Stony Brook University

Improving Capacity and Connectivity in Wireless Access Networks

A Dissertation Presented
by
Anand Prabhu Subramanian

to
The Graduate School
in Partial Fulfillment of the
Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science
Stony Brook University

May 2009

Stony Brook University
The Graduate School

Anand Prabhu Subramanian

We, the dissertation committee for the above candidate for
the Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dr. Samir R. Das, Dissertation Advisor
Associate Professor, Department of Computer Science

Dr. Himanshu Gupta, Chairperson of Defense
Associate Professor, Department of Computer Science

Dr. Jie Gao, Committee Member
Assistant Professor, Department of Computer Science

Dr. Milind M. Buddhikot, Outside Committee Member
Member of Technical Staff, Wireless Network Elements Research,
Bell labs, Alcatel Lucent Inc.

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Improving Capacity and Connectivity in Wireless Access Networks

by

Anand Prabhu Subramanian

Doctor of Philosophy

in

Computer Science

Stony Brook University

2009

There is an increasing demand for high quality ubiquitous mobile broadband wireless access in recent times. More and more computing devices are becoming mobile and are equipped with high processing power, memory capability, optimized user interfaces, and a variety of communication interfaces. So people using these devices expect good connectivity and high quality usage experience. In addition to hand held mobile devices, vehicles are also equipped with devices that need wireless access. Applications that are used by mobile users also demand high bandwidth and very good link quality.

IEEE 802.11 based wireless network access is very attractive due to high data rate connectivity, low deployment cost, and free or low-cost network access to users. 802.11 based wireless networks, however, suffer from serious interference problems limiting their capacity due to the broadcast nature of the wireless medium and their use of the unlicensed spectrum band. They also suffer from poor connectivity due to shorter coverage area and highly varying wireless channel. To achieve good performance, two important issues need to be addressed – *capacity* improvement in dense deployments such as wireless infrastructures as well as efficient *connectivity* of mobile clients to the infrastructures.

In this dissertation, we address both these challenges by proposing efficient MAC and network layer solutions that exploit physical layer diversities such as *multi-channel* support and *directional communication*.

First, we address the capacity issue in broadband wireless mesh infrastructures where each wireless node is equipped with multiple radio interfaces. We develop practical centralized and distributed channel assignment techniques that minimize overall network interference and perform very close to lower bounds on the optimal. Experimental study on a multi-radio mesh testbed revealed interesting practical challenges such as interface and channel heterogeneity. We propose mechanisms to incorporate heterogeneous channel information to improve capacity further. To combat interference further, we study the benefits of using directional communication at each radio in the mesh nodes and propose a directional MAC protocol that reduces wireless interference through higher spatial reuse. We study various scenarios in which deafness and directional hidden terminal problems could occur and solve them efficiently.

Next, we investigate mechanisms to improve connectivity in wireless networks in highly mobile scenarios. We study the use of steerable beam directional antennas to improve the duration and quality of connectivity between moving vehicles and fixed infrastructures (*V2I scenarios*), and between multiple vehicles (*V2V scenarios*) when they move at high speed. The main challenge here is that the directional antenna should be steered appropriately on a continuous basis when the vehicles are moving in order to maintain good link quality throughout the entire drive. To address this challenge, we develop a framework called *MobiSteer* that provides practical approaches to perform beam steering and access point selection in a vehicle-to-infrastructure (V2I) setting. The goal is to improve the signal-to-noise ratio (SNR) and reduce handoff overheads. MobiSteer is then extended to vehicle-to-vehicle (V2V) scenarios. On experimental testbeds, both these techniques are demonstrated to provide tremendous improvement in average SNR with resultant improvement in physical layer data rates and/or communication ranges. Finally, as a by-product of our MobiSteer framework, we develop a triangulation based technique to localize roadside 802.11 APs. Such localizations will be useful

in obtaining realistic data sets about large-scale chaotic WiFi deployments in urban areas for use in modeling and analysis.

Dedicated to my Amma, Appa and Raja.

Contents

List of Figures xi

List of Tables xvi

Acknowledgements xvii

1 Introduction 1
- 1.1 IEEE 802.11 based broadband wireless access network architecture 2
- 1.2 Improving capacity in wireless multi-hop backbone 4
- 1.3 Improving client connectivity to wireless infrastructure 5
- 1.4 Obtaining realistic WiFi network topologies 6
- 1.5 Dissertation Organization 7

2 Channel Assignment in Wireless Mesh Networks 8
- 2.1 Introduction 8
- 2.2 Problem Formulation 11
- 2.3 Related Work 15
- 2.4 Centralized Tabu-based Algorithm 18
- 2.5 Distributed Greedy Algorithm (DGA) 21
- 2.6 Bounds on Optimal Network Interference 25
  - 2.6.1 Semidefinite Programming Formulation 25
  - 2.6.2 Linear Programming Formulation 30
- 2.7 Generalizations 33
- 2.8 Simulation Results 35

2.8.1 Graph-Theoretic Performance Metric 35
2.8.2 ns2 Simulations 38
2.9 Experimental Evaluation 41
2.9.1 Testbed 42
2.9.2 Generating communication graph and conflict graph . . 43
2.9.3 Performance Results 44
2.10 Understanding Channel and Interface Heterogeneity 45
2.10.1 Testbeds 46
2.11 Channel Diversity 47
2.11.1 Long term variation of RSS 47
2.11.2 Relation between RSS and delivery ratio 48
2.11.3 Link behavior in different channels 48
2.12 Interface Diversity 51
2.13 Channel Assignment Algorithm 53
2.14 Summary 55

3 Directional MAC Protocol for Wireless Mesh Networks 57
3.1 Introduction 57
3.2 Deafness and Directional Hidden Terminal Problem Scenarios 59
3.3 Related Work 60
3.4 Antenna Model and Assumptions 62
3.4.1 Packet Transmission 62
3.4.2 Packet Reception 63
3.5 Directional MAC Protocol 63
3.5.1 Discussion 66
3.6 Performance Evaluation 68
3.7 Summary 78

4 Vehicle to Roadside Communication 79
4.1 Introduction 79
4.1.1 MobiSteer Design Goals 81
4.1.2 Contributions 82
4.2 MobiSteer Architecture 83
4.2.1 Hardware Setup 83

4.2.2 Software Setup 86
4.2.3 Operational Modes 87
4.2.4 Data Collection 89
4.3 Cached Mode Operation 91
4.3.1 Optimal AP and Beam Selection 91
4.4 Cached mode: Experimental Results 95
4.4.1 Scenarios 95
4.4.2 Collecting and Analyzing Data 98
4.4.3 Optimal Steering Results 102
4.4.4 Experiments with *In Situ* Networks 103
4.5 Online Mode 104
4.6 Related Work 107
4.7 Summary 109

5 Inter-Vehicular Communication 111
5.1 Introduction 111
5.2 Experimental Setup and Scenarios 114
5.2.1 Sender and Receiver Programs 114
5.2.2 Beam Switching Latency 116
5.2.3 Beam Index Annotation 117
5.2.4 Measurements Scenarios 118
5.3 Experimental Results 119
5.3.1 Evidence of Directionality 122
5.3.2 Best Beam Combination Vs. Omni-Omni 123
5.3.3 Best Beam Combination Vs. LOS 125
5.3.4 Cars Driving In Opposite Direction 126
5.4 Practical Beam Steering 128
5.4.1 Computation of LOS Beam 130
5.4.2 Experimental Results 132
5.5 Related Work 134
5.6 Summary 136

6 Localizing Roadside WiFi Networks . . . 138
6.1 Introduction . . . 138
6.2 Experimental Scenarios and Data Collection Method . . . 141
6.2.1 Experimental Scenarios . . . 141
6.2.2 Data Collection . . . 141
6.2.3 Determining Orientation . . . 145
6.3 Localization Algorithm . . . 145
6.3.1 Preliminaries . . . 145
6.3.2 Understanding and Modeling Reflections . . . 147
6.3.3 Modeling Reflections by k-Means Clustering . . . 149
6.3.4 Choosing Real AP Location from k Images . . . 150
6.3.5 Learning k for Clustering . . . 151
6.4 Performance Evaluation . . . 152
6.4.1 Benefit of Using Directional Antennas and AOA . . . 152
6.4.2 Benefit of Modeling Reflection Using Clustering . . . 153
6.4.3 Impact of GPS Accuracy . . . 154
6.4.4 Impact of Car Speed . . . 155
6.5 Related Work . . . 157
6.6 Summary . . . 158

7 Conclusion . . . 160

Bibliography . . . 161

List of Figures

1 IEEE 802.11 based Broadband Wireless Access Network Architecture 3

2 Communication graph and corresponding conflict graph. 12

3 Merge operation of second phase. The two figures are the communication graphs of the network before and after the merge operation. Labels on the links denote the color/channel. Here, the merge operation is started at node i by changing all its 1-colored links to color 2. 22

4 Fractional network interference of solutions delivered by various algorithms compared with the lower bounds in dense or sparse networks for 3 or 12 channels. 36

5 Saturation throughput in ns2 simulations for 12 channels and various traffic models, viz., (a) Single hop, (b) Multi-hop Peer-to-Peer, (c) Multi-hop Gateway. 39

6 Saturation throughput in ns2 simulations when using non-orthogonal channels with 802.11b-like multi-channel model (11 channels with varying degrees of interference; 3 channels are mutually orthogonal). 41

7 Aggregate network throughput in the network when transmit power of each node is set to 11dBm (a) Two radio case (b) Three radio case. 42

8 Aggregate network throughput in the network when transmit power of each node is set to 15dBm (a) Two radio case (b) Three radio case. 42

9 Multi radio wireless mesh node constructed using a Soekris net4801 embedded computer consisting of 4 Atheros chipset based miniPCI 802.11a/b/g wireless cards. The antennas are separated from each other of by a distance approximately 0.5 m. 43
10 Characteristics of RSS metric. 49
11 Variation of RSS and delivery ratio using different channels on sample links in our two testbeds. 49
12 Link behavior across different channels in the two testbeds. . . 51
13 Interface heterogeneity in multi-radio nodes in 802.11a testbed. 52
14 Aggregate number of packets received when a set of 10 links transmit packets simultaneously. Each sample run consists of a different set of 10 links in the network. 54
15 Deafness scenarios. 58
16 Directional hidden terminal problem due to unheard RTS/CTS. 60
17 Control window mechanism to prevent directional hidden terminal problem 64
18 Scenario (i) : Scenario allowing high spatial reuse. 68
19 Throughput of different flows in Scenario (i). 69
20 Scenario (ii) : Scenario to study deafness. 70
21 Throughput of different flows in scenario (i). The curves for CW-DMAC Flow 3 → 2 and CW-DMAC Flow 1 → 2 overlap. 71
22 Snapshot of backoff values chosen by node 1 and node 3 72
23 Scenario (iii) : Scenario to study directional hidden terminal problem. 72
24 Throughput of flow from node 2 to node 3 in scenario (ii). . . 73
25 Linear topology used to demonstrate the cascading effects of deafness. 73
26 Throughput of the flow 1 → 4 in the linear topology. 74
27 RTS Retransmission in a linear topology. 75
28 5 x 5 Multihop Grid topology. 75
29 Aggregate throughput in grid topology. 76
30 Aggregate throughput in a 30 node random topology. 77

31 Beam steering and AP selection to improve connectivity. . . . 80
32 Block diagram of directional antenna components. 82
33 The beam patterns for Phocus Array antenna: (a) omni-directional; (b) two directional beams. The concentric circles are 10db per division. Taken from [1]. 84
34 Hardware setup. 85
35 *MobiSteer* operation. 88
36 Performance of active probing vs. passive scanning in building RF signature database. 90
37 (a) Notion of representative trajectory. (b) Mapping of GPS samples. 91
38 Experimental environments for the controlled experiments: (a) a large empty parking lot in Stony Brook university campus, (b) the graduate students' apartment complex in the same campus. The black dots show the locations of the APs and the black arrow shows the driving route. 94
39 Performance for individual beams for a specific AP along the drive. 96
40 Demonstrating sampling error: (a) average PHY-layer data rate at every segment with 90% confidence interval; (b) variations in the best beam selection. 97
41 Estimated and observed performance on the best <*AP,beam*> along with the omni-directional beam performance for the entire path 98
42 Measurement data demonstrating the relationship between SNR and PHY-layer data rate in our setup. 102
43 (a) Average SNR from *In Situ* APs in each 40 meter segment along the route with beam steering and with omni-directional beam. (b) Distance along the route each AP was heard with beam steering and with omni-directional beam. 103
44 Total Number of bytes received along the drive in "controlled" experiments (apartment complex scenario). 106

45 Average SNR in each segment along the drive in *In Situ* experiments. 107

46 Beam steering to improve quality of V2V links between two moving vehicles in opposite direction. 111
47 Transmission and reception protocol stack architecture. . . . 115
48 Measurement scenarios: (a) Highway (b) Suburban. The red arrow shows the driving routes. 117
49 Average SNR (in dB) in each of 9×9 beam combinations for two representative samples. 120
50 SNR improvement (in dB) for the best beam combination over the omni-omni combination on a scale of link distance. 121
51 CDF of SNR improvement (in dB) for the best beam combination compared to the omni-omni combination (cars driving in the same direction). 123
52 How often do LOS beams match with the best beams? (Cars driving in the same direction.) 124
53 SNR differential (in dB) between the best beam combination and LOS beams (cars driving in the same direction. 125
54 Average communication ranges when using the best beam combination and the omni-omni combination (cars driving in opposite direction). 126
55 CDF of SNR improvement (in dB) for the best beam combination compared to the omni-omni combination (cars driving in opposite direction). 127
56 (a) Frequency of match between the the best beam combination and LOS beams; (b) SNR differential (in dB) between the best beam combination and LOS beams (cars driving in opposite direction). 128
57 LOS beam computation. Beam indices increase in the anti-clockwise direction. 130
58 SNR improvement (in dB) when using LOS beams vs omni-omni combination in (a) Highway (b) Suburban scenario. 131

59 Physical layer data rate improvement when using LOS beams vs omni-omni combination in (a) Highway (b) Suburban scenario. 131

60 Drive-by localization of roadside APs. 139
61 Three experimental scenarios along with the AP locations and the driving trajectories. The black stars are the locations of the APs. The black curve is the trajectory followed by the *MobiSteer* Node. 142
62 (a) Angular error using center of directional beam. (b) Angular error considering beam width. (c) A scenario demonstrating reflection. 144
63 (a), (b): AP location and directions of the strongest beams in two scenarios demonstrating the impact of reflections. (c): CDF of the angular error of the strongest beam from the actual direction towards the AP in two scenarios. 146
64 It is impossible to tell the true location of the AP apart from the images, as the same information may admit two (or more) feasible configurations. 151
65 Relationship between distance and RSS in (a) parking lot (b) apartment complex. 152
66 Comparison of DrivebyLoc and VORBA. 153
67 (a) Localization errors for DrivebyLoc and trilateration.(b) CDF of the GPS errors in two scenarios. (c) Impact of speed and number of runs on localization error. 156

List of Tables

1 Aggregate TCP Throughput in Random Topology 78

Acknowledgements

First and foremost, I am greatly indebted to my adviser Prof. Samir R. Das for his invaluable advice and guidance during my graduate studies. He is an extremely approachable person and is always open for interesting discussions. He gave me great freedom to independently think about new research problems and provided several informative pointers to literature articles to improve my problem solving skills. We had innumerable brainstorming meetings that were extremely lively and productive. His constructive criticisms were very useful in directing my ideas in the right direction. Prof. Das encouraged me to work in the area of experimental wireless networking research and with his guidance I developed invaluable skills to build practical and efficient networking systems and critically evaluate their performance using systematic measurements. Apart from research, he also guided me in several personality development aspects. He taught me about good work habits and team spirit which I believe would be very helpful in my future career. I also thank his wife June for inviting me to several get-togethers and parties at their home that were extremely enjoyable.

I am very fortunate to have worked with several professors at Stony Brook on several research projects. In particular, I worked with Prof. Himanshu Gupta on several projects in the area of dynamic spectrum allocation and channel assignment in mesh networks. Prof. Gupta greatly helped me in developing strong algorithmic skills to solve several optimization problems efficiently. I also closely worked with Prof. Jie Gao. I am always amazed by her strong background in computational geometry and the way she applies them to networking problems. I learned several interesting geometric techniques while working with her in the localization project and would like to thank her

for her guidance.

I spent three exciting summer semesters in Bell Labs, NJ. My mentor Milind Buddhikot at Bell labs is a very accommodating and inspiring person who I highly respect and thank for providing me with invaluable support during my graduate studies and particularly during my internship periods. He introduced me to cutting edge industry problems in the area of dynamic spectrum management in cellular networks. I believe, the experiences I gained while working at Bell labs helped me become a practical problem solver who can build efficient networking systems. My manager Scott Miller at Bell labs helped me in several aspects during my internships. It was a great pleasure to work with him and I greatly thank him for all his help. I also would like thank Theodoros Salonidis, Henrik Lundgren and Christophe Diot who I worked with during my internship at Thomson, Paris for their great support and guidance.

At Stony Brook, I worked with several students. Notably my senior students at WINGS lab such as Vishnu Navda, Bin Tang, Shweta Jain, Zhongheng Zhou, Xianjin Zhu and Anand Kashyap, Ritesh Maheshwari were extremely helpful and exciting to work with. When I joined WINGS lab in summer 2005, they welcomed me with so much warmth and helped me get acquainted to wireless systems research. I also enjoyed the company of my other lab mates Kannan Dhanasekaran, Praveen Durairaj, Pralhad Deshpande, Utpal Kumar Paul, Kanthimathinathan Sambandan, Naveen Ramaraj who kept a lively atmosphere and an enjoyable working environment throughout my graduate studies. I also like to thanks Jing Cao, Andreas Timm-Giel and Markus Becker who visited our lab during my stay in Stony Brook for their exicting discussions.

Last but not the least, I would like to express my heart-felt gratitude and sincere thanks to my mother Mrs. Leela Subramanian, father Dr. Subramanian and my brother Raja whose unconditional love and support, drove me forward to make this far in my life and career, and to continue on the road ahead. I dedicate this disseration to them.

Chapter 1

Introduction

In recent times, there is a surge in the use of mobile computing devices like laptops, pocket PCs, PDAs, mobile phones etc. among users. The processing power, memory capabilities, user interfaces and communication interfaces (like WiFi, bluetooth, cellular etc.) of these devices have evolved a lot, so users desire better connectivity and high quality usage experience 'anytime and anywhere'. In addition to hand-held mobile devices, vehicles such as cars and mass transportation services like buses and trains are also equipped with devices that need wireless access. Providing network connectivity to moving vehicles can generate several interesting applications like general purpose internet connectivity to passengers in mass transport systems, providing emergency traffic and road safety information, rear-seat passenger infotainment etc. With the increase in mobility of computing devices in one end, applications that are used by mobile users demand high bandwidth and very good quality link on the other end. This includes realtime downloading of maps, multimedia contents, high-definition audio and video, video chats, traffic and weather information. Voice over IP (VOIP) based phones are becoming more popular and they require very good quality links for a good call experience. So there is a huge interest in developing technologies that can provide high quality ubiquitous broadband wireless access to users 'on the fly'.

Improving *capacity* and *connectivity* in wireless networks are two grand challenges to achieve this goal. In dense deployments and high traffic scenarios, wireless networks are often interference-limited necessitating techniques in

capacity improvement. In highly mobile scenarios on the other hand, the focus is often on maintaining good link layer connectivity to other mobile nodes or to a physical infrastructure. In this dissertation, we address both these challenges in the contest of IEEE 802.11 based wireless access networks by proposing efficient MAC layer and network layer solutions that exploit lower layer diversities such as *multi-channel* support and *directional communication*.

IEEE 802.11 based wireless network access is highly attractive due to the wide spread availability of WiFi interfaces in user devices, high data rate connectivity, very low deployment cost and free or cheap network access to end users. Several enterprises and universities have deployed large scale 802.11 networks to service a number of users. Numerous urban areas are already covered with dense deployments of 802.11 wireless networks and efforts to form community mesh networks using them are under way [6]. In this dissertation, first, we explore the use of multiple channels and directional communication to improve the capacity of the wireless infrastructures. Next, we study the use of steerable beam directional antennas on mobile clients to improve the duration and quality of connectivity between the mobile clients and the wireless infrastructure, and between multiple mobile clients.

1.1 IEEE 802.11 based broadband wireless access network architecture

We consider a broadband wireless access network architecture based on IEEE 802.11 [90] based networks. Consider a typical urban neighborhood, where several wireless routers are deployed in people's home or as part of commercial deployments [4, 13]. These wireless routers form an multi-hop wireless mesh infrastructure. Wireless mesh networks [30] are multi-hop networks of wireless routers as shown in Figure 1 (shown by the blue color dashed lines). Each mesh router consist of multiple radio interfaces out of which one of the interface acts as an 'access' interface used to connect to the clients. The other interfaces are called the 'relay' interfaces and form part of the wireless backbone. Some of the wireless routers (called as Gateways) have Internet backhaul connectivity

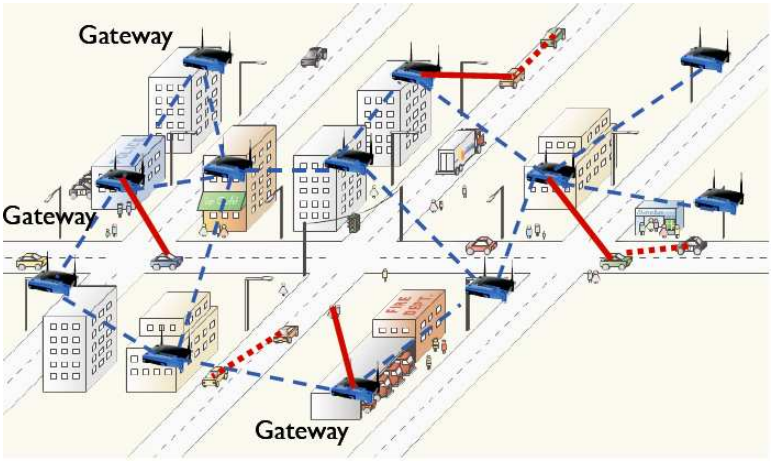


Figure 1: IEEE 802.11 based Broadband Wireless Access Network Architecture

to the wired network. The gateways are connected to the Internet via wired (Ethernet) or wireless (1xRTT, EV-DO, 802.16) uplinks. The relay interfaces in routers are used to construct a self-configuring, secure, managed, power-adaptive packet forwarding backbone between the relay and gateway nodes.

Each mesh router replays packets from wireless clients if the gateway nodes are not within range. Mesh clients like laptops, PDAs, pocket PCs and mobile vehicles connect to their nearest mesh router using the 'access' interface and reach the wired network through the multi-hop mesh backbone. We consider connectivity from highly mobile clients such as cars and mass transportation services like trains and buses that connect to the wireless infrastructure (shown by red color solid lines in Figure 1). We also consider connectivity between multiple mobile clients (shown by red color dotted lines in Figure 1) to support interesting vehicle-to-vehicle applications.

1.2 Improving capacity in wireless multi-hop backbone

An important design goal for dense wireless deployments such as wireless mesh networks is *capacity*. It is well-known that wireless interference severely limits network capacity in multi-hop settings [82] due to the broadcast nature of the wireless channel. Use of multiple channels provide a simple opportunity to perform bandwidth aggregation to increase capacity [102]. The basic idea is to use orthogonal (non-interfering) channels for neighboring wireless transmissions. The current 802.11 standard [90] (also used for mesh networks) indeed provides several orthogonal channels to facilitate this. There are about 3 orthogonal channels in 802.11b standard and 12 orthogonal channels in 802.11a standard. We develop centralized and distributed quasi-static channel assignment algorithms whose objective is to minimize the overall interference in the network there by improving the capacity of the network. We compare their performance with lower bounds achieved using a linear and semi-definite formulation of our channel assignment problem and show they perform very close to the bounds on the optimal solution. Experimental study of our channel assignment algorithms on a multi-radio mesh testbed revealed several practical challenges that were not addressed in previous studies based on simulations or theoretical analysis. In practice, different channels for the same link exhibit different link qualities depending on the propagation environment and different interfaces on the same node also exhibit link quality variations due to hardware differences and required antenna separations. We propose a general technique to use channel specific link quality information in channel assignment algorithms to improve capacity further.

Next, we study the use of directional communication to improve the capacity of the mesh backbone. Directional antennas can concentrate radio signal energy in a particular direction (main lobe), instead of radiating it in all directions like their omni-directional counterpart [154, 153, 132]. So the transmission on a directional antenna can potentially cause much lesser interference (along side and back lobes), thereby giving a significant capacity advantage in multi-hop wireless networks. Similarly, reception on directional antenna is

subject to lesser interference. We design an efficient directional MAC protocol that addresses traditional problems like deafness and directional hidden terminal problems [57] and solve them using a single radio interface and single channel. Our directional MAC protocol can work on each relay interface in the mesh router, below our channel assignment protocol to improve the capacity of the backbone further.

1.3 Improving client connectivity to wireless infrastructure

We study the use of steerable beam directional antennas to improve the duration and quality of connectivity between moving vehicles and fixed infrastructures (*V2I scenarios*), and between multiple vehicles (*V2V scenarios*) when they move at high speed. We study a physical layer enhancement – directional communication – to improve network connectivity in vehicular context. By focusing energy in one direction, a directional antenna can get a better transmit or receive gain for a targeted direction compared to an omni-directional antenna. Each mobile node is equipped with a steerable beam directional antenna and communicates with infrastructure using directional beams. However, having a directional antenna alone is not sufficient for a moving vehicle. The direction must be steered appropriately for the best link quality. The steering must be done on a continuous basis as the car moves so that good connectivity can be maintained to the appropriate network node. The beam steering should be done in such a way so as to increase the duration of connectivity and improve the link quality. To address this challenge, we develop a framework called *MobiSteer* that provides practical approaches to perform beam steering and AP selection. The goal is to select the best AP and beam combination at each point along the drive given the available information, so that the throughput can be maximized. We conduct extensive field studies and provide experimental results in controlled as well as in-situ scenarios. We demonstrate that our approach can give up to $2-4$ times improvement in link throughput and more than 2 times improvement in duration of connectivity

compared to using omni-directional communication.

Beam steering in vehicle-to-vehicle communication scenarios is more challenging as both the communicating nodes move at high speed. We conduct systematic experiments using steerable beam directional antenna mounted on two vehicles in different multi-path environments and with various drive patterns. From our measurement data, we observe that when the antenna beams in the two vehicles directly point to each other, SNR improvements are similar to the case when the best beam combination was chosen via a comprehensive, but expensive, scanning and probing method. Using this observation, we come up with a simple and practically feasible beam steering approach that provides more than 2 times improvement in duration and quality of connectivity.

1.4 Obtaining realistic WiFi network topologies

Many urban regions have a high density of WiFi APs – deployed in a "chaotic" fashion [29] in homes and businesses, in campuses and hotspots, or as a part of a metro or municipal WiFi effort [4]. There is little knowledge about the nature of these networks, e.g., density, connectivity, interference properties, etc. The first step in understanding their nature is estimating locations of the APs. We use the MobiSteer framework to localize roadside WiFi networks using a triangulation based approach augmented with a clustering algorithm to account for multi-path reflections in cluttered environments. Such localizations will be useful in obtaining realistic data sets about large-scale chaotic WiFi deployments in urban areas for use in modeling and analysis. Experimental evaluation of our approach in highly reflective environments shows an order of magnitude improvement in localization accuracy compared to existing trilateration based approaches using omni-directional antenna.

1.5 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we present our channel assignment algorithms and show the performance improvement obtained using extensive simulation and experimental studies. Chapter 3 describes our directional MAC protocol that solves the deafness and directional hidden terminal problem in multi-hop wireless networks and improves network capacity. In Chapter 4, we present our MobiSteer framework and describe our beam steering and AP selection algorithms. In Chapter 5, we study the use of steerable beam directional antenna for inter-vehicular communication. Chapter 6 describes our localization technique using the MobiSteer framework. We conclude this dissertation in Chapter 7.

Chapter 2

Channel Assignment in Wireless Mesh Networks

2.1 Introduction

In this chapter, we address the challenge of improving capacity in the wireless mesh network backbone by using multiple channels to minimize wireless interference. Essentially, wireless interference can be minimized by using orthogonal (non-interfering) channels for neighboring wireless transmissions. The current IEEE 802.11 standard for WLANs (also used for mesh networks) indeed provides several orthogonal channels to facilitate the above. Presence of multiple channels requires us to address the problem of which channel to use for a particular transmission; the overall objective of such an assignment strategy is to minimize the overall network interference.

Dynamic Channel Assignment. One of the channel assignment approaches is to frequently change the channel on the interface; for instance, for each packet transmission based on the current state of the medium. Such *dynamic channel assignment* approaches [156, 143, 36, 141] require channel switching at a very fast time scale (per packet or a handful of packets). The fast-channel switching requirement makes these approaches unsuitable for use with commodity

hardware, where channel switching delays itself can be in the order of milliseconds [50] which is an order of magnitude higher than typical packet transmission times (in microseconds). Some of the dynamic channel assignment approaches also require specialized MAC protocols or extensions of 802.11 MAC layer, making them further unsuitable for use with commodity 802.11 hardware.

Static or Quasi-static Channel Assignment. Due to the difficulty of use of above dynamic approach with commodity hardware, there is need to develop techniques that assign channels statically [135, 134, 110, 28, 147]. Such static assignments can be changed whenever there are significant changes to traffic load or network topology; however, such changes are infrequent enough that the channel-switching delay and traffic measurement (see Section 2.2) overheads are inconsequential. We refer to the above as *quasi-static channel assignments*. If there is only one radio interface per router, then the above channel assignment schemes will have to assign the *same* channel to all radios/links in the network to preserve network connectivity. Thus, such assignment schemes require use of multiple radio interfaces at each node. Due to board crosstalk or radio leakage [28, 138], commodity radios on a node may actually interfere even if they are tuned to different channels. However, this phenomena can be addressed by providing some amount of shielding or antenna separation [138], or increased channel separation (as is the case in 802.11a) [134].

Problem Addressed. In this chapter, we address the problem of quasi-static assignment of channels to links in the context of networks with multi-radio nodes. The objective of the channel assignment is to minimize the overall network interference. Channel assignment is done as some variation of a graph coloring problem; but it has an interesting twist in the context of mesh networks. The assignment of channels to links must obey the *interface constraint* that the number of different channels assigned to the links incident on a node is at most the number of interfaces on that node. Different variations of this problem have been shown to be NP-hard [135, 110] before. Thus, efficient algorithms that run reasonably fast and provide good quality solutions are of interest. Since computing the optimal is intractable and approximation

algorithms are still an open question, we take the approach of computing a *bound on the optimal* using mathematical programming approaches, and develop heuristics that perform very close to the obtained bounds on the optimal.

Our Contributions. For the above described channel assignment problem, we develop a centralized and a distributed algorithm. The centralized algorithm is based on a popular heuristic search technique called Tabu search [86] that has been used in the past in graph coloring problems. The distributed approach is motivated by the greedy approximation algorithm for Max K-cut problem in graphs [73]. To evaluate their performances, we develop two mathematical programming formulations, using semidefinite programming (SDP) and integer linear programming (ILP). We obtain *bounds* on the optimal solution by relaxing the ILP and SDP formulations to run in polynomial time. Finally, detailed ns-2 simulations as well as experimental study in a 11-node multi-radio testbed demonstrate the full performance potential of the channel assignment algorithms in 802.11 based multi-radio mesh networks.

The *salient features of our work* that set us apart from the existing channel assignment approaches on multi-radio platforms are as follows.

- Our approach is "topology preserving," i.e., all links that can exist in a single channel network also exist in the multichannel network after channel assignment. Thus, our channel assignment does not have any impact on routing.
- Our approach is suitable for use with commodity 802.11-based networks without any specific systems support. We do not require fast channel switching or any form of MAC layer or scheduling support. While our algorithms indeed use interference and traffic models as input, such models can be gathered using experimental methods.
- Our work generalizes to non-orthogonal channels, including channels that are supposedly orthogonal but interfere because of crosstalk or leakage [138].
- Ours is the first work that establishes good lower bounds on the optimal network interference, and demonstrates good performance of the

developed heuristics by comparing them with the lower bounds.

2.2 Problem Formulation

In this section, we first present our network model and formulate of our channel assignment problem.

Network Model. We consider a wireless mesh network with stationary wireless routers where each router is equipped with a certain (not necessarily same) number of radio interfaces. We model the *communication graph* of the network as a general undirected graph over the set of network nodes (routers). An edge (i, j) in the communication graph is referred to as a *communication link* or *link*, and signifies that the nodes i and j can communicate with each other as long as both the nodes have a radio interface each with a common channel. There are a certain number of channels available in the network. For clarity of presentation, we assume for now that the channels are orthogonal (non-interfering), and extend our techniques for non-orthogonal channels in Section 2.7.

Interference Model. Due to the broadcast nature of the wireless links, transmission along a communication link (between a pair of wireless nodes) may interfere with transmissions along other communication links in the network. Two interfering links cannot engage in successful transmission at the same time if they transmit on the same channel. The *interference model* defines the set of links that can interfere with any given link in the network. There have been various interference models proposed in the literature, for example, the physical and protocol interference models [82, 91]. The discussion in this chapter is independent of the specific interference model used as long as the interference model is defined on pairs of communication links.

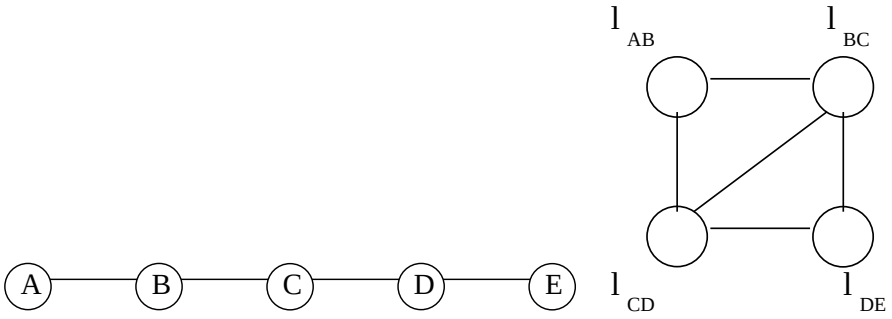
For clarity of presentation, we assume a *binary interference model* for now (i.e., two links either interfere or do not interfere), and generalize our techniques to fractional interference in Section 2.7. Moreover, in our approach of quasi-static channel assignment, the level of interference between two links actually depends on the traffic on the links. However, for clarity of presentation,

we assume uniform traffic on all links for now, and generalize our techniques to non-uniform traffic in Section 2.7.

Conflict Graph. Given an interference model, the set of pairs of communication links that interfere with each other (assuming them to be on the same channel) can be represented using a *conflict graph* [91]. To define a conflict graph, we first create a set of vertices V_c corresponding to the communication links in the network. In particular,

$$V_c = \{l_{ij} \mid (i,j) \text{ is a communication link}\}.$$

Now, the conflict graph $G_c(V_c, E_c)$ is defined over the set V_c as vertices, and a *conflict edge* (l_{ij}, l_{ab}) in the conflict graph is used to signify that the communication links (i,j) and (a,b) interfere with each other if they are on the same channel. The above concept of a conflict graph can be used to represent any interference model. As defined above, the conflict graph does not change with the assignment of channels to vertices in the conflict graph.



(a) Communication graph. (b) Conflict graph.

Figure 2: Communication graph and corresponding conflict graph.

We illustrate the concept of conflict graph in Figure 2. The wireless network represented in Figure 2 has five network nodes $A, B, \ldots, E$ and four communication links as shown in the communication graph (see Figure 2(a)). The conflict graph (see Figure 2(b)) has four nodes each representing a communication link in the network. In this figure, we assume an 802.11 like interference model where the transmission range and interference range are equal. When

RTS/CTS control messages are used links within two hops interfere. Thus, the communication link (A, B) interferes with the communication links (B, C) and (C, D), and not with (D, E).

Notations. Here, we introduce some notations that we use throughout this chapter.

- N, the set of nodes in the network.
- R_i, the number of radio interfaces on node $i \in N$.
- $\mathcal{K} = \{1, 2, \ldots, K\}$, the set of K channels.
- $V_c = \{l_{ij} \mid (i, j)$ is a communication link$\}$.
- $G_c(V_c, E_c)$, the conflict graph of the network.
- For $i \in N$, $E(i) = \{l_{ij} \in V_c\}$, i.e., $E(i)$ is set of vertices in V_c that represent the communication links incident on node i.

In addition, throughout this chapter, we use variables u, v to refer to vertices in V_c, variables i, j, a, b to refer to nodes in N, and the variable k to refer to a channel. Since assigning channel can be thought of as coloring vertices, we use the terms channel and colors interchangeably throughout in this chapter.

Channel Assignment Problem. The problem of channel assignment in a multi-radio wireless mesh network can be informally described as follows. Given a mesh network of router nodes with multiple radio interfaces, we wish to assign a unique channel to each communication link[1] in the network such that the number of different channels assigned to the links incident on any node is at most the number of radios on that node. Since we assume uniform traffic on all links for now, we assign channels to all links, and define the *total network interference* as the number of pairs of communication links that are interfering (i.e., are assigned the same channel and are connected by an edge in the conflict graph). The objective of our problem is to minimize the above

[1]Note that merely assigning channels to radios is not sufficient to measure network interference/capacity, since a link still can use one of many channels for transmission.

defined total network interference, as it results in improving overall network capacity [82].

