
193 ##acg: mark sets_this_dw_iter done
194 param sets_this_dw_iter default 0;
195
196 #####
197 # Heuristics to identify primal feasible solutions #
198 #####
199
200 # Given current estimates, if link were used, would constraint be
201 # satisfied? Doesn't look promising as an approach. Especially
202 # because, if Vbar = 0.99995, it still fails.
203
204 param HatS {i in Nodes, j in Nodes} default 0;
205
206 # Just CLAP-SINR_constraint_real with S[i,j]=1
207 param SINR_given_link {(i,j) in Links} =
208     #part A
209     ((P[i]*Dbar[i,j]*Dbar[j,i]*1) / (Lb[i,j]*Nr) +

```

```

210     Gamma1*(1+M[i , j]) *0
211     -
212     #part B
213     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*Dbar[k , j]*Dbar[j , k
          ]*Vbar[k]))/(Lb[k , j]*Nr))
214     ));
215 param SINR_wouldbe_ok {(i , j) in Links} = if(SINR_given_link[i , j] >= 0) then 1
          else 0;
216
217 # Vbar >= S[i , j] satisfied if S[i , j] = 1
218 param Vbar_wouldbe_ok {(i , j) in Links} = if (Vbar[i] >= 1) then 1 else 0;
219
220
221 param duplex_given_link{(i , j) in Links} = (
222     #other links involving i
223     sum{xj in Nodes: xj <> j} Sbar[i , xj] +
224     sum{xj in Nodes: xj <> j} Sbar[xj , i] +
225     #other links involving j
226     sum{xi in Nodes: xi <> i} Sbar[xi , j] +
227     sum{xi in Nodes: xi <> i} Sbar[j , xi]);
228
229 param duplex_wouldbe_ok{(i , j) in Links} = if(duplex_given_link[i , j] == 0) then
          1 else 0;
230
231 param link_wouldbe_ok {(i , j) in Links} =
232     SINR_wouldbe_ok[i , j] *
233     Vbar_wouldbe_ok[i , j] *
234     duplex_wouldbe_ok[i , j];
235
236 # Find closest primal feasible point

```



```

237 ## S — activation of link ij
238
239 param ewma_hats_alpha = 0.02;
240
241 # Apply EWMA to global Dbar
242 param ewma_dbar_alpha = 0.02;
243
244 var PrimalS {i in Nodes, j in Nodes} binary;
245
246 ## V — activation of node i
247 var PrimalV {i in Nodes} >= 0 <= 1;
248
249 #minimize closest_primal: sum{(i,j) in Links} abs(PrimalS[i,j] - HatS[i,j]);
250 minimize closest_primal: sum{(i,j) in Links}((HatS[i,j] -PrimalS[i,j])^2);
251 ###XXX even w/o constraints, process seems not to work!
252
253 subject to primal_duplex {i in Nodes}:
254     sum{j in Nodes} PrimalS[i,j] +
255     sum{j in Nodes} PrimalS[j,i] <= 1;
256
257 subject to primal_coupling {i in Nodes}:
258     sum{j in Nodes: (i,j) in Links} PrimalS[i,j] <= PrimalV[i];
259
260 ###XXX subject to primal SINR!;

```

Listing D.9: trial-SINR-logic.mod

```

1 param trial_V {i in Nodes} = sum{j in Nodes: (i,j) in Links} trial_S[i,j];
2 param trial_D {i in Nodes, j in Nodes};
3 param trial_B {i in Nodes, p in Pats} >= 0 <=1;
4

```

```

5 param trial_duplex_constraint {i in Nodes} =
6     sum{j in Nodes: (i,j) in Links} trial_S[i,j] +
7     sum{j in Nodes: (j,i) in Links} trial_S[j,i] - 1;
8
9 param tsc_a {(i,j) in Links} = ((P[i]*trial_D[i,j]*trial_D[j,i]*trial_S[i,j])
    / (Lb[i,j]*Nr) + Gamma1*(1+M[i,j])*(1-trial_S[i,j]));
10
11 param tsc_b {(i,j) in Links} = (Gamma1*(1+ sum{k in Nodes: k <> i and k <> j
    }((P[k]*trial_D[k,j]*trial_D[j,k]*trial_V[k])/(Lb[k,j]*Nr))));
12
13 param trial_SINR_constraint {(i,j) in Links}=
14     #part A
15     ((P[i]*trial_D[i,j]*trial_D[j,i]*trial_S[i,j]) / (Lb[i,j]*Nr) +
16     Gamma1*(1+M[i,j])*(1-trial_S[i,j])
17     -
18     #part B
19     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}((P[k]*trial_D[k,j]*
    trial_D[j,k]*trial_V[k])/(Lb[k,j]*Nr))
20     ));
21
22
23 param trial_links_ok =
24     if ((forall {i in Nodes} trial_duplex_constraint[i] <= 0) and
25         (forall {(i,j) in Links} trial_SINR_constraint[i,j] >=0))
26     then 1 else 0;
27
28 param trial_rc = (1 - sum{(i,j) in Links} (trial_S[i,j] * beta_t[i,j]));
29
30
31 # Output information. t_ prefix indicates that these are calculated

```

```

32 # over the trial_ variables
33 param t_rssMW{i in Nodes, j in Nodes: i <> j} = P[i]*trial_V[i]*trial_D[i,j]*
      trial_D[j,i]/Lb[i,j];
34
35 param t_interfereMW{(i,j) in Links} =
36     sum{k in Nodes: k <> i and k <> j}(t_rssMW[k,j]);
37
38 param t_trueSINR{(i,j) in Links} = t_rssMW[i,j] / (t_interfereMW[i,j]+Nr);
39
40 param t_trueSINR_dB{(i,j) in Links} = if (trial_S[i,j] > 0) then (10*log10(
      t_trueSINR[i,j])) else (-Infinity);

```

Listing D.10: post-schedule-SINR-opt.mod

```

1 #####
2 # Post-scheduling antenna tweaking: Basic idea is to improve gain,
3 # even on links where the SNR constraint was not binding. Note that
4 # this is heuristic and by no means guaranteed to produce an optimal
5 # anything, though it will not hurt the optimality of the schedule.
6 #####
7
8 # Precondition: trial_S must be the just-accepted schedule.
9 # trial_SINR_constraint is then correct for this schedule.
10
11
12
13 maximize signal_margin: sum{(i,j) in Links}
14 ##part A
15     ((P[i]*log(D[i,j]*D[j,i])*trial_S[i,j]) / (Lb[i,j]*Nr)
16     -
17 ##part B

```



```

45         (max{p in Pats} (pat_gain[i,j,p]) / Lb[i,j]));
46
47
48 # Question 2: What's my (k's) "pissing on link i->j budget?" Naively,
49 # just divide the current (pre-optimization) budget evenly among all
50 # possibly involved nodes.
51
52 # note that it doesn't matter which link we're considering: Every
53 # transmitter, plus one receiver, is "involved"
54 param num_involved_nodes = 1 + sum{i in Nodes} trial_V[i];
55
56 param pissing_budget {(i,j) in Links, k in Nodes} =
57     trial_SINR_constraint[i,j] / num_involved_nodes;
58
59
60 ## Objective is missing: Not common between centralized and distributed code.
61
62 #Link(s) from me
63 subject to dist_maintain_signal_margin1 {(i,j) in Links: i == this_node}:
64     trial_SINR_constraint[i,j] -
65     ((P[i]*local_D[j]*trial_D[j,i]*trial_S[i,j]) / (Lb[i,j]*Nr) +
66     Gamma1*(1+M[i,j])*(1-trial_S[i,j])
67     -
68     Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}(
69         (P[k]*trial_D[k,j]*trial_D[j,k]*trial_V[k])/(Lb[k,j]*Nr)
70     )) <= pissing_budget[i, j, this_node] ;
71
72 #Link(s) to me
73 subject to dist_maintain_signal_margin2 {(i,j) in Links: j == this_node}:
74     trial_SINR_constraint[i,j] -

```