A note is in order regarding our choice of optimization objective. While a natural objective for the channel assignment problem would be to directly maximize overall network throughput, it turns out that modeling network throughput analytically in a random access based medium access model is hard. The previous works in the literature ([91, 101, 32, 106]) that maximize overall network throughput assume a time-slotted synchronized medium access model with scheduling as one part of the problem. In a time-slotted synchronized medium access model, modeling throughput is much easier relative to a CSMA based random access model. There indeed has been several attempts in recent literature to model link capacity [92, 129, 136] in a CSMA based network using measurements from real 802.11-based networks. However, these models are quite complex and it is difficult to use them as objective functions for an optimization problem and at the same time develop efficient solution approaches. In our work, our interest is in developing solutions for use with commodity systems based on 802.11. Thus, in our channel assignment problem we use an objective function than can be formally defined using the conflict graph model and optimized efficiently. With this argument, network interference is a more practical choice relative to network throughput.

Consider a wireless mesh network over a set N of network nodes. Formally, the *channel assignment problem* is to compute a function $f : V_c \rightarrow \mathcal{K}$ to minimize the *overall network interference* $I(f)$ defined below while satisfying the below *interface constraint*.

Interface Constraint.

$$\forall i \in N, \quad |\{k \mid f(e) = k \text{ for some } e \in E(i)\}| \leq R_i.$$

Network Interference $I(f)$.

$$I(f) = |\{(u, v) \in E_c \mid f(u) = f(v)\}|. \tag{1}$$

If we look at assignment of channels to vertices as coloring of vertices, then the network interference is just the number of monochromatic edges in

the vertex-colored conflict graph. The channel assignment problem is NP-hard since it reduces to Max K-cut (as discussed below).

Input Parameters – Measuring Interference and Traffic. Note that, under the simplying assumption of uniform traffic, the only input to our channel assignment problem is the network conflict graph. The conflict graph (along with the edge weights for fractional interference; see Section 2.7) can be computed using methods similar to recently reported measurement-based techniques in [126, 136]. These techniques are localized, due to the localized nature of interference, and hence, can be easily run in a distributed manner. Also, in most cases (for static network topologies), the above measurements need to be done only one-time. For the case of non-uniform traffic, we need to measure average (over the time scale of channel assignment) traffic (i.e., the function $t(.)$ of Section 2.7) on each link. Such traffic measurements can be easily done using existing software tools (e.g., COMO [20]).

Relationship with Max K-cut. Given a graph G, the Max K-cut problem [73] is to partition the vertices of G into K partitions in order to maximize the number of edges whose endpoints lie in *different* partitions. In our channel assignment problem, if we view vertices of the conflict graph assigned to a particular channel as belonging to one partition, then the network interference is actually the number of edges in the conflict graph that have endpoints in *same* partition. Thus, our channel assignment problem is basically the Max K-cut problem with the added interface constraint. Since Max K-cut is known to be NP-hard, our channel assignment problem is also NP-hard.

2.3 Related Work

The use of multiple channels to increase capacity in a multihop network has been addressed extensively. Generally, there have been two types of approaches, viz., (i) Fast switching of channels (possibly, on a per-packet basis) on a single radio, or (ii) Assigning channels to radios for an extended period of time in a multi-radio setting.

Fast Switching of Channels. In MMAC protocol [143], the authors augment the 802.11 MAC protocol such that the nodes meet at a common channel periodically to negotiate the channels to use for transmission in the next phase. In SSCH [36], the authors propose dynamic switching of channels using pseudo-random sequences. The idea is to randomly switch channels such that the neighboring nodes meet periodically at a common channel to communicate. In DCA [156], the authors use two radios - one for the control packets (RTS/CTS packets) and another for data packets. The channel to send the data packet is negotiated using the control packets and the data packets are sent in the negotiated channels. In AMCP [141], the authors uses similar notion of a control channel, but a single radio and focus on starvation mitigation. In [78] the authors use a channel assignment approach using a routing protocol and then use these channels to transmit data. For coordination, control channels are used. In [109] two radio and single radio multichannel protocols are proposed, but separate control channels are not needed.

All the above protocols require a small channel switching delay (of the order of hundred microseconds or less), since channels are switched at a fast time scale (possibly, on a per-packet basis). But, the commodity 802.11 wireless cards incur a a channel switching delay of the order of milliseconds (based on our observations), as channel switching requires a firmware reset and execution of an associated procedure. Similar experiences were reported in [50], and in particular, it has been shown in [36, 143] that packet-based channel assignment may not be feasible in a practical setting [150]. In addition, the above approaches require changes to the MAC layer. Thus, the above approaches are not suitable with currently available commodity hardware.

Static/Quasi-Static Channel Assignment in Multiradio Networks. There have been many works that circumvent fast channel switching by assigning channels at a much larger time scale in a multiradio setting. This solution is deemed more practical as there is neither a need to modify the 802.11 protocol or need for interfaces with very low channel switching latency.

In particular, [134] assume a tree-based communication pattern to ease

coordination for optimizing channel assignment. Similar tree-based communication patterns have been used in [158]. The above schemes do not quantify the performance of their solutions with respect to the optimal. In addition, [147] considers minimum-interference channel assignments that preserve k-connectivity. None of the above schemes preserve the original network topology, and hence, may lead to inefficient assignments and routing in a more general peer-to-peer communication.

Topology Preserving Schemes. To facilitate independent routing protocols, our work focusses on developing quasi-static channel assignment strategies that preserve the original network topology. Prior works on topology preserving channel assignment strategies are as follows. Adya et al. [28] propose a strategy wherein they assume a hard-coded assignment of channels to interfaces, and then determine which channel/interface to use for communication via a measurement-based approach. They do not discuss how the channels are assigned to interfaces. In [135], Raniwala et al. propose a centralized load-aware channel assignment algorithm; however, they require that source-destination pairs with associated traffic demands and routing paths be known a priori. In [62], Das et al. present a couple of optimization models for the static channel assignment problem in a multi-radio mesh network. However, they do not present any practical (polynomial time) algorithm. In [130], the authors propose a linear optimization model channel allocation and interface assignment model. Their model differs from ours in the sense that they assign channels to interfaces, and then, assign interfaces to neighbors so that neighors having interfaces with common channels can communicate. In contrast, in our model, we assign channels to links directly. In addition, [29] assumes binary interference and a uniform traffic model.

In [131], a purely measurement-based approach is taken for channel assignment to radios (instead of links). Here, one radio at each node is tuned to a common channel to preserve the original topology; however, this can be wasteful when only a few interfaces are available. Moreover, assignment of channels to radios still leaves the problem of which channel to use for a transmission/link. In [96], the authors propose a simple greedy algorithm for

channel assignment in multi-radio networks. They assume a binary interference model and do not show any performance bounds.

In the most closely related work to ours, Marina and Das in [110] address the channel assignment to communication links in a network with multiple radios per node. They propose a centralized heuristic for minimizing the network interference. We compare the performance of our proposed algorithm with this heuristic, and show a significant improvement.

Other Related Works. In other related works, [103] proposes a hybrid channel assignment strategy: some interfaces on a node have a fixed assignment, and the rest can switch channels as needed. To put things in perspective, our work presents algorithms for making these fixed assignments. Authors in [91, 101, 32, 106] address joint channel assignment, routing, and scheduling problems. These papers make an assumption of synchronized time-slotted channel model as scheduling is integrated in their methods. This makes modeling network throughput straightforward and consideration of a joint channel assignment and routing problem practical. However, the synchronized time-slotted model is hard to implement in commodity radios that use 802.11, as in 802.11 scheduling is done following a CSMA-based random access paradigm. In addition, these works often make impractical assumptions. For example, [91]'s approach requires enumeration of all maximal sets of non-interfering links (independent sets), and [101] considers networks with bounded "interference degrees."

2.4 Centralized Tabu-based Algorithm

In this section, we describe one of our algorithms for the channel assignment problem, based on the Tabu search [86] technique for coloring vertices in graphs. Our Tabu-based algorithm is centralized. Centralized algorithms are quite practical in "managed" mesh networks where there is already a central entity. Moreover, they are amenable to a higher degree of optimization, easier to upgrade, and use of "thin" clients. Centralized approaches have indeed been proposed in various recent works [110, 135, 147], and have also become prevalent in the industry (e.g., WLAN and mesh products from Meru Networks [14],

Tropos [22]).

Algorithm Overview. Recall that our channel assignment problem is to color the vertices V_c of the conflict graph G_c using K colors while maintaining the interface constraint and minimizing the number of monochromatic edges in the conflict graph. In other words, the channel assignment problem is to find a solution/function $f : V_c \rightarrow \mathcal{K}$ with minimum network interference $I(f)$ such that f satisfies the interference constraint. Our Tabu-based algorithm consists of two phases. In the first phase, we use Tabu search based technique [86] to find a good solution f without worrying about the interface constraint. In the second phase, we remove interface constraint violations to get a feasible channel assignment function f.

First Phase. In the first phase, we start with a random initial solution f_0 wherein each vertex in V_c is assigned to a random color in $\mathcal{K}$. Starting from such a random solution f_0, we create a sequence of solutions $f_0, f_1, f_2, \ldots, f_j, \ldots$, in an attempt to reach a solution with minimum network interference. In the j^{th} iteration ($j \geq 0$) of this phase, we create the next solution f_{j+1} in the sequence (from f_j) as follows.

The j^{th} Iteration. Given a solution f_j, we create f_{j+1} as follows. First, we generate a certain number (say, r) of random neighboring solutions of f_j. A random neighboring solution of f_j is generated by picking a random vertex u and reassigning it to a random color in $(\mathcal{K} - \{f_j(u)\})$. Thus, a neighboring solution of f_j differs from f_j in the color assignment of only one vertex. Among the set of such randomly generated neighboring solutions of f_j, we pick the neighboring solution with the lowest network interference as the next solution f_{j+1}. Note that we do not require $I(f_{j+1})$ to be less than $I(f_j)$, so as to allow escaping from local minima.

Tabu List. To achieve fast convergence, we avoid reassigning the same color to a vertex more than once by maintaining a *tabu list* τ of limited size. In particular, if f_{j+1} was created from f_j by assigning a new color to a vertex u, then we add $(u, f_j(u))$ to the tabu list τ. Now, when generating random neighboring solutions, we ignore neighboring solutions that assign the color k to u if (u, k) is in τ.

Termination. We keep track of the best (i.e., with lowest interference) solution f_{best} seen so far by the algorithm. The first phase terminates when maximum number (say, i_{max}) of allowed iterations have passed without any improvement in $I(f_{best})$. In our simulations, we set i_{max} to $|V_c|$. Since network interference $I(f)$ takes integral values and is at most $(|V_c|)^2$, the value $I(f_{best})$ is guaranteed to decrease by at least 1 in $i_{max} = |V_c|$ iterations (or else, the first phase terminates). Thus, the time complexity of the first phase is bounded by $O(rd|V_c|^3)$, since each iteration can be completed in $O(rd)$ time where r is the number of random neighboring functions generated and d is the maximum degree of a vertex in the conflict graph. Note that network interference of a neighboring solution can be computed in $O(d)$ time. A formal description of the first phase is shown in Algorithm 1.

Algorithm 1: First Phase of Tabu-based Algorithm.

Input : Conflict Graph $G_c(V_c, E_c)$; Set of channels $\mathcal{K}$.
Output: Channel Assignment Function $f_{best} : V_c \rightarrow \mathcal{K}$.

```
Start with a random assignment function f_0;
f_best = f_0;  I_best = I(f_0);  τ = null;  j = 0;  i = 0;
while I(f_i) > 0 and i ≤ i_max do
    Generate r random neighbors of f_j;
     Each neighbor is generated by randomly picking
     a u in V_c and k ∈ K s.t. k ≠ f_j(u) and (u, k) ∉ τ,
     and changing f_j(u) to k
    Let f_{j+1} be the neighbor with lowest interference.
    Add (u, f_j(u)) to τ.
    If τ is full, delete its oldest entry;
    if (I(f_{j+1}) < I_best)
      then  I_best = I(f_{j+1});  f_best = f_{j+1};  i = 0;
      else i = i + 1;
    endif;
    j = j + 1;
end while
RETURN f_best;
```

Second Phase. Note that the solution f returned by the first phase may violate interface constraints. Thus, in the second phase, we eliminate the interface constraints by repeated application of the following "merge" procedure.

Given a channel/color assignment solution f, we pick a network node for the merge operation as follows. Among all the network nodes wherein the interface constraint is violated, i.e, whose number of radios is less than the number of distinct colors assigned to the incident communication links, we pick the node wherein the difference between the above two terms is the maximum. Let i be the node picked as above for the merge operation. We reduce the number of colors incident on i by picking (as described later) two colors k_1 and k_2 incident on i, and changing the color of all k_1-colored links to k_2. In order to ensure that such a change does not create interface constraint violations at other nodes, we *iteratively* "propagate" such a change to all $k_1 - colored$ links that are "connected" to the links whose color has been just changed from k_1 to k_2. Here, two links are said to be connected if they are incident on a common node. Essentially the above propagation of color-change ensures that for any node j, either *all or none* of the k_1-colored links incident on j are changed to color k_2. See Figure 3. Completion of the above described color-change propagation marks the completion of *one* merge procedure. The above described merge procedure reduce the number of distinct colors incident on i by one, and does not increase the number of distinct colors incident on any other node (due to the all or none property). Thus, repeated application of such a merge operation is guaranteed to resolve all interface constraints. Note that a merge operation probably will result in increase in network interference. Thus, for a given node i, we pick those two color k_1 and k_2 for the merge operation that cause the least increase in the network interference due to the complete merge operation.

2.5 Distributed Greedy Algorithm (DGA)

In this section, we describe our Distributed Greedy Algorithm (DGA) for the channel assignment problem. Our choice of greedy approach is motivated by the following two observations.

Max K-cut Problem in Random Graphs. As described before, the Max K-cut problem on a given graph G is to partition the vertices of G into K disjoint subsets such that the sum of number of edges with endpoints in different

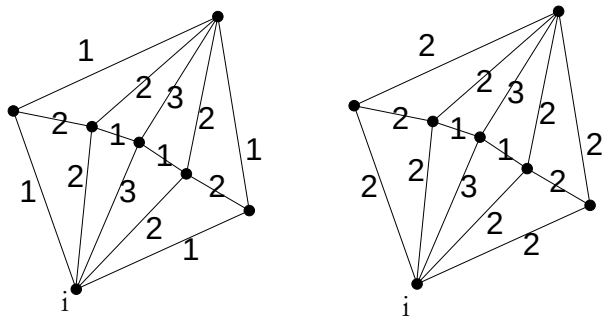


Figure 3: Merge operation of second phase. The two figures are the communication graphs of the network before and after the merge operation. Labels on the links denote the color/channel. Here, the merge operation is started at node i by changing all its 1-colored links to color 2.

partitions is maximized. In [59], the authors consider $G_{n,p}$ graphs which are defined as random graph over n vertices where each edge exists with a uniform probability of p. The authors design an algorithm with an approximation ratio $1 - \frac{1}{Kx}$ (where $x \geq 1$) for the Max K-cut problem in such $G_{n,p}$ graphs. In particular, they obtain a lower bound on the size of the Max K-cut in $G_{n,p}$ graphs problem using a simple greedy heuristic, and obtain an upper bound using a relaxed semidefinite program given by [73]. They show that the lower and upper bounds are close with very high probability. In effect, the authors show that the greedy heuristic delivers a $1 - \frac{1}{Kx}$ factor approximation solution with very high probability. The greedy heuristic proposed in [59] for Max K-cut works by deciding the partition of one vertex at a time in a greedy manner (i.e., place the vertex in the partition that results in maximizing the number of edges with endpoints in different partitions).

Conflict Graph is $G_{n,p}$. It can be shown that a network formed by randomly placed nodes in a fixed region generates a random conflict graph G_c which is also $G_{n,p}$. Here, we assume an interference model wherein two communication links (u, v) and (r, s) interfere with each other depending on the locations of

the nodes u, v, r, and s (as is the case with protocol interference model [82]). Now, the vertices $l_{u,v}, l_{r,s} \in V_c$ representing the communication links (u, v) and (r, s) are connected in G_c if and only if the communication links (u, v) and (r, s) interfere with each other. Thus, the probability of an edge between two vertices of V_c depending only on the locations of the involved network nodes, and since the network nodes are randomly placed, the probability of an edge between two vertices in V_c is uniform.

The above observations motivate use of a greedy approach for our channel assignment problem.

Centralized Greedy Algorithm. We start with presenting the centralized version, which yields a natural distributed implementation. In the initialization phase of our greedy approach, each vertex of V_c is colored with the color 1. Then, in each iteration of the algorithm, we try to change the color of some vertex in a greedy manner without violating the interface constraint. This strategy is different from the Tabu-based algorithm, where we resolve interface constraint violations in the second phase while not worrying about introducing them in the first phase. In each iteration of the greedy approach, we try to change the color of some vertex $u \in V_c$ to a color k. We look at all possible pairs of u and k, considering only those that do not result in the violation of any interface constraint, and pick the pair (u, k) that results in the largest decrease in network interference. The algorithm iterates over the above process, until there is no pair of u and k that decreases the network interference any further. Note that a vertex in V_c may be picked multiple times in different iterations. However, we are guaranteed to terminate because each iteration monotonically decreases the network interference. In particular, as noted in previous section, since the network interference takes integral values and is at most $(|V_c|)^2$, the number of iterations of the greedy algorithm is bounded by $(|V_c|)^2$. Since each iteration can be completed in $O(dK|V_c|)$, where K is the total number of colors and d is the maximum degree of a vertex in the conflict graph, the total time complexity of the greedy algorithm is $O(dK|V_c|^3)$. The pseudocode for the centralized verison of the greedy algorithm is shown in Algorithm 2.

Algorithm 2: Centralized Greedy Algorithm.

Input : Conflict Graph $G_c(V_c, E_c)$; Set of channels $\mathcal{K}$.
Output: Channel Assignment Function $f : V_c \rightarrow \mathcal{K}$.
Initialization:
$f(u) = 1, \forall u \in V_c$
Repeat
(1) Choose the pair $(u, k) \in (V_c \times \mathcal{K})$, such that when $f(u)$ is assigned to k, the
interference constraint is not violated and the total network interference $(I(f))$
decreases the most
(2) Set $f(u) = k$
Until $I(f)$ cannot be decreased any further.

Distributed Greedy Algorithm (DGA). The above described greedy approach can also be easily distributed by using a localized greedy strategy. The distributed implementation differs from the centralized implementation in the following aspects. Firstly, in the distributed setting, multiple link-color pairs may be picked simultaneously across the network by different nodes. Secondly, the decision of which pair is picked is based on the local information. Lastly, to guarantee termination in a distributed setting, we impose additional restriction that each pair (u, k) is picked at most once (i.e., each vertex $u \in V_c$ is assigned a particular color k at most once) in the entire duration of the algorithm.

In the distributed implementation, each vertex $u = l_{ij} \in V_c$ corresponding to the link (i, j) is *owned* by i or j, whichever has the higher node ID. This is done to ensure consistency of color information across the network. Initially, each vertex in V_c is assumed to colored 1. Let $m \geq 1$ be the parameter defining the local neighborhood of a node. Based on the information available about the colors of links in the m-hop neighborhood of i, each network node i selects (after waiting for a certain random delay) a (u, k) combination such that (i) $u = l_{ij}$ is owned by i, (ii) changing the color of u to k does not violate the interface constraint at node i or j, (iii) the pair (u, k) has not been selected before by i, and (iv) the pair (u, k) results in the largest decrease in the "local" network interference. Then, the node i sends a `ColorRequest` message to node

j. The node j responds with the `ColorReply` message, if and only if changing the color of u to k still does not violate the interface constraint at node j. On responding with the `ColorReply` message, the node j *assumes*[2] that the color of u has been changed to k. On receiving the `ColorReply` message for j, the node i sends a `ColorUpdate`(u, k) message to all its m-hop neighbors. If a `ColorReply` message is not received within a certain time period, the node i abandons the choice of (u, k) for now, and starts a fresh iteration. Since each pair (u, k) is picked at most once, then the total number of iterations (over all nodes) in the above algorithm is at most $O(|V_c|K)$. The pseudocode for the distributed greedy algorithm that runs in every node $i \in V$ is shown in Algorithm 3.

The above Distributed Greedy algorithm is localized, and can be made to work in dynamic topologies. Our simulation results showed that the above distributed algorithm performs almost same as the centralized version, due to the localized nature of the network interference objective function. The input network parameters of traffic and interference are measured as discussed in Section 2.2.

2.6 Bounds on Optimal Network Interference

In this section, we derive lower bounds on the minimum network interference using semidefinite and linear programming approaches. These lower bounds will aid in understanding the quality of the solutions obtained from the algorithms presented in previous two sections.

2.6.1 Semidefinite Programming Formulation

In this section, we model our channel assignment problem in terms of a semidefinite program (SDP).

[2]Such an assumption may need to be later corrected through communication with i if the `ColorUpdate(u,k)` message is not received from i within a certain amount of time.

Algorithm 3: Distributed Greedy Algorithm for each node $i \in V$

Input : "Local" network and conflict graph; set of channels $\mathcal{K}$.

Output: Channel Assignment (i.e., $f(u)$) for all links $u \in V_c$ incident on node i.

Repeat

Among all pairs (u, k) where $u \in V_c$ is owned by i and $k \in \mathcal{K}$ that is not already chosen and does not violate interface constraint at i

choose the one which produces largest decrease in local interference.

Send `ColorRequest`(u, k) to node j where $u = (i, j)$.

Wait for `ColorReply`(u, k) message from node j.

If `ColorReply`(u, k) message is not received within a certain time

Abandon the choice (u, k).

Until Local interference cannot be decreased any further, or all (u, k) combinations have already been chosen.

When `ColorRequest`(u, k) message is received from node j, where $u = (i, j)$:

If assigning channel k to link u does not cause interface constraint violation

Send `ColorReply`(u, k) message to node j.

When `ColorReply`(u, k) message is received from node j:

Set $f(u) = k$ and send `ColorUpdate(u,k)` message to "local" neighborhood

When `ColorUpdate`(u, k) message is received:

Update locally maintained channel assignment of links in the local network graph.

Semidefinite Programs. A *semidefinite program* [77] is a technique to optimize a linear function of a symmetric positive-semidefinite matrix[3] subject to linear equality constraints. Semidefinite programming is a special case of convex programming [149], since a set of positive semidefinite matrices constitutes a convex cone. Semidefinite programs can be solved in polynomial time using various techniques [81]. The reader is referred to [77, 33] for further details on semidefinite programming and its application to combinatorial optimization. The standard form of semidefinite program is as follows.

$$\begin{array}{lll} \text{Minimize} & \text{C.X} & \\ \text{such that} & \text{A}_\text{i}.\text{X} = b_i, & 1 \leq i \leq m, \text{ and} \\ & X \succeq 0 & \end{array}$$

where C, A_i $(\forall i)$, and X are all symmetric $n \times n$ matrices, and b_i is a scalar vector. The constraint $X \succeq 0$ implies that the variable (to be computed) matrix X must lie in the closed, convex cone of a positive semidefinite matrix. Also, the . (dot) operation refers to the standard inner product of two symmetric matrices.

As mentioned in Section 2.2, our channel assignment problem is essentially the Max K-cut problem in the conflict graph with the additional interface constraint. Below, we start with presenting the SDP for the Max K-cut problem from [73]. We then extend it to our channel assignment problem by adding the interface constraint.

SDP for Max K-cut. Let y_u be a variable that represent the color of a vertex $u \in V_c$. Instead of allowing y_u to take 1 to K integer values, we define y_u to be a vector in $\{a_1, a_2, ..., a_K\}$, where the a_i vectors are defined as follows [73]. We take an equilateral simplex Σ_K in $\mathbf{R}^{K-1}$ with vertices $b_1, b_2, ..., b_K$. Let $c_K = \frac{(b_1+b_2+...+b_K)}{K}$ be the centroid of Σ_K, and let $a_i = b_i - c_K$ for $1 \leq i \leq K$. Also, assume $|a_i| = 1$ for $1 \leq i \leq K$. Now, the Max K-cut problem can be formulated as an integer quadratic program as follows [73].

[3]A matrix is said to be *positive semidefinite* if all its eigen values are nonnegative.

IP$_{\mathbf{Max-K}}$:

$$\text{Maximize} \qquad \frac{K-1}{K} \sum_{(u,v)\in E_c} (1 - y_u.y_v)$$
$$\text{such that} \qquad y_u \in \{a_1, a_2, ..., a_K\}$$

Note that since $a_i.a_j = \frac{-1}{K-1}$ for $i \neq j$, we have:

$$1 - y_u.y_v = \begin{cases} 0, & \text{if } y_u = y_v \\ \frac{K}{K-1}, & \text{if } y_u \neq y_v. \end{cases}$$

Interface Constraint. We now add the interface constraint to the above formulation for Max K-cut. For each $i \in N$, let

$$\Phi_i = \sigma(E(i), R_i) - (\binom{|E(i)|}{2} - \sigma(E(i), R_i))/(K-1),$$

where $\sigma(E(i), R_i)$ is as defined as follows:

$$\sigma(S, K) = \frac{\beta\alpha(\alpha+1) + (K-\beta)\alpha(\alpha-1)}{2}, \tag{2}$$

where $\alpha = \lfloor \frac{|S|}{K} \rfloor$ and $\beta = |S| \ mod \ K$. It can be shown [112] that the number of monochromatic edges in the clique of size $|S|$ when colored by K colors is at least $\sigma(S, K)$. Now, we add the following constraint to represent the interface constraint.

$$\sum_{u,v\in E(i)} y_u.y_v \geq \Phi_i \quad \forall i \in N \tag{3}$$

Recall that vertices in $E(i)$ form a clique in the conflict graph, and cannot be partitioned into more than R_i partitions to satisfy our interface constraint. Now, $\sigma(E(i), R_i))$ gives a lower bound on the number of monochromatic edges in this clique ($E(i)$) [112], and thus, $\binom{|E(i)|}{2} - \sigma(E(i), R_i))$ is an upper bound on the number of non-monochromatic edges. Since we know that $y_u.y_v = 1$ for any monochromatic edge (u, v) and $y_u.y_v = \frac{-1}{K-1}$ for any non-monochromatic edge, we have constraint in the above Equation 3.

Note that even though Equation 3 is a valid constraint, it does not necessarily restrict the number of colors assigned to vertices of $E(i)$ to R_i. Thus, the IP_{Max-K} augmented by the above Equation 3 only gives an upper bound on the number of non-monochromatic edges.

Relaxed SDP for Channel Assignment. Since we cannot solve the integer quadratic program IP_{Max-K} for problems of reasonable size, we relax it by allowing the variables y_u to take any unit vector in $R^{|V_c|}$. Since $y_u.y_v$ can now take any value between 1 and -1, we add an additional constraint to restrict $y_u.y_v$ to be greater than $\frac{-1}{K-1}$. The relaxed SDP for the channel assignment is as follows.

$$\text{Maximize} \quad \frac{K-1}{K} \sum_{(u,v)\in E_c} (1 - y_u.y_v)$$

such that

$$y_u \in R^{|V_c|} \text{ and } |y_u| = 1$$

$$y_u.y_v \geq \frac{-1}{K-1}, \quad \forall u \neq v, \text{ and}$$

$$\sum_{u,v\in E(i)} y_u.y_v \geq \Phi_i, \quad \forall i \in N.$$

Standard SDP Formulation. Now, we convert the above relaxed version into the standard SDP formulation. Let W be the $|V_c| \times |V_c|$ symmetric matrix representing the adjacency matrix of the graph G_c, and let e be the $|V_c| \times 1$ vector containing all 1's. Now, let $L = d(W.e) - W$ denote the Laplacian of the W matrix, where $d(W.e)$ is the $|V_c| \times |V_c|$ matrix with $W.e$ as the main diagonal. Finally, let

$$C = -\frac{L(K-1)}{2K},$$

X be the semidefinite $|V_c| \times |V_c|$ matrix representing $y_u.y_v$ for all $u, v \in V_c$. Now, the semidefinite program for the channel assignment problem in the

standard SDP form (Matrix Notation) [59] can be represented as follows.

$$
\begin{aligned}
&\text{Minimize} \qquad C.X \\
&\text{such that} \\
&\quad diagonal(X) = e \\
&\quad X_{u,v} \geq \frac{-1}{K-1}, \quad \forall u \neq v \in V_c, \\
&\quad A_i.X \geq 2\Phi_i, \quad \forall i \in N, \text{ and} \\
&\quad X \succeq 0,
\end{aligned}
$$

where each $A_i (i \in V)$ is a $|V_c| \times |V_c|$ matrix representing $E(i)$. In particular, the $A_i[u, v] = 1$ if $(u, v) \in E_i$, and 0 otherwise. Also, the inequalities in the above constraints can be converted into equalities by subtracing linear positive variables from the left hand side.

The solution to the above semidefinite program gives an upper bounds on the number of non-monochromatic edges, and the lower bound on the optimal network interference can be obtained by subtracting it from $|E_c|$. This semidefinite program can solved using standard SDP solver such as DSDP 5.0 [40].

2.6.2 Linear Programming Formulation

In our simulations, we observed that solving the semidefinite program formulation presented in the previous section can take a long time (12 hours on a 2.4 GHz Intel Xeon machine with 2GB RAM for a 50 node network) and memory, and hence, may not be feasible for very large network sizes. Thus, in this section, we formulate our channel assignment problem as an integer linear program (ILP), and use the relaxed linear program with additional constraints to estimate the lower bound on the optimal network interference. The LP formulation can be solved in a much less time (less than an hour vs. 12 hours) than the SDP formulation, but yields a slightly looser lower bound than SDP on the optimal network interference. Note that the SDP and LP formulations are used only to demonstrate the performance of our Tabu-based and Greedy algorithms.

Integer Linear Programming. Recall that N is the set of network nodes, R_i is the number of radio interfaces for a node i, $\mathcal{K}$ is the set of available channels, and $G_c(V_c, E_c)$ is the conflict graph. Also, $E(i)$ represents the set of vertices in V_c that represent the communication links incident on node $i \in N$.

We use the following set of binary integer (taking values 0 or 1) variables and constraints in our ILP formulation.

- Variables Y_{uk}, for each $u \in V_c$ and $k \in \mathcal{K}$. The variable Y_{uk} is 1 if and only if the vertex $u \in V_c$ is assigned the channel k. Essentially, the variables Y_{uk} define the channel assignment function. Since, each vertex in V_c is given exactly one channel, we have the following constraints.

$$Y_{uk} = \{0,1\}, \qquad \forall u \in V_c,\ \forall k \in \mathcal{K} \tag{4}$$

$$\sum_{k \in \mathcal{K}} Y_{uk} = 1, \qquad \forall u \in V_c \tag{5}$$

- Variables X_{uv}, for each edge $(u,v) \in E_c$. The variable X_{uv} is 0 only if the vertices $u, v \in V_c$ are assigned different channels.[4] The following equation defines the value of X_{uv} in terms of Y variables.

$$X_{uv} = \{0,1\}, \qquad \forall (u,v) \in E_c \tag{6}$$

$$X_{uv} \geq Y_{uk} + Y_{vk} - 1,\ \forall (u,v) \in E_c, \forall k \in \mathcal{K} \tag{7}$$

  The variables X_{uv} are used to define the network interference (the objective function defined later).

- Variables Z_{ik}, for each network node $i \in N$ and channel $k \in \mathcal{K}$. The variable Z_{ik} is 1 if and only if some $u \in E(i)$ has been assigned a channel k; note that, u represents a communication link incident on $i \in N$.

$$Z_{ik} = \{0,1\}, \qquad \forall i \in N,\ \forall k \in \mathcal{K} \tag{8}$$

$$Z_{ik} \geq Y_{uk}, \quad \forall u \in E(i),\ \forall i \in N,\ \forall k \in \mathcal{K} \tag{9}$$

$$Z_{ik} \leq \sum_{u \in E(i)} Y_{uk}, \qquad \forall i \in N,\ \forall k \in \mathcal{K} \tag{10}$$

[4] If vertices u and v in V_c are assigned same channel, then X_{uv} can be 0 or 1. However, X_{uv} will be chosen to be 0 to minimize the objective function (see below), as there are no additional constraints involving X_{uv}. The additional constraints in Equation 12 and 13 can be looked upon as derivations of Equation 7.

The last equation above is used to enforce that Z_{ik} is 0 if there is indeed no vertex $u \in E(i)$ that has been assigned a channel k. The below equation enforces the interface constraint using Z variables.

$$\sum_{f=1}^{k} Z_{if} \leq R_i \qquad \forall i \in N \tag{11}$$

Objective Function. Our objective function for the above ILP is to

$$\text{Minimize} \sum_{(u,v)\in E_c} X_{uv}.$$

Linear Programming. Due to NP-hardness of integer linear programming, solving the above ILP is intractable for reasonably sized problem instances. Thus, we relax the above ILP to a linear program (LP) by relaxing the integrality constraints. In particular, we replace the Equations 6, 4, and 8 by the following equation.

$$0 \leq X_{uv}, Y_{uk}, Z_{ik} \quad \leq \quad 1.$$

The solution to the relaxed linear program gives only a lower bound on the optimal solution to the ILP. Through simulations, we have observed that the lower bound obtained by the above LP formulation is very loose. Thus, in order to obtain a tighter lower bound, we add additional constraints as follows.

Clique Constraint. For each vertex $u \in V_c$, let S_u be the set of vertices in a maximal clique containing u. As discussed in Section 2.6.1, we can lower bound the number of monochromatic edges in a complere graph of size $|S_u|$ when colored by K colors as $\sigma(S_u, K)$ using Equation 2. The above observation yields the following additional constraint.

$$\sum_{v,w\in S_u} X_{vw} \geq \sigma(S_u, K) \quad \forall u \in V_c \tag{12}$$

Since the set of vertices $E(i)$ in V_c forms a clique in G_c and uses at most R_i colors (due to the interface constraint on node i), we also have the following constraint.

$$\sum_{(u,v)\in E(i)} X_{uv} \geq \sigma(E(i), R_i) \quad \forall i \in N \tag{13}$$

The above two additional constraints pose a lower bound on the interference on clique like subgraphs. This helps to reduce the gap between the actual integer optimum and the relaxed linear solution.

Number of Variables and Constraints. The number of variables in the above LP formulation is $|E_c| + K(|V_c| + N)$, and the total number of equations/constraints are $2(|V_c|+|N|)+K(2|V_c|+2|N|+|E_c|)$ including the integrality constraints. We solve the linear program using GLPK [8], a public-domain MIP/LP solver.

2.7 Generalizations

In the previous sections, for sake of clarity, we made various assumptions, viz., uniform traffic on all communication links, a binary interference model, and orthogonal channels. In this section, we generalize our techniques to relax these assumptions. These generalizations are quite useful in practical deployments. For example, the links in the network communication graph may carry different amounts of traffic. Thus, the average interference must be weighted by traffic as interfering traffic is not the same for all interfering link pairs. Also, channels – even when they are orthogonal in theory – do interfere due to device imperfections (e.g., radio leakage, improper shielding, etc.) [138]. Thus, modeling of non-orthogonal (i.e., interfering) channels is a good idea. In addition, this also allows us to explicitly utilize non-orthogonal channels. Finally, regardless of traffic and use of different channels, path loss effects can influence the degree of interference between two links – and thus, result in fractional interference between two links.

Non-uniform Traffic and Fractional Interference. Let u and v be two vertices in the conflict graph, $r(u, v)$ (a real number between 0 and 1) be the level of interference between two links corresponding to the vertices u and v when both links carry saturated traffic. The level of interference $r(u, v)$ between pairs of links u and v can be computed using techniques similar to [126]; Section 2.9 gives a detailed description of how it is computed in our experimental study. Let $t(u)$ and $t(v)$ denote the normalized traffic (with respect to saturated traffic) on the links corresponding to the vertex u and

v respectively. Note that in our network model, we assume that the traffic is known a priori. Measurements of these parameters was discussed in Section 2.2. Based on the above notations, the overall network interference $I(f)$ for a given channel assignment function $f : V_c \rightarrow \mathcal{K}$ can be defined as follows. Let $M = \{(u, v)|u, v \in V_c \text{ and } f(u) = f(v)\}$. Then,

$$I(f) = \sum_{(u,v)\in M} t(u)t(v)r(u, v).$$

Note that $t(u)t(v)r(u, v)$ is a reasonable way to model the level of interference between the nodes u and v with given traffic loads, since $r(u, v)$ is the level of intereference with saturated traffic and $t(u)$ and $t(v)$, the respective traffic loads, are normalized with respect to the saturated traffic.

For the generalized interference and traffic model, the Tabu-based and Greedy algorithms use the above definition of network interference; no additional changes are required. Similarly, the LP and SDP formulations of the channel assignment problem can be generalized by appropriately extending the objective function; no other changes are required in the list of variables and constraint equations.

Non-orthogonal Channels. Let $c(k_1, k_2)$, a value between 0 and 1, denote the level of interference between two channels k_1 and k_2. For non-orthogonal channels, the overall network interference can be further generalized as follows for a given channel assignment function $f : V_c \rightarrow \mathcal{K}$.

$$I(f) = \sum_{(u,v)\in E_c} t(u)t(v)r(u, v)c(f(u), f(v)).$$

As before, Tabu-based and Greedy algorithms can use the above definition of network interference without any additional changes. However, in the LP formulation, we need to replace the Equations 7 by the following.

$$X_{uv} \geq Y_{uk_1} + Y_{vk_2} - 2 + c(k_1, k_2),\ \forall (u, v) \in E_c, \forall k_1, k_2 \in \mathcal{K}$$

Unfortunately, the SDP formulation cannot be generalized easily for non-orthogonal channels. The problem arises from the difficulty in choosing appropriate vectors a_i such that $a_i.a_j$ is proportional to $c(i, j)$ for all channels $i, j \in \mathcal{K}$. The values $c(i, j)$ are characteristics of the channel spectrum, and can be measured independently.

2.8 Simulation Results

In this section, we study the performance of our designed algorithms for the channel assignment problem through extensive simulations. The main goal of the simulation based evaluation is to understand the performance of our algorithms in large-scale network scenarios. In Section 2.9, we will present detailed experimental evaluation of our algorithms on a 11-node multi-radio mesh testbed. We present our performance results for two different settings. First, we evaluate a graph-theoretic performance metric, and then, evaluate throughput improvement using ns2 simulations. We start with discussing various algorithms used for comparison.

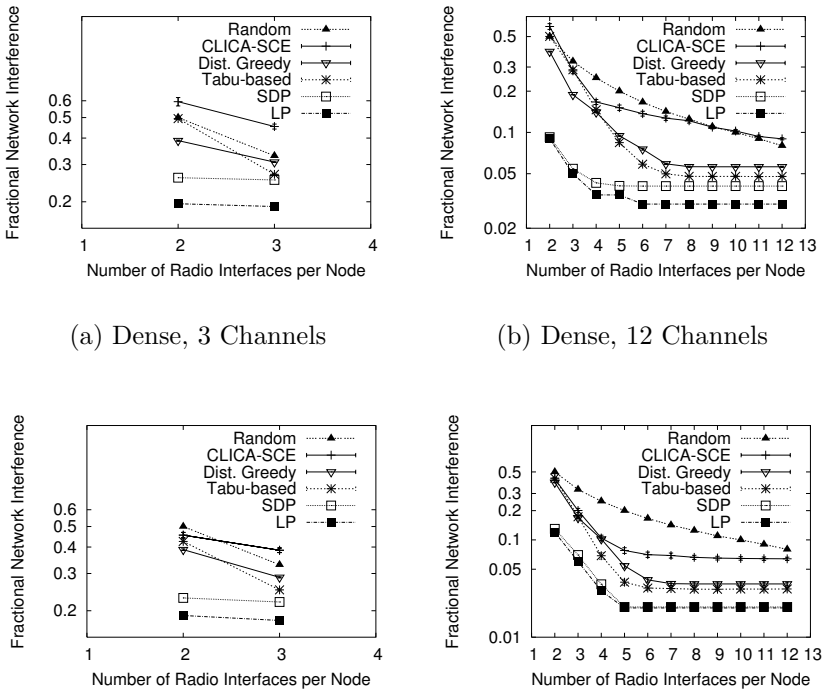
Algorithms. In addition to our designed algorithms (Tabu-based and Distributed Greedy) and the lower bounds obtained from the linear and semidefinite programming techniques, we also present results for two other algorithms for comparison. In particular, we simulate a modified version of the centralized CLICA heuristic presented in [110] for a slightly different version of the channel assignment problem.[5] We refer to the modified algorithm of [110] as CLICA-SCE. We also simulate a *random* algorithm which uses only a limited number of channels (equal to the number of radio interfaces), assigns a different channel to each radio interface, and then, selects a random interface (and hence, channel) for transmitting a packet. See Section 2.3 for a discussion on other related works.

We note here the network interference metric is actually a localized metric since a communication link interferes with only "neighboring" communication links. Thus, we observed that the centralized version of the greedy algorithm performed almost exactly the same as the Distributed Greedy algorithm.

2.8.1 Graph-Theoretic Performance Metric

In this set of experiments, we generate random networks by randomly placing a number of nodes in a fixed region, and evaluate various algorithms based

[5]In CLICA [110], a communication link may multiplex between multiple channels, but in our network model each communication link uses exactly one channel for transmission. We modify CLICA to use our network model.



(a) Dense, 3 Channels

(b) Dense, 12 Channels

(c) Sparse, 3 Channels

(d) Sparse, 12 Channels

Figure 4: Fractional network interference of solutions delivered by various algorithms compared with the lower bounds in dense or sparse networks for 3 or 12 channels.

on a certain graph-theoretic performance metric. To solve linear programs, we used GLPK [8] which is a public-domain MIP/LP solver, while to solve semidefinite programs, we used DSDP 5.0 [40] [41] which uses an efficient interior-point technique.

Graph Parameters. We consider two sets of random network, viz., dense and sparse networks, generated by randomly placing 50 nodes in 500×500 and 800×800 square meters of area respectively.[6] In dense networks, the average node degree is around 10, while in sparse networks the average node degree is around 5. Each node has the same number of radio interfaces, and has

[6]We evaluated networks of size up to 750 nodes and varying densities, with similar performance results for all algorithms. However, the LP and SDP formulations for networks of size larger than 50 nodes took unreasonably long computation time.

a uniform transmission and interference range of 150 meters. Two nodes are connected by a communication link if they lie within each other's *transmission range*. Also, two communication links (i, j) and (g, h) interfere with each other if and only if either g or h lies within the *interference range* of i or j; this is based on the protocol interference model [82]. We assume orthogonal channels and uniform traffic on all links.

Performance Metric. We evaluate the performance of our algorithms in random networks using the metric "fractional network interference." Given a channel assignment function f computed by an algorithm, the *fractional network interference* is defined as the ratio of network interference ($I(f)$) and the total number of edges in the conflict graph. This represents the number of conflicts that remain even after channel assignment relative to the number of conflicts in the single-channel network. The fractional network interference for the random algorithm is given by $\frac{1}{R}$, where R is the number of radios on each node. Note that the above performance metric is purely graph-theoretic and hence, we do not use any network simulator for these experiments.

Results. In Figure 4, we plot the fractional network interference for varying number of radio interfaces/node, in dense and sparse networks using 3 and 12 channels. In general, both our algorithms perform extremely well compared to the CLICA-SCE and random algorithms. The Tabu-based algorithm almost always performs better that than the Distributed Greedy algorithm, except when the number of radios is very small. When the number of radios is very small, the second phase of Tabu-based algorithm is forced to perform many inefficient merge operations which leads to performance degradation.

The performance of our algorithms compared to the lower bounds obtained from the LP and SDP formulations shows that our algorithms deliver very good solutions, particularly for larger number of radios. Note that the vertical axis of the plots is presented in log-scale for ease of viewing. The performance difference between the Tabu-based algorithm and the SDP lower bound is about 1% to 4% when the number of radios is large. We can also see that the SDP formulation delivers a much better lower bound than the LP formulation, for all parameter values. However, as we noted before, running SDP is significantly more computationally expensive (in terms of time and

memory) than LP.

The comparison of plots for dense and sparse networks bring out interesting features. The fractional interference reduces with increase in number of radios per node; however, this trend saturates beyond a certain number of radios. This saturation point is reached with smaller number of radios for sparse networks than for dense networks, for the same number of channels. This is because the denser networks can potentially support more concurrent transmissions than the sparse networks. Similar trends were observed in [110].

2.8.2 ns2 Simulations

In this set of experiments, we study the impact of channel assignment in improving throughput in an 802.11-based mesh network. We compare the performance of various algorithms by measuring the *saturation throughput* using ns2 simulations over randomly generated networks. We consider networks of 50 nodes randomly placed in a 1000×1000 square meters area. The transmit power, receive and carrier sense thresholds in the default setting of ns2 are such that the transmission range is 250 meters and the interference range is 550 meters. We used the same default radio parameters as in ns2 [21], except that we set the channel data rate to 24Mbps. All transmissions are unicast transmissions following the 802.11 MAC protocol with RTS/CTS, and the packet size is fixed to 1000 bytes.

Performance for Various Traffic Models. We use three different traffic models.

- Single-hop traffic model: This model consists of identical poisson traffic for each communication link. The single-hop traffic model is useful to evaluate the performance in the case when all links in the network carry the same load.

- Multi-hop peer-to-peer traffic model: In this model, 25 randomly selected source-destination pairs communicate using multihop routes. The routes are computed statically using the shortest number of hops as the metric, and do not change for the lifetime of the simulation.

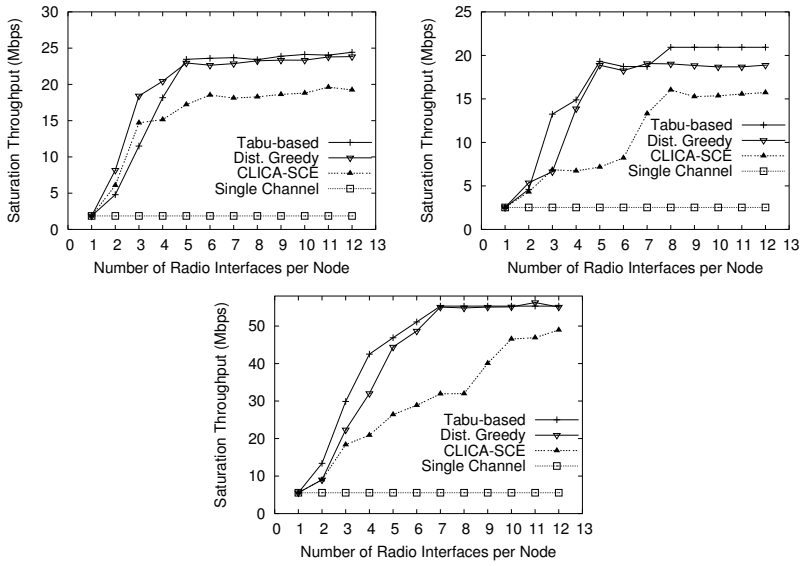


Figure 5: Saturation throughput in ns2 simulations for 12 channels and various traffic models, viz., (a) Single hop, (b) Multi-hop Peer-to-Peer, (c) Multi-hop Gateway.

- Multi-hop gateway traffic model: In this model, 4 random nodes are selected as gateways, and 25 source nodes send traffic to their nearest (in terms of hops) gateway. Routes are determined as in the previous traffic model. Such a traffic model will be common when the mesh network is used for Internet gateway connectivity.

Note that in the last two traffic models the traffic on the links is non-uniform. The traffic information is used in the channel assignment algorithms as suggested in Section 2.7.

Figure 5 plots *saturation throughput* against number of radio interfaces per node for the three traffic models and 12 channels (as we are experimenting with an 802.11a like system). We obtain the saturation throughputs as follows. For a particular number of radios and channels, we run a series of simulations, increasing the offered load each time, starting from a low value. We stop when the throughput does not increase any further with increase in the offered load.

We note that in all the three traffic models, our algorithms perform very

well. We also see that the observations we made from the earlier graph-theoretic evaluations translate well into the ns2 results. The saturation throughput remain same after a certain number of radios, as inferred in the graph-theoretic simulations. Also, the relative performance of the algorithms in the ns2 simulations is the same as observed in the graph-theoretic simulations. This indirectly establishes the merit of the chosen interference model, optimization objective, and use of graph-theoretic measures as a method of performance evaluation.

Modeling Non-Orthogonal Channels. So far, we have used only perfectly orthogonal channels. This however is a limitation in systems such as 802.11b where few orthogonal channels are available. Since our techniques are general enough to handle non-orthogonal channels (Section 2.7), we now model a non-orthogonal channel situation.

We assume an 802.11b like system where there are 11 channels, with only 3 of them being mutually orthogonal. For modeling the interference between non-orthogonal channels, we follow the technique outlined in Section 2.7. We use the data from [58] to model the "weighted" nature of conflicts. This data is obtained based on a simple analysis of the amount of overlapped spectrum between every pair of channels in 802.11b. We also did direct measurements on an 802.11b testbed to estimate interference between non-orthogonal channels and the values we obtained are similar to those quoted in [58]. Since such measurements can be very much hardware and environment specific, we stick to the data in [58].

In the ns2 simulator, we model inter-channel interference as follows. Physical layer frames transmitted on channel k_1 arriving at a radio interface tuned to channel k_2 are reduced in power depending on the degree of non-interference. For example, if a k_1-frame arrives at a k_1-interface, the frame does not undergo any power reduction. On the other hand, if a k_1-frame arrives at a k_2-interface, where k_1 and k_2 are perfectly orthogonal, then the k_1-frame is completely silenced. Power reduction between 0% and 100% occur for other intermediate cases. In the simulator, the interference (e.g., carrier-sense or collisions) is calculated only after such power reduction.

We use the peer-to-peer multihop traffic model (as defined before) to show

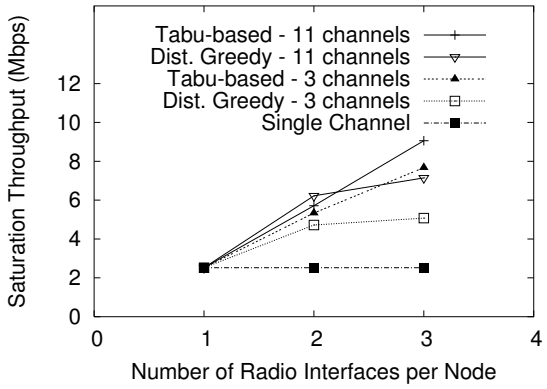


Figure 6: Saturation throughput in ns2 simulations when using non-orthogonal channels with 802.11b-like multi-channel model (11 channels with varying degrees of interference; 3 channels are mutually orthogonal).

the performance of our algorithms with non-orthogonal channels. See Figure 6. We observe that both our algorithms perform better when using all available 11 channels than when using only the 3 mutually orthogonal channels. The factor of improvement is less in the Tabu-based algorithm compared to the Distributed Greedy algorithm due to the inefficiency of the merge operations. Overall, use of non-orthogonal channels is a better choice than restricting channel assignments to only orthogonal channels.

2.9 Experimental Evaluation

In this section, we present an experimental evaluation of our channel assignment algorithms compared to the random channel assignment algorithm and CLICA-SCE algorithm. We start by describing our testbed and then discuss about generating the conflict graph for a given topology of mesh nodes. We then describe the channel assignment procedure and present the performance results.

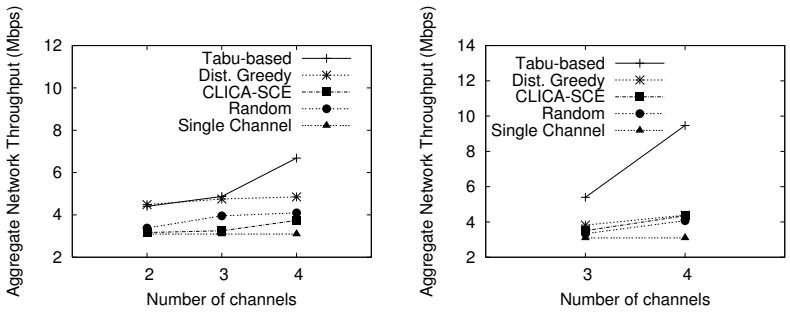


Figure 7: Aggregate network throughput in the network when transmit power of each node is set to 11dBm (a) Two radio case (b) Three radio case.

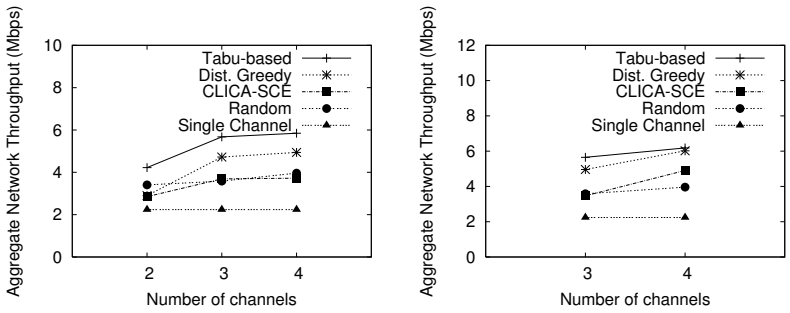


Figure 8: Aggregate network throughput in the network when transmit power of each node is set to 15dBm (a) Two radio case (b) Three radio case.

2.9.1 Testbed

The mesh testbed is an indoor wireless testbed that consists of 11 nodes each of which is a Soekris [18] net4801 embedded computer running Pebble Linux [15] with the Linux 2.4.31 kernel. The PCI-slot in the embedded computer is expanded into 4 mini-PCI slots using RouterBoard 14 [17] which allows us for using 4 mini-PCI wireless cards. We use three 802.11 a/b/g mini-PCI wireless cards based on Atheros [3] chipset with external antennas in each mesh node. Figure 9 shows our multi-radio mesh node prototype. We use the latest `madwifi` [12] driver for the 802.11 interfaces. In our experiments, we configured the 802.11 interfaces in 802.11a mode as there are twelve orthogonal channels and the communication range of each wireless node is less compared

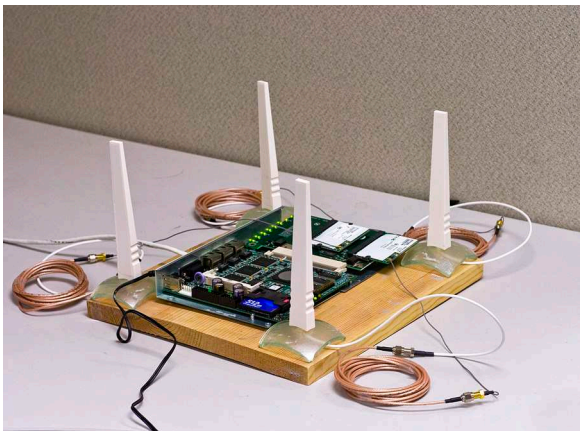


Figure 9: Multi radio wireless mesh node constructed using a Soekris net4801 embedded computer consisting of 4 Atheros chipset based miniPCI 802.11a/b/g wireless cards. The antennas are separated from each other of by a distance approximately 0.5 m.

to 802.11b/g modes so that we could get different multi-hop topologies within indoors. Due to board crosstalk or radio leakage [28, 138] there is interference between the radios in the same node even when they are configured to separate orthogonal channels. In order to over come this, we separated the three external antennas at a distance of about 0.5 m based on measurements similar to [138]. With this setup, we could use 7 (channels 36, 44, 52, 60, 149, 157, 165) out of the 12 orthogonal channels in each mesh node without interference between different radios on the same node.

2.9.2 Generating communication graph and conflict graph

The nodes in our mesh tesbed are static and in order to generate different topologies, we used different transmit powers. In the experiments reported in this chapter, we have used two different transmit powers (11dBm and 15dBm) to generate one sparse and one dense topology. The first step in our experimental procedure is to generate the communication graph for a given configuration of nodes. The communication graph is generated by allowing each node to transmit broadcast packets one after another and all other nodes measuring

the delivery ratio and througput. This takes about $O(n)$ time, where n is the number of nodes in the network. In our study, we repeat the above procedure multiple times and choose links with delivery ratio more than 80% as the stable links in the network and use only these links. The reason behind this is that the links that have poor delivery ratio are very unstable making it harder to get good statistical confidence in the results with runs of reasonable lengths. Also, the number of stable links is much less compared to the total number of possible links in the network. This also reduces the time to compute the conflicts between link pairs. Among the 7 orthogonal channels, channels 149, 157, 165 are in the upper band of the 5GHz spectrum and the link characteristics using these channels are significantly different from the rest of the 4 channels we considered in the lower and middle band. Since all the channel assignment algorithms considered for the study here assume that all channels to be alike, we had to restrict our study to the 4 channels only (i.e, channels – 36, 44, 52, 60).

Once the links in the communication graph are decided, we schedule pairs of links simultaneously and measure the throughput on each link in the presence of transmission on the other link. If there are m stable links in the network, this measurement takes $O(m^2)$ time to complete. The conflict graph is then computed using the method used in [126]. The whole process of generating the conflict graph is automated and the conflict graph is fed as input to the channel assignment algorithms.

2.9.3 Performance Results

In our performance study, we use saturated load on all stable links as determined while generating the communication graph for each transmit powers simultaneously. Each experiment is repeated 10 times and the average value is reported. Figure 7 and 7 show the aggregate throughput in the network when using channel assignments from our Tabu-based and Greedy algorithm[7] compared to the random channel assignment and CLICA-SCE algorithms for

[7]In order to make the experimental procedure simple, we considered a centralized version of our greedy algorithm. As noted in Section VIII, the performance of both versions are similar.

two different topologies (sparse topology using 11dBm transmit power in all node and dense topology using 15dBm transmit power in all nodes). We also show the aggregate network throughput when using the same channel in all links. This serves as a base case.

For the sparse topology, Figure 6(a) shows the case when each node uses two radios and Figure 6(b) shows the case when each node uses three radios. We see the Tabu-based algorithm perform extremely well compared the other algorithms in both the cases. There is a notable difference in aggregate throughput when using 4 channels in this topology using our Tabu-based algorithm. This is because the level of interference resulting from the channel assignment when using 4 channels is much less compared to when using 2 or 3 channels. The performance of the Greedy algorithm is not significantly better when compared to the random and CLICA-SCE algorithms . In the dense topology, both our algorithms perform well and there is a notable increase in the aggregate throughput when using 3 channels compared to using 2 channels due to higher reduction in level of interference. Beyond 3 channels, the improvement is negligible. The improvement in aggregate throughput when using more number of channels and interfaces is largely dependent on the actual topology and the way links interfere in the network. Compared to the single channel case, there is a significant increase in aggregate throughput in the network. This shows the effectiveness of using good channel assignment algorithms.

2.10 Understanding Channel and Interface Heterogeneity

One singular limitation of all channel assignment works in existing literature (including our proposed Tabu based and greedy channel assignment algorithms) is that they all assume that the channels and radio interfaces are all homogeneous. However in practice, the 802.11 channels vary significantly in Signal-to-Noise Ratio (SNR). Also, different radio interfaces on the same mesh nodes often provide different SNR measures even for the same channel. We

learned that channel and interface heterogeneity can impact the performance of channel assignment algorithms while conducting experiments presented in section 2.9. In the rest of this chapter, we present an exhaustive set of measurement results that demonstrate heterogeneity in channels and interfaces via a set of careful measurements on two different wireless mesh network testbeds (802.11g and 802.11a) covering a wide-spectrum of possibilities. We show experimentally that the homogeneity assumptions often lead to very poor channel assignment. We followup the measurements with techniques to incorporate channel-specific link quality information in channel assignment algorithms to improve their performance.

In current literature, there are very few works thats explore channel heterogeneity. Das et al [63] observed variation in routing metrics in different channels in wireless mesh networks. However, their work primarily focuses on comparing different routing metrics and understanding their dynamics. In [121], the author observed variation in link quality in multiple channels when studying interference maps in 802.11 networks. This paper studied one 802.11a link and showed variation in delivery ratio in different channels. In contrast to these works, we quantifies the variation in using both different channels and interface pairs using extensive measurements in two different mesh testbeds operating 802.11g and 802.11a bands and using different hardware platforms. We also show that the variations in link quality are not correlated to frequency of the channels. Finally, we experimentally demonstrate that utilizing channel and interface-specific information in channel assignment algorithms improves performance significantly.

2.10.1 Testbeds

The measurements reported in the rest of this chapter are from two different wireless mesh network testbeds (802.11g and 802.11a) set up in our departmental building as described below. The 802.11g testbed uses 10 Dell latitude D510 laptops each with one Atheros chipset based D-link DWL AG660 PCMCIA 802.11a/b/g card with an internal antenna. The transmit powers are fixed to 15 dBm and data rate to 11 Mbps. Measurements from this testbed

were collected on 40 different links on three orthogonal channels 1, 6, 11 (2412, 2437 and 2462 MHz respectively) in the 802.11g band. The second 802.11a test bed is an expanded setup of the tested described in section 2.9.1. For this experiments, we expanded the testbed to consist 13 nodes.

All nodes in both the testbeds run Linux (kernel 2.6.22 in laptops and kernel 2.4.29 in the Soekris boxes) and the widely used `madwifi` device driver (version v0.9.4) for the 802.11 interfaces. We used standard linux tools such as `iperf` to send UDP packets on the sender node for each link measured and `tcpdump` on the receiver node running on a raw monitoring interface to capture the packets. This gives us the additional prism monitoring header information such as the received signal strength (RSS), noise, channel and data rate for every received packet.

2.11 Channel Diversity

This section shows the results of our measurement study to understand the heterogeneity in channels due to varying path loss of different frequency bands. In the following, we first show that Received Signal Strength (RSS) of packets in each link is relatively stable in each channel and is a 'good' metric to compare the performance of any given link when using different channels.

2.11.1 Long term variation of RSS

We study a single link in the 802.11a testbed for a *24 hour* period by sending 1000-byte UDP packets at a rate of 100 packets per second. We repeat this experiment on 7 different 802.11a channels for the same link. Figure 10(a) shows the Allan deviation in the RSS values in each of the 7 channels at different time intervals ranging from 100 ms to 10 hours. Allan deviation is used as a metric to quantify the burstiness of variation in any quantity. The median variation is about 1.5 dBm and the 90% variation is about 2.5 dBm in a single channel. The variations are similar across all 7 channels. We see that the variation at different intervals are small considering the minimum granularity of RSS measurements is 1 dBm. This figure shows that in *any*

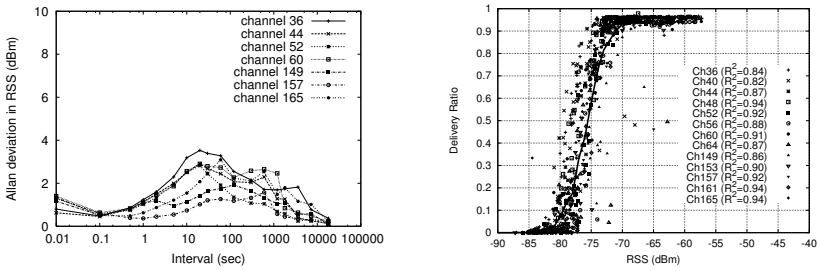
given channel, the variation in RSS value is minimal and sampling RSS values at smaller intervals (in the order of tens of seconds) can be representative of longer measurements. We also see similar results in the 802.11g testbed which are not reported here due to space constraints.

2.11.2 Relation between RSS and delivery ratio

Now that we have seen that RSS is relatively stable over long periods of time, next our goal is to show that RSS is a good predictor of link performance in each channel. For this, we studied 78 different links in the 802.11a testbed by sending back-to-back 1000-byte packets in each link using the 13 orthogonal channels for a period of 60 seconds one after another and measured the average RSS value and delivery ratio for each link in different channels. Figure 10(b) shows the relationship between average RSS and the delivery ratio of the links in our 802.11a testbeds. It shows a scatter plot of average RSS vs. delivery ratio of each link for all channels. The interpolations (the dark lines) of the aggregated data are also shown. Visually it appears that the RSS vs. delivery ratio statistics is independent of channels – no definite channel specific pattern emerges. We have also computed the R^2 value for each individual channel data with respect to the interpolation (noted in the plots). The R^2 values are similar across channels - varying between 0.82–0.94. This shows that RSS is a good predictor of delivery ratio and this relationship is relatively independent of the channel used. Note that delivery ratio (or, throughput) is a commonly accepted performance metric for the upper layer protocols. We observed similar characteristics from measurements in the 802.11g testbed. Thus, we can focus on RSS alone to understand channel and interface specific behavior as this fundamental metric is influenced by the *propagation environment*.

2.11.3 Link behavior in different channels

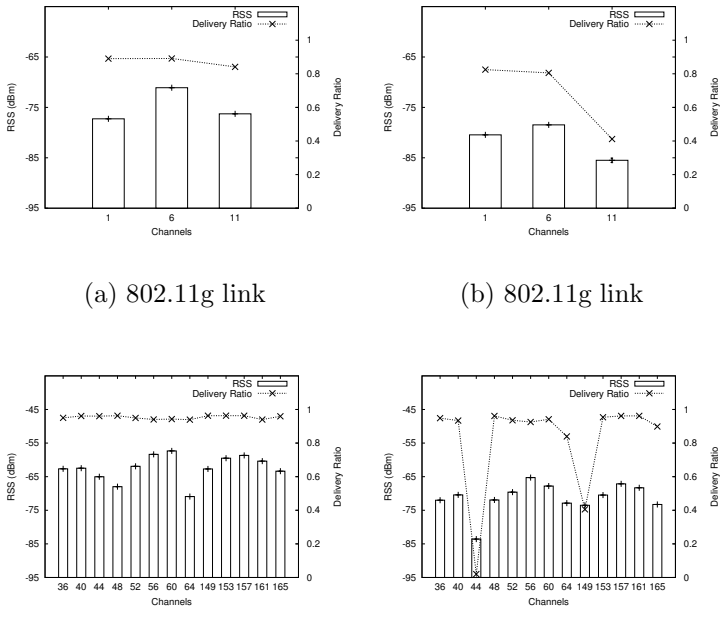
Now we look at the average RSS value (with 95% confidence interval) on each channel for two sample links in each testbed. See Figure 11. Figures 11(a) and 11(b) show the performance of two 802.11g links. In both cases, we see considerable variation in RSS in different channels. In the first case, even



(a) Long term variation of RSS values for a single link in 7 different 802.11a channels.

(b) Relationship between average RSS value and delivery ratio in different channels in our 802.11a testbed.

Figure 10: Characteristics of RSS metric.



(a) 802.11g link

(b) 802.11g link

(c) 802.11a link

(d) 802.11a link

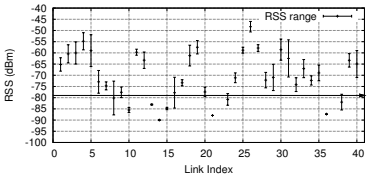
Figure 11: Variation of RSS and delivery ratio using different channels on sample links in our two testbeds.

though there is variation in RSS, the delivery ratios do not vary much. This is because the RSS values are already quite high. In the second case, we see that the delivery ratio of the link is good in channel 1 and 6 but is quite poor in channel 11. A similar behavior is observed in the 802.11a testbed. See Figures 11(c) and 11(d) for two sample links. These results demonstrate that RSS on a link could be channel-specific and this can impact the delivery ratio significantly.

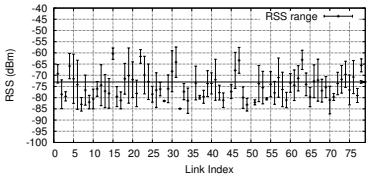
It is now interesting to study how much variation is there in RSS values for each of the 40 links in the 802.11g testbed and 78 links in the 802.11a testbed. In Figure 12(a) we show the range of variation in RSS value for each link in the 802.11g testbed. The bars show the maximum and minimum RSS value for each link considering all channels. The median RSS range (i.e., the median of the differences between the maximum and minimum over all links) is about 6 dBm and the 90-percentile RSS range is about 12 dBm. Figure 12(b) shows the RSS variation in the 802.11a testbed. In this case, the median RSS range is about 11 dBm and the 90-percentile RSS range is about 18 dBm. This is significantly higher than the variation of RSS in a single channel as noted previously. *Evidently, there are considerable variations in RSS values across channels.* The variation in the 802.11a testbed is higher. This is because the path loss characteristics are frequency specific and the 802.11a band (5180-5825MHz) is much wider compared to the 802.11g band (2412-2462MHz).

In both the plots, the horizontal arrow shows the RSS threshold values. Note that many links the RSS range crosses the threshold indicating *such links perform poorly in some channels, while performing quite well in some others.*

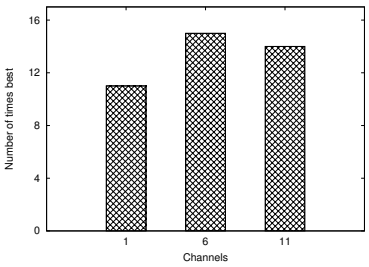
Now, it will be interesting to find out whether there is any one channel that is good for all links. In Figure 12(c) and 12(d), we show how many times each channel is the best based on the RSS values considering all links studied. We see that in both testbeds, there is no clear winner among channels. Each link performs differently in different channels. The RSS values are not correlated with the channel frequency. If this was the case, the channel 36 in the 802.11a band and channel 1 in the 802.11g band should have the best RSS values in all links. Some channels do exhibit better overall performance relative to their peers (e.g., channels 165 and 64 for 802.11a testbed). But generally speaking,



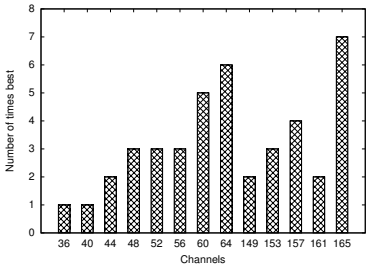
(a) Range of RSS variation in each link in 802.11g testbed across all 3 orthogonal channels.



(b) Range of RSS variation in each in the 802.11a testbed across all 13 orthogonal channels.



(c) Number of times each channel is best based on the RSS values on each link in the 802.11g testbed.



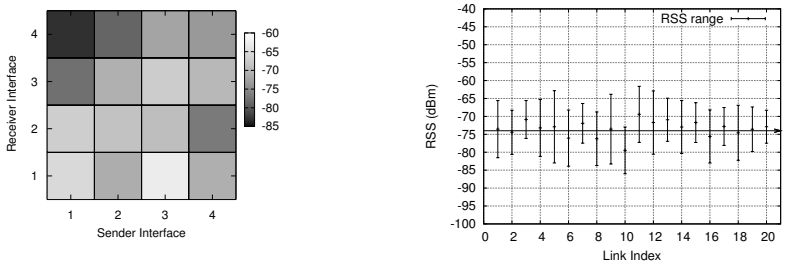
(d) Number of times each channel is best based on the RSS values on each link in the 802.11a testbed.

Figure 12: Link behavior across different channels in the two testbeds.

any channel could be the best for some link. *This makes it impossible to judge which channels to use for a given link without doing actual measurements on the links.*

2.12 Interface Diversity

For a given link between two multi-radio nodes, the choice of actual radio interfaces to use for this link could impact the link performance. The reason for this is two fold. First, there could be inherent manufacturing variations between the interfaces even though they use the same card model. Second,



(a) RSS values (in dBm) for 16 possible interface pair combinations on a sample link.

(b) Range of RSS value between different interface pair combinations for each link.

Figure 13: Interface heterogeneity in multi-radio nodes in 802.11a testbed.

the antennas for the interfaces need to be situated at a distance to prevent radio leakage issues so that the orthogonal channels do remain orthogonal in practice [138]. This makes the actual distance between different antenna pairs for the same node pair slightly different (noted in Section 2.10.1). This issue is more significant in 802.11a as it provides shorter ranges relative to 802.11g. On the other hand, 802.11a is indeed attractive for multichannel work, as it provides many more orthogonal channels.

To understand the variations caused by interface selection, we study 20 links (a subset of the 78 links studied before) in our 802.11a testbed using 16 possible interface pairs for each link. We select the same channel (channel 64, one of the good performing channels) for this measurement on all links in order to isolate the effect of interface selection.

Figure 13(a) shows the RSS values on all 16 possible interface pair combinations for a sample link. Here we see that the RSS value varies between -60 dBm to -85 dBm. Considering the RSS threshold (about -74 dBm), the link shown here has a very poor delivery ratio when certain interfaces are used (e.g., 1 to 4). However, some other interfaces would have a good delivery ratio (e.g., 3 to 1). *It is also interesting to note that we cannot say that a specific interface has poor performance.* For example, if we consider the interface 1 on

the sender node, it has varying performance based on the receiver interface.

In Figure 13(b), we show the range of variation in RSS values between the 16 possible interface combinations for each of the 20 links studied. Each bar shows the maximum and minimum RSS value for each link considering all 16 combinations. Note the significant variation in RSS values among different interface pairs. The median and 90-percentile RSS variation is about 12 dBm and 16 dBm respectively. Also note that most of these ranges straddle the RSS threshold (-74 dBm). This means the delivery performance can indeed significantly vary depending on the interface choices. *A channel assignment algorithm unaware of such variations can easily choose a bad interface pair for a link even though there are better interface pairs that could be potentially used.*

2.13 Channel Assignment Algorithm

In this section, we demonstrate the potential of using channel-specific link quality information in existing channel assignment algorithms to get better performance. For this purpose, we modify the greedy channel assignment algorithm described in section 2.5 to use the channel-specific link quality information when assigning channels for links. Recall that the greedy channel assignment algorithm assigns channels to *links* [8] in a greedy fashion trying to minimize the overall interference in the network. At the same time it satisfies the interface constraint, i.e., ensures that the number of channels assigned to links incident on a node does not exceed the number of interfaces on the node.

Recall that the greedy channel assignment algorithm works as follows: Initially, none of the links are assigned channels. The algorithm iterates through each link that is not assigned a channel yet and chooses a feasible set of channels that obey the interface constraint. From this feasible set of channels, it selects a channel that minimizes the overall *network interference* which is modeled using a conflict graph. The algorithm terminates when no further assignment of channels to links can reduce the network interference. Note

[8] Since it assigns channels to links directly, it is difficult (but not impossible) to incorporate the interface-specific information in this algorithm.