```

75      ((P[i]*trial_D[i,j]*local_D[i]*trial_S[i,j]) / (Lb[i,j]*Nr) +
76      Gamma1*(1+M[i,j])*(1-trial_S[i,j])
77      -
78      Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}(
79      (P[k]*trial_D[k,j]*trial_D[j,k]*trial_V[k])/(Lb[k,j]*Nr))
80      )) <= pissing_budget[i, j, this_node];
81
82 #Others' links
83 subject to dist_maintain_signal_margin3 {(i,j) in Links: i != this_node and j
      != this_node}:
84      trial_SINR_constraint[i,j] -
85      ((P[i]*trial_D[i,j]*trial_D[j,i]*trial_S[i,j]) / (Lb[i,j]*Nr) +
86      Gamma1*(1+M[i,j])*(1-trial_S[i,j])
87      -
88      Gamma1*(1+ sum{k in Nodes: k <> i and k <> j}(
89      (P[k]*(if (k==this_node) then local_D[j] else trial_D[k,j])*trial_D[
          j,k]*trial_V[k])/(Lb[k,j]*Nr))
90      )) <= pissing_budget[i, j, this_node];
91
92
93 subject to dist_real_pats {j in Nodes}:
94      local_D[j] - sum{p in Pats} (pat_gain[this_node,j,p] * local_B[p]) =
          0;
95
96      ## Force full power again.
97 subject to dist_one_pat:
98      sum{p in Pats} local_B[p] == 1;

```

Listing D.11: compute-m.mod

```
1 #####
```

```

2 # Routine to pre-compute  $M[i, j]$ 
3 #####
4 param target_p {(i, j) in Links} default 0;
5
6
7 # Compute the worst possible power from other node to our victim
8 ##acg: mark worst_case_power done
9 param worst_case_power{k in Nodes, j in Nodes: k <> j} = (P[k]*
10     (max {p in Pats} pat_gain[k, j, p]) *
11     (max {p in Pats} pat_gain[j, k, p]))/Lb[k, j];
12
13 ##acg: interference_obj calls worst_case_power
14 ##acg: mark interference_obj done
15 maximize interference_obj: sum {(i, j) in Links} S[i, j]*worst_case_power[i, j]*1
16     e9;
17 ##acg: mark victim_off done
18 subject to victim_off{(i, j) in Links}: target_p[i, j] * S[i, j] = 0;
19
20 param max_i{(i, j) in Links};
21 param Mij{(i, j) in Links} = (max_i[i, j]/(Nr*1e9));

```

## D.2 Command Files

These command files are AMPL's imperative component.

### D.2.1 On-Line System

Listing D.12: master-daemon.ampl

```

1 ## Test of interaction between AMPL and invoking program
2
3 ## Source STMDA model definitions

```

```
4 model ea_stdma.mod;
5 data stdma-defaults.dat;
6
7 option times 0;
8 option show_stats 0;
9
10 ## Source table defintions
11 commands "tables.ampl";
12
13 #data nodeconf.dat;
14
15 set changed_nodepairs in {Nodes, Nodes};
16 param out_sbar {i in Nodes, j in Nodes: i == flap_node};
17 let AnncLinkSets := {};
18
19
20 read table nodevars;
21 read table step;
22 read table nodepairvars;
23 read table linkvars;
24 read table patterns;
25
26 write table x;
27 write table m_antennas;
28 write table link_in_set;
29 write table beta_t;
30
31 print "Initializing" > master_log.txt;
32 print "Initializing" > dw_log.txt;
33
```



```

34 #####
35 #           Compute M[ij]
36 #####
37 for {(i,j) in Links}{
38     let {(k,l) in Links} target_p[k,l] := 0;
39     let target_p[i,j] := 1;
40     solve other_links;
41     let max_i[i,j] := interference_obj;
42 }
43 let {(i,j) in Links} M[i,j] := Mij[i,j];
44
45
46
47 display M;
48
49
50 #####
51 #   Initialize LinkSets with simple TDMA
52 #####
53
54 param index in LinkSets;
55 for {(i,j) in Links}{
56     let num_link_sets := num_link_sets +1;
57     if (num_link_sets == 1) then let index := first(LinkSets);
58     else let index := next(index, LinkSets);
59     let link_in_set[index,i,j] := 1;
60 }
61
62 param dw_iter integer >0;
63 let dw_iter := 1;

```

```

64 param old_num_link_sets integer >0;
65
66
67
68 repeat {
69     read table nodevars;
70     read table step;
71     read table nodepairvars;
72     read table linkvars;
73     read table patterns;
74
75     #####
76     #   Solve RMP: Find the optimal link set activations
77     #####
78     solve RMP;
79     let updated_links := Links;
80     let {(i,j) in Links} beta_t[i,j] := demand_coverage[i,j].dual;
81     ## XXX contemplate significance of negative dual values (for >=
82     constraint)!
83     let {(i,j) in Links} new_beta_t[i,j] := max(0, beta_t[i,j]);
84
85     #display link_in_set;
86     display x;
87     display beta_t;
88     display new_beta_t;
89
90     # Cause x and link_in_set tables to be filled in
91     let AnnLinkSets := LinkSets;
92     write table x;
93     write table m_antennas;

```

```

93     write table link_in_set;
94     let AnncLinkSets := {};
95     write table beta_t;
96
97
98     print "top";
99     repeat{
100         read table nodevars;
101         read table step;
102         read table nodepairvars;
103         read table linkvars;
104         read table patterns;
105
106         #display local_Dbar;
107         #display local_Lbar;
108         #display local_Sbar;
109         #display local_mu_bar;
110
111         printf "New_loop._.Step_=%d", Step[1] >> master_log.txt;
112
113         #####
114         # Find best feasible primal solution from current dual
115         #####
116         printf "Attempting_primal_recovery\n";
117         ## Binary search on act_threshold for feasible set with lowest
118             reduced cost
119         let act_threshold := 0.5;
120         let bs_ss := 0.25;
121         let best_rc := 1;

```

```

121      let best_thresh :=1.1; #Demand the impossible, until something
           possible is proven OK.
122      printf "\tthreshold\tfeas.\treduced_cost\n";
123      repeat {
124          let {i in Nodes, j in Nodes: i <> j} trial_D[i,j]:=
           local_Dbar[i,j];
125          printf "\t%f\t%d\t%d\n", act_threshold, trial_links_ok
           , trial_rc;
126
127          print "P.R._loop" >> master_log.txt;
128          display act_threshold >> master_log.txt;
129          display trial_S >> master_log.txt;
130          display trial_V >> master_log.txt;
131          display trial_D >> master_log.txt;
132          display trial_duplex_constraint >> master_log.txt;
133          display P >> master_log.txt;
134          display Lb >> master_log.txt;
135          display Nr >> master_log.txt;
136
137          display tsc_a >> master_log.txt;
138          display tsc_b >> master_log.txt;
139
140          display trial_SINR_constraint >> master_log.txt;
141          display trial_links_ok >> master_log.txt;
142          display trial_rc >> master_log.txt;
143
144          display t_rssMW >> master_log.txt;
145          display t_trueSINR >> master_log.txt;
146          display t_trueSINR_dB >> master_log.txt;
147

```

```

148         if ((trial_links_ok == 1) and
149             (trial_rc <= best_rc)) then {
150             let best_rc := trial_rc;
151             let best_thresh := act_threshold;
152         };
153
154         if (trial_links_ok == 1) then {
155             # reduce threshold as long as feasibility is
156                 maintained
157             let act_threshold := act_threshold - bs_ss;
158         }
159         else {
160             # increase threshold until feasible
161             let act_threshold := act_threshold + bs_ss;
162         };
163
164         let bs_ss := bs_ss/2;
165     } while (bs_ss > 0.002);
166     let act_threshold := best_thresh;
167
168
169     #####
170     # Evaluate primal solution as possible stopping point
171     #####
172
173     if (trial_rc < rc_thresh) then {
174
175         display act_threshold;
176         display best_thresh;

```