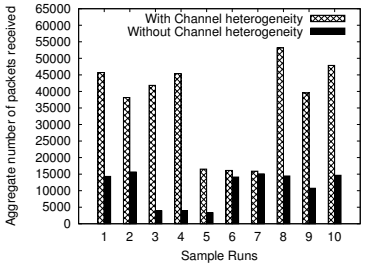


Figure 14: Aggregate number of packets received when a set of 10 links transmit packets simultaneously. Each sample run consists of a different set of 10 links in the network.

that among the channels in the feasible set, it is often the case that more than one channel can lead to the minimum interference. Since the algorithm is unaware of possible difference in link quality in different in channels, it chooses one channel arbitrarily. *Note that this is a singular limitation in all channel assignment algorithms in current literature as they do not use channel specific link quality information to make a choice.* In the new version of the greedy channel assignment algorithm, we use the channel-specific link quality information (e.g. RSS on different channels) to make this choice. Given RSS values are relatively stable, short term measurements (one time or periodic) are good enough to estimate the link quality in different channels. These measurements can be done whenever the channel assignments are recomputed. Estimating the periodicity of channel assignment depending on the environment and channel conditions is one of our future work.

In our 802.11a multi-radio testbed, we use 7 orthogonal channels (channels 36, 44, 52, 60, 149, 157, 165) and 4 interfaces in each node to study the performance of the channel assignment algorithm. In Figure 14, we show the performance of the greedy channel assignment algorithm with and without the channel-specific link quality information. We used periodic probes sent at 100 packets per second in each channel for 1 second to measure the link quality in different channels on each link before running the greedy algorithm that uses channel-specific link quality information. The horizontal axis shows 10 different experimental runs. In each run, we send back-to-back UDP packets

on 10 randomly chosen links simultaneously. The two versions of the channel assignment are used to assign channels for these 10 links. For each channel assignment, the experiment is run for 60 seconds and the aggregate number of packets received is measured. Note that the channel assignment algorithm using the channel-specific link quality information performs very well in all experimental runs compared to the case when all channels are considered homogeneous. Except in two cases (runs 6 and 7), the improvements are quite substantial - varying between 2-8 times. We noted that in the two cases where performance improvements are marginal, use of channel-specific information did not result in a very different channel assignment. Overall, the average improvement was by a factor of about 3.

2.14 Summary

In this chapter, we formulated and addressed the channel assignment problem in multichannel wireless mesh networks where each node may be equipped with multiple radios. We presented centralized and distributed algorithms that assign channels to communication links in the network with the objective of minimizing network interference. Using linear programming and semidefinite programming formulations of our optimization problem, we obtain tight lower bounds on the optimal network interference, and empirically demonstrate the goodness of the quality of solutions delivered by our algorithms. Using simulations on *ns2* and detailed experimental study on a 11-node multi-radio mesh testbed, we demonstrated the effectiveness of our approaches in improving the network throughput.

Next, we presented a detailed measurement study of channel and interface heterogeneity using measurements from two mesh testbeds using different hardware platforms and frequency bands (2.4GHz for 802.11g and 5GHz for 802.11a). We quantified the variation in link quality when using different channels and interface pairs and demonstrated that choosing the right channel and interfaces for a link can improve its performances significantly. We also demonstrated that this variation is 'non-trivial' in the sense that same channel does not perform uniformly well for all links, or the same interface does not

perform uniformly well for all other interfaces it is paired up with. Finally we proposed a mechanism to incorporate channel heterogeneity information in channel assignment algorithms to improve its performance. An important future direction of our work is to develop methods to measure these variations efficiently, understand how often they need to be repeated and design channel assignment schemes that take both channel and interface variations into account and come up with efficient solutions.

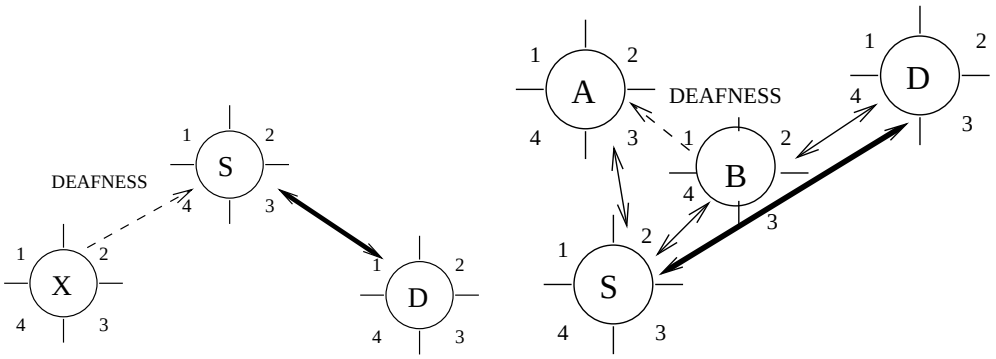
Chapter 3

Directional MAC Protocol for Wireless Mesh Networks

3.1 Introduction

In this chapter, we study the use of directional communication to improve the capacity of wireless multi-hop networks. Directional antennas can concentrate radio signal energy in a particular direction, instead of radiating it in all directions like their omni-directional counterpart. So the transmission on a directional antenna can potentially cause much lesser interference, thereby giving a significant capacity advantage in multi-hop wireless networks. Similarly, reception on directional antenna is subject to lesser interference. Thus, advances in directional antenna technology have motivated researchers to revisit the design of medium access control (MAC) protocols to fully exploit their advantages. Recently many approaches [39] [56] [57] [69] [87] [97] [98] [132] [145] have been proposed that aim to benefit from the ability to communicate in a desired direction.

Though directional antennas offer many benefits such as better spatial reuse, increased coverage and better link reliability, they also present new problems. *Deafness* and *directional hidden terminal problem* are two such problems. These problems if left unaddressed can have a serious effect on



(a) Type I (b) Type II

Figure 15: Deafness scenarios.

network performance. Deafness occurs when the transmitter fails to communicate to its intended receiver, because the receiver's antenna is oriented in a different direction. Directional hidden terminal problem occurs when a transmitter is unaware of the state of the channel when it orients its antenna to a new direction. This occurs when a potential transmitter fails to hear the RTS/CTS exchange[1] between another pair of nodes (because of its antenna orientation) and then initiates a transmission to the receiver of the ongoing communication. This can cause a collision.

The deafness problem has indeed been studied extensively in recent literature (see [55] and the references therein). However, the current approaches to solve this problem uses additional resources such as additional channels, radios or busy tones [69] [55]. On the other hand, the directional hidden terminal problem due to unheard RTS/CTS is largely left unaddressed. While deafness leads to lost channel utilization due to increasing backoff intervals, the directional hidden terminal problem causes collision that impacts performance adversely. In this chapter, our goal is to develop a MAC protocol that solves both problems comprehensively using a *single channel* and *single radio interface*.

[1] As in all related literature, we will use the IEEE Standard 802.11 as the baseline MAC layer protocol, and thus will use 802.11-related terminology.

3.2 Deafness and Directional Hidden Terminal Problem Scenarios

In this section we study various scenarios in which deafness and directional hidden terminal problem would occur. In general, deafness is caused when the transmitter repeatedly tries to send RTS to a destination but the destination does not reply with a CTS. In Figure 15(a), if node S is transmitting to node D, it sends a directional RTS to node D using beam 3. Node D then sends a directional CTS using beam 1. Node X is not aware of this transmission. If it initiates a transmission to node S, node S will not respond as it is transmitting data directionally to node D. This causes node X to backoff unnecessarily resulting in poor channel utilization. Here, deafness arises because node S has its beam oriented in a different direction and node X assumes that the RTS packet is lost due to congestion. Node X cannot initiate a transmission to node S immediately when the transmission between node S and node D is over, because it has to wait for the entire backoff interval. The directional MAC protocols that send RTS or CTS in a directional manner (DRTS/DCTS) [146] [57] [98] suffer from this problem.

Another variant of the deafness problem occurs when the receiver has its beam blocked due to another transmission in its vicinity and hence it cannot respond with a CTS to prevent collision with the ongoing transmission. See Figure 15(b). In this scenario, node S starts a transmission to node D. When node S sends the directional RTS, beam 4 of node B and beam 3 of node A are blocked. When node D sends the directional CTS, beam 2 of node B is blocked. Node S then starts its data transfer to node D. Now if node B wants to transmit to node A, it can send the directional RTS using beam 1 that is free. But node A cannot send a CTS back as its beam 3 is blocked. Thus node B assumes its RTS is lost due to congestion and goes into a backoff.

In Figure 2, we demonstrate the hidden terminal problem that arises due to the use of directional antennas. Suppose node S wants to communicate with node D while node A and node B are already communicating. Node A's antenna beam is oriented towards node B. If node S sends a directional RTS to node D, node A cannot hear it and will not block its beam 1. When node D

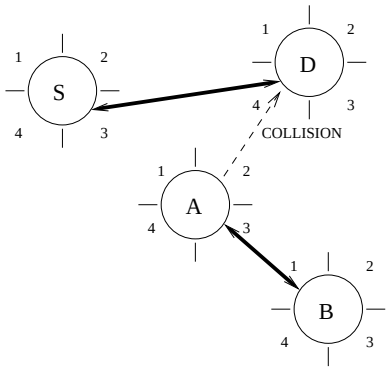


Figure 16: Directional hidden terminal problem due to unheard RTS/CTS.

sends the directional CTS packet, node A does not hear it and will not block its beam 2. Now after the data transfer between node A and node B is over, if node A wants to transmit to node D, it senses its beam 2 to be free. If it tries to send a directional RTS packet to node D, the RTS packet may collide with the data packet sent by node S. This kind of hidden terminal problem arises because a node misses an RTS/CTS exchange in its neighborhood and initiates a transmission to the receiver of an ongoing transmission. This scenario can turn into deafness, if collision does not occur. To the best of our knowledge, the directional hidden terminal problem due to unheard RTS/CTS has not been solved in current literature using a *single channel* and *single radio interface*.

3.3 Related Work

In the past, majority of the research using directional antenna were focused on single hop networks and cellular networks [38] [162]. Recently, many researchers have started to use directional antenna for multi-hop ad hoc or mesh networks [39] [56] [57] [69] [97], [98] [132] [145].

In an early proposal on the use of directional antenna in multi-hop networks [117], Nasipuri et al. have proposed to send the RTS and CTS packets omni-directionally so that the transmitter and receiver can locate themselves,

and then send the DATA and ACK packet directionally. This solves the deafness problem but results in poor spatial reuse. Ko et al. have proposed that the nodes send directional RTS while the CTS is sent omni-directionally [97]. They assume that the transmitter knows the location of the receiver. Directional RTS leads to deafness around the neighborhood of the sender as described in Figure 15(a).

Takai et al. have proposed the DVCS mechanism in which a node performs directional virtual carrier sensing [146]. This achieves good spatial reuse. In an similar work, Roychoudhury et al. have proposed the DMAC (directional MAC) protocol that performs all MAC layer operations in directional mode [57] . This combined with the DVCS mechanism achieves maximum spatial reuse, but it suffers from both deafness and directional hidden terminal problems.

Elbatt et al. have introduced the idea of blocked beam and unblocked beams for a node [69]. They have proposed to include the beam index (of the beam in which the DATA packet can be sent) in the RTS and CTS packets and send them in all unblocked beams. They send RTS/CTS in the blocked beams using a different channel. Thus each node needs to be equipped with two radios which are tuned to two different channels. They solve deafness but do not solve the directional hidden terminal problem.

Korakis et al. have proposed the circular-DMAC protocol [98] that tries to address the deafness problem. A node initiating a transmission, sends RTS packets directionally in all beams and the receiver node sends a single directional CTS packet. This protocol prevents deafness only in the neighborhood of the transmitter. The RTS packets are sent sequentially, so the receiver has to wait to send the CTS until the sender has sent RTS in all its beams. Some amount of inefficiency is introduced if the RTS packet towards the destination is lost, as all other nodes hearing the RTS will set their NAVs.

Huang et al. has proposed a busy tone approach [87] using multiple transceivers, capable for transmitting data packets as well as busy tones simultaneously. But this protocol suffers from both deafness and directional hidden terminal problem.

Recently, Roychoudhury and Vaidya [55] have addressed the deafness

problem by sending a tone omni-directionally after the data transmission between any two nodes complete. This is a corrective approach rather than a preventive one. This approach allows a node that suffers from deafness to go into repeated backoffs, and then terminate the backoff after the data transmission is over. Also, tone aliasing can happen in this protocol. This approach does not solve the second vaiant of the deafness problem as described in Figure 15(b), nor the directional hidden terminal problem due to unheard RTS/CTS.

In [79], Gossain et al. have identified the problems mentioned earlier and proposed modifications to existing directional MAC protocols to address the deafness problem. Their approach addresses both the deafness scenarios but not the directional hidden terminal problem. In [60], Cordiero et al. have proposed an optimization to the circular DMAC protocol to solve deafness and hidden terminal problem due to asymmetry in gain between directional and omni-directional antennas [57].

3.4 Antenna Model and Assumptions

In our directional MAC protocol design, we assume a switched beam antenna model which consists of N beams covering the entire 360 degrees. We assume two passive antenna elements attached to a *single radio* similar to the antenna model in [55]. One antenna offers *omni-directional* mode of operation and the other offers *directional* mode of operation. Practical phased array antennas that electronically switch to different beams and change to omnidirectional mode are available as commodity hardware [1].

3.4.1 Packet Transmission

A node can send a packet either omni-directionally or directionally. Whenever a node wants to transmit omni-directionally, it uses the omni-directional antenna to transmit the packet. When a node wants to transmit directionally, it selects the appropriate beam using the directional antenna and transmits in the desired direction.

3.4.2 Packet Reception

When a node is idle, it senses the medium in omni-directional mode when it does not know the direction from which the signal might arrive. When it detects a signal, the antenna performs an azimuthal scan [55] in order to select the beam on which the impinging signal is maximum and switch off its other beams. Once the direction is known, packet reception is done in a directional manner.

In our protocol, we assume that the directional gain is equal to the omni-directional gain, though usually the former is higher. This is achieved by reducing the transmit power when transmitting directionally. This also conserves power. We also assume that each node knows the direction to its one-hop neighbors. Each node knows the beam index used to communicate with its neighbors and the beam index used by its neighbors to communicate with it [57] [69] [146]. We do not assume that the antenna elements in different nodes should be aligned.

3.5 Directional MAC Protocol

In this section, we describe our directional MAC protocol design and techniques to solve deafness and directional hidden terminal problem. The following intuitions form the basis of our directional MAC protocol design.

1. The deafness scenario as shown in Figure 15(a) can be solved if the transmitter and the receiver can somehow inform their neighboring nodes about their impending transmission.

2. The deafness scenario as shown in Figure 15(b) can be solved if the receiver that has a blocked beam finds a way to inform the sender that the transmission cannot happen without disturbing other ongoing transmissions in its neighborhood.

3. The directional hidden terminal problem due to unheard RTS/CTS can be solved if the nodes do not miss the RTS/CTS exchanges happening in their neighborhood.

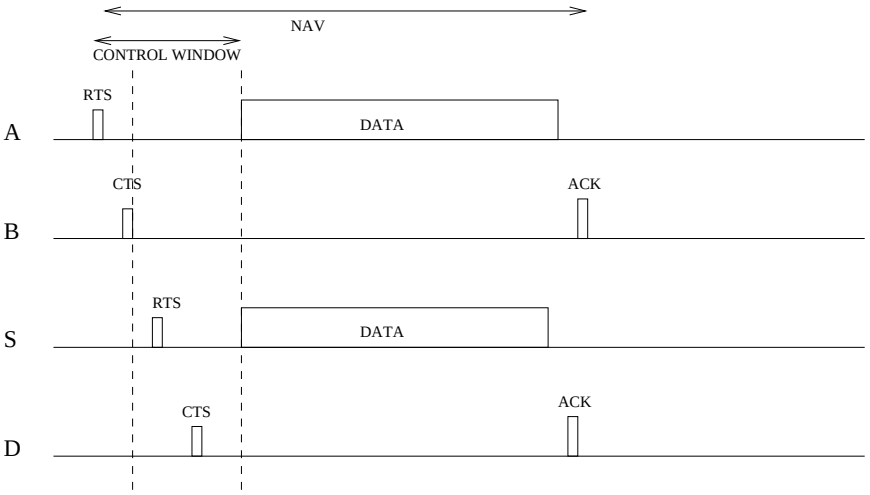


Figure 17: Control window mechanism to prevent directional hidden terminal problem

The detailed description of our directional MAC protocol is presented next.

In our protocol, the RTS and CTS packets are sent *omni-directionally* while the DATA and ACK packets are sent *directionally*. This way all the neighboring nodes of the sender and the receiver are informed about their communication. However, omni-directional RTS/CTS can decrease spatial reuse, as all neighboring nodes regardless of direction will now set their DNAV[2]. In order to prevent this, we overload the RTS/CTS packets with the beam index in which the actual DATA/ACK transmission will happen directionally.

As mentioned in section 3.4, each node knows the beam index used to communicate to its one hop neighbors. This beam index is put in the RTS packet and sent omni-directionally. Any node that receives this RTS packet, blocks the beam towards the sender of the RTS packet only if the beam index in the packet is same as the one used by the sender of the RTS packet to communicate with it. Similarly the receiver puts the beam index it uses to send the ACK packet in the CTS packet and send it omni-directionally. Now

[2]DNAV or directional network allocation vector is the mechanism to denote how long the channel has been reserved by other ongoing transmissions in a particular direction.

the neighbors of the receiver set their DNAVs accordingly after hearing the CTS packet. Each node maintains a neighborhood transmission table in which it stores the sources of the RTS/CTS packets it heard and the corresponding NAV durations. Here, we assume that the CTS packet also carries the address of its sender.

There is a chance that the RTS/CTS packets sent omni-directionally might collide with any other ongoing data transmission in the neighborhood. We prevent this by seperating the transmission of control and data packets in time. After the exchange of the control packets, the nodes wait for a duration called *control window* before transmitting the DATA packet as shown in Figure 3. The value of the control window is put in the RTS/CTS packets and informed to the neighboring nodes. We use the control window for the following two purposes. Firstly, it avoids collisions between control and data packets. Secondly, it allows multiple data transmissions to start in a neighborhood simultaneously if the data transmissions are in different directions and do not interfere with each other.

A node that sends the first RTS/CTS packet in a neighborhood, defines the control window and informs it to its neighbors during the control packet exchange. Any other node in the same nieghborhood that wants to start a data transmission that will not interfere with the previously reserved transmission can start a RTS/CTS exchange if it can complete it within the defined control window. Note that the control window is defined by the sender of the first RTS/CTS packet. The nodes sending the subsequent RTS/CTS packets do not redefine the control window. Once the control window is over, all the nodes that reserved a transmission can send their data in different directions simultaneously.

The size of the control window is an important factor in this approach. If the control window is large and there are only few non-interfering transmissions that can take place, it results in poor utilization of the channel. If it is small and there are more number of non-interfering transmissions that can take place, it results in poor spatial reuse. So the size of the control window is made adaptive depending on the traffic in the network. Based on our simulation experience, we define the size of the control window as a multiple of the time

for a control packet exchange. A node that defines the control window chooses a value for it as α × number of control packet exchanges it heard in the previous window × time for a control packet exchange, where $1 \leq \alpha \leq 2$.

Since we seperate the transmission of control and data packets in time, the nodes do not miss the RTS/CTS exchange in their neighborhood, thereby preventing the *directional hidden terminal problem*. We note that the use of a time window between the reservation and actual data transfer has been used in MACA-P protocol [27]. However their objective was to increase parallelism in CSMA based MAC protocols rather than solving deafness and directional hidden terminal problem.

We address the deafness problem as described in Figure 15(b) using a special packet called NCTS (*negative* CTS) which is sent omni-directionally when a node receives an RTS but cannot send the CTS as the beam it will use to send the ACK packet is already blocked. The NCTS packet can be sent omni-directionally because there is no data transmission going on during the control window. This informs the sender of the RTS packet that the data transmission cannot happen without interfering with the already reserved transmissions. When the node that sent the RTS packet gets the NCTS, it sends a TC (*Transmission Cancel*) packet omni-directionally. The TC packet signifies that the current transmission has been canceled; so all the neighbors now can cancel their NAV that was set due to the original RTS packet. They also cancel the control window, if sender of the TC packet has previously defined it.

3.5.1 Discussion

We now show how our MAC protocol solves the deafness and directional hidden terminal problem as shown in Figure 1 and 2.

In Figure 15(a), node S sends an omni-directional RTS packet with the beam index 3 to node D. When node X hears this packet, it adds an entry for node S in its neighborhood transmission table. Node X need not block its beam 2, because the beam index in the RTS packet is 3 but the beam used by node S to communicate with node X is 4. If node X has a packet to node

S, it knows that node S is already involved in another transmission from its neighbor transmission table and waits until it completes. This way of solving deafness is a preventive approach in contrast to a corrective approach in [55].

In Figure 15(b), node S sends an omni-directional RTS packet with beam index 2. Now node A blocks its beam 3 and node B blocks its beam 4. When node D sends the CTS packet with beam index 4, node B blocks its beam 2. Node S, then waits for the control window to get over. If node B, has data to transmit to node A, it sends a RTS packet within the control window. Now node A replies with a NCTS packet to node B informing it that the transmission cannot occur without interfering with the already reserved transmission between node S and D. Node B sends a TC packet to notify its neighbors that its transmission has been cancelled. This way we prevent the deafness scenario as described using Figure 15(b).

In Figure 2, if node A wants to transmit to node B, it sends an omni-directional RTS with beam index 3 and defines the control window as shown in Figure 3. Node B sends an omni-directional CTS packet with beam index 1 echoing the same control window. Both these packets are sent omni-directionally, so that all their neighbors are aware of this transmission and set their directional NAVs appropriately. Node S and node D need not block their beams 3 and 4 respectively, as the beam index 3 in the RTS packet sent by node A is different from the beam index used by node A to send to node S and node D. If node S wants to transmit to node D, it sends an RTS packet within the control window defined by node A as this transmission does not interfere with the transmission between node A and node B. Thus both the DATA transmissions can start simultaneously as shown. Note that when an RTS/CTS exchange happens, no node in the neighborhood is involved in a DATA transmission and will not miss them. Thus the directional hidden terminal problem does not occur. The hidden terminal problem due to asymmetry in gain [57] is trivially solved as both the directional and omni-directional ranges are same in our protocol.

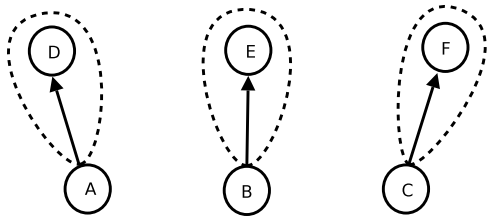


Figure 18: Scenario (i) : Scenario allowing high spatial reuse.

3.6 Performance Evaluation

In this section, we evaluate the performance of our directional MAC protocol (henceforth refered to as CW-DMAC) using the qualnet network simulator [148] (version 4.0). We use the DMAC protocol [57] and IEEE 802.11 standard as a baseline for performance comparison. In DMAC, all the MAC layer operations are done in a directional manner. It is aimed towards maximum spatial reuse but it suffers from all the problems mentioned in Section 3.2. In the IEEE 802.11 standard, all the MAC layer operations are done in an omni-directional manner. It does not suffer from either deafness or directional hidden terminal problem but suffers from poor spatial reuse. Our directional MAC protocols aims to achieve as much spatial reuse as possible at the same time solving both deafness and directional hidden terminal problem using a *single radio* and *single channel*.

In all our simulations, the transmit power is adjusted such that both the omni-directional and directional transmission ranges are approximately 280 meters. We use UDP as well as TCP traffic in our experiments with packet size 1024 bytes. We use the 802.11b physical layer model as implemented in qualnet and fix the data rate to 11Mbps. We use 8 directional beams (each of width 45°) in all our experiments. We do not consider node mobility in our simulation scenarios.

Intially, we study the performance of the protocols in different scenarios that bring out the advantages of our MAC protocol over DMAC and IEEE 802.11 and later show that our MAC protocol performs much better than the

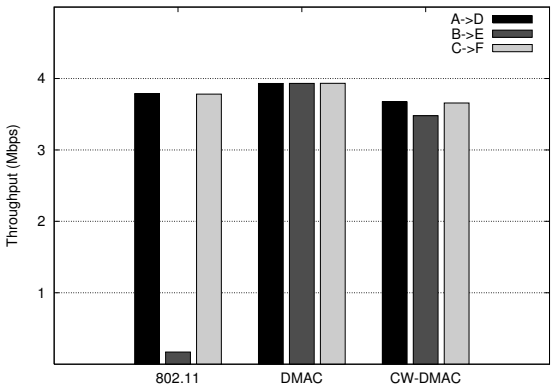


Figure 19: Throughput of different flows in Scenario (i).

other protocols in generic traffic scenarios and network topologies.

Spatial Reuse: In the first set of performance comparison, we use the scenario in Figure 4 that allows for high spatial reuse. The individual throughputs of the three flows when sending at a rate of 8Mbps are shown in Figure 5. As the IEEE 802.11 standard uses omnidirectional RTS/CTS packets, node B has to contend with transmissions from node A and node C. Node A and node C are out of interference range of each other. We can see that the throughput of the flow $B \rightarrow E$ drasticaly reduces because of the other two flows. DMAC does all its MAC operation in a directional manner, so it exploits the high spatial reuse in this scenario and all the three flows operate as if the other two flows are not in the neighborhood.

Eventhough we use omnidirectional RTS/CTS transmissions in our MAC protocol, we use the beam index of the corresponding DATA/ACK transmission and do directional virtual carrier sensing. So when node B receives the RTS packets from node A or node C it does not need to wait for the entire transmission to complete. If it can complete an RTS/CTS transmission within the control window, it can transmit along with nodes A and C. We can see that the throughputs of all the three flows are almost equal. The throughputs are slightly lesser than that achieved when using DMAC because we suffer a

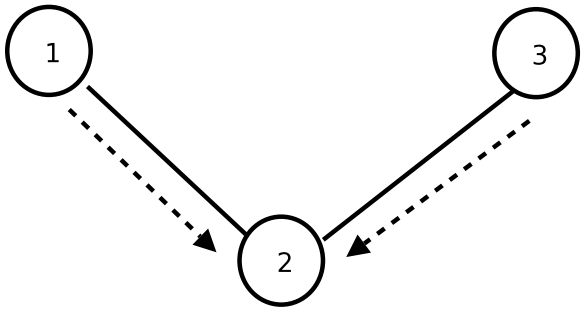


Figure 20: Scenario (ii) : Scenario to study deafness.

little when the control window is not fully utilized. Note that this scenario does not suffer from deafness or directional hidden terminal problem and is aimed to demonstrate the spatial reuse that can be exploited by our MAC protocol even when we use omnidirectional RTS/CTS transmissions.

Deafness: Next we use the scenario in Figure 6 that brings out the deafness problem. In the scenario (ii), nodes 1 and 3 transmit to node 2. In this scenario, when one of the senders initiates a successful transmission, the receiver turns towards that sender and the other sender suffers from deafness in the DMAC protocol. In our experiments, node 3 initiates a successful transmission with node 2, so node 1 suffers from deafness. Even when the transmission between node 3 and node 2 is over, node 1 is in backoff. As DMAC uses directional RTS and directional CTS, node 1 is not aware of this transmission and assumes that it's RTS packet is lost due to congestion.

Figure 7 compares the aggregate throughput and throughputs of individual flows with DMAC and CW-DMAC protocols. In the case of DMAC, node 1 suffers from extended deafness and its throughput decreases when the sending rates of the flows are increased. We observed that the number of packets dropped by node 1 is very high compared to the number of packets dropped by node 3 as node 1 exceeded the RTS retransmission limit many times and dropped packets. In CW-DMAC, node 2 sends an omni-directional CTS, so node 1 knows about the transmission between node 3 and node 2. Thus node

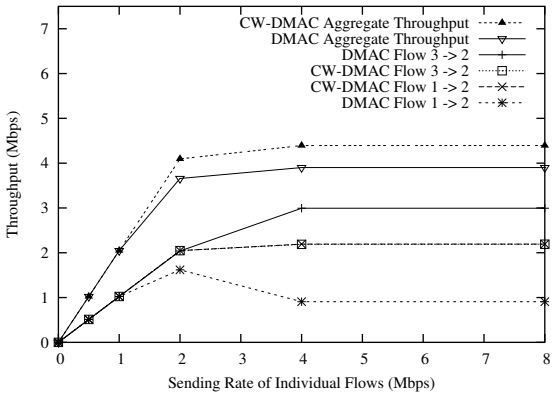
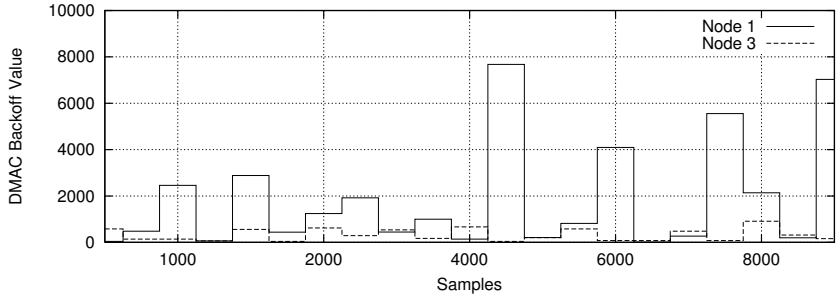


Figure 21: Throughput of different flows in scenario (i). The curves for CW-DMAC Flow 3 → 2 and CW-DMAC Flow 1 → 2 overlap.

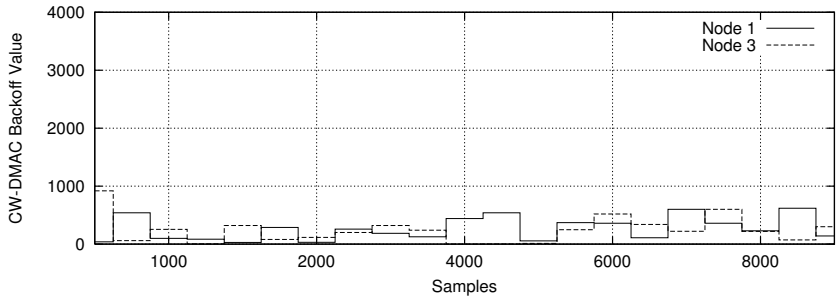
1 does not suffer from deafness. We can see that the throughput curves of flows 1 → 2 and 3 → 2 overlap in the case of CW-DMAC. CW-DMAC also performs better than DMAC in terms of aggregate throughput.

To get a better understanding of the scenario, we show a snapshot of the backoff value chosen by nodes 3 and 1 when running DMAC and CW-DMAC in Figure 8. When using DMAC, we can see that node 1 suffers from deafness and has to choose very high backoff values compared to node 3. In the case of CW-DMAC, the backoff values chosen by nodes 1 and 3 are comparable.

Directional Hidden Terminal Problem: The scenario (iii) in Figure 9 is used to study the effects of the directional hidden terminal problem. Here node 1 has a flow to node 3 and node 4, and node 2 to node 3. In the case of DMAC, when node 1 is transmitting to node 4, it is not aware of the transmission between node 2 and node 3. Since node 1 has a flow to node 3, if it transmits an RTS packet to node 3 it collides with the DATA packet from node 2 to node 3. In Figure 10, we compare the throughput of the flow 2 → 3 which is affected by directional hidden terminal problem due to unheard RTS/CTS. In the case of DMAC, we can see that when the sending rate of the flows are increased, the throughput of flow 2 → 3 decreases drastically due



(a) DMAC



(b) CW-DMAC

Figure 22: Snapshot of backoff values chosen by node 1 and node 3 .

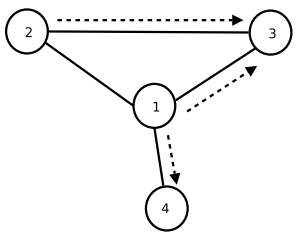


Figure 23: Scenario (iii) : Scenario to study directional hidden terminal problem.

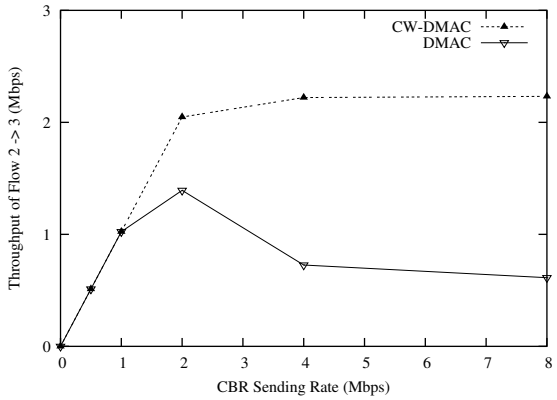


Figure 24: Throughput of flow from node 2 to node 3 in scenario (ii).

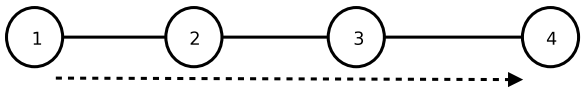


Figure 25: Linear topology used to demonstrate the cascading effects of deafness.

to collision between the DATA packets from node 2 and RTS packets from node 1 at node 3. In the case of DMAC, we observed an increasing fraction of packet drops at node 2 when the rate of the flows were increased. As Figure 10 demonstrates, CW-DMAC effectively solves this problem. We do not study the performance of IEEE 802.11 in the above two scenarios as it does not suffer from deafness and directional hidden terminal.

Cascading Effect of Deafness: We now analyze the performance of DMAC and CW-DMAC in a linear topology (Figure 11). We consider a flow between node 1 and 4 along the three hop path. This scenario is to demonstrates the cascading effect of deafness in a multi-hop scenario.

Figure 12 shows the throughtput of the flow for the different MAC protocols. DMAC suffers from the cascading effect of deafness [57] and performs

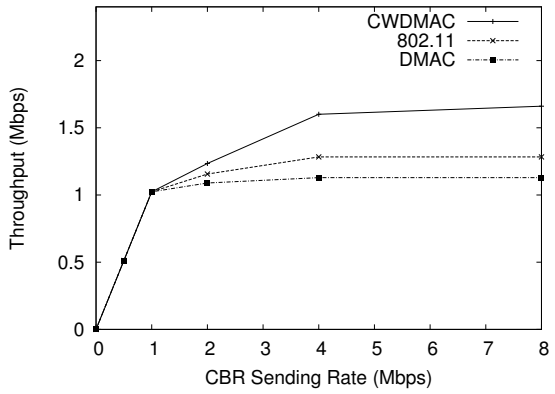


Figure 26: Throughput of the flow $1 \rightarrow 4$ in the linear topology.

poorer than 802.11 and CW-DMAC. CW-DMAC performs better than 802.11 and DMAC as it can exploit the paralellism in transmitting packets between node $1 \rightarrow 2$ and node $3 \rightarrow 4$ as node 3 is out of the interference range of node 1.

To understand the cascading effect better, we show the RTS retansmissions at each node when using DMAC and CW-DMAC in Figure 13. In DMAC, a node remains in directional mode when it is in backoff [57]. Thus in this topology, when node 3 is transmitting to node 4, node 2 suffers from deafness and goes into backoff. Node 1 tries to send packets to node 2 which is already in backoff in a directional mode. So node 1 in turn suffers from deafness as well. Thus, the periods of deafness are likely to be prolonged for the upstream nodes. Figure 13 shows the fraction of RTS packets retransmitted by each node for both protocols. In the case of DMAC, we see overall about 60% of the RTS packets are retransmitted, as nodes suffer from deafness. The bulk of this is in node 1 due to the above cascading effect. The number of RTS retransmissions for CW-DMAC is very low. This demonstrates the effectiveness of CW-DMAC to address the deafness problem.

The above experiments use scenarios that target specific deafness and hidden terminal problems. We now compare the performance of DMAC and

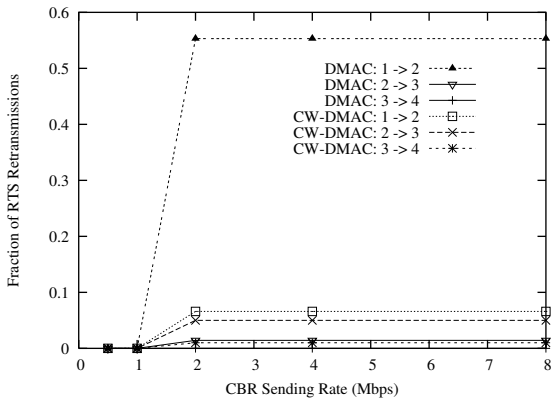
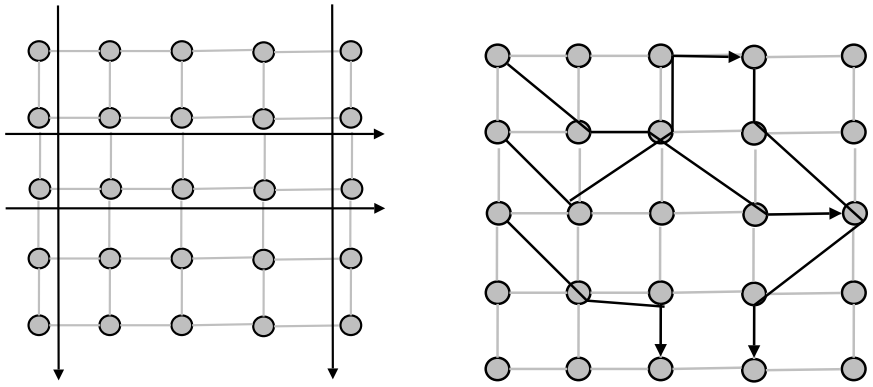


Figure 27: RTS Retransmission in a linear topology.

CW-DMAC for larger grid and random multihop topologies.

Grid Topology:



(a) Alligned flows.

(b) Random unalligned flows.

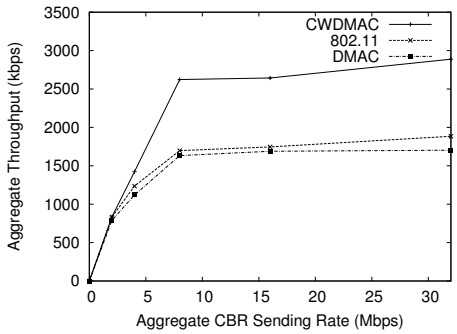
Figure 28: 5 x 5 Multihop Grid topology.

Figure 14 shows a 5x5 grid topology. The distance between adjacent nodes in the grid is about 180m so that both adjacent nodes and diagonal nodes can communicate with each other. We study two kinds of traffic patterns in the

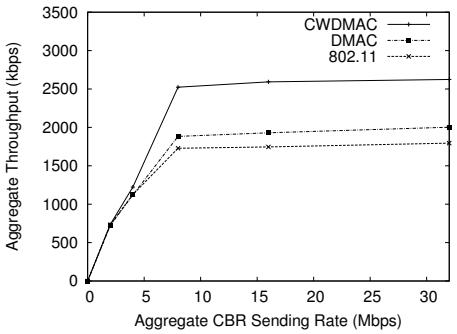
grid topology to understand the performance of CW-DMAC and DMAC.

In Figure 14(a), we have 4 multi-hop flows which are alligned along a straight line. We fix the routes of each flow statically. Figure 15(a) shows the comparison of the aggregate throughput of the four flows when using CW-DMAC, IEEE 802.11 and DMAC. DMAC performs poor compared to others because of higher interference in the direction of subsequent hops and due to the allignment of the hops along the chosen paths. The cascading effect of deafness also plays a role in DMAC's poor performance.

In the next experiment, we generate random multihop flows which are not alligned in the grid topology as shown in Figure 14(b). The performance of CW-DMAC, IEEE 802.11 and DMAC are shown Figure 15(b). It is clear from the plot that CW-DMAC outperforms both DMAC and IEEE 802.11 by exploiting spatial reuse and not suffering from deafness and directional hidden terminal problem. DMAC performs better than IEEE 802.11 as the flows are not alligned and the subsequent hops can transmit in different directions. The results shown in Figure 15(a) and 15(b) are an average of 5 different runs with flows between different set of nodes.



(a) Aggregate throughput for the alligned traffic flows in grid topology.



(b) Aggregate throughput for the random unalligned traffic flows in grid topology.

Figure 29: Aggregate throughput in grid topology.

Random Topology Next we study the performance of the different protocols in a large random multihop topology. 30 nodes were randomly placed in an

area of 1500m × 1500m. We set up 5 CBR flows simultaneously between randomly chosen source and destination pairs and study the throughput behavior. The routes were assigned statically. The simulation results were averaged over 10 runs. Figure 16 shows the aggregate throughput as the load increases. CW-DMAC clearly outperforms DMAC by providing approximately 20% higher saturation throughput. This demonstrates that deafness and hidden terminal problems occur frequently enough in random multihop scenarios and our protocol effectively solves them and improves network performance. IEEE 802.11 performs the worst due to its poor spatial reuse.

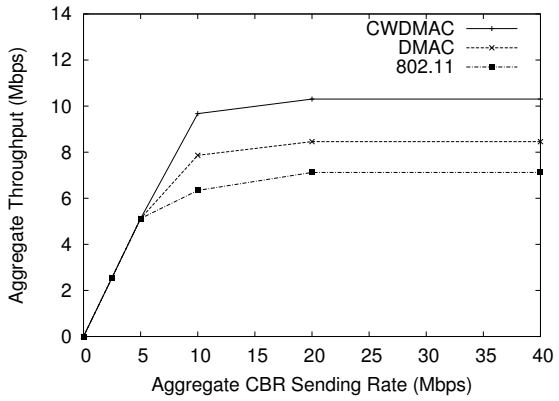


Figure 30: Aggregate throughput in a 30 node random topology.

Finally, we study the effect of different MAC protocols on TCP performance in the random topology. We set 5 FTP flows with packet size 1024 bytes between random pairs of nodes. Table 1 shows the aggregate TCP throughput achieved when using the different MAC protocols. The table shows that CW-DMAC performs better than DMAC which is better than IEEE 802.11. The packet drops and end-to-end delay in DMAC vary drastically because of collisions due to directional hidden terminal problem and backoffs due to deafness. Thus the round trip time estimated by TCP increases which there by affects the congestion window size and results in poor TCP throughput.

Table 1: Aggregate TCP Throughput in Random Topology

CW-DMAC	*DMAC*	*IEEE 802.11*
6.83	4.34	3.83

3.7 Summary

We studied the use of directional communication to improve capacity in multi-hop wireless networks. We addressed the issue of deafness and directional hidden terminal problem that occur when MAC protocols are designed for directional antenna based wireless multi-hop networks. We studied various scenarios in which deafness and directional hidden terminal problem could occur and proposed a new directional MAC protocol that address these problems comprehensively and solves them using a *single channel* and *single radio interface*. Current solutions in literature either do not address these issues comprehensively or use more than one radio/channel to solve them. Simulation results showed that our directional MAC protocol efficiently solves both the problems at the same time gaining from the advantages of the directional antennas.

Chapter 4

Vehicle to Roadside Communication

4.1 Introduction

Recently there is a strong interest in developing networking techniques for moving vehicles, to enable wireless communication between vehicles, or between vehicles and fixed infrastructures near the road. Generally, three types of applications are emerging. First, communications between vehicles can enable various traffic safety and traffic information applications, developing the need for ad hoc communication between vehicles [94, 113, 72, 157]. Second, cars can serve as mobile sensors, providing a range of sensed information involving information related to the car, driving condition, road condition, traffic and environment – that can be uploaded into some form of "infrastructure" database to be queried by other cars or by some monitoring application [105, 89]. Third, the ability to do a general-purpose Internet access from cars can keep the car occupants entertained and informed, and can potentially generate services and applications specifically for in-car use [47, 115, 140].

To succeed, these applications need good connectivity to other vehicles or fixed access points (AP). While delay-tolerant techniques [70, 125] can be designed to smooth out periods of disconnections, end-to-end delay and possibility of lost messages do increase with long periods of disconnections reducing

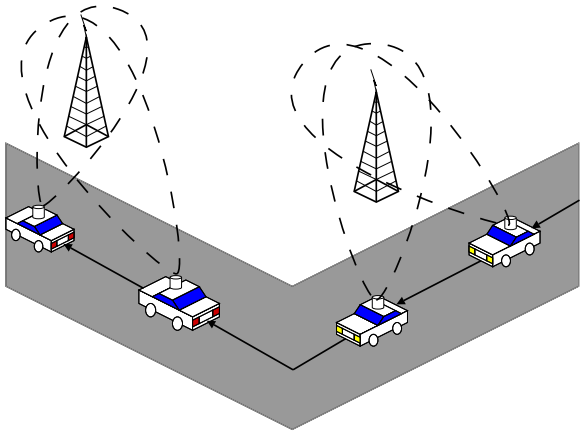


Figure 31: Beam steering and AP selection to improve connectivity.

the effectiveness of many applications. In experiments reported in [47] regarding vehicular access of existing WiFi networks in urban areas, the authors have observed that the median connectivity duration to an AP is about 13 seconds, but the average inter-arrival times for "associable" APs is about 75 seconds, indicating possible long periods of disconnections. In addition the link-layer delivery rate is about 80% even when connected, which would be considered quite poor for TCP.

In this chapter, we will study a physical layer enhancement – directional communication — to improve network connectivity in vehicular context. By focusing energy in one direction, a directional antenna can get a better transmit or receive gain for a targeted direction compared to an omni-directional antenna [154, 153, 132]. In addition, directional antenna has the potential to provide a better immunity from co-channel interference [154, 97, 132] and multipath fading [35]. However, having a directional antenna alone is not sufficient for a moving vehicle. The direction must be steered appropriately for the best link quality. The steering must be done on a continuous basis as the car moves so that good connectivity can be maintained to the appropriate network node. The beam steering should be done in such a way so as to increase the duration of connectivity and improve the link quality. See Figure 31.

4.1.1 MobiSteer Design Goals

Our goal here is to develop practical beam steering techniques so as to maximize the duration and quality of connectivity between the moving vehicle and fixed access points (AP). We address this goal by developing *MobiSteer*, a 802.11-based mobile network node that uses steerable-beam directional antenna to be specifically used in moving vehicles with the appropriate beam steering technology. The current design of *MobiSteer* addresses the scenarios driven by the second and third applications mentioned above, where the vehicles access a fixed network (Internet) using one-hop 802.11 links. We use 802.11b/g as the link layer because of its wide availability, though much of the techniques developed here are not link layer specific and would apply as well to 802.11p based DSRC [5, 19], for example.

In the model we consider, the roadside APs use regular (omni-directional) antennas and the vehicle uses a steerable-beam directional antenna. There are two reasons for choosing such a model. First, omni-directional antennas are appropriate on the APs, as they may be associated to multiple vehicular nodes in different directions; thus to use directional antennas, they have to perform certain coordinations with their clients so that the beam from the AP steer to the right client at the right time. This is hard to do in a random access environment. Second, use of this architecture opens up the possibility of using existing 802.11 networks [47] that are now deployed widely. This enables cars to directly use existing 802.11 public hotspots, mesh networks, or even home networks that now blanket many urban regions. Accessing home 802.11 networks from moving cars in the neighborhood streets is attractive as most such 802.11 APs carry a broadband backhaul that is often idle [47].[1]

Given this model, we address the question of how the directional antenna beam steering should be done for maximizing duration and quality of connectivity to fixed APs. The quality of the link is determined by the PHY-layer data rate that is possible in the link between the vehicle and AP.

[1]The issue of security and open access are important in this context. However, they are a matter of developing appropriate application and protocol support, and is not our direct concern in this chapter. Some of these discussions can be found in [47]. Community building efforts such as FON [6] are also relevant in this context.

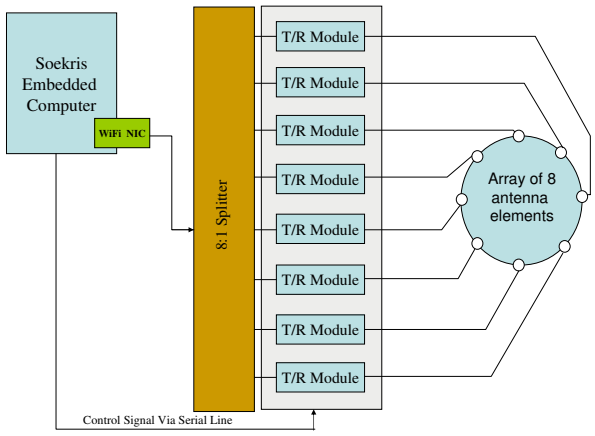


Figure 32: Block diagram of directional antenna components.

When the moving vehicle hears multiple access points, the decision of handoff arises. Essentially, the mobile node has to decide which access point to associate and which beam to use to communicate with the access point to get better connectivity. This boils down to the problem of designing beam steering techniques along with proper handoff decisions.

4.1.2 Contributions

The contributions in this chapter are three fold.

1. We show that use of steerable beam directional antenna with an appropriate beam steering technique can provide a significant performance advantage over using a fixed beam. An equivalent omni-directional beam pattern is used as a comparison point.

2. We develop practical beam steering approaches for two modes of operation – *cached* and *online*. In the cached mode, radio survey data is collected during the idle periods when the vehicle is not communicating with the fixed infrastructure and a geocoded RF signature database is created and maintained for frequently driven routes. This database is

used to drive an algorithm that generates a trace of how beams should be steered and handoffs initiated as the car moves along the known route. In the online mode, on the other hand, no such database exists. We develop simple heuristics for beam steering and handoff based on lessons learnt from our experimental study.

3. We perform extensive measurements in a *controlled environment*, where we deploy our own road-side APs to aid performance data collection. We also provide measurement data for accessing *in situ* networks [47], where we drive around in urban areas with reasonably dense population of 802.11 networks. The measurements show the power of our designed beam steering approaches.

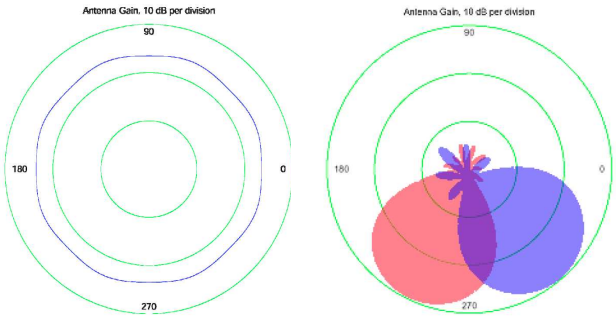
The rest of the chapter is organized as follows. In Section 4.2, we provide a detailed description of the architecture of *MobiSteer*. Section 4.3 describes the beam steering algorithm for the cached mode of operation. Section 4.4 presents detailed performance results for this mode. In Section 4.5 we discuss the online beam steering algorithm and also present the performance results. Section 4.6 discusses the related work. We conclude this chapter in Section 4.7.

4.2 MobiSteer Architecture

In this section, we describe the architecture of *MobiSteer*. First, we describe the hardware and software setup used in our experiments. Next, we describe the two different modes of operation, and then discuss methods of collecting data that drive the techniques.

4.2.1 Hardware Setup

Our directional antenna set up uses electronically steerable Phocus Array antennas from Fidelity Comtech [1] for the 2.4 GHz band used in IEEE 802.11b/g. The Phocus Array antenna system consists of eight element *phased arrays* driven by eight individual T/R (transmit-receive) boards that receive radio signals from the wireless card via an eight way RF splitter (See Figure 32).



(a) (b)

Figure 33: The beam patterns for Phocus Array antenna: (a) omni-directional; (b) two directional beams. The concentric circles are 10db per division. Taken from [1].

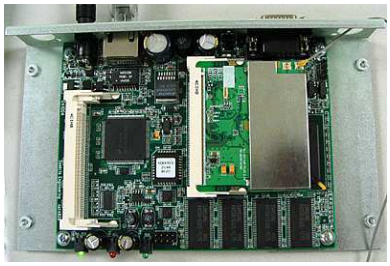
The phased arrays combine radio waves by introducing different phase differences and gains in the eight arrays [154, 1]. A T/R board is essentially a vector modulator with bi-directional amplifier controlled by software. Various beam patterns are possible by setting the phases and gains in different boards. The antenna is set to behave identically for transmit and receive, i.e., *the antenna gains for transmit and receive are the same.*

The software control on the antenna to produce different beam patterns is achieved via serial-line commands from an embedded computer (a Soekris net4511 board [18]). The Soekris net4511 embedded computer has a 100/133 Mhz AMD processor, 64MB SDRAM and a compact flash card interface used for storage. It also has a miniPCI and a PCMCIA interface. We use a 802.11 a/b/g miniPCI card based on Atheros [3] chipset with an external antenna interface. The embedded computer runs pebble Linux [15] with the Linux 2.4.26 kernel and the widely used `madwifi` [12] device driver for the 802.11 interface.

While many beam patterns are possible using the phased array, the manufacturer ships the antenna with 17 pre-computed patterns – one omnidirectional beam and 16 directional beams, each with an approximately 45° half-power beam-width and low side lobes. Each directional beam is overlapping with the next beam and is rotated by 22.5° with respect to the next, thus

(a) *MobiSteer* node – open.

(b) *MobiSteer* node – in cover and mounted on a car.



(c) Soekris embedded router platform for APs in controlled experiments.

Figure 34: Hardware setup.

covering the 360° circle with 16 beam patterns. The directional gain is about 15dBi. Figure 33 shows the manufacturer provided beam patterns. In the experiments reported in this chapter, we have used only the 8 non-overlapping beams out of the 16 beams to limit the number of experiments. Because of the overlapped nature of the beams, we found in our early experiments that the advantage gained from use of all 16 beams is marginal. We refer to the omni-directional beam with beam index 0 and the 8 directional beams we use with beam indices 1 to 8. Adjacent beams are numbered successively.

In order to get the location of the vehicle along the route it travels, we

use a USB-based Garmin [7] GPS receiver inside the car that is connected to the embedded computer via a PCMCIA-to-USB converter. This GPS receiver provides a position accuracy of less than 3 meters 95% of the time. The embedded computer is powered via a PoE interface. We used the car battery and a PoE injector for power. For convenience, we will refer to the entire vehicular setup, including the embedded computer with 802.11 and GPS interfaces and the directional antenna as the *MobiSteer node*, in the rest of this chapter. The setup is shown in Figure 34(a) and (b).

In Figure 34(c), we show the embedded router platform for the APs we use for our controlled experiments described in Section 4.4. The APs are Soekris [18] net4826 router boards with similar Atheros based 802.11 a/b/g miniPCI cards connected to regular rubber duck omnidirectional antennas. The APs also run the same base software (pebble Linux and `madwifi` driver) as the *MobiSteer* node.

4.2.2 Software Setup

The directional antenna beam pattern is changed via sending a serial line command from the embedded computer. The antenna vendor, Fidelity Comtech, supplied us with a patched `madwifi` driver, which implements an interface for user level programs to control the antenna beam patterns through the `/proc` virtual file system in Linux. On receiving a command via the `/proc` interface, the madwifi driver sends the command to the antenna controller over the serial interface and initiates a busy loop in the driver for 150 μs in order for the beams to stabilize. This is the nominal beam switching latency. We found that the `/proc` interface method added a huge delay in user to kernel communication(75 ms). So, we implemented a Linux `ioctl()` interface for controlling the antenna that incurred only an additional delay of 100 μs. Including this overhead, the total beam switching latency is 250 μs.

The `madwifi` driver allows creation of additional raw *virtual* interface (`ath0raw`) for a physical wireless interface. The virtual interface allows reception of all 802.11 frames (control, management, data) as if in the monitor mode, while the main interface can still operate in the ad hoc or infrastructure

mode. We modified Kismet [11], a popular wireless packet sniffer software to *optionally* capture all packets received on the raw virtual interface. Kismet communicates with the GPS server, running as a daemon (`gpsd`), and stamps the current time and GPS coordinates with each received frame from any AP. Each received frame (from APs) is also annotated with an index for the current beam pattern on the antenna, the current channel, the PHY-layer data rate and the SNR (Signal to Noise Ratio). The SNR, data rate, channel information for each received frame is obtained from the radio-tap header appended by the `madwifi` driver for each received frame. The entire frame with this annotated information is logged during the drive. We refer to this logged data as the *RF signature database*. This is built *during the idle periods* of the *MobiSteer* node when it is not communicating. This is used for later analysis.

4.2.3 Operational Modes

MobiSteer operates in two modes – *cached* and *online*. The cached mode is applicable when the car is in a familiar driving route (e.g., home to work). As noted before, the RF signature database is populated and maintained whenever the *MobiSteer* node is not in communication. This database is used later to drive an optimal beam steering and AP selection algorithm discussed in Section 4.3, when the *MobiSteer* node wishes to communicate and is on a "familiar" route. Thus, any "idle" drive provides *MobiSteer* with "samples" for the multidimensional <*location, timestamp, AP, channel, data rate, beam, SNR*> dataset. More drives on the same route provides more samples and thus better statistical confidence. We use an averaging technique (described in Section 4.3.1) for use in our algorithm. We have ignored a few practicalities such as AP churn in our current work; however, this can be easily accommodated in our technique by providing more weight to recent samples, ignoring APs not heard recently, etc.

The applicability of the cached mode technique is not limited to the routes driven by the same user. It is possible to share RF signature databases by multiple *MobiSteer* nodes. For example, databases could be uploaded to a central server and all nodes can benefit from such shared database. Whenever

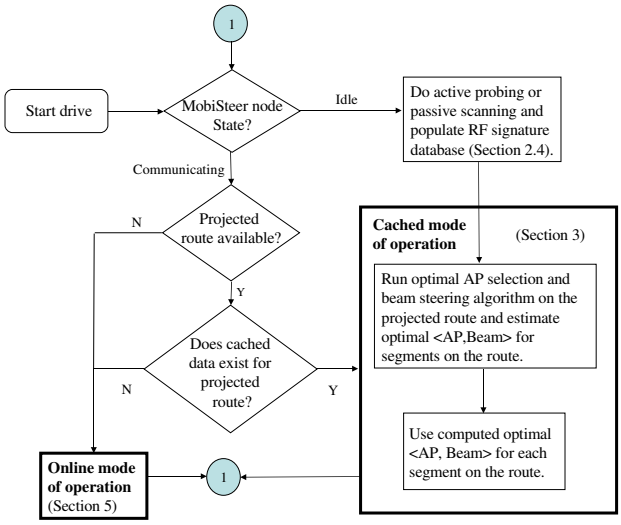


Figure 35: *MobiSteer* operation.

a user wants to travel a particular route, she can download the RF signature database, run our algorithm on it to get an optimal beam steering and AP selection pattern for the route. The CarTel [89] architecture is a perfect framework of how this can be done. The issue of sharing, however, is orthogonal to our work here. For the cached mode of operation, *MobiSteer* must know the projected route and an estimated speed of the car along every point on the route. We assume that such information is available from the navigation system and/or prior driving history.

In the online mode of operation, no such database is available. This mode of operation is used when the user travels in a previously untraveled route and wants to communicate with APs in the route. Here, the *MobiSteer* node scans the environment in all the beams and channels using active probing (discussed in the next section) and chooses the best beam and AP combination depending on the SNR values of the probe response frames received. This mode of operation is discussed in detail in Section 4.5. Figure 35 presents an overview of the modes of operation of *MobiSteer*. Note the possibility of switching between the modes of operation during the drive. A hybrid mode of operation is also possible, though we do not explore this here. *MobiSteer* switches to data collection whenever it is idle.

4.2.4 Data Collection

Two methods of data collection are used to build the RF signature database – *passive scanning* and *active probing*. Active probing is also used for the online mode of operation. In both these methods, the frames received at the *MobiSteer* node are used to infer the quality of the link between the AP and the *MobiSteer* node in both uplink and downlink directions. It is acceptable to assume that uplink and downlink qualities are similar. This is because the radio propagation characteristics are symmetric and the antenna transmit and receive processing are identical by default.

In passive scanning, the *MobiSteer* node scans for *any frame from APs* using the monitor interface (`ath0raw` in our case) on all antenna beams staying on each beam for about 200ms. This is done for every channel in sequence. The interval 200ms is selected due to the fact most APs broadcast beacons at the default interval of 100ms. Thus, 200 ms provides enough opportunity for the car to receive at least one frame from any neighboring AP when the surrounding medium is idle. For a fast moving vehicle, however, 200ms holding time for each beam can be too long. For example, with 9 beams (8 directional and 1 omni) the complete scan on 11 channels takes $9 \times 11 \times 200$ ms $= 19.8$ sec. At 100 km/hr the car can move 550m during this time. Compare this with the typical range of an AP, which is 150m according to [29]. This distance can be enough for the car to miss beacons from some APs. From our experience in data collection using passive scanning, we feel that this rarely would present a problem, as the data collection is likely to be repeated many times along the same route. Given the randomness in beacon generation and driving speeds, we expect that all APs would be heard over time on the beams/channels they are supposed to be heard. Also, several optimizations can be used. For example, from our own wardriving experience in connection with this work, we found over 90% of the APs are in channels 1,6, and 11. Here, the scanning time can be reduced significantly by scanning only in these three channels.

The second method of data collection is to use active probing to build the RF signature database faster. In active probing, we do not wait for beacons; instead the *MobiSteer* node sends out periodic *probe request* frames and record *probe responses* from the APs. Our software generates probe request frames

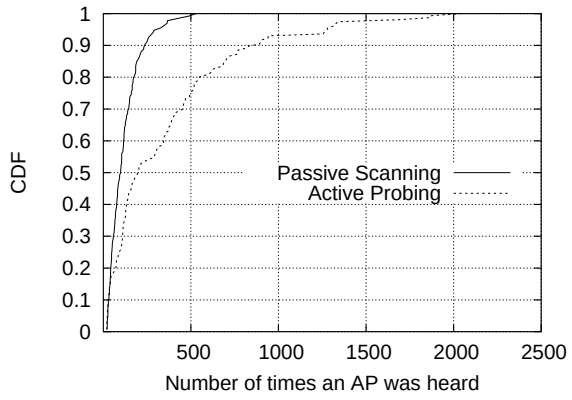


Figure 36: Performance of active probing vs. passive scanning in building RF signature database.

that can be sent at customizable intervals. These probe request frames have the same format as those used in 802.11 probe request frames. Similar to passive scanning, in each channel, the *MobiSteer* node cycles through all 9 beams. Whenever a node sends a probe request frame, it takes about 30ms for the AP to send back the probe response frames. So with each beam it first sends out a probe request frame and wait for 30ms to gather probe responses, before switching over to the next beam.

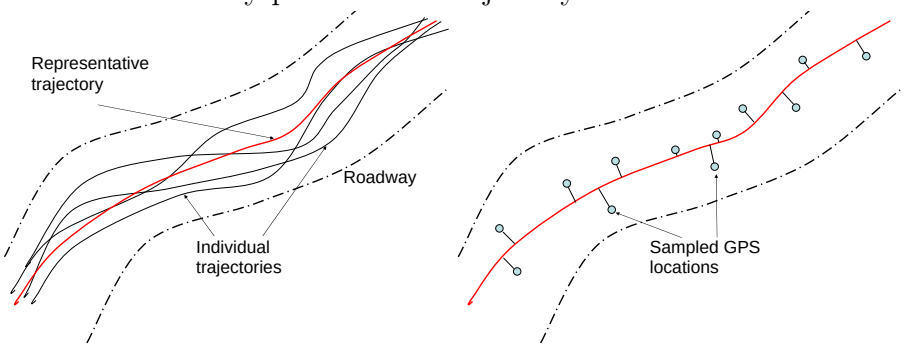
Evidently, active probing allows quicker sampling, and thus gets more samples per drive. Figure 36 shows this. This data is based on 8 drives on the same 5 km route with active and passive probing near Stony Brook university campus with a fairly dense population of APs. It is thus possible that a stable RF signature database can be built with lesser number of drives on the same route when active probing is used. Either passive scanning or active probing can be used to support the cached mode of operation. However, active probing is exclusively used in the online mode to reduce the scan time. It is also possible to update the RF signature database when the *MobiSteer* node is actually communicating. Data packets then simply act as probes.

4.3 Cached Mode Operation

The cached mode of operation rely on an existing RF signature database collected during "idle" drives. The idea here is to utilize this existing data to choose (i) APs along the route to connect to and (ii) appropriate directional beams to use. This is to be done at all points on the route. The goal is to maximize performance in terms of data transfer rate.

4.3.1 Optimal AP and Beam Selection

The data collection phase (either passive scanning or active probing) builds the RF signature database with tuples like <*location, time stamp, AP, channel, data rate, beam, SNR*> for a trajectory of interest, where *AP* denotes the BSSID and MAC address of the AP, *channel* is its channel, *beam* is an indicator for the beam used to receive this frame,[2] *SNR* is the Signal to Noise ratio value of the received frame, *location* is the GPS coordinates where the frame is received, *time stamp* is the time at which the frame was received and *data rate* is the PHY-layer data rate of the frame. The cached mode of operation uses an optimal AP and beam selection algorithm that computes the best AP to associate with at every point in the trajectory and the best beam to use to commu



(a) (b)

Figure 37: (a) Notion of representative trajectory. (b) Mapping of GPS samples.

[2]We assume that the antenna orientation with respect to the car is fixed in order to have a common representation for a *beam*.

To develop this algorithm, we first discretize the problem from practical consideration. We break the trajectory into segments of length Δ and assume that any AP or beam selection decision is taken only at the start of a segment. In each segment, we scan all beams on all channels to gather data samples. The value of Δ depends on the average speed of the car along the route and the time it takes to completely scan all beams on all channels. Two values of Δ are used in the experiments we report, 5m and 40m, depending on the average speed of the car in the environments the studies were done.

In the data collection phase for the same route, the vehicle is likely to drive on slightly different trajectories on different runs. This is true even when the car is always driven on the same lane. Even if it is on the exact same trajectory, there is no guarantee that the locations provided by the GPS is exactly the same because of GPS accuracy issues. Thus, there is a need to compute some notion of a *representative trajectory* for a driving route. See Figure 37. There could be many ways to do this – for example, (i) by merging all location coordinates in the RF signature database for the same route and then interpolating a smooth path through them, or (ii) computing an average trajectory from the individual trajectories for each run, or (iii) by simply picking any one trajectory as the representative. We used the third method to save computational effort. The trajectories used in our study were quite close, and the actual method will have little influence on the results, if any.[3]

Every RF database tuple, collected during different runs of the same route, is mapped onto a point on the representative trajectory that is closest to the location of the tuple. See Figure 37. For every segment of length Δ on the representative trajectory, all such tuples mapped onto this segment are analyzed to compute the best AP and beam for this segment. This is done by computing the average SNR for each AP and beam combination and selecting the *best beam* for each AP for every segment. This will be used in the algorithm in Section 4.3.1.1.

Recall that the beam steering latency is about 250μs. This is negligible

[3]In our experiments, we always stuck to the same lane. We have anecdotal evidence that two lane roadways can still be approximated quite well using a single trajectory. We do not have experience with wider roadways, where it may be possible that we need more than one trajectory to represent a driving route.

compared to the travel time in a segment. For example, for a segment of length (Δ) 5m, even at a very high speed of 120km/hour, the travel time is about 150ms. Even at this small segment size and high speed, the beam steering latency is around 0.1% of the travel time. We have used 5m as the lower bound of the segment length Δ. A smaller length is meaningless as it becomes comparable to GPS accuracy limits. Sometimes we have used higher lengths (upto 40m) because of some practical limitations we had. A small segment length implicitly requires a rich RF signature database with a dense set of samples. When driving speed is higher, samples collected become sparser – thus, for the same sample density, more runs are required.

While beam steering latency can be ignored, handoff latency may not be negligible. This may vary from a few ms to several 10s of ms, depending on whether channel is changed and/or authentication is used [118].[4] Since handoffs could be expensive at high speeds, the latency must be factored in the optimal AP and beam selection algorithm. This can be done in the following fashion.

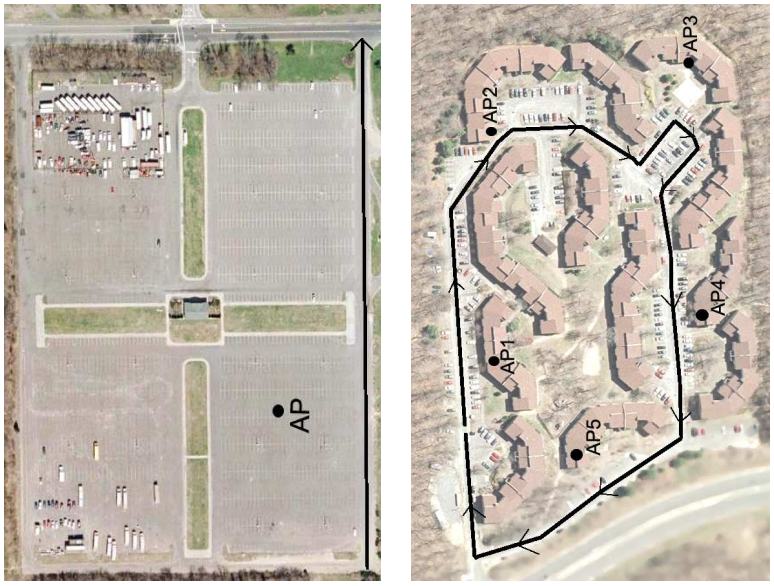
4.3.1.1 Optimal Handoff Algorithm

Assume that the vehicle speed is known for each segment. This can be estimated from the RF signature database tuples at the time of computing the representative trajectory. At start of a segment i, we need to make a decision about the AP to use for this segment. Assume that the speed of the car is s_i at segment i and that the handoff latency when handing off from AP k to AP j is $h(k, j)$. Then, with this handoff latency, the useful time in segment i is given by

$$t(i, k, j) = \frac{\Delta}{s_i} - h(k, j).$$

Note that $h(k, k) = 0$. If the estimated average PHY-layer bit rate when associated with AP j in segment i *with the best beam* is $r(i, j)$, then the maximum number of bits that could be transferred in segment i is $r(i, j)t(i, k, j)$ with this handoff. Here the quantity $r(i, j)$ represents the link quality between AP j

[4]Note that in WLAN deployments handoff latency is much higher because of probing. But probing is not needed here.



(a) (b)

Figure 38: Experimental environments for the controlled experiments: (a) a large empty parking lot in Stony Brook university campus, (b) the graduate students' apartment complex in the same campus. The black dots show the locations of the APs and the black arrow shows the driving route.

and the car within segment i.

Based on this, a simple dynamic programming algorithm can compute the best AP for each segment of the trajectory such that the aggregate number of bits transferred can be maximized. The algorithm is as follows:

for $i = 2$ to Number of segments **do**
  for $j = 1$ to Number of APs **do**
    $best(i, j) = \max_{\forall k}(r(i-1, k)t(i-1, k, k) + r(i, j)t(i, k, j))$
    $prev(i, j) = k$ value for which the previous quantity is maximized.
  end for
end for

After this algorithm is run, the maximum $best(N, j)$ is picked for the last segment (N). Assume that this is $best(N, m)$. Then, $prev(i, j)$ is traced backwards as $prev(N, m)$, $prev(N-1, prev(N, m))$, etc., thus enumerating

the best AP for each segment i. If no AP is visible for some segment i on the trajectory, a designated null AP is assumed (with bit rate $r(i, null) = 0$) so that the algorithm can run correctly.

This computed AP and beam combination for each segment is used to drive the beam steering and handoff of the *MobiSteer* node. Note that the algorithm is quite general. We have justifiably ignored beam steering latency and pre-selected the best beam for each AP in each segment. However, if the beam steering latency is high in a different hardware set up, it can be accounted for in a similar fashion as handoff latency.

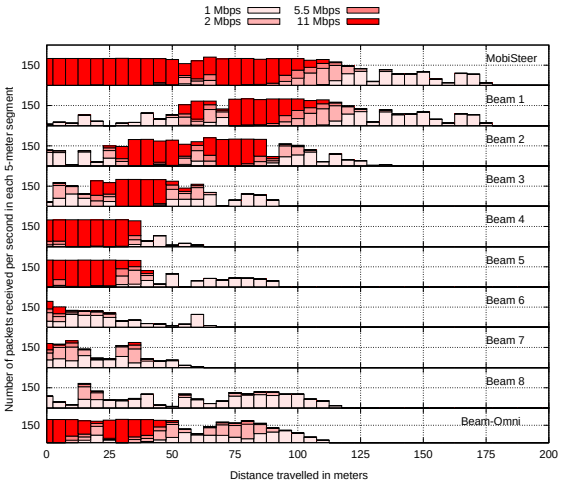
In certain scenarios we can only measure the average SNR values within a segment and cannot estimate the average PHY-layer data rate. For these cases we can run the above optimal algorithm using average SNR values for $r(i, j)$. The algorithm determines the optimal AP and beam combination for each segment that maximizes the average SNR for the trajectory. Note also that the speed of the car is an input the algorithm and thus must be estimated. The algorithm must also know the route. We assume that this information is available from the navigation system or from prior driving history that can be cached.

4.4 Cached mode: Experimental Results

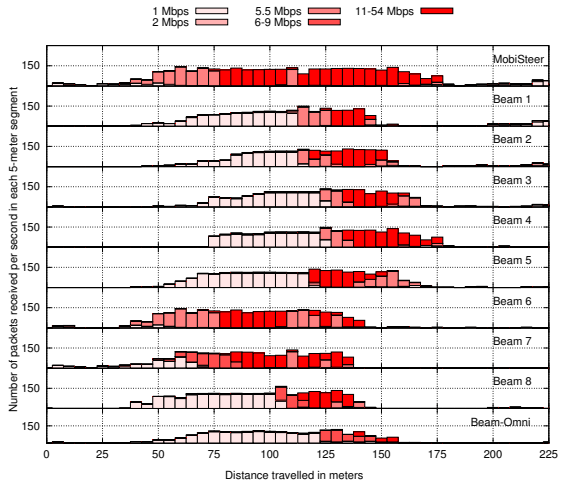
In this section, we provide a detailed performance evaluation of the optimal AP and beam selection procedure described in the previous section.

4.4.1 Scenarios

We will use two scenarios for our experiments to evaluate our beam steering algorithm. The first is a "controlled scenario" where we deploy our own APs. See Figure 34(c) and associated description for the AP architecture. We use two specific controlled scenarios - (a) a large empty parking lot in Stony Brook University campus without any neighboring buildings and large trees



(a) Parking lot (802.11b is used).



(b) Apartment complex (802.11g is used; only one AP is shown).

Figure 39: Performance for individual beams for a specific AP along the drive.