```

177     display trial_S;
178     display trial_D;
179     display trial_SINR_constraint;
180     display trial_duplex_constraint;
181     display trial_links_ok;
182     display trial_rc;
183
184     if (exists {l in LinkSets} (not exists {(i,j) in Links}
185         (trial_S[i,j] > link_in_set[l,i,j]))) then { #must not
           be equal to OR DOMINATED BY existing link set
186
187         print "Rejected_link_set!";
188         display {l in LinkSets}: {(i,j) in Links} link_in_set[
           l,i,j];
189         display trial_S;
190     } else {
191         let num_link_sets := num_link_sets+1;
192         let index := next(index,LinkSets);
193         let {(i,j) in Links} link_in_set[index,i,j] := trial_S
           [i,j];
194         let sets_this_dw_iter := sets_this_dw_iter + 1;
195
196
197         print "Added_link_set_Continuing";
198         display trial_S;
199
200         print "Added_link_set_Continuing" >> dw_log.txt;
201         display {(i,j) in Links} trial_S[i,j] >> dw_log.txt;
202

```

```

203     printf "\nPost-optimized_antennas_for_candidate_
        schedule:\n" >> dw_log.txt;
204     printf "Pre-opt_SINR_info\n" >> dw_log.txt;
205     display trial_SINR_constraint >> dw_log.txt;
206     display t_trueSINR_dB >> dw_log.txt;
207
208     for {blah in Nodes} {
209         let this_node := blah;
210         display blah >> dw_log.txt;
211         display d_s_m_dbg_nolog >> dw_log.txt;
212         expand dist_signal_margin >> dw_log.txt;
213
214         # "warm start" with non-exploding values.
215         let {j in Nodes: j < this_node} local_D[j] :=
            trial_D[this_node, j];
216         let local_D[this_node] := 0;
217         solve dist_improve_antennas;
218         let {j in Nodes} tmp_trial_D[this_node, j] :=
            local_D[j];
219         display local_D >> dw_log.txt;
220         display local_B >> dw_log.txt;
221         ## Record the most-selected pattern. Should
            always be selected
222         ## with value of 1, within some small epsilon.
223         for {p in Pats} {
224             if (local_B[p] == max {xp in Pats}
                local_B[xp]) then
225                 {
226                     let master_b[index, this_node] :=
                        p;

```

```

227         }
228     }
229 }
230 let {i in Nodes, j in Nodes} trial_D[i,j] :=
    tmp_trial_D[i,j];
231 printf "Post-dist-opt_SINR_info\n" >> dw_log.txt;
232 display trial_SINR_constraint >> dw_log.txt;
233 display t_trueSINR_dB >> dw_log.txt;
234 display master_b >> dw_log.txt;
235
236 if (sets_this_dw_iter >= num_sets_per_dw_iter) then {
237     #set Sbar so that reduced_cost parameter is
        correct.
238     let {(i,j) in Links} Sbar[i,j] := trial_S[i,j]
        ];
239     write table x;
240     write table m_antennas;
241     write table link_in_set;
242     write table beta_t; ## otherwise, "break"
        throws off sync
243     break;
244 }
245 }
246 }
247
248 ## Jitter beta_t
249 ## Jitter beta_t to break ties
250 let {(i,j) in Links} beta_t[i,j] :=
251     if (beta_t[i,j] < max_beta_jitter)
252     then 0

```



```

253             else beta_t[i,j] + Uniform(-max_beta_jitter ,
                                           max_beta_jitter);
254
255             write table x;
256             write table m_antennas;
257             write table link_in_set;
258             write table beta_t;
259             print "rump";
260         } while (1 < 2);
261         print "bottom";
262     } while (1 < 2);
263     print "tail";

```

Listing D.13: solver-daemon.ampl

```

1  ## Test of interaction between AMPL and invoking program
2
3  ## Source STMDA model definitions
4  model ea_stdma.mod;
5  data stdma-defaults.dat;
6
7  #param local_Vbar {n in Nodes} >= 0 <= 1;
8  #param local_Sbar {(src, dst) in Links};
9  #param local_Lbar {(src, dst) in Links};
10 param whatever {n in Nodes};
11
12 option times 0;
13 option show_stats 1;
14 option display_precision 0;
15
16 ## Source table defintions

```

```

17 commands "tables.ampl";
18
19 data nodeconf.dat;
20
21 set changed_nodepairs in {Nodes, Nodes};
22 param out_sbar {i in Nodes, j in Nodes: i == flap_node};
23
24
25 ## Do one "null cycle" to get information for Mij calculation. It's
26 ## critical to do the same reads and writes, in the same order, as
27 ## the normal cycle.
28
29 read table nodevars;
30 read table step;
31 read table nodepairvars;
32 read table linkvars;
33 read table patterns;
34
35 #Dummy values so computed_nodevars doesn't output "." and make Python barf
36 let {n in Nodes} local_mu_bar[n] := 0.5;
37
38 write table computed_nodevars;
39 write table computed_Sbar;
40 write table computed_Lbar;
41 write table computed_Dbar;
42 write table computed_Bbar; ##It's really local_B directly
43
44
45 ## Compute M[i, j]
46 for {(i, j) in Links}{

```

```

47     let {(k,l) in Links} target_p[k,l] := 0;
48     let target_p[i,j] := 1;
49     solve other_links;
50     let max_i[i,j] := interference_obj;
51 }
52 display max_i;
53 display Mij;
54 let {(i,j) in Links} M[i,j] := Mij[i,j];
55
56 display Lb;
57
58 repeat {
59     read table nodevars;
60     read table step;
61     read table nodepairvars;
62     read table linkvars;
63     read table patterns;
64
65     let {i in Nodes} mu[i] := local_mu_bar[i];
66
67     print "****_New_Iteration_****";
68
69
70     display local_Sbar;
71     display local_Lbar;
72     display local_mu_bar;
73     display mu;
74
75     ## Perform an STDMA link iteration!
76

```

```

77     ## Local/Global name stuff
78     let {i in Nodes, j in Nodes} Sbar[i,j] := 0; #non-links initialized to
           zero
79     let {(i,j) in Links} Sbar[i,j] := local_Sbar[i,j];
80     let {(i,j) in Links} Lbar[i,j] := local_Lbar[i,j];
81     let {(i,j) in Links} local_beta_t[i,j] := beta_t[i,j];
82     let {i in Nodes, j in Nodes: i <> j} Dbar[i,j] := local_Dbar[i,j];
83
84     display {j in Nodes, p in Pats} pat_gain[snarp_node,j,p];
85     display {i in Nodes, j in Nodes: (i == snarp_node or j == snarp_node)
           and i <> j} Lb[i, j];
86
87     display {j in Nodes, p in Pats} pat_gain_dB[snarp_node,j,p];
88     display {i in Nodes, j in Nodes: (i == snarp_node or j == snarp_node)
           and i <> j} Lb_dB[i, j];
89
90     display local_QNA_a;
91     display local_QNA_b;
92     display LocalVbar;
93     display local_Dbar;
94
95     expand single_dual_qna_surr_flap_obj;
96
97     solve LOCALSDQ_FLAP;
98     let {xj in Nodes: (flap_node, xj) in Links} local_Sbar [flap_node,xj]
           := local_S[xj];
99     let LocalVbar[flap_node] := sum {xj in Nodes: (flap_node, xj) in Links
           } local_S[xj];
100    let {i in Nodes} Vbar[i] := LocalVbar[i];
101

```