— offering a virtually multipath-free environment with little, if any, external interference, (b) the graduate students' apartment complex in the same

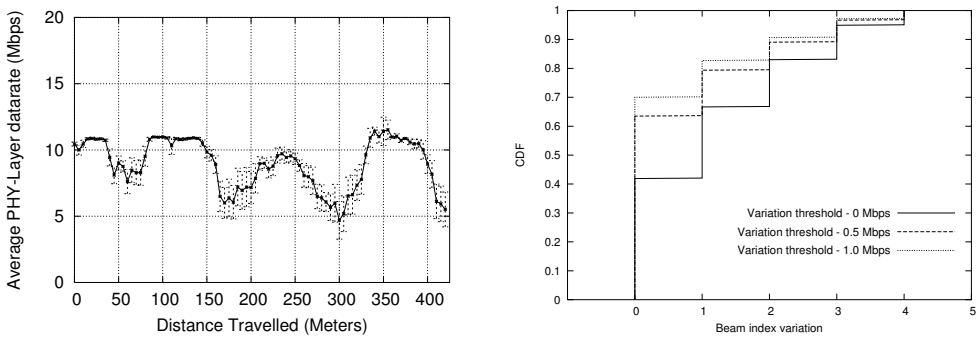


Figure 40: Demonstrating sampling error: (a) average PHY-layer data rate at every segment with 90% confidence interval; (b) variations in the best beam selection.

campus — offering diametrically opposite environment, rich in both multipath and external 802.11 traffic. See Figure 38 for the satellite image to get some understanding of the environments.

We use only one AP in the parking lot. It has been hard to use more than one AP gainfully in such a "clean" environment! However, we use five APs in the apartment complex. Here, the APs are carefully located so that at each point on our driving route, typically two APs are always heard and all points on the driving route are covered by at least one AP. This controlled set of experiments demonstrates the beam steering advantage by doing actual measurements of link-layer data transfer rate between the *MobiSteer* node and the APs. The APs are run on the same channel. Using just one channel in the experiments removes the channel variable from our experiments and lets us concentrate on only the beam steering aspect – the main focus of our work.

The second scenario uses *"In Situ"* wireless networks [47] in various urban roadways near Stony Brook University campus. This set of experiments demonstrate the beam steering advantage when using the APs that are deployed in an uncoordinated fashion in urban areas. Here also one single channel is used for the experiments. The most popular channel (channel 6, which is configured as the default channel in most commercial wireless routers) is used so that we can have most APs visible to the *MobiSteer* node.

In the *In Situ* experiments, we did not perform actual data transfers. There are a couple of reasons for this. First, this would restrict us using only

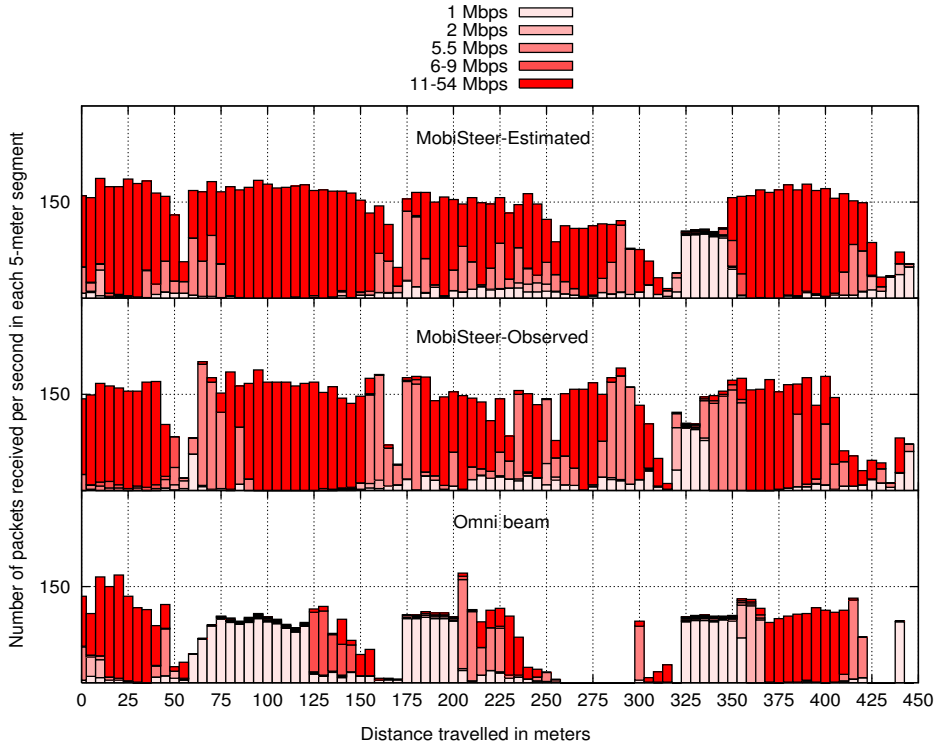


Figure 41: Estimated and observed performance on the best <*AP,beam*> along with the omni-directional beam performance for the entire path

open APs. We found that only 22% were open in our drive! Such a low fraction is atypical. We conjecture that in university and high-tech industrial areas (as ours) more wireless users know and care about security. Second, this would also require us to do statistically significant load tests that may interfere with the activities of the owners of the concerned networks. Thus, we have restricted load tests to only controlled scenarios. In *In Situ* scenarios SNR measurements coupled with coverage results are used to estimate performance advantages.

4.4.2 Collecting and Analyzing Data

In this section we first describe how we have collected data in the controlled scenario. Then we analyze the collected data for interesting properties.

The APs operate in pseudo-ad hoc mode and continuously unicast 1000 byte UDP packets to the *MobiSteer* node at a constant rate of 300 packets/sec. The ad hoc mode is chosen instead of infrastructure mode so that the *MobiSteer* node can receive packets from any AP rather than only the specific AP it is associated to. This enables us to exclusively study the beam steering part of our algorithm. If and when the *MobiSteer* node receives any packet it records the tuple <*location,time stamp, AP, channel, data rate, beam, SNR*> as indicated before. The default auto-rate algorithm [42] in the card driver is used for rate adaptation.

The data collection in the controlled experiments is done fairly conservatively to eliminate any source of error. In order to eliminate any possibility of missing packets due to beam steering delays (which is already negligible), only fixed beams are used for each drive and beams are switched only between drives. So a set of 9 drives on the same path gives us data on each of the 8 directional beam plus the omni-directional beam. Each drive is done in a very slow speed (about 10 miles/hour). Segment length Δ is assumed to be 5m. Because of the slow speed, each segment receives enough packets to populate the RF signature database. We have done 8 such sets of drives on different days and times in order to analyze the variability of the data. Recall that we are using one channel as all our deployed APs are in the same channel.

Figure 39 shows the number of packets received per second on different beams in every segment of the drive based on the average statistics from the entire RF signature database collected over all the 8 drives. We also show the PHY-layer data rates with which these packets were received. When we did the parking lot experiments (Figure 39(a)) we used 802.11b. When we moved over to the apartment complex experiments (Figure 39(b)) we realized that we could take advantage of the higher data rate in 802.11g. So the plots have different data rate ranges. In any case, we will only qualitatively study them and not compare across them. In the apartment complex experiments (Figure 39(b)) we show the results for only one AP out of the five available for brevity. Plots for the other APs are qualitatively similar.

In the parking lot experiments (Figure 39(a)), we can clearly see how the performance of different beams vary as the *MobiSteer* node moves along

the path. The performance improves as the car approaches the AP and then slowly fades as the car moves away from the AP. Also, the location of the peak performance shifts gradually as beams are changed from 1 to 2, 2 to 3 etc. Beams 6, 7 and 8 do not perform well as they are pointing to the opposite direction. The omni beam has only modest performance. The best beam (plot labeled *MobiSteer*) performance at every segment clearly outperforms any fixed beam and is significantly better than the omni beam. The aggregate throughput improvement over omni is more than twice.

In the apartment complex experiments (Figure 39(b)), the results are similar. However, given the rich multipath environment the shift in the peak performance point from beam to beam is not as clear any more. Also, no beam is clearly very poor. Unlike the parking lot, all beams achieve good data rate and offer similar connectivity durations. However, the best beam (labeled *MobiSteer*) clearly outperforms any individual beam and the omni beam. Here, the aggregate throughput improvement over omni is more than four times.

This set of plots brings out the potential of using beam steering as it exploits the beam diversity. There is also an improvement in the duration of connectivity by using beam steering compared to using omni-directional beam. This improvement is about 75% and 50% respectively in the two experiments. In addition, we notice that the unicast transmissions occur more often at higher PHY-layer data rates when using the best beam as compared to using the omni-directional beam. This is because the auto-rate algorithm switches to higher PHY-layer data rate when the number of packet retransmissions reduce. Since unicast data transmissions from an AP to the *MobiSteer* node includes synchronous ACK transmissions on the reverse direction, lower packet losses for unicast transmissions also imply that both uplink and downlink quality have improved, Going forward, for brevity we restrict our analysis to the apartment complex experiments only. This evidently offers a more challenging environment. Also, we have more APs in this environment to study AP selection.

Next, we analyze the variability of the collected data in the RF signature database over different drives in the same route. This is important, as more

variability will require collection of a large number of samples and will make the entire process less reliable. Figure 40(a) shows the average PHY-layer bit rate for the best <*AP,beam*> combination at each segment. The average is made over the 8 runs. The 90% confidence interval is also shown. While variations are indeed present, they are not significant. On careful analysis we also noted that the variations are a bit higher in the parts of the route which was surround by buildings *on all the sides*. We conjecture that this variation is due to the severity of multipath fading problem due to too many reflections.

Figure 40(b) shows the variability from another perspective. It shows how much the best <*AP,beam*> selection would vary if the algorithm is run separately on the data set of each individual drive. It uses one of the drives as a reference and plots the difference in beam numbers for the best beam (recall that adjacent beams have successive beam numbers) over segments for each drive. The *AP* selections are not shown as the runs rarely differed on the selection of APs. The results show that both AP and beam selections are quite stable, and whenever the beam selection does vary, often an adjacent beam is selected. The plots use a notion of variation threshold where the beam selection process ignores variations in the bit rate values that are below certain threshold. For example, the plot shows that if a difference of 1 Mbps is acceptable, then 70% of times there is no difference in the best beam selections for different drives, and 83% of the times the beam selections remain within adjacent beams.[5]

So far, we have only analyzed the collected data. To evaluate the performance of the optimal beam steering and handoff algorithm as presented in the previous section, we run the algorithm on the aggregate RF signature database. Since the PHY-layer data rate is available, the data rate – rather than SNR – is used for the best beam selection. The algorithm provides the best <*AP,beam*> for each segment that would maximize the overall throughput.

[5]It is quite possible that we will get much better results here if we use all 16 beams provided by the antenna.

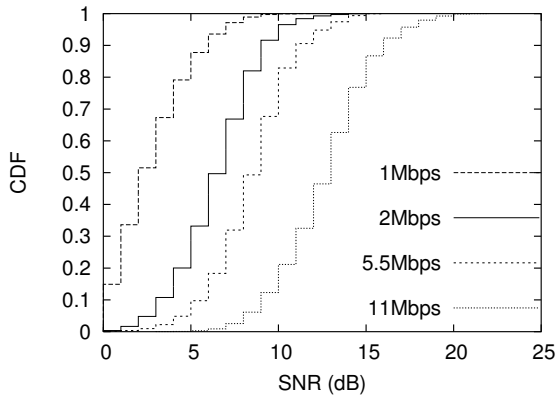


Figure 42: Measurement data demonstrating the relationship between SNR and PHY-layer data rate in our setup.

4.4.3 Optimal Steering Results

We run the APs in pseudo-ad hoc mode as before with the same traffic. The *MobiSteer* node now steers the beam and selects APs following the algorithm output. No real handoff is done, as the *MobiSteer* node receives packets from all APs in the ad hoc mode. The *MobiSteer* node simply ignores packets from APs other than the one selected for the current segment. The beam steering is done as follows. The GPS receiver on the *MobiSteer* node continuously samples locations, maps the sampled location to the nearest point on the representative trajectory and determines the segment the vehicle is on. At the time of a segment change, the *MobiSteer* node steers the beam indicated by the best beam pattern from the algorithm.

Figure 41 shows the observed and estimated performances on the best $<AP,beam>$ for every segment on the drive. The estimated performance is derived from the dataset that generated Figure 40(a). The observed performance is an average of 4 runs. The observed performance is close to the estimated performance with some variations due to the temporal changes in the propagation environment. A careful visual inspection also reveals that the difference is greater at the portions where large variability has been observed in Figure 40(a). The observed performance is significantly higher than the performance when using the omni-directional beam alone.

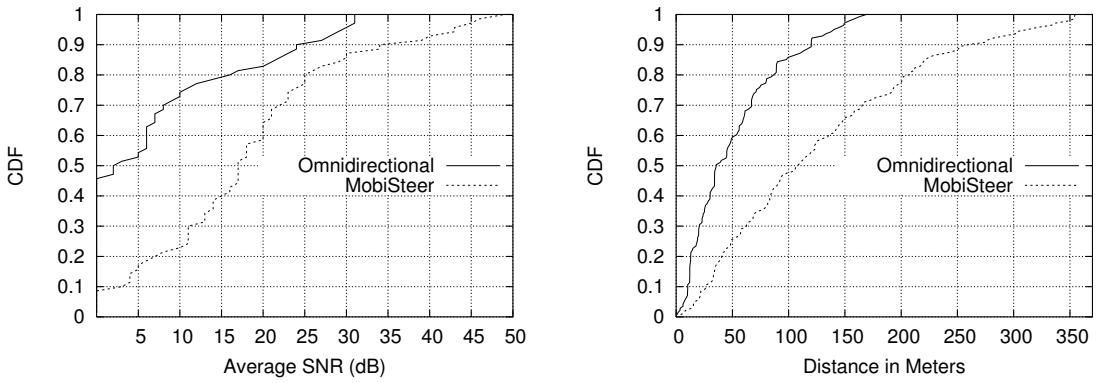


Figure 43: (a) Average SNR from *In Situ* APs in each 40 meter segment along the route with beam steering and with omni-directional beam. (b) Distance along the route each AP was heard with beam steering and with omni-directional beam.

4.4.4 Experiments with *In Situ* Networks

The previous experiments in a controlled setting have established the power of directional beam steering and the viability of cached mode of data collection and computation of optimal steering and handoff. However, the experiments have been done in a controlled setting with carefully planned deployment of the 5 APs with good visibility everywhere on the driving route. We will now study the power of our cached mode technique using 802.11 APs normally present in urban environment.

For these experiments, we drove around the urban areas in the surroundings of Stony Brook university campus on different days and collected AP information using active probing. The route through which we drove (about 5 km) have both offices and homes on both sides of the road. The average speed along this route is around 30–40 miles/hour. Initial scouting runs through these routes revealed that a very high percentage of the APs were tuned to channel 6. So, for the experimental data collection, we used channel 6 alone. Our dataset consists of 307 unique APs. Since the probe response frames are all sent using the same data rate (1Mbps in our case), we use the SNR of

the received frames to determine the quality of the link. A set of independently performed micro-benchmarks demonstrates that SNR indeed impacts on PHY-layer data rates in our setup. See Figure 42. Clearly, with higher SNR the probability of the link using a higher data rate is higher.

We employ our algorithm on this data set to study the performance of *MobiSteer* and using omni-directional beam. A segment length (Δ) of 40m is used to factor in the higher driving speed. Figure 43(a) shows the CDF of the average SNR in each 40 meter segment along the route driven during the data collection. For each segment, we compute the average SNR for the frames received using omni-directional beam and the maximum of the average SNR for the frames received using each directional beam. This clearly shows the advantage of using directional beams as better SNR helps in achieving better PHY-layer data rate improving the quality of connection between the vehicle and the fixed access point. The median SNR on omni beam is around 2dB and using directional beams is around 17dB. Referring to Figure 42 this presents a significant improvement in data rate.

Figure 43(b) shows the distance for which each AP along the route is heard using omni-directional beams and beam steering. This plot ignores some outliers, where some APs are heard only for a very brief interval (1m or less). This plot shows the usefulness of *MobiSteer* to improve the connectivity duration. The median connectivity duration with an AP (note that this is different from AP range) improves from 40m with omni beam to more than 100m with *MobiSteer.*

4.5 Online Mode

The cached mode of operation described in Section 4.3 is applicable only when the RF signature database for the route is available. When the *MobiSteer* node ventures into a completely new environment, AP selection and beam steering must be done in an online fashion. In this section we present a simple online heuristic to scan the environment on-the-fly and choose the best <*AP,beam*> combination. We will study only the beam steering aspect as handoff and AP selection techniques are well investigated in literature [111, 120].

For online beam steering, active probing is performed over all beams and channels and the SNR values on all probe responses are recorded as described in Section 4.2.4. This is called a *probing* phase. After the probing phase, the <*AP,beam*> combinations, where any probe response is received, are ranked in a table T according to the average SNR values of the probe responses. The *MobiSteer* node then associates with the AP AP_i, if $< AP_i, beam_j >$ has the highest SNR in table T. This $< AP_i, beam_j >$ combination is continued until d consecutive packet drops, when the next best beam (say, $beam_k$) for the same AP_i is selected, if $< AP_i, beam_k >$ exists in table T. This continues until no other beam for AP_i exists in T, or the last selected beam for AP_i failed to transmit a single packet successfully. At this time, the next best AP is selected from table T to hand off to and the corresponding beam is used for this AP. This strategy continues. If such an AP does not exist in T or if the handoff fails to associate, another probing phase is started to refresh table T. The size of the table T and the threshold of d drops that determines when to switch to the next $< AP, beam >$ combination are parameters for the heuristic.

The main penalty incurred when using the online mode compared to the cached mode of operation is the probing time. In order to probe in all the beams and channels, the probing time is around 3080 ms when using active probing. It takes 270ms to scan in all the 9 beams and there are 11 channels (assuming 802.11 b/g). The channel switching delay is around 10ms for our wireless cards. For the online mode of operation using only the omni-directional beam, the probing time is 440 ms as only channel scan and no beam scan is needed. Also, in the online mode, at every instant *MobiSteer* may not be able to use the optimal $< AP, beam >$ combination as these combinations are only discovered during probing and not during communication. However, online operation may have one advantage. Since the probing is done right before communication, it is relatively immune to temporal variabilities due to changes in vehicle locations, propagation environment, etc.

Figure 44 shows the median total number of bytes received by the *MobiSteer* node using various modes of operation in the controlled experiments in the apartment complex scenario from the 5 APs over 8 runs. It also shows the break-up of bytes received at different PHY-layer data rates. The figure

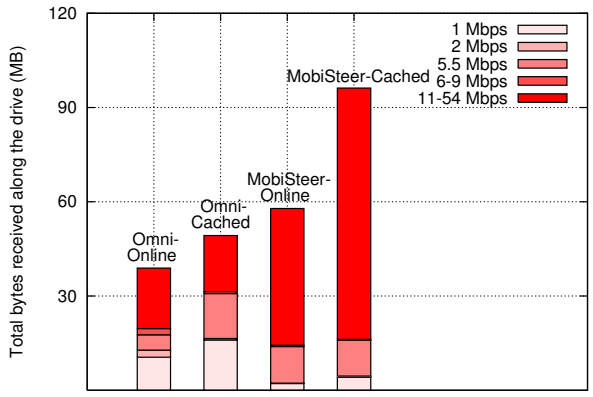


Figure 44: Total Number of bytes received along the drive in "controlled" experiments (apartment complex scenario).

clearly shows that beam steering outperforms omni-directional communication significantly in both online and cached modes. It also shows that cached mode is superior to online. About 39% less packets are received when using online mode as opposed to cached mode. This is due to the time spent in the probing phase and possible use of suboptimal $<AP,beam>$ combination. Also there is over 50% improvement in using *MobiSteer* in online mode compared to using only omni-directional beam in online mode.

In Figure 45 we show the average SNR in each 40 meter segment along the route for the in situ experiments. For the online modes, the zero SNR value indicates a probing phase or periods of no probe response along the route. This figure shows the SNR improvement in using beam-steering compared to omni-directional communication and also the performance benefits of using cached mode compared to online mode. In this set of experiments, the average SNR using *MobiSteer* in online mode is around 11 dB compared to 4 dB when using omnidirectional in online mode. The average SNR using *MobiSteer* in cached mode is 18 dB with a 7 dB gain over *MobiSteer* in online mode. The online mode sometimes performs better than cached mode due to availability of fresher channel estimates, but, in general, scanning all beams and channels in online mode incurs a large penalty.

The main insight from these experiment is that using *MobiSteer* cached

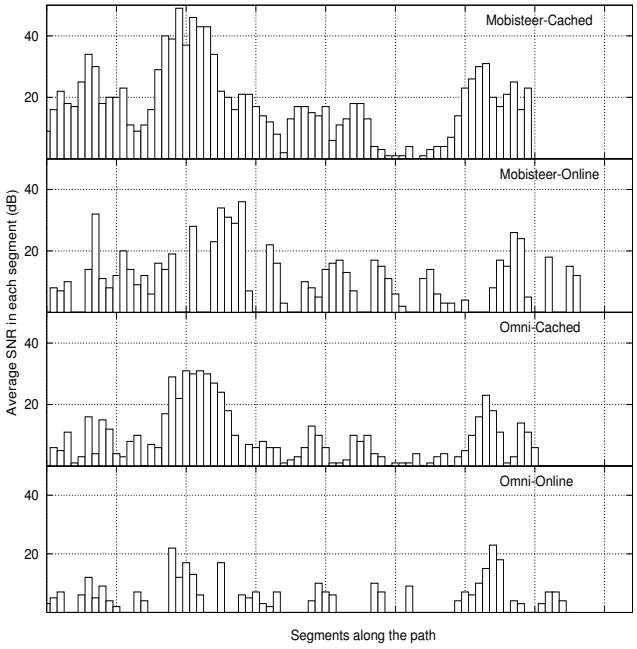


Figure 45: Average SNR in each segment along the drive in *In Situ* experiments.

mode of operation is far superior to online mode, and should be used whenever RF signature database is available. The *MobiSteer* online mode is still superior compared to using omni-directional antenna.

4.6 Related Work

Several works have demonstrated the feasibility of IEEE 802.11 based communication from moving vehicles. In [124] the authors show how long a connection between a moving car and a roadside AP can be maintained while driving at different speeds between 80–180 km/hour. They show that approximately a third of the connections can be reasonably used and communication is possible for about 4–9 seconds in these speeds. Up to 9 MB of data could be

transmitted at speeds around 80 km/hour. These experiments have been conducted in a well-planned environment, similar to our controlled experiments. More recently in [47], experiments are performed to connect to unprotected access points in metro areas in normal driving environments and use them for Internet connectivity. The experiments – performed over 290 drive hours – show that the average duration of connectivity to an AP is about 13 seconds at prevalent driving speeds, and several 100s of KBytes data on average can be transferred during this period using TCP to a server on the Internet. While this seems to be a good news, the data also show that that there could be significant periods of disconnections, necessitating disconnection tolerant protocols at upper layers. In particular, while the median connectivity duration is about 13 seconds, the average inter-arrival times for "associable" APs is about 75 seconds, indicating possible long periods of lack of connectivity. In addition, the link-layer delivery rate is about 80%, which would be considered quite poor for TCP. These observations originally provided motivation for our work.

Several studies investigate the characteristics of unplanned deployment of 802.11 APs in dense urban areas. Prominent of them are several "war driving" studies [24, 26] and extensive measurement studies in [29]. However, these studies do not directly consider feasibility of vehicular access. Other studies have considered vehicular access, but are interested in developing disconnection tolerant protocols [125]. Several architectural and application-layer works have also appeared in literature that considers many uses of urban vehicular grid, including vehicular Internet access [75].

None of the above works consider use of directional antenna to improve connectivity or any other performance measures. However, in other contexts – such as ad hoc or mesh networks – several papers have considered use of directional antenna to improve spatial reuse and immunity from interference. See, for example [97, 57, 132, 55]. These papers mainly considers modifications and extensions of 802.11 for use with directional antennas. They, however, do not directly consider mobility. All of these works are simulation studies on using directional communication.

Experimental studies combining 802.11 and directional antennas are quite

limited. Some researchers appear to do experimental work [54, 133, 49, 119] using various steerable and switched beam antennas. However, actual experimental data has been reported only in [54]. In [49, 119], the authors use the same directional antenna we use in our work. In [49], they study how to enhance security using directional communication and in [119] present ways of exploiting directional communication for better spatial reuse. But no real experimental results are reported. Another paper [133] demonstrates significant throughput improvements in mobile environments using directional antennas with respect to omni-directional. A design is presented, but the results use only high-fidelity simulations.

In wireless communications, signal propagation and antenna literatures, the concept of adaptive beam steering from vehicles or from fixed base stations have indeed appeared [68, 152, 160]. However, experimental work in real environments has been limited. Also, this set of work is not related to either 802.11 or Internet access. To the best of our knowledge, ours is the first experimental study that considers steerable beam directional antennas for 802.11 networks for accessing roadside APs for the purpose of using the Internet.

4.7 Summary

In this chapter, we investigated the use of directional beam steering to improve performance of 802.11 links in the context of communication between a moving vehicle and roadside APs. To do this, we used a framework called *MobiSteer*. *MobiSteer* can operate in the cached mode – using prior radio survey collected during "idle" drives – or, it can operate in an online mode, using probing. The goal is to select the best AP and beam combination at each point along the drive given the information available so that throughput can be maximized. We used extensive experiments – controlled scenarios with our own APs, in two different multipath environments, as well as *in situ* scenarios, where we use APs already deployed in an urban region – to demonstrate the performance advantage of using *MobiSteer* over using an equivalent omni-directional antenna. *MobiSteer* improves the connectivity duration as well as

PHY-layer data rate due to better SNR provisioning. Summarizing the results, *MobiSteer* has improved the throughput in our controlled experiments by a factor of 2 – 4. In *in situ* experiments, it has improved the connectivity duration by more than a factor of 2 and average SNR by about 15 dB. We also demonstrated that cached mode of operation is superior to online mode giving more than 50% improvement in throughput.

The concept of *MobiSteer* can be used in several related vehicular applications. For example, it can be used for ad hoc communications among vehicles. This, of course, adds to the complexity as both communicating nodes have to steer their beams, needing coordination techniques. We present a practical beam steering approach for inter vehicular communication in the next chapter. Another interesting application of *MobiSteer* is localization of roadside APs. Accurate localization is possible, as the moving vehicle is able to take many SNR samples at different directions and locations, thus providing diversity. Though multipath propagation can complicate the measurements, statistical estimation techniques can be used to improve accuracy in such cases. Accurate localization of roadside APs can provide useful datasets for the wireless networking research community to understand better the nature of chaotic WiFi network deployments in urban areas. This will also be useful to create realistic topologies for wireless mesh networking research. We present our localization technique using the *MobiSteer* architecture in Chapter 6.

Chapter 5

Inter-Vehicular Communication

5.1 Introduction

Vehicular ad hoc networking (VANET) has been an emerging area of interest in recent times. The idea here is to form a mobile ad hoc network where a

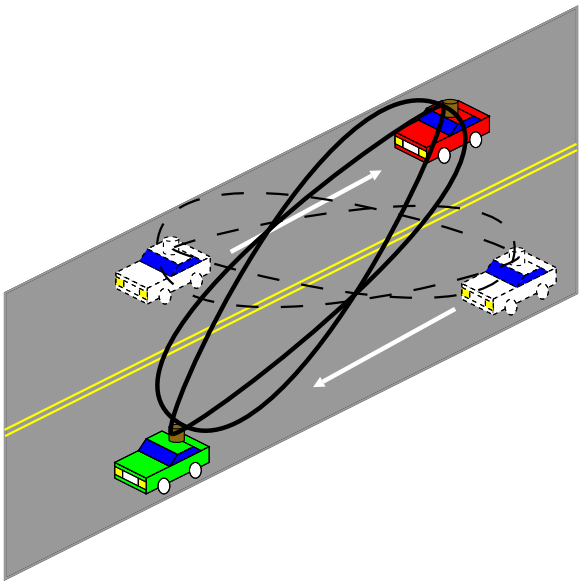


Figure 46: Beam steering to improve quality of V2V links between two moving vehicles in opposite direction.

group of vehicles form the network nodes, possibly with connectivity to static infrastructure. Such networks can enable applications of tremendous value, ranging from safety (e.g., collision warning) to infotainment (e.g., traffic information, just-in-time navigation, multimedia download, etc.). VANET research has progressed mainly in two areas (not necessarily independent) – vehicle-to-infrastructure (V2I), connecting moving vehicles to road-side access points (APs) in a one-hop fashion, and vehicle-to-vehicle (V2V), where a network is to be maintained among a set of moving vehicles. Our focus in this chapter is on V2V network.

For a V2V network to succeed, good connectivity must be maintained among peer vehicles. While delay or disconnection tolerant protocols [70, 125] are possible to tackle periods of disconnection, end-to-end delay and possibility of lost messages do increase with long periods of disconnection. In this chapter, we study the use of a physical layer enhancement – directional communication – to improve the connectivity of a V2V network. Our goal in this chapter is twofold – (i) to understand the potential benefit of using steerable beam directional antennas in the context of V2V communication, and (ii) to develop practical beam steering techniques in the same context. We use the *MobiSteer* system presented in chapter 4 as the experimental platform. In the previous chapter, we used *MobiSteer* in connection with V2I communication alone. In this chapter, we augment it for use in V2V communications. This increases the complexity in two dimensions. First, both end points of the link are now mobile. Second, we now use directional antennas on both ends of a link rather than at only one end as considered in chapter 4. We continue to use 802.11b/g as the link layer because of its wide availability, though much of the techniques developed here are not link layer specific and would apply as well to 802.11p based DSRC [5, 19], for example.

A significant research has been done in V2I and V2V communications in recent years – many using 802.11-like link layer [47, 124, 83]. However, much of existing research uses simulations or analytical modeling for evaluation. See, for example, proceedings of the VANET workshops [23]. Evaluations on real testbeds have been relatively few. Even then, much of the experimental work have focused on V2I communications. See, for example, [89, 47, 124, 83]. The

research community has relatively little experience in experimental study on V2V platforms, exceptions being [142, 155, 114, 88]. This is understandable, as experimental study involving moving vehicles is logistically complex, more so in the V2V context as multiple moving vehicles must be involved.

To put certain amount of focus on our work and also to make the logistics of experiments manageable, our work investigates the performance of a *single V2V link in isolation* and not an entire multihop network. The expectation is that if the performance of a single link can be improved, multiple instantiation of the the same underlying technique would improve the performance of all network links. We will revisit extensions of our techniques for a network-wide use in the Section 5.4. Our technique resides in the link layer, right above the MAC, and is completely transparent to routing protocols.

Our work makes the following contributions.

1. This is the first work to systematically study the use of steerable beam directional antenna in the context of V2V communication with 802.11-based link layer.

2. We report extensive measurement study using two mobile nodes (cars) in two different environments. Noting from our previous experiences reported in chapter 4 that the multipath behavior and hence the performance of directional antennas are dependent on the surrounding environments, we have chosen a suburban area and a highway for the experiments. They also provide different driving patterns (speed, turns, etc) that impact the performance of our techniques.

3. Based on this study, we show that the best practical strategy for beam steering is to point the directional beams at the two link end points directly towards each other. We call this the *line-of-sight or LOS* strategy.[1] While this may not be always the optimal in the sense of SNR, we show that the cost of finding the optimal beam directions is significant in practical platforms.

[1]Note that this a slight abuse of terminology as the straightline between the two nodes may not always provide a line-of-sight when there is an obstruction in between.

5.2 Experimental Setup and Scenarios

We use a similar hardware setup as described in section 4.2.1 in this work, though now both the ends of the link (cars) are equipped with a directional antenna. Here we describe the software measurement tools that we developed for our evaluation. We also describe the measurement scenarios.

5.2.1 Sender and Receiver Programs

Our measurement tool primarily consists of a *sender* and a *receiver* program that operate on the two MobiSteer nodes that are mounted on the two moving vehicles. They serve as the end points of the link under evaluation. The *sender* program is a UDP application that transmits unicast packets of a given size at a constant rate to the receiver application. In addition to transmitting packets, it also steers the antenna beam appropriately depending on the operating mode. There are three operating modes – *fixed*, *switching* and *LOS*. In the fixed mode, there is no beam steering. The antenna uses a specific fixed beam out of the 9 possible (one omni and 8 directional beams). In the switching mode, the antenna beam is switched periodically, cycling through all the 9 beams, staying in each beam for a specified *hold* time. This mode will be used in our work for scanning and probing. In the LOS mode, the beam chosen for communication is always the one such that the centerline of the beam is the closest to the straight line joining the two nodes. The LOS beam is computed based on the last known GPS coordinates of the sender and receiver node. A more detailed discussion on computing the LOS beam will be given in Section 5.4.

Each packet sent is annotated with (i) a sequence number, (ii) the sender nodes's current GPS coordinates, and (iii) the index of the antenna beam used to transmit the packet. The sender program communicates with `gpsd` running on the laptop every 200 ms for an update of the GPS coordinates. We will say more about accurate beam index annotation in Section 5.2.3.

The *receiver* program is a modified version of Kismet [11], a popular wireless packet sniffing software. The `madwifi` driver allows creation of additional raw *virtual* interfaces (e.g., `ath0raw`) for a physical wireless interface. The

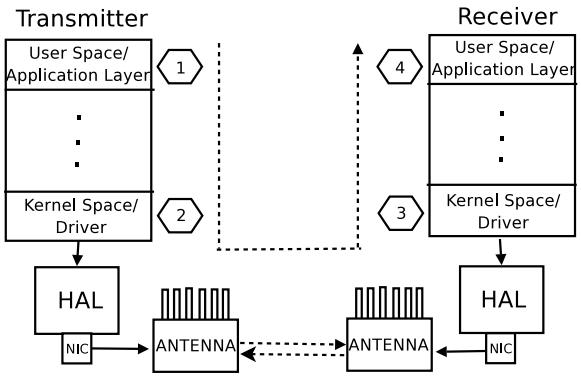


Figure 47: Transmission and reception protocol stack architecture.

virtual interface allows reception of all 802.11 frames (control, management, data) as if in the monitor mode, while the main interface can still operate in the ad hoc or infrastructure mode. The raw device type of the virtual interface is set to type 1, so that the driver appends the *prism monitoring header* containing information about RSSI (received signal strength indicator), noise, PHY-layer data rate and channel of each received packet. Similar to the sender program, the modified Kismet on the receive side also operates in three modes – *fixed*, *switching*, and *LOS*. For every received packet, Kismet annotates it with the current time (time in the prism monitoring header provided by the driver) and the receiver node's GPS coordinates. The GPS coordinates are obtained in the same fashion as the sending node. It also annotates the packet with the receive beam index. For each received packet all relevant information is logged for later analysis. It includes sequence number, sender beam index, sender GPS coordinates, receiver beam index, receiver GPS coordinates, receive timestamp, RSSI, noise, PHY-layer data rate and channel.

We have addressed a couple of technical issues for efficient and correct operation of the above methods. We describe them in the following.

5.2.2 Beam Switching Latency

Beam switching latency is an important issue for continuously switching beams. As mentioned before, beam switching is achieved via a software command sent over the serial line from the SBC to the directional antenna. The antenna vendor, Fidelity Comtech [1], supplies a patched `madwifi` driver that implements an interface for user level programs to control the antenna beam patterns through the `/proc` virtual file system. On receiving a command to the antenna interface, the driver sends a control signal to the antenna controller over the serial interface to choose the appropriate beam pattern and initiates a busy loop for about 150 μs in order to wait for the beams to stabilize. This is the nominal beam switching latency. We found that using the `/proc` interface method added a significant delay due to user space to kernel space communication (75 ms). In order to reduce this delay, we implemented a Linux `ioctl()` interface for controlling the antenna that incurred only an additional delay of 100 μs. Including this overhead, the total beam switching latency becomes about 250 μs.

In the switching mode experiments, we ensure that the hold time in each beam combination is much larger than this time, so that beam switching latency does not play an important role in our evaluations. However, for switching mode, fast scanning (i.e., small hold time) is also important as this mode is used in probing. Thus, probes must be sent and received and thereby allowing RSSI samples to be collected on all beam combinations without the cars moving significant distances so that RSS (received signal strength) can be compared meaningfully for all beam combinations. This issue will be clearer when we discuss our choice of hold time in Section 5.3.

For concerns related to the LOS mode, note that in 250 μs, a car can move only 7 mm at the highway speed of 100 km/hr! This distance is already several order of magnitude smaller than GPS error. Thus, in real applications GPS samples would be taken at a significantly longer intervals relative to the beam switching latency. Thus, the beam switching latency will not play any significant role in the LOS mode, if at all.

(a) (b)

Figure 48: Measurement scenarios: (a) Highway (b) Suburban. The red arrow shows the driving routes.