```

102
103     print "Post-solve_(FLAP):";
104     display local_S;
105     display local_Sbar;
106     #display LocalVbar;
107
108     display snarp_node;
109     display {(i,j) in Links} local_Lbar[i,j]*1e5;
110     display lbar_epsilon;
111     display local_mu_bar;
112     expand snarp_obj;
113     expand snarp_real_pats;
114     expand snarp_one_pat;
115
116     solve SNARP;
117     print "Post-solve_(SNARP):";
118     display local_D;
119     let {j in Nodes} local_Dbar[snarp_node,j] := local_D[j];
120     let {i in Nodes, j in Nodes: i <> j} Dbar[i,j] := local_Dbar[i,j];
121     display local_B;
122
123     print "SNARP_DEBUGGING:";
124     if (snarp_node == 99) then {
125         print "FORCIBLY_SETTING_dbg_D_and_dbg_B_for_testing!";
126         let {p in Pats} dbg_B[p] := (if (p == 11) then 1 else 0);
127         let {j in Nodes} dbg_D[j] := pat_gain[snarp_node, j, 11];
128     } else {
129         print "MIRRORING_local_D_and_local_B_for_testing!";
130         let {p in Pats} dbg_B[p] := local_B[p];
131         let {j in Nodes} dbg_D[j] := local_D[j];

```

```

132
133     }
134
135     display snarp_real_pats_dbg;
136     display snarp_one_pat_dbg;
137
138     # Compute subgradients
139     #let ssize := 1e-4; ##Lbar step size (WRONG)
140     #let mu_ss := 1e-3; ##Mu step size (WRONG)
141
142     let ssize := (1e-13)/((Step[1]+10)**2);
143     let mu_ss := (1e2)/((Step[1]+10)**2);
144
145     # CRUCIAL, otherwise all the subgradient stuff may be miscalculated
146     let {(i,j) in Links} Sbar[i,j] := local_Sbar[i,j];
147
148     for {(i,j) in Links} let local_Lbar[i,j] := max(0,Lbar_suggest[i,j]);
149     for {(i,j) in Links} let Lbar[i,j] := local_Lbar[i,j];
150
151     for {i in Nodes} let mu[i] := max(0, mu_suggest[i]);
152     for {i in Nodes} let local_mu_bar[i] := mu[i];
153
154     print "Post-solve_(subgradient):";
155     display duplex_constraint;
156     display local_mu_bar;
157
158     display CLAP_SINR_constraint;
159
160
161     display Lbar_suggest;

```

```

162     display local_Lbar;
163
164     write table computed_nodevars;
165     write table computed_Sbar;
166     write table computed_Lbar;
167     write table computed_Dbar; ##It's really local_D directly
168     write table computed_Bbar; ##It's really local_B directly
169
170
171 } while (1 < 2);
172
173 display Nodes;
174 display local_Sbar;

```

Listing D.14: stdma\_subproblem.ampl

```

1 ## Start with subproblem
2 ##
3 display worst_case_power;
4
5 ## Compute M[i,j]
6 ##acg: - calls other_links
7 ##acg: - calls interference_obj
8 for {(i,j) in Links}{
9     let{(k,l) in Links} target_p[k,l] := 0;
10    let target_p[i,j] := 1;
11    solve other_links;
12    let max_i[i,j] := interference_obj;
13 }
14 display max_i;
15 display Mij;

```

```

16 let {(i,j) in Links} M[i,j] := Mij[i,j];
17
18 ## Jitter beta_t to break ties
19 let {(i,j) in Links} local_beta_t[i,j] :=
20     if (beta_t[i,j] < max_beta_jitter)
21     then 0
22     else beta_t[i,j] + Uniform(-max_beta_jitter , max_beta_jitter);
23
24 display local_beta_t;
25 display local_beta_t >> dw_log.txt;
26
27 # Data for evaluating performance
28 printf "iteration\tclap.obj\treduced.cost\tflap.obj\tfarp.obj\tstep.norm\ttempl
    .time\tflap.solver.time\tfarp.solver.time\tlbar.step\tmu.step\tstep\t" > (
    "runlog." & dw_iter & ".txt");
29 printf {(i,j) in Links}: "lambda.%d.%d\t", i, j >> ("runlog." & dw_iter & ".
    txt");
30 printf {(i,j) in Links}: "S.%d.%d\t", i, j >> ("runlog." & dw_iter & ".txt");
31 #printf {i in Nodes, p in Pats}: "pat.%d.%d\t", i, p >> ("runlog." & dw_iter &
    ".txt");
32 printf "\n" >> ("runlog." & dw_iter & ".txt");
33
34 printf "SNR_information_log\n" > ("debug_SNR." & dw_iter & ".txt");
35
36 #Link use
37 printf "iteration\tsrc.node\tdst.node\tSbar\tHatS\ttrue.SINR\tCLAP.SINR\n" > (
    "link_log." & dw_iter & ".txt");
38
39 #Rewards

```



```

40 printf "iteration\tsrc.node\tdst.node\tlambda\tuse\tSINR\tnull\tlagrange\tmax.
    lbar\n" > ("rewards." & dw_iter & ".txt");
41
42 # Antenna patterns specifically
43 printf "iteration\tnode\tpat\tval\n" > ("antenna_log." & dw_iter & ".txt");
44 printf "iteration\tref.node\tother.node\tgain\n" > ("gain_log." & dw_iter & ".
    txt");
45
46 # Subgradient FLAP
47 printf "iteration\tnode\tmu\n" > ("dual_flap." & dw_iter & ".txt");
48
49 #(Re) initialize Lbar to 0 between Dantzig-Wolfe iterations
50 let {(i,j) in Links} Lbar[i,j] := 0;
51 let {(i,j) in Links} Sbar[i,j] := 0;
52 let {i in Nodes} V[i] := 0;
53
54 #(Re) initialize mu
55 ##acg: mu calls mu_initial_est
56 let {i in Nodes} mu[i] := mu_initial_est[i];
57
58 ##acg: display calls flap_obj
59 let sets_this_dw_iter := 0;
60 for {iteration in Steps}{
61     display Dbar[0,0];
62     display Dbar[1,1];
63     display Dbar;
64     display Lbar;
65     expand flap_obj;
66 ## Solve FLAP
67

```

```

68
69 #           solve FLAP;
70 #           let flap_iter_time := _solve_time;
71 #           display FLAP.result;
72 #           display flap_obj;
73 #           display S, V;
74 #           display sum{(i,j) in Links} S[i,j];
75
76 #           let {i in Nodes, j in Nodes} Sbar[i,j] := S[i,j];
77 #           let {i in Nodes} Vbar[i] := V[i];
78
79 #   ## Solve QNA_FLAP for comparison;
80
81 #           expand qna_flap_obj;
82 #           solve QNA_FLAP;
83 #           solve DUAL_FLAP;
84
85 #   And really use it!  Overwrites FLAP results!
86 #           let {i in Nodes, j in Nodes} Sbar[i,j] := S[i,j];
87 #           let {i in Nodes} Vbar[i] := V[i];
88
89 #           let {i in Nodes, j in Nodes} Other_Sbar[i,j] := S[i,j];
90 #           #expand farp_obj;
91
92 #           if (snarp_not_farp == 1) then {
93 #               ## Solve SNARP instead of FARP
94 #               let {(xi, xj) in Links} local_Lbar[xi,xj] := Lbar[xi,xj];
95 #               let {xi in Nodes, xj in Nodes} local_Dbar[xi,xj] := Dbar[xi,xj];
96
97 #               for {i in Nodes}{

```

```

98
99     let flap_node := i;
100     display flap_node;
101     print "Pre-solve:";
102     display Sbar;
103     display {p in Nodes} sbar_guan_penalty_term[0, p];
104     display Lbar;
105
106     expand single_coupling;
107     expand single_dual_qna_surr_flap_obj;
108     ##acg: - calls SDQ_FLAP
109     solve SDQ_FLAP;
110     #let {xj in Nodes} Other_Sbar [flap_node, xj] := S[flap_node, xj
111         ];
112     let {xj in Nodes} TmpSbar [flap_node, xj] := S[flap_node, xj];
113     #let {xj in Nodes} Sbar [flap_node, xj] := S[flap_node, xj];
114     let LocalVbar[flap_node] := V[flap_node];
115     #let Vbar[flap_node] := V[flap_node];
116     print "Post-solve:";
117     display S;
118     display TmpSbar;
119     display LocalVbar;
120
121     let snarp_node := i;
122     display Dbar;
123
124     ## Debugging output for SNARP. Recall:
125     # maximize snarp_obj: sum{(i, j) in Links: i == snarp_node}(
126     #     (Lbar[i, j]*Sbar[i, j]*(P[i]/(Lb[i, j]*Nr))*local_D[j]*Dbar[j, i])
127     #     - (sum{(k, l) in Links: k <> i and l <> j}(

```