5.2.3 Beam Index Annotation

When beams are switching fast, we found that the beam index annotation must be carefully done. Doing this annotation in the application program can often lead to errors. This is true even though the actual beam steering commands are issued from the application layer. When the beam index is annotated in the packet at the application layer (marker 1 in Figure 47), the difference between the time at which this annotation is done to the time at which the packet actually reaches the device driver for transmission (marker 2 in Figure 47) can vary significantly and non-deterministically due to varying system load and OS process scheduling. During this time the beam could switch to a different pattern. This problem was also observed in [45]. To avoid this, we modified the `madwifi` driver to annotate the packet with the current beam index right at the point the driver handles the packet to the Hardware Abstraction Layer (HAL). This is the final point we have any control over the

packet, as HAL is implemented in binary.[2] While this is the best we could do, annotation errors could not be completely eliminated without some more work. This is because packet retransmissions are possible and this could take time in order of tens of milliseconds at low data rates for large packet. This time is of the same order of hold times that we consider. Thus, at this time scale beams can still switch. To prevent this problem, we set the retransmission count to be 0 so that lost packets are never retransmitted. Note that we can safely ignore backoff times, as they are in microseconds. Also, our UDP application sends data only at a very moderate rate (one packet every 10 ms) so packet queueing delay within HAL would be very small if any. Packet queuing in the card can lead to a packet to be sent in a different beam than the beam indended for it. We took care to maintain a constant inter-packet interval so that we can get the same number of packet samples on all beam combinations for the switching mode experiments.

Similarly on the receiver side, the time at which the packet is received by the kernel (marker 3 in Figure 47) and the time it reaches the receiver program (marker 4 in Figure 47) can be quite different. We again annotate the packet in the driver right after the HAL delivers the packet.

5.2.4 Measurements Scenarios

In the previous chapter in using *MobiSteer*, we observed that in congested urban environments (with buildings and foliage in close proximity), the directionality the antenna is poor and reflections often dominate. Reflections have the ability to make the best directional beam on one end of a link point away from the other end. This makes the determination of the best beam difficult. While in these prior studies, we used only one directional antenna (the other end point used a regular omni-directional antenna), we hypothesized that similar situation would be likely even with two directional antennas. Thus, we carefully selected the measurement scenarios. Our first scenario is a *highway* (2-3 lanes on each direction) that is fairly straight with only few turns along

[2]We indeed experimented with an open source HAL, OpenHAL [16]; but found it to be unstable in our system.

our driving route. The cars here have less obstructed, straightline path between them. The second scenario is a *suburban* scenario – a dense residential neighborhood with narrow streets (1 lane) with houses and trees on both sides fairly close to the street. There are also quite a few turns along our driving route often obstructing the straightline path between the cars. The satellite pictures of the two scenarios and the driving routes are shown in Figure 48 so that the reader can get a clear idea.

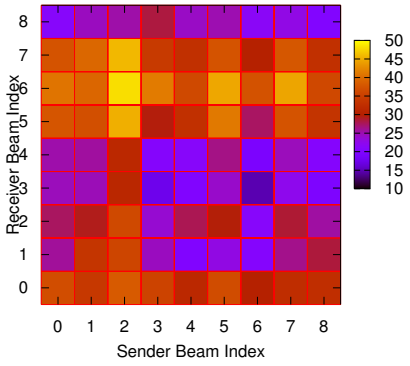
5.3 Experimental Results

One car runs the sender program and the other the receiver program. The *switching* mode experiment is done to collect data in all $9 \times 9 = 81$ beam combinations between the sender and receiver. A fixed transmit power of 18 dBm is used on both the nodes.[3] The PHY-layer data rates of both the nodes is 'fixed' at 1 Mbps so as to have a longer link. It is important to disable auto-rate control, as packet loss on one beam combination, for example, may trigger a rate decrease, and thus improved packet delivery performance on another beam. This will make performance across different beams hard to compare. Evaluation with auto-rate control enabled is reported in a more appropriate context in a Section 5.4. As explained before, the packet send rate is kept constant – 1 packet every 10 ms or 100 packet/s. Each packet has a 512 bytes payload.

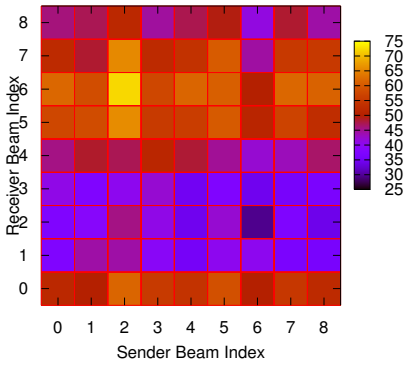
After careful consideration, the hold time at the sender side is fixed at 30 ms and on the receiver side it is set 9 times this value, that is 270 ms. This is to ensure that once the receiver switches to beam i, it has opportunity to receive packets on all 9 beams from the sender before it switches to the next beam. A complete scan of 81 beam combinations thus take 81×30 ms $=$ 2.43 s. Since the packets are sent at 10 ms interval, we need to hold on each beam for a duration longer than this so that in each beam combination, some packets are received to make meaningful comparisons. The cars can travel about 65 meters within 2.43 s even at 60 miles/hour (The fastest speed we

[3]The equivalent isotropically radiated power (EIRP) when using this transmit power is 33 dBm, well below the FCC limit of 45 dBm [127].

used in highway scenario) and about 30 meters at 30 mile/hour (The fastest speed we used in suburban scenario). We found that the environment doesnt change much within these distances in the two scenarios considered.



(a) Highway (180 m distance between cars).

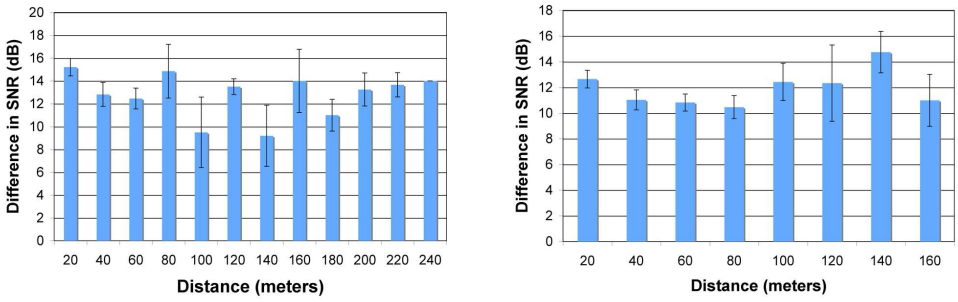


(b) Suburban (25 m distance between cars).

Figure 49: Average SNR (in dB) in each of 9×9 beam combinations for two representative samples.

The experiments are repeated 6 times in each scenario for (i) the two cars

following each other and (ii) the cars driving in opposite directions. The cars drive normally appropriate for the road and traffic conditions. When the cars drive in the same direction, care is taken to produce as much variability as possible in the intervening distance between cars, as we sometimes categorize the performance results in terms of this distance. The car speed varied between 15–30 miles per hour in the suburban scenario and between 35–55 miles per hour in the highway scenario so as to take measurements with a wide range of distances between the two cars.



(a) Highway (180 m distance between cars).

(b) Suburban (25 m distance between cars).

Figure 50: SNR improvement (in dB) for the best beam combination over the omni-omni combination on a scale of link distance.

For each experimental run, information about all the received packets are logged. The log contains all relevant information about the packet as described in Section 5.2.1. The log is postprocessed to group all received packets for one complete scan of both send and receive beams, i.e., 81 beam combinations spanning over a continuous time of $81T_h$, where T_h is the hold time in each beam combination. Each group provides us with one 'sample.' In our analysis, we consider only those samples that have atleast one received packet in all beam combinations so that we can do meaningful comparisons. The 'best beam combination' in a sample is the one that provides the best 'average' SNR (Signal to Noise Ratio) for this sample. SNR is computed from the difference between RSSI and noise values reported in the prism monitoring

header for each received packets. The averaging is done over all received packets within the hold period. Note that during the period both beams stay fixed. The inter-packet time being constant, all beam combinations are given equal opportunity for communication.

We will first present the results for the cars driving in the same direction. The results for the cars driving in opposite direction will be presented subsequently in Section 5.3.4.

5.3.1 Evidence of Directionality

To understand the effect of directionality we present two representative samples from the log in Figure 49. The figure shows the average SNR on all 81 beam combinations in a color-coded fashion. The first is a long link with the cars about 180 m apart and the second one is short link with the cars about 25 m apart. Note that the difference between the SNR between the best and worst beam combination is quite large — about 40 dB and 50 dB in the two samples respectively. Note also SNRs for all beam combinations around the best beam combination (sender 2, receiver 6) are relatively strong as well, going down about 5-10 dB for each beam difference. Note also the worst beam combination is approximately on the opposite direction (sender 6, receiver 2) as expected. The omni-omni combination (0,0) provides only a mediocre SNR, about halfway between the best and the worst. The combination with omni on one side and directional on the other side follow a predictable pattern. The line for beam index 0 on sender peaks at index 6 on receiver, and the line for beam index 0 on receiver peaks at index 2 on receive.

While qualitatively both scenarios produce similar results, there is a significant quantitative difference. The highway scenario gives similar SNR values even at much longer distance. This is due to the unobstructed view and openness of the surrounding space. Note that while we show here only two samples here, similar behavior is observed throughout the drives.

5.3.2 Best Beam Combination Vs. Omni-Omni

Now we study how much improvement one can expect when using the best beam combination vs the omni-omni combination. The latter serves as the baseline as it represents the performance of using regular antennas without beamforming or beam steering abilities. Figure 50 shows the average SNR improvement (in dB) when using the best beam combination compared to the omni-omni case for each sample categorized by link distance (in 20 m divisions). We plot the improvement of the average SNR along with 95% confidence intervals for each distance category.

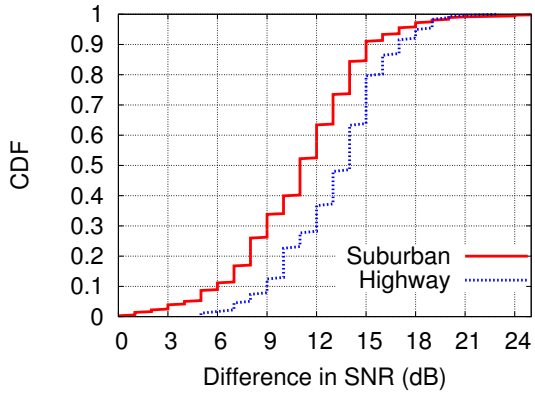


Figure 51: CDF of SNR improvement (in dB) for the best beam combination compared to the omni-omni combination (cars driving in the same direction).

Note that over a wide range of link distances the best beam combination provides excellent improvement when averaged, often between 10-14 dB. No specific behavioral difference is observed along the distance scale. However, there is some amount of scenario specific difference. See Figure 51 for the CDF of the SNR improvements in the two scenarios for all samples. Note that the median SNR improvements are about 11 dB and 14 dB in the suburban and highway scenarios, respectively. Also, note that there is at least 6 to 10 dB SNR improvement 90% of the times. Slight better performance in the highway scenario (about 2-3 dB) is due to less scattering and longer segments of straight roads.

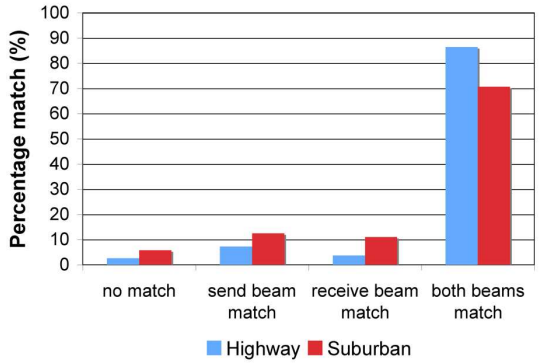


Figure 52: How often do LOS beams match with the best beams? (Cars driving in the same direction.)

Now, we will address a natural question: how often the best beam combination corresponds to the beams along LOS. We define the LOS beam as the beam whose main lobe is closest to the straight line direction to the other node in terms of angles. A detailed discussion on computing the LOS beams is described in Section 5.4.1. To determine the relation between LOS beams and the best beam combintion, we look at the entire set of samples and present the fraction of times they match in Figure 52. We see a significant fraction of times (86% and 71% in highway and suburban scenarios, respectively) the LOS beams match with the best beams. Again, the higher fraction of matches in highway is due to less multipath and scattering. Also, quite often (about 96% of times in either scenario) there is a match of least one beam. *This means that very often choosing the LOS beams actually provides the best beams.* With the availability of the GPS coordinates LOS beams are easier to determine relative to the best beams. The latter requires certain amount of probing and coordination among communicating nodes.

To see the dB difference between the two choices (best beams vs LOS), look at Figure 53. This figure shows the SNR differential between the best beams and LOS beams. The 90-percentile difference is less than 2 dB in both scenarios. Thus, very little improvement is expected by using the actual

best beam relative to using the LOS beams. This means even when the LOS beams are different from the best beam combination, the SNR differences are not significant.

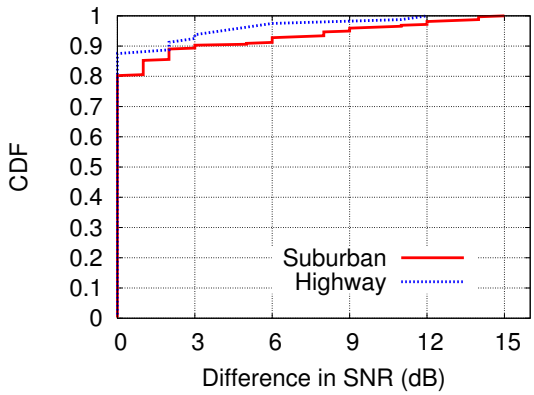


Figure 53: SNR differential (in dB) between the best beam combination and LOS beams (cars driving in the same direction.

5.3.3 Best Beam Combination Vs. LOS

Now, we will address a natural question: how often the best beam combination corresponds to the beams along LOS. We define the LOS beam as the beam whose main lobe is closest to the straight line direction to the other node in terms of angles. A detailed discussion on computing the LOS beams is described in Section 5.4.1. To determine the relation between LOS beams and the best beam combintion, we look at the entire set of samples and present the fraction of times they match in Figure 52. We see a significant fraction of times (86% and 71% in highway and suburban scenarios, respectively) the LOS beams match with the best beams. Again, the higher fraction of matches in highway is due to less multipath and scattering. Also, quite often (about 96% of times in either scenario) there is a match of least one beam. *This means that very often choosing the LOS beams actually provides the best beams.* With the availability of the GPS coordinates LOS beams are easier to determine

relative to the best beams. The latter requires certain amount of probing and coordination among communicating nodes.

To see the dB difference between the two choices (best beams vs LOS), look at Figure 53. This figure shows the SNR differential between the best beams and LOS beams. The 90-percentile difference is less than 2 dB in both scenarios. Thus, very little improvement is expected by using the actual best beam relative to using the LOS beams. This means even when the LOS beams are different from the best beam combination, the SNR differences are not significant.

5.3.4 Cars Driving In Opposite Direction

Now, we repeat similar experiments as presented so far except that now the cars drive in opposite directions. Again the experiments are repeated 6 times in each of the scenarios as before. These results give us insights as to whether the direction of travel influences our conclusions. Also, it is easier to study the range improvements when cars are driven in opposite directions. They come in range, stay in range for some time and then go out of range.

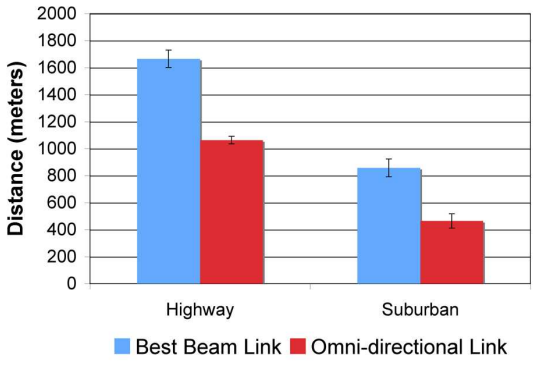


Figure 54: Average communication ranges when using the best beam combination and the omni-omni combination (cars driving in opposite direction).

The range improvements are shown in Figure 54 for the best beam combination versus omni-omni combination. Note approximately 50% and 80%

range improvements from use of directional antennas in highway and suburban scenarios respectively. Also, note in accordance with our previous observation with SNR, the communication range is almost twice in highways compared to suburban scenarios, with an average range of about 1.6 km.

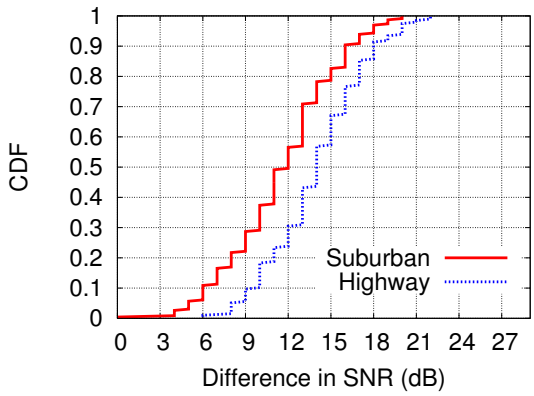
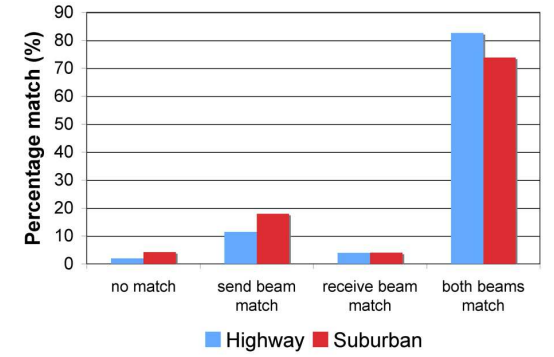
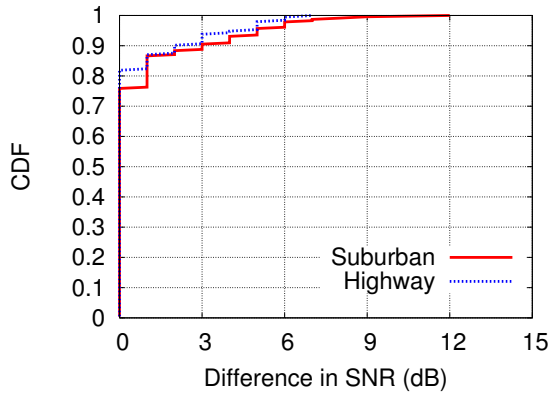


Figure 55: CDF of SNR improvement (in dB) for the best beam combination compared to the omni-omni combination (cars driving in opposite direction).

We next present similar data sets as in the previous experiments for studying (i) SNR improvements for the best beam combination over omni-omni combination and (ii) relationship between the best beams and the LOS beams. The results are presented in the same fashion in Figures 55 and 56(a) and (b). We note that a high degree of similarity (both qualitative and quantitative) with the previous results (i.e., Figures 51, 52 and 53). *Thus, we conclude that while surrounding environment plays a key role (e.g., highway vs. suburban), relative direction of travel does not play any significant role in use of directional antennas for V2V communications.*



(a)



(b)

Figure 56: (a) Frequency of match between the the best beam combination and LOS beams; (b) SNR differential (in dB) between the best beam combination and LOS beams (cars driving in opposite direction).

5.4 Practical Beam Steering

So far, we have studied the performance of V2V links when using directional beams on both ends. While the performance improvement over using omni-directional beams is impressive, the performance potential has been demonstrated by scanning and probing on all beam combinations. Online scanning and probing on all beam combinations to select the best is evidently a high overhead operation. As discussed before in Section 5.2, the packet transmit time in the lowest bit rate (1 Mbps) of 802.11b is in the order of several ms. Thus, even if the hold time is in order of a single packet transmit time, it

will take an order of 100ms for probing on all 81 beam combinations. When cars drive in opposite direction, the best beam directions could change rapidly generating considerable overhead. Also, in a VANET setting a node may have more than one links. Each of them must be evaluated separately. This increases the overheads further. However, we have seen that the performance of the LOS beams very closely match the performance of best beam combinations in both scenarios and various driving patterns and speeds. LOS beams are relatively straightforward to compute. They only need the GPS coordinates of the both end points. Thus, each node just need to know only the coordinates of its neighbors. This is likely not an extra overhead, as some neighbor discovery protocol (e.g., hello messages [2]) must be present in the routing protocol. The GPS coordinates could simply be piggybacked onto this protocol messages. Also, neighbor location updates does not need to be very frequent. For example, predictions could be used based on speed history and driving route (from the navigation system, e.g.). Since the predictive model is known to all nodes involved, location updates can be sent only when the prediction deviates sufficiently.[4]

Here, we do not get into the discussion of what nodes should be neighbors. This generates a subtle issue when directional communication is used. This is because many neighbor discovery protocols use broadcast messages. A question arises as to which beam is to be used for broadcast. While omni-directional beam would be a natural choice, reduced range of the omni-directional beam presents a problem. There could be many solutions for this. For example, (i) such broadcasts could be sent with a higher transmit power on the omni beam; (ii) multiple broadcasts could be made on directional beams on all directions; or (iii) reduced range is deemed acceptable, and only nodes reachable via the omni beam at regular power are deemed as neighbors (here, the extra reach of the directional beams cannot be utilized, but higher SNR can certainly be

[4]This can work as follows. Consider, two end points of a link U and V. U receives a location update from V, and from this point, predicts V's location using a given model. The model can use a range of information about V, e.g., speed history and route. Since the model is also known to V, V uses the same model to determine whether U's prediction is sufficiently accurate. This can be done without any communication. When the deviation becomes significant, another location update is sent. See [100] for further details of this technique in a general setting.

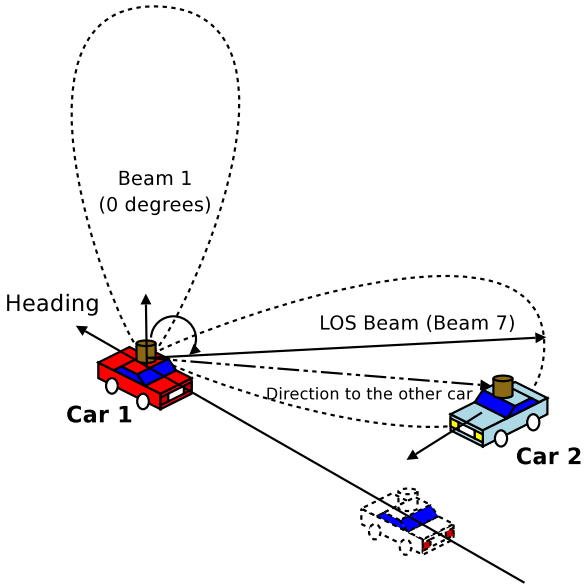


Figure 57: LOS beam computation. Beam indices increase in the anti-clockwise direction.

used for better link quality and thus higher PHY-layer data rate). An appropriate choice here should take into account upper layer aspects, including the routing protocol and the needs of the application.

5.4.1 Computation of LOS Beam

For our experimental work here we use choice (iii) above. We use periodic broadcast beacons (at 1 sec intervals) on the omni beam to disseminate GPS coordinates. The same transmit power is used. No prediction is used. Now, there are two technical issues to be address: (a) determining the LOS beam indices on either sides, and (b) consideration of the GPS error.

Since the antenna is fixed on the car, the orientation of the car needs to be determined to determine the LOS beam indices. We compute the car's orientation by computing its 'heading'. Heading can be computed based on past GPS locations. Note that we assume that GPS samples are taken every 200 ms intervals as we have done in the previous set of experiments. This

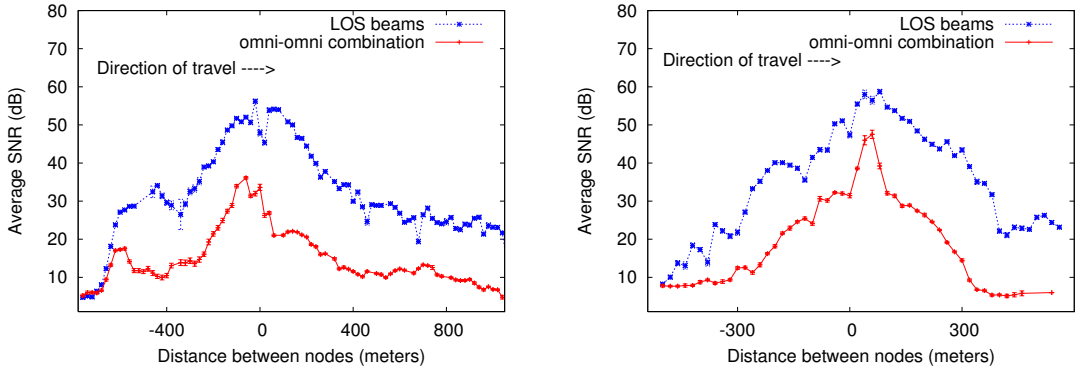


Figure 58: SNR improvement (in dB) when using LOS beams vs omni-omni combination in (a) Highway (b) Suburban scenario.

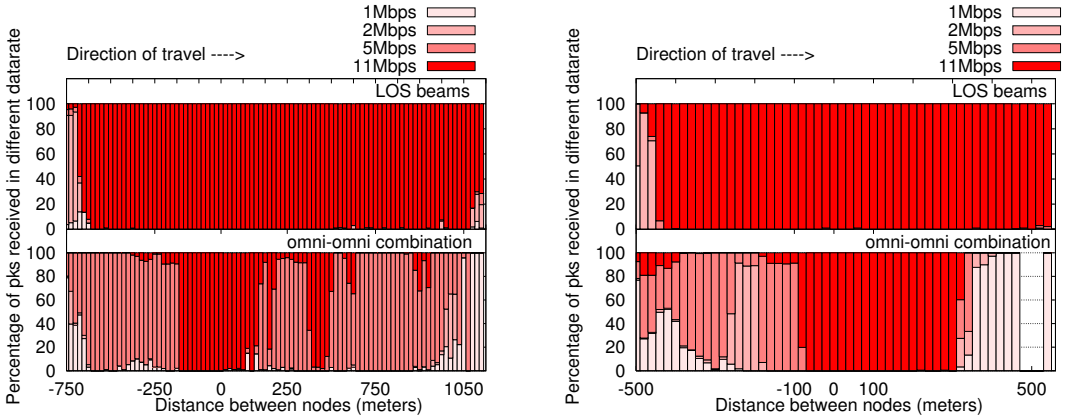


Figure 59: Physical layer data rate improvement when using LOS beams vs omni-omni combination in (a) Highway (b) Suburban scenario.

sampling interval is deemed sufficient, as within this time the car can move a distance roughly equivalent to GPS error ranges even at highway speeds. The direction of the other car is determined using the GPS coordinates of both cars. The beam with its center line closest to this angle is chosen to be the LOS beam. See Figure 57. Since we are considering only a single link in our experiments, LOS beams are computed whenever the car(s) move sufficiently and the antenna is always kept steered to this beam.

To determine when a new LOS beam computation should be done, we use estimates of GPS error. The goal is not to compute the beam too frequently that can be influenced heavily by GPS errors. We evaluated the errors in our GPS unit and the errors were found to be environment specific. Highest errors were observed in congested suburban scenarios. The median error was found to be 5.5 m and 90-percentile error 7 m. Maximum error was always less than 8 m. Considering these values and several trials, we determined a 25 m linear difference between two GPS readings of the same car is sufficient to trigger a new beam computation. This threshold is used to compute a heading or a new LOS direction.

5.4.2 Experimental Results

Our interest here is to do a realistic throughput experiment where the LOS beam is computed online. Note that we have computed the LOS beams only in the post processing step in the previous experiments using the logged GPS coordinates. Since all beams are no longer scanned as before, comparison between different beam choices (e.g., omni-omni and LOS beams) now presents a problem. This is because multiple drives are now needed with different beam choices for comparison. Since the position of two moving cars can vary a lot during different drives, it is hard to compare across experiments. To address this, we keep one car fixed and drive the other car such that the cars come in range, remain in range for some time and then go out of range. This does require fairly rapid change in the LOS direction when the cars cross each other. Thus, the LOS computation technique is exercised quite well. Also, now different experiments use the same car location and drive paths, making

them directly comparable. The default auto-rate control algorithm [42] in the `madwifi` driver was enabled instead of using a fixed rate as we are not scanning in all beams. This is to emulate a realistic operational environment, and also to study improvement in the PHY-layer data rate using this technique. Otherwise, the software setup for these experiments is similar to that used in the previous section. Experiments are done again using the same two scenarios.

Two different experiments are performed - (i) the sender program and Kismet operate in the *fixed* mode with the beam index set to 0; (ii) the sender program and Kismet operate in the *LOS* mode where they use the LOS beams computed using the GPS coordinates of the two nodes as described before. In both the experiments the sender node is static and the receiver node moves at a constant speed. It comes in range, crosses the sender node and then goes out of range of the sender node.[5] The speed of the car for both the experiments was kept constant (50 mph in highway scenario and 30 mph in the suburban scenario) as much as possible though our analysis is not affected by little variation in speed. The sender node transmitted 512 byte packets at a rate of 100 pkts/second. MAC level retransmissions were turned on since we are not switching between beams frequently. Each experiment was repeated 4 time in both scenarios.

Figure 58 shows the average SNR for each 20 meter segment along the path travelled by the mobile car. The x-axis shows the distance between the two nodes as the mobile node (receiver) comes in range, crosses the sender and goes out of range. Packets recieved in the 4 runs of each experiment were grouped based on the distance between the sender and receiver node and we plot the average SNR for each 20 meter segment. In the highway scenario, note a peak SNR of 56 dB and 36 dB in LOS beams and omni-omni combination respectively with a median SNR improvement of 18 dB. Note a peak SNR of 58 dB and 47 dB in LOS beams and omni-omni combination respectively in the suburban scenario with a median SNR improvement of 16 dB.

In Figure 59, we show the performance of LOS beams and omni-omni combination in terms of the PHY-layer data rate improvement. For each 20 meter segment, we plot the percentage of received packets in different data

[5]The results were similar when the roles of the sender and receiver are changed.

rates. The darker color in the plot denotes higher data rates. The PHY-layer data rate of the link is chosen by the auto-rate control algorithm based on the quality of the link. As seen from Figure 58, better SNR is observed even at a longer distance between the communicating nodes using the LOS beams and this is reflected on the data rate plot. While using the omni-omni combination, the auto-rate control algorithm switches the data rate to the highest value (11Mbps in 802.11b) when the distance between the communicating nodes is about 200 m or less. However, when using the LOS beams, the highest data rate is used as far as 800 m and 450 m distance between the sender and receiver nodes in the highway and suburban scenarios respectively. At farther distance, lower data rates are used due to poor link quality. As V2V links are short-lived due to high mobility of nodes, it is important to have a good quality link so as to transfer as much data as possible within the limited time.

5.5 Related Work

There is currently a significant interest in VANET research. The focus has been on developing accurate radio propagation models [64], mobility [53] and simulation models for VANETs [51], applications such as safety [43, 137], location aware services [65], traffic management services [95] and on developing routing [66, 61, 67] and MAC protocols [159] to deal with high mobility and poor link quality. Much of the current research, however, has focused on analytical or simulation modeling for evaluation. Experimental studies are less common.

Among the experimental studies reported in literature, work on V2I communication is more mature. Several works have demonstrated the feasibility of IEEE 802.11 based communication from moving vehicles. In [124] the authors show how long a connection between a moving car and a roadside AP can be maintained while driving at different speeds between 80–180 km/hour. In [74] perform a similar study and evaluate the performance of TCP and UDP transfers between the vehicle and the AP. More recently, Hadaller et. al [83] again perform similar experiments and demonstrate that current MAC and transport protocols can achieve only 50% of overall throughput possible and

identify ten different problems in existing protocols that are responsible for the throughput loss. Our MobiSteer work presented in chapter 4 also target communication between moving vehicle and roadside APs. It proposes practical beam steering and AP selection algorithms. In the CarTel project [47], experiments are performed to connect to unprotected access points in metro areas in normal driving environments and use them for Internet connectivity. The duration and quality of connectivity are evaluated. In [108], using measurements from a moderate sized vehicular testbed, the authors characterize the nature of the connectivity between moving vehicles and roadside APs. In [93] antenna placement and use of diversity for V2I communication are studied.

Fewer experimental studies have been reported in the V2V domain. In [142], the authors measure the performance of 802.11b-based V2V communication in different scenarios like freeways, urban and suburban environments. They show that the performance of the V2V links is greatly affected by the nature of the environment. Our experience in this regard has been similar. In [155], the authors demonstrate single hop and multihop V2V link with a single forwarder node. They present preliminary results on throughput vs. distance of V2V communication in a highway environment. Authors in [114] report experiences with static 1 and 3-hop scenarios and a mobile 3-hop scenario. TCP and UDP performance results are presented in a 2-hop vehicular network in a highway and suburban scenarios in in [88]. In the DieselNet project (see, e.g., [163, 46]), a bus-based disruption-tolerant networking (DTN) testbed using WiFi equipped nodes are used for many interesting studies – related to routing, mobility modeling, security and upper-layer protocol design. In comparison to these papers, our work adds a new dimension of using steerable beam directional antennas to improve the quality of V2V links.

Measurement studies using steerable beam directional antennas for 802.11-based networks are also quite limited. Authors in [45] present a measurement tool using the same antenna used in our work and report results from static experiments in two environments. In [49], the authors study how to enhance security using directional communication and in [119] they present how directional communication can gain better spatial reuse. A model of the same antenna is used in these two papers, but no real experimental results are

reported. In [54], the authors report measurement results using beamforming antennas to improve network capacity in the context of static mesh networks and wireless backbones.

In the general context of ad hoc and mesh networks, there is a body of literature primarily focusing on use of directional antennas to improve spatial reuse and to provide immunity from interference. See, for example [97, 57, 132, 55]. These papers mainly considers modifications and extensions of the 802.11 protocol for use with directional antennas. They, however, do not directly consider mobility. All of these works use simulation studies for evaluation.

5.6 Summary

In this chapter, we investigated the use of directional antenna beam steering to improve performance of 802.11 links in the context V2V communication. We considered a single V2V link, and used extensive experiments in two different multipath environments to demonstrate the performance advantage over using regular omni-directional antenna. We showed that directional antenna improves the link SNR significantly, which also translates to at least 11 dB SNR improvements (median) and at least 50% range improvement. The improvements are scenario specific. As expected, in highways with lower possibility of multipath reflections the improvements are considerably higher – relative to congested suburban environments.

We also demonstrated that determining the best beams for communication can be simplified by a simple heuristic, where the beam pointing directly to the other node (LOS beam) is used. While in theory the LOS beams are not the best always, our results show that even in the suburban environment they are indeed the best significant fraction of times. Thus, instead of using an expensive scanning and probing method, we have developed a simple protocol, by exchanging the GPS coordinates, to determine the LOS beams in a continuous fashion. Our experience of using this protocol directly shows that the expected gain from directional communications is achievable in a practical setting. We showed significant improvement in SNR and PHY-layer data rate, when auto-rate control is used with higher data rates used at 2 to

4 times farther distances between the sender and receiver node compared to omnidirectional communication.

While we have used a single V2V link, the technique used in this work, can be applied easily in a more general ad hoc network environment. Use of LOS beams only needs dissemination of GPS coordinates of the nodes in the neighborhood. Such dissemination can be piggybacked in routing control messages, for example. Also, as we have discussed, predictive approaches can be used using historical information and knowledge of driving route to predict GPS coordinates within a given accuracy. Note that our work has focused solely on improving link quality, when the link is noise-limited rather than interference-limited. In a high load situation, when the link is interference-limited, the same steering technique should still be useful. However, appropriate MAC protocols [97, 57, 132, 55] must also be used for the best throughput performance.

Chapter 6

Localizing Roadside WiFi Networks

6.1 Introduction

Localization of the nodes in a WiFi (802.11) network using radio-based information – such as signal strength, angle of arrival (AoA) etc. – is an important problem. The simple reason for this is that devices with WiFi interfaces are ubiquitous and localization of wireless/mobile devices enables interesting applications. GPS receivers are not available with most wireless clients making radio-based localization the only viable option. Needless to mention that GPS may not always work in indoors and in urban canyons.

In the most common approach for radio-based localization for WiFi, location-tagged RF fingerprinting is used for signals from *infrastructure nodes* (i.e., access points or APs). This location information must be independently determined. RADAR [37] and many followup papers [104, 99, 161] have used this basic method for indoor localization. On the other hand, Intel's Place Lab work [52] used a similar approach for outdoor localization. In a different approach called VORBA [123], rotating directional antennas are used in APs and a combination of signal strength and AoA information is used to localize clients in indoor environments. VORBA does not require RF fingerprints, but needs multiple APs with rotating directional antennas. All these

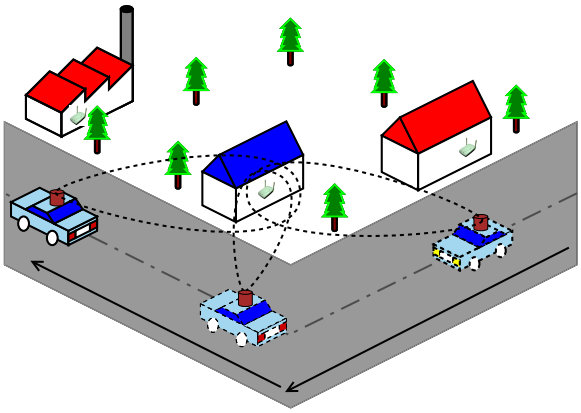


Figure 60: Drive-by localization of roadside APs.

approaches have been primarily used for localizing wireless client nodes, and not the infrastructure.