```

127 #       $\Gamma_1 * S_{ij} * L_{kl} * (P[i] / (L_{il} * N_r))$ 
128 #       $* local\_D[l] * D_{li}$ 
129 #       $))) +$ 
130 #       $sum\{(i, j) \text{ in Links: } j == snarp\_node\}$ 
131 #       $(L_{ij} * S_{ij} * (P[i] / (L_{ij} * N_r)) * D_{ij} * local\_D[i])$ 
132 #       $- (sum\{(k, l) \text{ in Links: } k \diamond i \text{ and } l \diamond j\}$ 
133 #       $\Gamma_1 * S_{ij} * L_{kl} * (P[k] / (L_{kj} * N_r))$ 
134 #       $* local\_D[k] * D_{kj}$ 
135 #       $))));$ 
136
137 #      display  $N_r$ ;
138 #      display  $\{xi \text{ in Nodes}\} P[xi]$ ;
139 #      display  $\{xj \text{ in Nodes}\} local\_D[xj]$ ;
140 #      display  $\{(xi, xj) \text{ in Links}\} S_{ij}$ ;
141 #      display  $\{(xi, xj) \text{ in Links}\} L_{ij}$ ;
142 #      display  $\{(xi, xj) \text{ in Links}\} D_{ij}$ ;
143 #      display  $\{(xi, xj) \text{ in Links}\} L_{ij}$ ;
144 #      display  $\{(xi, xj) \text{ in Links, } (xk, xl) \text{ in Links: } xj == snarp\_node$ 
      and  $xk \diamond xi \text{ and } xl \diamond xj\} L_{xk, xj}$ ;
145 #      display  $\{xi \text{ in Nodes, } xj \text{ in Nodes: } xi \diamond xj\} L_{ij}$ ;
146 #      display  $pat\_gain$ ;
147 #      display  $local\_B$ ;
148 # Comes up NaN:
149 #      display  $\{(xi, xj) \text{ in Links: } xi == snarp\_node\} ((L_{ij} * S_{ij} * (P[xi] / (L_{ij} * N_r)) * local\_D[xj] * D_{xj, xi})$ 
       $- (sum\{(xk, xl) \text{ in Links: } xk \diamond xi \text{ and } xl \diamond xj\}$ 
150 #       $\Gamma_1 * S_{ij} * L_{kl} * (P[xi] / (L_{il} * N_r))$ 
151 #       $* local\_D[k] * D_{kj})))$ ;
152 #
153

```

```

154 #           display{(xi, xj) in Links: xi == snarp_node} ((Lbar[xi, xj]*
Sbar[xi, xj]*(P[xi]/(Lb[xi, xj]*Nr))*local_D[xj]*Dbar[xj, xi]));
155 #           display{(xi, xj) in Links: xi == snarp_node} (sum{(xk, xl) in
Links: xk <> xi and xl <> xj and xl <> xi}(
156 #           Gamma1*Sbar[xi, xj]*Lbar[xk, xl]*(P[xi]/(Lb[xi, xl]*Nr))
157 #           *local_D[xl]*Dbar[xl, xi]));
158 #           display{(xi, xj) in Links, (xk, xl) in Links: xi == snarp_node
and xk <> xi and xl <> xj and xl <> xi} (Gamma1*Sbar[xi, xj]*Lbar[xk, xl]*(
P[xi]/(Lb[xi, xl]*Nr))*local_D[xl]*Dbar[xl, xi]);
159 #           display{(xi, xj) in Links, (xk, xl) in Links: xi == snarp_node
and xk <> xi and xl <> xj and xl <> xi} (Dbar[xl, xi]);
160 #           display Dbar;
161
162
163
164           ## Set / update local version of variables
165           let {xi in Nodes, xj in Nodes} local_Sbar[xi, xj] := Sbar[xi, xj
];
166           let {(xi, xj) in Links} local_Lbar[xi, xj] := Lbar[xi, xj
];
167           let {xi in Nodes, xj in Nodes} local_Dbar[xi, xj] := Dbar[xi, xj
];
168
169           printf "SNARP: _Iteration_%d, _at_node_%d\n", iteration ,
snarp_node >> ("debug_SNR." & dw_iter & ".txt");
170           display local_Sbar >> ("debug_SNR." & dw_iter & ".txt");
171           display Sbar >> ("debug_SNR." & dw_iter & ".txt");
172           display local_Lbar >> ("debug_SNR." & dw_iter & ".txt");
173           display local_Dbar >> ("debug_SNR." & dw_iter & ".txt");
174           display Dbar >> ("debug_SNR." & dw_iter & ".txt");

```

```

175     display LocalVbar >> ("debug_SNR." & dw_iter & ".txt");
176     display Vbar >> ("debug_SNR." & dw_iter & ".txt");
177     display {other in Nodes} snarp_component[snarp_node, other] >>
        ("debug_SNR." & dw_iter & ".txt");
178     expand alternate_snarp_obj >> ("debug_SNR." & dw_iter & ".txt"
        );
179     expand snarp_obj >> ("debug_SNR." & dw_iter & ".txt");
180     print "Alternate_SNARP_formulation" >> ("debug_SNR." & dw_iter
        & ".txt");
181
182     ##acg: - calls alt_SNARP
183     solve alt_SNARP;
184     display local_D >> ("debug_SNR." & dw_iter & ".txt");
185     display alternate_snarp_obj >> ("debug_SNR." & dw_iter & ".
        txt");
186     print "Original_SNARP_formulation" >> ("debug_SNR." & dw_iter
        & ".txt");
187     ##acg: - calls SNARP
188     solve SNARP;
189     display local_D >> ("debug_SNR." & dw_iter & ".txt");
190     display snarp_obj >> ("debug_SNR." & dw_iter & ".txt");
191
192     let {k in Nodes} temp_D[snarp_node, k] := local_D[k];
193     let {p in Pats} Bbar[snarp_node, p] := local_B[p];
194 }
195 #Apply EMWA to Dbar here, (instead of where temp_D is set, for no
        particular reason)
196
197 let {i in Nodes, j in Nodes} Dbar[i, j] :=
198     ((ewma_dbar_alpha * temp_D[i, j]) +

```

```

199             ((1 - ewma_dbar_alpha) * Dbar[i,j]));
200     }
201     else {
202     ## Solve FARP
203         ##acg: - calls FARP
204         ##acg: - calls _solve_time
205         ##acg: - calls farp_obj
206         solve FARP;
207         let farp_iter_time := _solve_time;
208         display FARP.result;
209         display farp_obj;
210         let {i in Nodes, j in Nodes} Dbar[i,j] := D[i,j];
211     }
212
213
214     display Dbar;
215
216
217     #SINGLE-NODE-FLAP
218     let {i in Nodes, j in Nodes} Sbar[i,j] := TmpSbar[i,j];
219     let {i in Nodes} Vbar[i] := LocalVbar[i];
220
221     display guan_curr_lagrangian;
222     display (guan_theta_w_star - guan_curr_lagrangian);
223     display guan_g_j**2;
224     display guan_max_sj;
225
226
227 ## Debugging / Evaluation output

```

```

228     printf "Iteration_%d,_post_FARP\n—————\n\n",
        iteration >> ("debug_SNR." & dw_iter & ".txt");
229     display {(i,j) in Links} S[i,j] >> ("debug_SNR." & dw_iter & ".txt");
230     if (snarp_not_farp == 1) then {
231         display Bbar >> ("debug_SNR." & dw_iter & ".txt");
232     }
233     printf "In-use_links\n" >> ("debug_SNR." & dw_iter & ".txt");
234     display {i in Nodes, j in Nodes: ((i,j) in Links or (j,i) in Links)
        and Sbar[i,j]==1} Dbar[i,j] >> ("debug_SNR." & dw_iter & ".txt");
235     printf "*Possible*_Interference_links\n" >> ("debug_SNR." & dw_iter &
        ".txt");
236     display {k in Nodes, j in Nodes: ((k,j) not in Links and (j,k) not in
        Links) and V[k]==1} Dbar[k,j] >> ("debug_SNR." & dw_iter & ".txt")
        ;
237     display rssMW >> ("debug_SNR." & dw_iter & ".txt");
238     display {(i,j) in Links} rssMW[i,j] >> ("debug_SNR." & dw_iter & ".txt
        ");
239
240     display rss_dBm >> ("debug_SNR." & dw_iter & ".txt");
241     display {(i,j) in Links} rss_dBm[i,j] >> ("debug_SNR." & dw_iter & ".
        txt");
242
243     display interfereMW >> ("debug_SNR." & dw_iter & ".txt");
244     display interfere_dBm >> ("debug_SNR." & dw_iter & ".txt");
245
246     display Nr >> ("debug_SNR." & dw_iter & ".txt");
247     display trueSINR >> ("debug_SNR." & dw_iter & ".txt");
248     display trueSINR_dB >> ("debug_SNR." & dw_iter & ".txt");
249
250     display SNR >> ("debug_SNR." & dw_iter & ".txt");

```



```

251     display offOK >> ("debug_SNR." & dw_iter & ".txt");
252     display interference >> ("debug_SNR." & dw_iter & ".txt");
253
254     display S >> ("debug_SNR." & dw_iter & ".txt");
255     display Sbar >> ("debug_SNR." & dw_iter & ".txt");
256     display dbg_CSC_a1 >> ("debug_SNR." & dw_iter & ".txt");
257     display dbg_CSC_a2 >> ("debug_SNR." & dw_iter & ".txt");
258     display dbg_CSC_b >> ("debug_SNR." & dw_iter & ".txt");
259
260     display CLAP_SINR_constraint >> ("debug_SNR." & dw_iter & ".txt");
261     display CLAP_SINR_constraint_real >> ("debug_SNR." & dw_iter & ".txt")
        ;
262     display Lbar_step_unscaled >> ("debug_SNR." & dw_iter & ".txt");
263
264     for {i in Nodes, p in Pats}{
265         if (snarp_not_farp == 1) then
266             printf "%d\t%d\t%d\t%g\n", iteration, i, p, Bbar[i,p] >> ("
                antenna_log." & dw_iter & ".txt");
267         else
268             printf "%d\t%d\t%d\t%g\n", iteration, i, p, B[i,p] >> ("
                antenna_log." & dw_iter & ".txt");
269     }
270
271     for {i in Nodes, j in Nodes}{
272         printf "%d\t%d\t%d\t%g\n", iteration, i, j, Dbar[i,j] >> ("
                gain_log." & dw_iter & ".txt");
273     }
274     for {(i,j) in Links}{
275         printf "%d\t%d\t%d\t%g\t%g\t%g\t%g\n", iteration, i, j, Sbar[i
                ,j], HatS[i,j], trueSINR[i,j], CLAP_SINR_constraint[i,j]

```

```

275         >> ("link_log." & dw_iter & ".txt");
276     }
277
278     for {(i,j) in Links}{
279         printf "%d\t%d\t%d\t%g\t%d\t%g\t%g\t%g\t%g\n", iteration, i, j
280             , Lbar[i,j],
281             flap_reward_use[i,j], flap_reward_SINR[i,j], flap_reward_null[
282                 i,j],
283             flap_lagrange[i,j], max_lbar[i,j] >> ("rewards." & dw_iter & "
284                 .txt");
285     }
286
287     for {i in Nodes}{
288         printf "%d\t%d\t%g\n", iteration, i, mu[i] >> ("dual_flap." &
289             dw_iter & ".txt");
290     }
291
292     #All the _bar *parameters* now contain the most recent _
293     ##variables*, so the following *parameter* calculations works:
294
295     let ssize := stepsize[iteration];
296     let mu_ss := mu_stepsize[iteration];
297     display CLAP_subgradient;
298     display CLAP_sg_norm;
299     display Lbar_step_unscaled;
300     display Lbar_step;

```

```

301     display duplex_constraint;
302     display duplex_sg_norm;
303     display mu_step;
304
305 ## Multiplier Estimation (by subgradient method)
306
307     #Track previous values
308     for {(i,j) in Links} let Lbar_prev[i,j] := Lbar[i,j];
309
310     #Enforce non-negativity for "mu" multipliers — see McShane
311     #"The Lagrange Multiplier Rule", 1973, The American
312     #Mathematical Monthly
313     ##acg: Lbar calls Lbar_suggest
314     for {(i,j) in Links} let Lbar[i,j] := max(0,Lbar_suggest[i,j]);
315
316     ##acg: mu calls mu_suggest
317     for {i in Nodes} let mu[i] := max(0, mu_suggest[i]);
318     display mu;
319     display Lbar_last_step;
320     display step_l2_norm;
321
322     #for {(i,j) in Links} let Lbar[i,j] := Lbar_suggest[i,j];
323
324     # Logging — execute after subgradient update.
325     printf "%d\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t%%g\t", iteration ,
        CLAP_obj, reduced_cost, flap_obj, farp_obj, CLAP_sg_norm,
        _ampl_time, flap_iter_time, farp_iter_time, Lbar_step_l2_norm,
        mu_step_l2_norm, step_l2_norm >> ("runlog." & dw_iter & ".txt");
326     printf{(i,j) in Links}: "%g\t", Lbar[i,j] >> ("runlog." & dw_iter & ".
        txt");

```

```

327     printf {(i,j) in Links}: "%g\t", S[i,j] >> ("runlog." & dw_iter & ".
        txt");
328     #printf {i in Nodes, p in Pats}: "%g\t", B[i,p] >> ("runlog." &
        dw_iter & ".txt");
329     printf "\n" >> ("runlog." & dw_iter & ".txt");
330
331     #     display SINR_wouldbe_ok;
332     #     display Vbar_wouldbe_ok;
333     #     display duplex_wouldbe_ok;
334     #     display link_wouldbe_ok;
335
336     ## Maintain EWMA smoothed estimate of S[i,j]
337     let {(i,j) in Links} HatS[i,j] := (
338         (ewma_hats_alpha * Sbar[i,j]) +
339         ((1 - ewma_hats_alpha) * HatS[i,j]));
340
341     printf "Starting_primal_recovery\n";
342     #restore closest_primal;
343     #expand closest_primal;
344     #solve RECOVER_PRIMAL;
345
346     ## Binary search on act_threshold for feasible set with lowest reduced
        cost
347     let act_threshold := 0.5;
348     let bs_ss := 0.25;
349     let best_rc := 1;
350     let best_thresh :=1;
351     printf "\tthreshold\tfeas.\treduced_cost\n";
352     repeat {
353         ## Try antenna cfg.

```

```

354 #           for {i in Nodes} {
355 #               let snarp_node := i;
356 #               ## Set / update local version of variables
357 #               let {xi in Nodes, xj in Nodes} local_Sbar[xi, xj] :=
358 #                   trial_S[xi, xj];
359 #               let {(xi, xj) in Links} local_Lbar[xi, xj] :=
360 #                   if (trial_S[xi, xj] == 1) then Lbar[xi, xj] else 0;
361 #               let {xi in Nodes, xj in Nodes} local_Dbar[xi, xj] :=
362 #                   Dbar[xi, xj];
363 #               solve SNARP;
364 #               display snarp_node;
365 #               expand snarp_obj;
366 #               display local_Lbar;
367 #               display local_Sbar;
368 #               display local_D;
369 #               let {k in Nodes} trial_D[i, k] := local_D[k];
370 #               let {p in Pats} trial_B[i, p] := local_B[p];
371 #           }
372 let {i in Nodes, j in Nodes} trial_D[i, j] := Dbar[i, j];
373 printf "\t%f\t%d\t%d\n", act_threshold, trial_links_ok,
           trial_rc;
374 if ((trial_links_ok == 1) and
375     (trial_rc <= best_rc)) then {
376     let best_rc := trial_rc;
377     let best_thresh := act_threshold;
378 };
379 display trial_S;
380 display trial_duplex_constraint;
381 display trial_D;
382 display trial_SINR_constraint;

```

```

383         display trial_links_ok;
384         display trial_rc;
385
386         if (trial_links_ok == 1) then {
387             # reduce threshold as long as feasibility is maintained
388             let act_threshold := act_threshold - bs_ss;
389         }
390         else {
391             # increase theshold until feasible
392             let act_threshold := act_threshold + bs_ss;
393         };
394
395
396         let bs_ss := bs_ss/2;
397     } while (bs_ss > 0.002);
398     let act_threshold := best_thresh;
399     #display trial_duplex_constraint;
400     #display trial_SINR_constraint;
401     #display trial_links_ok;
402     #display trial_rc;
403
404     printf "Closest_primal_point:\n";
405     printf "—————\n";
406     display trial_S;
407     display act_threshold;
408     display trial_rc;
409     display trial_links_ok;
410     #display PrimalV;
411     printf "distance:_%f\n", closest_primal;
412     printf "—————\n";

```

```

413
414     if (trial_rc < rc_thresh) then {
415         printf "\nPost-optimized_antennas_for_candidate_schedule:\n";
416         #solve improve_antennas;
417         #display {i in Nodes, j in Nodes} (trial_D[i,j], D[i,j]);
418         printf "Pre-opt_SINR_info\n";
419         display trial_SINR_constraint;
420         display t_trueSINR_dB;
421     # let {i in Nodes, j in Nodes} trial_D[i,j] := D[i,j];
422     # printf "Post-opt SINR info\n";
423     # display trial_SINR_constraint;
424     # display t_trueSINR_dB;
425         display B;
426         display Lb;
427         display pat_gain;
428         display max_link_impact_factor;
429         display pissing_budget;
430     for {blah in Nodes} {
431         let this_node := blah;
432         ##acg: - calls dist_improve_antennas
433         solve dist_improve_antennas;
434         let {j in Nodes} tmp_trial_D[this_node,j] := local_D[j
435             ];
436         display blah, local_D;
437         display blah, local_B;
438     }
439     let {i in Nodes, j in Nodes} trial_D[i,j] := tmp_trial_D[i,j];
440     printf "Post-dist-opt_SINR_info\n";
441     display trial_SINR_constraint;
442     display t_trueSINR_dB;

```

```

442
443
444     if (exists {l in LinkSets} (not exists {(i,j) in Links}
445         (trial_S[i,j] > link_in_set[l,i,j]))) then { #must not be
           equal to OR DOMINATED BY existing link set
446
447         print "Rejected_link_set!";
448         display {l in LinkSets}: {(i,j) in Links} link_in_set[l,i,j];
449         display trial_S;
450     } else {
451         let num_link_sets := num_link_sets+1;
452         let index := next(index, LinkSets);
453         let {(i,j) in Links} link_in_set[index,i,j] := trial_S[i,j];
454         let sets_this_dw_iter := sets_this_dw_iter + 1;
455
456     #         close ("runlog." & dw_iter & ".txt");
457     #         close ("debug_SNR." & dw_iter & ".txt");
458     #         close ("antenna_log." & dw_iter & ".txt");
459     #         close ("gain_log." & dw_iter & ".txt");
460     #         close ("link_log." & dw_iter & ".txt");
461     #         close ("rewards." & dw_iter & ".txt");
462     #         close ("dual_flap." & dw_iter & ".txt");
463
464         print "Added_link_set. Continuing";
465         display trial_S;
466
467         print "Added_link_set. Continuing" >> dw_log.txt;
468         display {(i,j) in Links} trial_S[i,j] >> dw_log.txt;
469
470         if (sets_this_dw_iter >= num_sets_per_dw_iter) then {

```



```

471         #set Sbar so that reduced_cost parameter is correct.
472         let {(i,j) in Links} Sbar[i,j] := trial_S[i,j];
473         break;
474     }
475 }
476 }
477
478
479     ## Try short-circuit conditions
480     ##acg: - calls CLAP_SINR_constraint
481     ##acg: - calls duplex_constraint
482     ##acg: - calls reduced_cost
483     if ((iteration > abs_min_iter) and
484         (reduced_cost < rc_thresh) and
485         (forall {(i,j) in Links} CLAP_SINR_constraint[i,j] > 0) and
486         (forall {i in Nodes} duplex_constraint[i] <= 0) and
487         (forall {(i,j) in Links} (Sbar[i,j] == 0 or Sbar[i,j]==1)) #no
488             fractional activation
489         ) then {
490
491         for {l in LinkSets} {
492             if (not exists {(i,j) in Links} (Sbar[i,j] > link_in_set[l,i,j]
493                 ))) then {
494                 print l;
495                 print "LinkSet_>=_Sbar_";
496                 display Sbar;
497                 display {(xi,xj) in Links} link_in_set[l,xi,xj];
498             }
499         }

```

```

499     if (exists {l in LinkSets} (not exists {(i,j) in Links}
500         (Sbar[i,j] > link_in_set[l,i,j]))) then { #must not be
           equal to OR DOMINATED BY existing link set
501
502         print "Rejected_link_set!";
503         display {l in LinkSets}: {(i,j) in Links} link_in_set[l,i,j];
504         display Sbar;
505     } else {
506         let num_link_sets := num_link_sets+1;
507         let index := next(index, LinkSets);
508         let {(i,j) in Links} link_in_set[index,i,j] := Sbar[i,j];
509         let sets_this_dw_iter := sets_this_dw_iter + 1;
510
511     #         close ("runlog." & dw_iter & ".txt");
512     #         close ("debug_SNR." & dw_iter & ".txt");
513     #         close ("antenna_log." & dw_iter & ".txt");
514     #         close ("gain_log." & dw_iter & ".txt");
515     #         close ("link_log." & dw_iter & ".txt");
516     #         close ("rewards." & dw_iter & ".txt");
517     #         close ("dual_flap." & dw_iter & ".txt");
518
519         print "Added_link_set. Continuing";
520         display Sbar;
521
522         print "Added_link_set. Continuing" >> dw_log.txt;
523         display {(i,j) in Links} Sbar[i,j] >> dw_log.txt;
524
525         if (sets_this_dw_iter >= num_sets_per_dw_iter) then break;
526     }
527 }
```

```

528
529     ## Termination conditions: Note that counting up to max_iter steps
        will also break
530     if ((iteration > early_exit_min_iter) and
531         (Lbar_step_l2_norm <= small_enough_lbar_step) and
532         (mu_step_l2_norm <= small_enough_mu_step)) then {
533
534     ## Make sure HatS gets to catch up, and values are stable.
535         display Lbar_step_l2_norm;
536         display Lbar_step_l2_norm >> dw_log.txt;
537         display mu_step_l2_norm;
538         display mu_step_l2_norm >> dw_log.txt;
539         close ("runlog." & dw_iter & ".txt");
540         close ("debug_SNR." & dw_iter & ".txt");
541         close ("antenna_log." & dw_iter & ".txt");
542         close ("gain_log." & dw_iter & ".txt");
543         close ("link_log." & dw_iter & ".txt");
544         close ("rewards." & dw_iter & ".txt");
545         close ("dual_flap." & dw_iter & ".txt");
546         print "Breaking: _step_size_l2_norm->_0.";
547         print "Breaking: _step_size_l2_norm->_0." >> dw_log.txt;
548         break;
549     }
550
551
552 #break;
553
554 } #End "for iteration in Steps"
555 close ("runlog." & dw_iter & ".txt");
556 close ("debug_SNR." & dw_iter & ".txt");

```

```

557 close ("antenna_log." & dw_iter & ".txt");
558 close ("gain_log." & dw_iter & ".txt");
559 close ("link_log." & dw_iter & ".txt");
560 close ("rewards." & dw_iter & ".txt");
561 close ("dual_flap." & dw_iter & ".txt");

```

## D.2.2 Off-Line Evaluation

Listing D.15: compare-versions.ampl

```

1  ## Basic command file
2  reset;
3
4  # 1: JBSS-FLAP
5  # 2: Centralized Dantzig-Wolfe
6  # 3: Pseudo-distributed
7  param exec_mode;
8
9  data compare_mode.dat;
10
11
12 ##acg: include central_ea_stdma.mod
13 model central_ea_stdma.mod;
14 ##acg: include stdma_auto.dat
15 data stdma_auto.dat;
16
17 param snarp_not_farp = 1;
18
19 ## Restricted Master Problem
20
21 display q;
22

```

```

23 # Initialize columns with simple TDMA
24 ##acg: - calls index
25 ##acg: - calls LinkSets
26 param index in LinkSets;
27 for {(i,j) in Links}{
28     ##acg: - calls num_link_sets
29     let num_link_sets := num_link_sets +1;
30     if (num_link_sets == 1) then let index := first(LinkSets);
31     else let index := next(index, LinkSets);
32     let link_in_set[index,i,j] := 1;
33 }
34
35 param dw_iter integer >0;
36 let dw_iter := 1;
37 param old_num_link_sets integer >0;
38 param rc_alias default 0;
39 printf "D-W_Log:\n—————\n\n" > dw_log.txt;
40 repeat{
41     printf "DBG_0!\n";
42     printf "Iteration:_%d\n—————\n", dw_iter >> dw_log.txt;
43     printf "DW_Iteration:_%d\n—————\n", dw_iter;
44     ## display link_in_set >> dw_log.txt ;
45     ##acg: - calls RMP
46     solve RMP;
47     printf "Beginning_Objective:_%d\n", RMP_obj >> dw_log.txt;
48     printf "DW_iteration_beginning_objective:_%d\n", RMP_obj;
49     display x >> dw_log.txt;
50     ## Use real dual values for beta_t!
51     let {(i,j) in Links} beta_t[i,j] := max(demand_coverage[i,j].dual,0);
52

```

```

53     ##acg: display calls beta_t
54     display beta_t >> dw_log.txt;
55     ##acg: - calls old_num_link_sets
56     ##acg: - calls num_link_sets
57     let old_num_link_sets := num_link_sets;
58
59 ## Mode decision here!
60     if (exec_mode == 3) then {
61         ##acg: include stdma_subproblem.ampl
62         commands stdma_subproblem.ampl;
63     }
64     if (exec_mode == 2) then {
65         commands solve_clap.ampl;
66     }
67     printf "Subproblem_terminated\n" >> dw_log.txt;
68     printf "Subproblem_terminated\n";
69
70     ##acg: - calls rc_alias
71     ##acg: - calls reduced_cost
72     ##acg: rc_alias calls reduced_cost
73     let rc_alias := reduced_cost;
74     printf "DBG_1!\n";
75     ##acg: - calls num_link_sets
76     ##acg: - calls old_num_link_sets
77     if (num_link_sets != old_num_link_sets) then {
78         printf "Number_of_new_link_set(s):_%d\n", (num_link_sets -
79             old_num_link_sets) >> dw_log.txt;
80     }
81     else {

```

```

81         printf "No_new_link_sets_(even_with_good_reduced_cost)_->_no_
           primal_feasible_solution." >> dw_log.txt;
82         break;
83     }
84     #display Sbar >> dw_log.txt;
85     printf "DBG_2!\n";
86     ##acg: - calls dw_iter
87     let dw_iter := dw_iter + 1;
88     printf "DBG_3!\n";
89     ##acg: display calls dw_iter
90     display reduced_cost >> dw_log.txt;
91     printf "DBG_3.5!\n";
92     ##acg: - calls rc_alias
93     ##acg: - calls rc_thresh
94 } while (rc_alias < -0.9);      ###XXX made-up threshold. Otherwise, jittering
           beta_t allows infinite stream of infinitesimal improvements.
95 printf "DBG_4!\n";
96
97 printf "\n" >> dw_log.txt;
98 printf "-----\n" >> dw_log.
           txt;
99 printf "|_____Execution_Complete_____|\n" >> dw_log.
           txt;
100 printf "-----\n\n" >> dw_log.
           txt;
101 printf "Post-loop_status_(Iterations=%d)\n-----\n",
           dw_iter >> dw_log.txt;
102
103 display {z in LinkSets}: {(i,j) in Links} link_in_set[z,i,j] >> dw_log.txt;
104

```





```

26
27 display logS , logS.slack , logS.rc ;
28 display logD , logD.slack , logD.rc ;
29 #display logV , logV.slack , logV.rc ;
30 display B , B.slack , B.rc ;
31
32 let{(i,j) in Links} inferredS[i,j] := exp(logS[i,j]);
33 display logS ;
34 #display derivedS ;
35 display inferredS ;
36
37 # let{(i,j) in Links} inferredT[i,j] := exp(logT[i,j]);
38 # display logT ;
39 # display inferredT ;
40
41 # let{i in Nodes} inferredV[i] := exp(logV[i]);
42 # display logV ;
43 # display inferredV ;
44
45 #display CCLAP_SINR1.slack ;
46 #display CCLAP_SINR2.slack ;
47
48 expand CCLAP_SINR_NEW ;
49
50 display CCLAP_SINR_NEW ;
51 display CCLAP_SINR_NEW.slack ;
52
53 # display CCLAP_SV ;
54 # display CCLAP_SV.slack ;
55

```

```

56 display CCLAP_DUPLEX;
57 display CCLAP_DUPLEX.slack;
58
59
60 display CCLAP_DUPLEX_PAIRWISE;
61 display CCLAP_DUPLEX_PAIRWISE.slack;
62
63 # display  $\{(i,j) \text{ in Links}\} : \{k \text{ in Nodes} : (i,k) \text{ in Links and } k \diamond i \text{ and } k \diamond j\}$ 
        CCLAP_DUPLEX_PAIRWISE1[i,j,k];
64 # display CCLAP_DUPLEX_PAIRWISE1.slack;
65
66 # display CCLAP_DUPLEX_PAIRWISE2;
67 # display CCLAP_DUPLEX_PAIRWISE2.slack;
68
69 # display CCLAP_DUPLEX_PAIRWISE3;
70 # display CCLAP_DUPLEX_PAIRWISE3.slack;
71
72 # display CCLAP_DUPLEX_PAIRWISE4;
73 # display CCLAP_DUPLEX_PAIRWISE4.slack;
74
75
76 let {i in Nodes, j in Nodes} inferredD[i,j] := exp(logD[i,j]);
77 display logD;
78 display inferredD;
79
80 display B;
81
82 #solve CLAP;
83
84 # # Information output

```

```

85 # display beta_t;
86 # display Links;
87 # display VBIN;
88 # display SBIN;
89 # display D;
90 # display B;
91
92 let {(i,j) in Links} Sbar[i,j] := inferredS[i,j];
93 # let {i in Nodes} Vbar[i] := inferredV[i];
94 let {i in Nodes, j in Nodes} Dbar[i,j] := inferredD[i,j];
95
96 display Sbar;
97 display Dbar;
98
99 display CLAP_SINR_constraint_real;
100
101 display rx_signal_MW;
102 display rx_nonsignal_MW;
103 display better_interfere_MW;
104 display betterSINR;
105 display betterSINR_dB;
106
107 #Forcing to boolean
108 #let {(i,j) in Links} Sbar[i,j] := if (inferredS[i,j] >= almost_one) then 1
      else 0;
109 #let {i in Nodes} Vbar[i] := if (inferredV[i] >= almost_one) then 1 else 0;
110 #let {i in Nodes, j in Nodes} Dbar[i,j] := inferredD[i,j];
111
112 display Sbar;
113 display Dbar;

```

```
114
115 ## Add results to D-W master problem
116 let num_link_sets := num_link_sets+1;
117 let index := next(index, LinkSets);
118 let {(i,j) in Links} link_in_set [index, i, j] := Sbar[i, j];
```

There are no further listings.