In this chapter, we consider the opposite problem – localization of *infrastructure nodes (APs)*. Our goal is to localize them in a passive fashion, i.e., without their direct participations in the localization process. There are tremendous application for such localization. WiFi networks are growing in a viral manner. Many urban regions have a high density of WiFi APs – deployed in a "chaotic" fashion [29] in homes and businesses, in campuses and hotspots, or as a part of a metro or municipal WiFi effort [4]. There is little knowledge about the nature of these networks, e.g., density, connectivity, interference properties, etc. The first step in understanding their nature is estimating locations of the APs. We expect that at the minimum our effort will provide researchers significant datasets for simulations and modeling purposes. We hope that this will eventually lead to significant research as in understanding Internet topology [71, 31] in wired networking context. Other than datasets for research use, learning locations of APs may reveal interesting social aspects. We will show later the accuracy of our localization is good enough to localize APs within the boundary of a typical house or even an apartment. This can lead to interesting data sets for social science – correlating census data (say,

level of education or home price) to Internet usage.[1]

While several war-driving databases are in existence [25][26], the location information therein is very primitive. The database simply contains the locations where the APs are heard with a sniffer. As we will show later, even with the most sophisticated techniques this information can only provide very rough location estimates, with errors in hundreds of meters. Our goal here is to be able to improve such location estimates by *an order of magnitude*.

In our approach, we exploit the *MobiSteer* architecture presented in chapeter 4. Here, we use *MobiSteer* to gather frames originating at roadside APs on different directional beams to estimate the Angle of Arrival (AoA) of the frames. For robustness reasons, our strategy requires that many samples of AoA information be collected from different locations. Thus a moving car is indeed needed so that many such samples can be collected with relatively little effort. The general idea is driving the car in the neighborhood where APs need to be localized, and collecting GPS-tagged signal strength information on different directional antenna beams for the frames transmitted (e.g., beacons) by the APs. See Figure 60. The approach is purely passive and based on "sniffing" alone; APs are unaware of the localization effort. Because of the use of a car, our work naturally targets outdoor use. However, the APs can be anywhere – either indoor or outdoor. In fact, in most of our experiments they are indeed indoor.

The rest of the chapter is organized as follows. In Section 6.2 we describe our experimental scenarios and data collection method. In Section 6.3 we describe our localization approach. The performance results are presented in Section 6.4. We follow it up with related work and summary.

[1] We are assuming that having an WiFi AP at home means that residents have broadband connection and use the Internet heavily. Note that we are ignoring privacy aspects. Our technique simply sniffs WiFi frames from streets and public places. This is no different than Google's Street View [9] that takes pictures from streets.

6.2 Experimental Scenarios and Data Collection Method

We use the hardware and software setup of our MobiSteer architecture described in Sections 4.2.1 and 4.2.2 as our experimental platform to collect data for localizing roadside wireless APs. In the following we describe the experimental scenarios and the data collection procedure employed in our work.

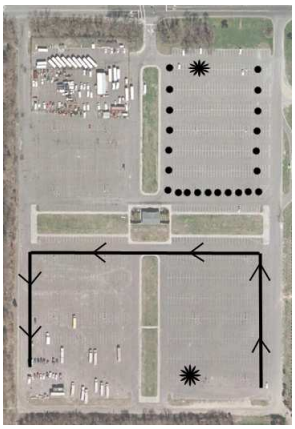
6.2.1 Experimental Scenarios

Figure 61 shows three representative environments in which we did our experiments. It also shows the actual location of the APs and the driving path of the *MobiSteer* node. Figure 61(a) is a large open empty parking lot with no surrounding building. This scenario is used to demonstrate the performance of our localization approach in an uncluttered environment and to create a best case scenario. This is representative of relatively rural or empty environment.

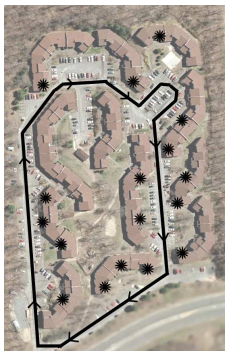
Figure 61(b) and 61(c) are complex environments where there are several two-storied buildings and large trees in close proximity. They are the graduate student apartment complex and the computer science department building in our university, respectively. The APs here are kept indoors as would be normal in such environments. These are relatively challenging scenarios for localization, as there are lots of possibilities of reflections and shadowing. They are representatives of urban homes and offices. The measurements reported here include data collected from APs deployed in 21 different locations in the three scenarios shown in Figure 61.

6.2.2 Data Collection

Ideally, we would like to have measurement samples for each AP on all beams at as many points on the roadway around the AP as possible. Samples on all beams let us estimate the AoA – by comparing the received SNRs on all beams. In free space or relatively uncluttered environment, the beam with the highest SNR would be the one pointing towards the AP. A little variation of



(a) Parking lot (2 APs).



(b) Apartment complex (17 APs).

(c) Office building (2 APs).

Figure 61: Three experimental scenarios along with the AP locations and the driving trajectories. The black stars are the locations of the APs. The black curve is the trajectory followed by the *MobiSteer* Node.

this also works well in cluttered environments with reflections, which we will describe in the next section.

The complexity in data collection comes from the fact that we have a single radio and single antenna system in the *MobiSteer* node. The radio operates on a given channel and the antenna uses a given beam at a time. Thus, all channels and beams need to be scanned. However, for each channel and beam combination the system must hold for certain duration T, where T should be long enough to receive a frame from all APs in range and also

long enough so that channel and beam switching latencies can be ignored. We have fixed T to be 100 ms. This time is the default beacon period in most APs and also much longer than the switching latencies. Since T is not insignificant, it is not possible for a moving car to gather measurement samples on all beams exactly at the same location. Our experience has shown that so long as the measurement samples on all beams are within 'close proximity', the errors introduced are not significant in our technique. We have defined 'close proximity' as 5 m, which is similar to the GPS error bound for the GPS receiver we have used. We will discuss more about GPS errors in Section 6.4.

Still, the car must be driven very slowly. To see this, consider that an entire scan on 16 beams take 1.6 s. Thus, the car should drive maximum 5 m in 1.6 s, i.e., 11.25 km/hr. Such slow driving may not always be practical. We propose to achieve the same effect by driving multiple times on the same route. This approach was also used in chapter 4 to build up an RF signature database, though the context and use of the database were different. One simple approach to reduce the number of runs in the data collection process is to use multiple fixed directional antennas oriented along different directions connected to multiple radios on the moving vehicle. This enables the vehicle to receive frames from APs in all directions throughout the drive.

To summarize, the idea is to drive the car at normal speed appropriate for the roadway used. The drive is simply repeated multiple times such that enough samples are collected. Samples are then clustered such that samples taken within 5 m from one another are assumed to be taken at the same point P. P is assumed to be the centroid of the locations of these samples that are in the cluster. For convenience we will refer to an instance of this point P as the *measurement point*. A larger number of samples naturally provide many such measurement points along the roadway and provides better accuracy for localizing roadside APs. Average SNR is used in the computation when there are multiple samples clustered on the same measurement point for the same beam. Here also, larger number of samples provide better immunity from outliers due to fading. Later in Section 6.4, we will provide an analysis of sensitivity of our method to the number of samples as number of samples is directly related to the measurement effort.

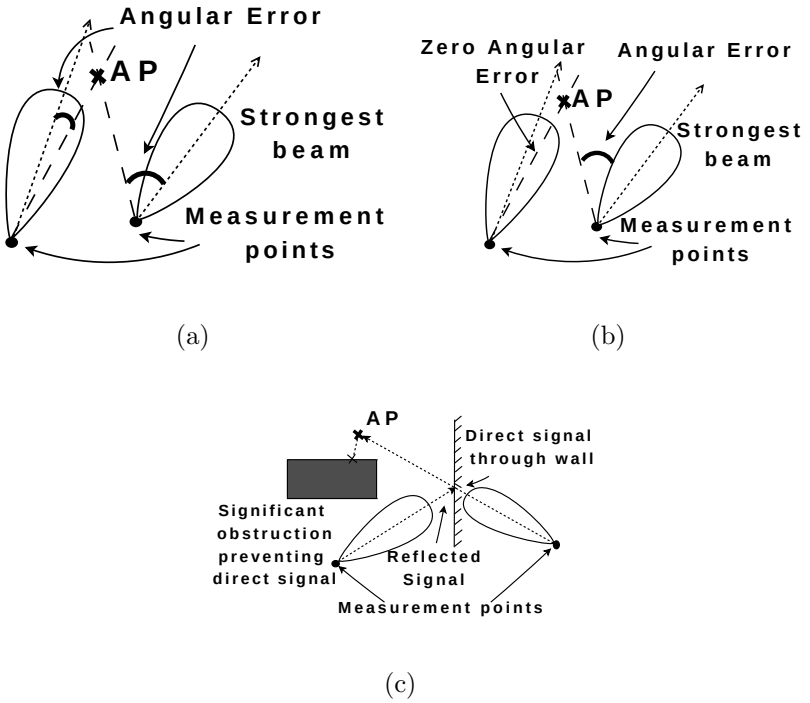


Figure 62: (a) Angular error using center of directional beam. (b) Angular error considering beam width. (c) A scenario demonstrating reflection.

To simplify the data collection process in the experimental results reported here, we have used a cart pushed at slow walking speed in some instances (scenarios (a) and (c) in Figure 61), instead of using a real vehicle and multiple drives. This enabled us to collect sufficient number of samples per AP on a single "walk". This also enabled us to utilize walking paths and open areas in the university campus where driving is not allowed. For the experiments in scenario (b) in Figure 61, we drove at a very slow speed (approx 10 mph) and repeated the runs multiple times. About 40-60 measurement points are used to localize each AP in the experiments reported here. We used our own APs for the experiments and made them broadcast UDP packets at 250 packets/sec. This let us "speed up" the experiments as we could get many samples on the same beam for the same measurement point and used the average SNR for each $<beam, AP>$ combination. We used the same channel for all the APs.

This speeds up the data collection process further as multiple channels need not be scanned. Note that use of carts or UDP broadcast packets from AP are only used to reduce measurement effort and does not have any fundamental impact on the results.

6.2.3 Determining Orientation

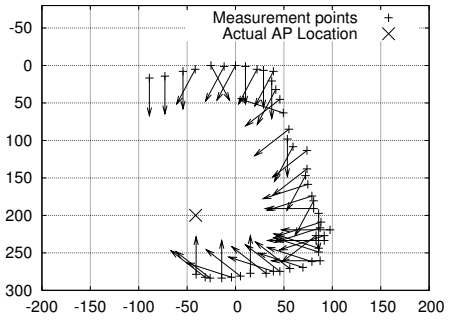
Since the antenna is mounted on the car in a fixed fashion, the orientation of the car (with respect to some absolute direction, say magnetic North) also provides an orientation of the antenna setup. This ensures that the absolute direction of the directional beams can be determined by knowing the car's orientation. Orientation of the car can be determined from the *heading* computed from the GPS locations – a method commonly used in navigation systems. This, however, may not provide enough accuracy for quick turns in small spaces. For better accuracy, a digital compass such as [10] could be used.

In the experiments, we indeed used a compass – not digital, but a regular compass with a magnetic needle. During the walks for data collection, we manually ensured that the antenna setup is always oriented in the same direction. In the car experiments, GPS headings were used to compute the orientation. We expect that use of digital compass will simplify the data collection process much and will likely provide better accuracy.

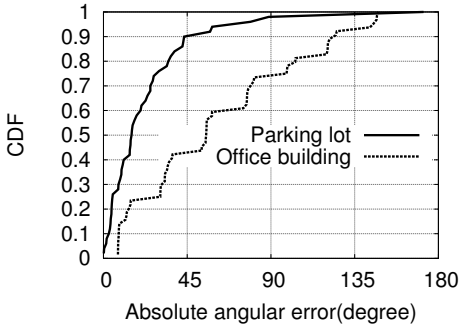
6.3 Localization Algorithm

6.3.1 Preliminaries

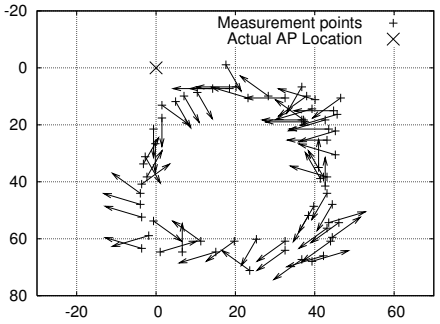
Our approach hinges on estimating the AoA of frames from a given AP at each measurement point. The AoA is estimated by noting the average SNR for the frames from a given AP on each directional beam for the same measurement point. The directional beam providing the strongest average SNR (we will call this the *strongest beam*) is expected to point directly to the AP discounting reflections. Thus, the AoA can be estimated by determining the strongest beam and then using the orientation information (Section 6.2.3) to determine the absolute direction of the strongest beam. (Unless mentioned otherwise,



(a)



(b)



(c)

Figure 63: (a), (b): AP location and directions of the strongest beams in two scenarios demonstrating the impact of reflections. (c): CDF of the angular error of the strongest beam from the actual direction towards the AP in two scenarios.

the beam direction corresponds to the center of the beam.) Let $\alpha(i)$ denote the absolute direction of the strongest beam at each measurement point (x_i, y_i) along the drive.

Let us first assume that signal reflections are not present. In this case, the AP can be localized at a point in the 2D plane for which the *sum-square of the angular error* from all the strongest beam directions is minimized similar to the approach used in [123]. See Figure 62(a). Two artifacts, however, complicate this scenario. They are described below.

- *Non-zero beam width:* The beams have non-zero width (about 45° between the half-power points on the main lobe). Using the center of the beam for AoA calculations may incur significant error; the non-zero beam width must be accounted for. One way to account for this would be to use angular error from the beam sides, i.e., the 'half-power' directions, and pick the minimum of these two errors. In case the direction to the localized point is contained within the beam width, then the error is assumed zero. See Figure 62(b).

- *Reflections:* In most realistic scenarios, radio obstructions and reflections would be present causing the strongest beam point away from the AP. See Figure 62(c). This phenomenon was quite evident in the measurements we did in chapter 4 in cluttered environments (same as Figure 61(b)).

6.3.2 Understanding and Modeling Reflections

To understand the impact of reflections, we use Figures 63(a) and (b) to show the measurement points ('+' symbols) and the direction of the strongest beam from each measurement point (arrows). For simplicity we are ignoring the beam width issue for now and using the center of the beam for direction. The actual location of the access point is also shown ('×' symbols). These figures correspond to measurements for the empty parking lot (Figure 61(a)) and office building (Figure 61(c)), as described in Section 6.2.1. To complement these figures, we also show the CDF of angular error of the strongest beam from the actual direction towards the AP from each measurement point (Figures 63 (c)).

Note that in the parking lot scenario most arrows are pointing roughly towards the AP as expected. However, the behavior is quite different in the office building scenario. Most of the arrows are pointing in a different direction, presumably due to reflections. Figure 63 (c) qualitatively demonstrates this. Note that for the parking lot scenario, the error CDF is rising sharply, and for the office building scenario, the rise is quite gradual denoting significant errors. The median angular error for the parking lot is about 15° (small) and for the office building is about 55° (unacceptably large). Note also that the

90-percentile error for parking lot is 43° — similar to the beam width of the antenna, while the 90-percentile error for office building is again very large (125°).

This study indicates that straight-forward approaches to minimize the sum-square of angular errors [123] using the estimated AoA values can give rise to significant errors in cluttered environments where the strongest signal reaches the *MobiSteer* node after one or more reflections. Since this situation will be the common case for roadside APs, modeling reflections is important.

An interesting observation in Figure 63(b) is that while arrows are all pointing to different directions, there seems to be a clustering effect. The arrows are not pointing to random directions but towards one of a handful of possible directions. This is as if each arrow is pointing towards either the real AP or one of its images arising out of presence of reflections. The complexity of modeling reflections now is that there is no knowledge of the number and locations of reflective surfaces that give rise to these images. Thus, there is no knowledge of number of images to look for, and the real AP is indistinguishable from any of its image.

We will approach this problem in the following fashion.

1. Use the well-known *k-means algorithm* [107] to group the measurement points into k clusters such that each group of measurement points have the strongest beam pointing (approximately) towards the same location. These k locations include the real location and the images of the AP. Since the number of reflective surfaces are not known *a priori*, we use the Anderson-Darling normality test [34] to learn the value of k while clustering (more on this in section 6.3.5).

2. Determine which one of these k images is the real AP. We show that it is impossible to determine in a general case, but heuristics can be used quite successfully.

We describe the details in the following subsection.

6.3.3 Modeling Reflections by k-Means Clustering

Given the set of tuples $< x_i, y_i, \alpha(i) >$ for each measurement point along the drive, the main idea is to cluster the measurement points into k clusters and find k locations which minimize the *aggregate of intra-cluster sum-square of angular errors*. The angular error is the error considering the beam width and is determined as described in Section 6.3.1

In addition, we use weights when minimizing the sum-square of angular errors. The intuition for this is as follows. Since the strongest beam is the one that is important for estimating AoA, we should make distinctions between measurement points with high SNR on the strongest beam compared to those with low SNR on the strongest beam. We use the average SNR (in dB) of packets received in the strongest beam as the weighting function. This intuitively puts emphasis on measurement points close to the actual AP location or on those received on a direct beam.

More formally, let w_i be the weights assigned to each measurement point. Let $\alpha_l(i)$ and $\alpha_r(i)$ denote the left and right half-power directions of a beam with absolute angle $\alpha(i)$. We seek to find a k-clustering of the measurement points into k clusters $(S_1, ..., S_k)$ and obtain k locations $L_i = (X_i, Y_i), 1 \leq i \leq k$, that minimize the following objective function:

$$\sum_{i=1}^{k} \sum_{j \in S_i} \begin{array}{l} w_j \cdot \min\{[\alpha_l(j) - \arctan(Y_i - y_j, X_i - x_j)]^2, \\ [\alpha_r(j) - \arctan(Y_i - y_j, X_i - x_j)]^2\} \end{array}$$

Note that the quantity within $\{.\}$ is the angular error using the non-zero beam width idea. Since the angular error can be between 0 and π, if the quantity within $\{.\}$ is larger than π, it is subtracted from 2π.

The k-means clustering algorithm works in the following way. For any given value of k, assume $L_1, \ldots, L_k$ are the k locations of the AP (i.e., real and the images) to be determined. Initially, L_i's are chosen randomly within the 'feasible region'.[2] Each measurement point is mapped to some L_i that provides the minimum angular error for this measurement point. Thus, the

[2]In our experiments, we defined the feasible region as a square of side 600m around the measurement points. We assume the transmission range of the AP is not more than 300m and thus the possible location of the AP should be within a region of radius 300m from the measurement points.

measurement points are now clustered into k clusters. The algorithm then repeats the following two steps until convergence.

- Compute a point for each cluster, denoted by C_i, in the feasible region that minimizes the weighted intra-cluster sum-square of the angular errors within each cluster.
- C_i's now become new L_i's. Re-cluster by mapping each measurement point to the L_i that provides the minimum angular error as before.

Convergence is obtained when the clustering does not change. Now we have k locations, $L_1, \ldots, L_k$, which represent the images of the AP including the real location. The next step is to choose one of these k points as the estimated location of the AP.

6.3.4 Choosing Real AP Location from k Images

The k-means clustering gives a set of possible locations of the AP, with one of them being the true location and the rest of them reflected images. However, it is hard to distinguish the true location from the image. A simple example is shown as in Figure 64. In the left figure, two measurement points A, B receive signal from the AP. The signal from AP to B is direct but the signal to A was reflected once. Thus at location A the strongest beam to the image of AP. Notice that in this figure everything is symmetric, thus we can swap the AP and its image and have another feasible configuration (right). In other words, given a set of k possible locations including the AP and its images, and the way the strongest beam at each measurement point points to these locations, it is impossible to tell the true location of the AP apart from the images, as the same information may admit two (or more) feasible configurations.

With this difficulty in mind, we use a heuristic to choose the true AP location. Notice that if the strongest beam at a point P points to an image, then the image is *always* farther away from P than the true location. Thus from a particular measurement point's view, the true location must be closer than any of the images. Based on this observation, we propose a simple heuristic that works very well in practice. Each measurement point ranks the k images

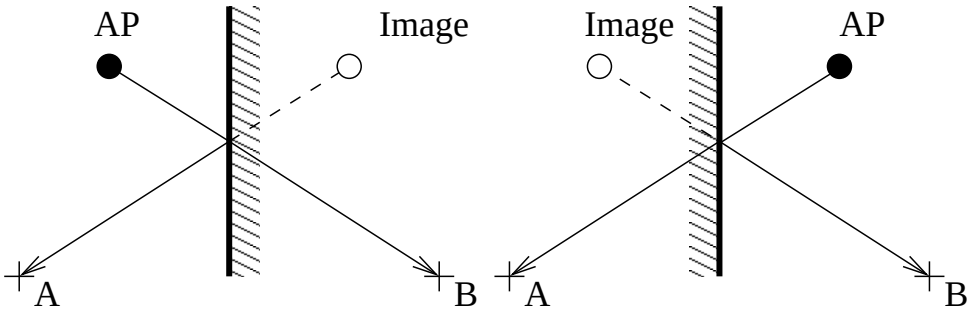


Figure 64: It is impossible to tell the true location of the AP apart from the images, as the same information may admit two (or more) feasible configurations.

based on their distances to itself. The nearest image is ranked 1st and the next is ranked 2nd and so on. We compute the weighted (weights being w_j's defined before) sum of the ranks for each image and choose the image with the least value of the weighted sum. In our experiments this always gives the location closest to the true AP location.

6.3.5 Learning k for Clustering

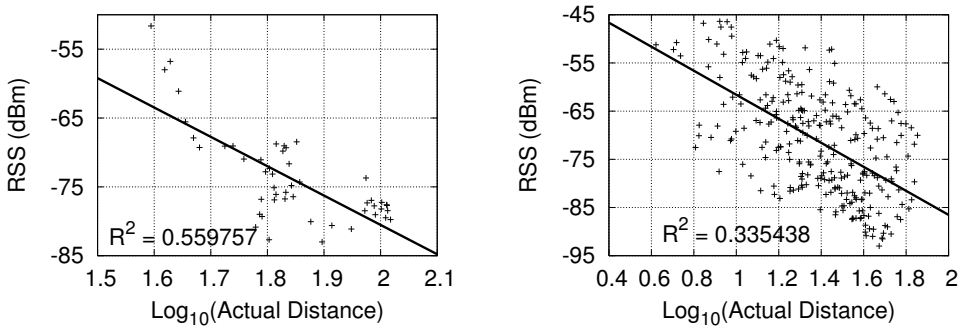
One remaining issue is to determine the right value of k to be used in the k-means clustering algorithm. We need some statistical means to estimate k from the measurement data as there is no *a priori* knowledge of number of images. Intuitively, we would like the measurement points to be clustered nicely such that in each cluster the strongest beams all point to the same location L_i. In other words, the angular error within one cluster should be *uni-modal*. Thus, we use the idea from the G-means algorithm [84] and learn the number of clusters, k, by checking whether the angular error values in each cluster follows a Gaussian distribution.

We start with the value of k=1 and successively increment k, performing a k-means clustering in each step as described before. After clustering, we check whether the error values in each cluster satisfy a statistical test for normality. If they do, we stop the procedure; otherwise, we increment k and repeat. We have used the standard Anderson-Darling normality test [34] [144]

with a significance level[3] of 1% to test for normality.

6.4 Performance Evaluation

In this section, we present a detailed performance evaluation of our Drive-by Localization approach (DrivebyLoc) using measurements from the 21 APs in three environments as shown in Figure 61. The main comparison points are (i) a trilateration approach using distance information [139] estimated using from signal strength information (SNR) using omni-directional antenna,[4] and (ii) VORBA [123], a localization approach using directional antenna and AoA information. We also study and quantify the effect of several factors that impact the performance of our system.



(a) (b)

Figure 65: Relationship between distance and RSS in (a) parking lot (b) apartment complex.

6.4.1 Benefit of Using Directional Antennas and AOA

In Figure 67 (a), we show the CDF of the localization errors for DrivebyLoc and the trilateration approach for the 21 cases studied. The trilateration approach fundamentally depends on deriving distance estimates from the received signal

[3]Significance level is the chance of incorrectly judging a set of values to be not Gaussian.
[4]The same antenna with omni-directional beam is used for this study.

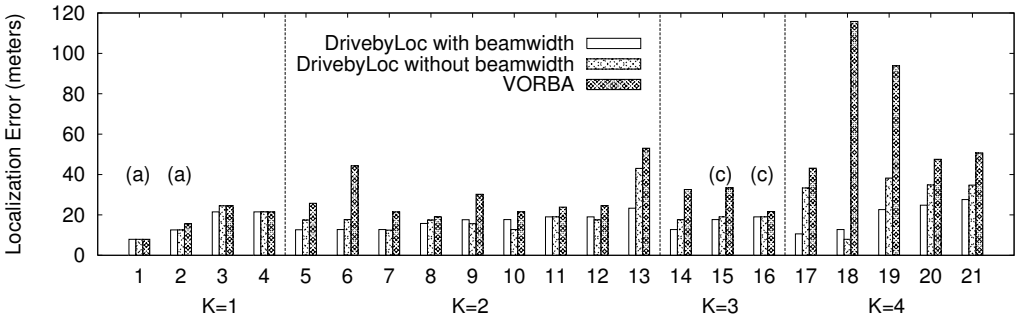


Figure 66: Comparison of DrivebyLoc and VORBA.

strength (RSS). RSS is derived from SNR assuming constant noise. A simple propagation path loss modeling approach was used to infer distance from RSS following the method used in [48]. The idea is to assume exponential decay of RSS with distance. Thus, RSS (in dBm) should have a linear relationship with the log of distance. <RSS, distance> tuples are collected in the data collection phase *assuming* that the AP locations are known. This provides the scatterplot in Figure 65. A linear regression (shown) provides the necessary path loss model to be used to estimate distance from RSS. Note that (i) the R^2 value of the regression is not high; and (ii) the parking lot scenario provides a relatively more accurate modeling (higher R^2). The latter is likely due to lesser reflections and shadowing problems.

Figure 67 (a) clearly shows that DrivebyLoc is about an *order of magnitude* better than trilateration. The median localization error in DrivebyLoc is about 15 m, while it is about 128 m in trilateration. In fact, the maximum error in DrivebyLoc is less than 30 m. This shows that it can localize APs within the accuracy of individual homes even in a very cluttered environment.

6.4.2 Benefit of Modeling Reflection Using Clustering

We now study the benefit of modeling reflection using the k-means clustering idea and also of modeling non-zero beam width. For this purpose, we compare

our results with those of VORBA [123]. In VORBA a similar approach is taken, except that a) signal reflections and b) non-zero beam width are not modeled and c) no weighting using SNR is used.

Figure 66 shows the localization errors obtained using three approaches in each of our 21 experiments categorized by the value of k learned using the Anderson-Darling normality test. For DrivebyLoc we show the performance with and without modeling of non-zero beam width as discussed in Section 6.3.1. Note that the same results for DrivebyLoc with beam width modeling was shown before in Figure 67(a) in CDF form.

Note that all three schemes perform almost similarly when $K = 1$. VORBA's performance worsens for larger K. Some very large errors are observed for VORBA $K = 4$ relative to to DrivebyLoc. This indicates the modeling reflections is critical for accurate localization. The impact of modeling beam width is usually small and in general it is beneficial, though a few outliers are indeed noticed. But overall it should be recommended that DrivebyLoc be used with modeling beam width. Note also that DrivebyLoc's performance with increasing K is relatively stable compared with VORBA. Overall it can be concluded that out of the three differences (see above) between DrivebyLoc and VORBA, modeling reflections has the most impact. Also, overall with the entire data set the median error for DrivebyLoc is about 15 m, while for VORBA it is about 30 m.

For the curious reader, we make a note here which scenarios these 21 sets of experiments correspond to. Note the labels (a) and (c) on top of some of the columns. These correspond to scenarios Figure 61(a) and (c). The rest correspond to the scenario in Figure 61(b). Thus, as expected all schemes get the best results in the parking lot.

6.4.3 Impact of GPS Accuracy

As we take measurements while moving in a car, the location of a measurement point is obtained through a GPS unit. GPS devices are known to have errors [80]. In this section, we study the impact of GPS error in our localization approach. In order to quantify the effect of GPS error, we did one experiment

in an open parking lot as shown in Figure 61(a) by manually measuring the distance between pairs of measurement points and finding *absolute coordinates* for each point. For this purpose, we used the cart setup as mentioned in Section 6.2.2 instead of a car. We moved the cart and stopped at fixed measured distances and took measurements on all directional beams. The dotted lines in Figure 61(a) show the points where the measurements were taken. We also noted the GPS coordinates at each measurement point. The localization error for this particular experiment using the manually measured absolute coordinates is around 12.6 m while the localization error using the GPS coordinates is around 16.4 m. This shows that the GPS errors indeed worsen the performance of our localization approach. To get a better insight about the GPS accuracy, we show the CDF of GPS errors for this particular experiment in Figure 67 (b). The GPS error is computed by finding the difference between actual distance between two points and the distance computed using the GPS coordinates of the two points. The median GPS error is about 1.25 m and this causes an increase in our localization error by about 4 m.

The GPS error could be higher in a cluttered environment with buildings and trees blocking the GPS signals from the satellites. To understand this better, we repeated similar GPS error measurements for the office building scenario. As expected, the errors were higher (median error about 5.5 m) Figure 67 (b), shows the CDF of GPS errors in these two scenarios. This error is quite comparable to the errors DrivebyLoc has in the same scenario (about 20 m). It remains unclear how the GPS errors are contributing to our localization errors. But given the parking lot experience above, the results presented in Figure 66 likely underestimates DrivebyLoc's performance.

6.4.4 Impact of Car Speed

As mentioned in Section 6.2.2, the number of measurement samples could impact the accuracy of DrivebyLoc. The number of samples is inversely proportional to the car speed and directly proportional to the number of runs. Recall that in our data collection approach we either used a car with slow drive (10 mph) or used a cart to walk. Also, in order to 'speed up' the experiments

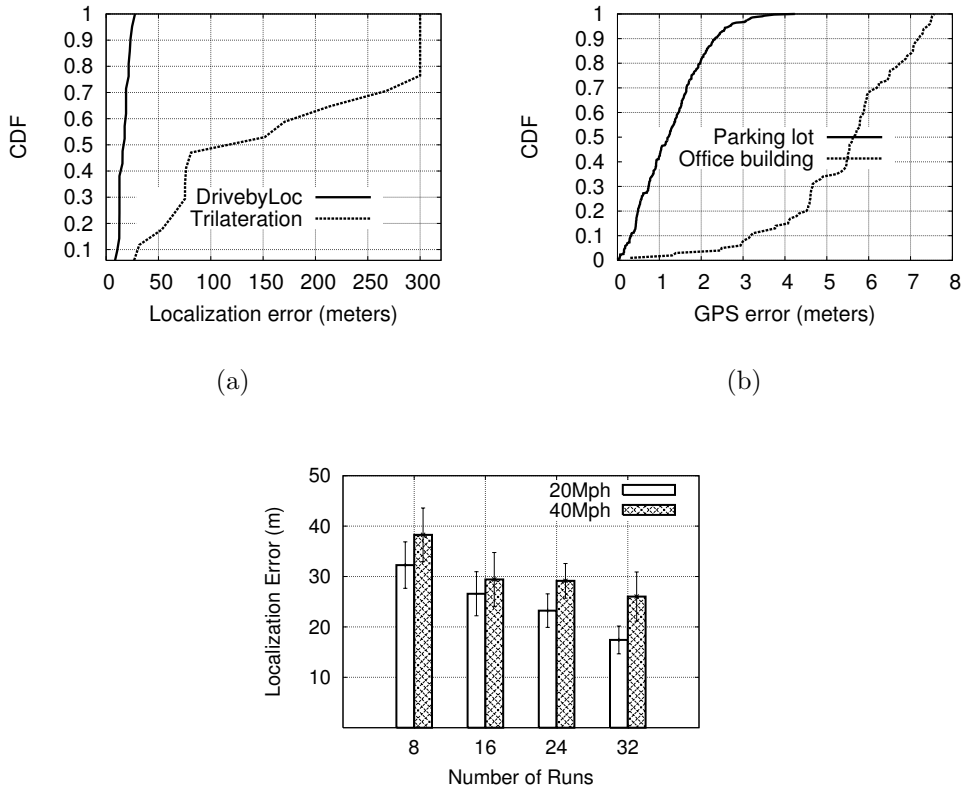


Figure 67: (a) Localization errors for DrivebyLoc and trilateration.(b) CDF of the GPS errors in two scenarios. (c) Impact of speed and number of runs on localization error.

we had the APs broadcast UDP packets at 250 packets/sec which might happen in practice. We are now interested in evaluating what would happen if the car is driven at a more normal speed and APs do not transmit any frame other than beacons at 100ms intervals. We use our existing data to 'simulate' data collection at different speeds and then repeat it to simulate multiple runs. To do this simulation experiment, the existing data is laid out on an imaginary 2D map as points, with each point as if 'lit up' at 100 ms intervals to simulate the corresponding AP beacons. An imaginary car is driven on the same roadway with slightly randomly varying speed (to simulate reality) about an

average. Any point lit up within 5 m from the current car location is counted as measurement at the car location. This can be repeated to simulate multiple drives.

In Figure 67 (c), we show the mean localization errors for the 17 experiments in the apartment complex scenario along with the 95% confidence intervals. Two different speeds and multiple drives are shown. As expected, more runs and slower speeds provide better mean localization error and smaller confidence intervals. As discussed in Section 6.2.2, use of multiple directional antennas and multiple radios on the moving vehicle would help to reduce the number of runs significantly.

6.5 Related Work

RADAR [37] is one of the first systems to do indoor localization of WiFi clients. The key idea is to do an RF fingerprinting *a priori* to collect signal strength values from different APs tagged with location information. When the wireless client needs to be localized, it uses the current signal strength values it receives from different APs and do a lookup on the RF fingerprints. There are several follow up works similar to RADAR [104, 99, 161]. In particular, Ladd et al [104] improve the accuracy of indoor localization from about 10 m by RADAR to within 1 m by using probabilistic inference of positions from noisy signal information. Using a similar idea, Intel's Place lab work [52] localize wireless clients in outdoor settings. All these ideas suffer from the problem of carefully conducting RF fingerprinting. Also in these works, the emphasis is on localizing WiFi clients unlike our approach.

Use of directional antennas for localization is not new. VORBA [123], one of the significant work in AoA based indoor WiFi localization uses WiFi APs equipped with a rotating directional antenna and estimates AoA information from packets transmitted from clients and uses a simple triangulation approach to find the position of the wireless clients. Our approach compared to VORBA is more robust to AoA information suffering from reflections in cluttered environments that are representative of most urban WiFi deployments.

There have been approaches for indoor localization using other mediums such as ultra sounds, infrared, optical waves ets. Active Badge [151] is an indoor localization system that employs infrared medium. Each user is given an infrared badge and can be localized by IR stations that read the badges. Active Bat [85] uses a similar idea but employs ultrasound medium and has extremely high accuracy in the order of centimeters. The Cricket [128] system from MIT is another indoor positioning system that uses ultrasound combined with RF. It uses several beacons that transmit ultrasound waves deployed in the ceiling of each room in the building. The mobile nodes receiving these waves infer the range and localize themselves. In [116], Nasipuri and El Najjar propose an angle based indoor localization system employing optical waves. They use three rotating optical beacon signal generators that generate regular beacons and wireless sensor nodes equipped with photo sensors determine their locations from the estimated angular separations between the optical sources. This idea is somewhat related our approach in the sense that they also use directional beams and angle information.

Finally, localization in multihop adhoc and sensor networks has been studied in a number of works [139, 122, 44, 76] and they differ from each other depending on the type of information used for localization such as angles, ranges and connectivity. The idea is to come up with a consistent embedding of the multihop network in either 2D or 3D plane.

6.6 Summary

In this chapter, we proposed and tested with abundant experiments a system with directional antennas for localizing roadside WiFi access points, by simply driving though the neighborhood where the APs need to be localized. The power of the technique is its complete passive nature. It also does not depend on any prior collection of RF fingerprinting data. While similar approaches have been investigated before [123], our major contribution is identifying that signal reflections can cause significant localization errors and then developing a clustering method to solve this problem. The idea is to recognize *a priori* that there could be images of the AP, and the real AP might be indistinguishable

from the images. Thus, we localize these – possibly multiple – images and then use a heuristic to identify the real one among the set of images. The method has demonstrated very satisfactory localization accuracies even in complex environments, compared with existing approaches with omni-directional or directional antennas. In spite of using the toughest scenarios for localization, the localization errors are roughly between 10-30 m in spite of the fact many of our APs were indoors. This is very good compared to the localization error observed in Intel Place Lab outdoor localization effort [52], where the *median* error is between 13-40 m in spite of very careful radio fingerprinting of the environment. This is an impressive performance given that we observed up to about 7.5 m of GPS error in similar environments.

While our work is limited to 2D, this can be extended to 3D using a combination of antennas so that directivity is on both horizontal and vertical axes can be obtained. We expect that an important fallout of our work will be the creation of a very accurate 'WiFi map' of urban APs with a minimum effort, and eventually motivate novel applications.

Chapter 7

Conclusion

Developing technologies for ubiquitous broadband wireless network access is very important to provide high quality 'anytime and anywhere' wireless connectivity to mobile users. In this dissertation, we explored the potential of IEEE 802.11 based ubiquitous wireless network access and addressed two key challenges - improving *capacity* and *connectivity*.

Improving capacity is a fundamental problem in any form of wireless network design. We proposed efficient channel assignment and directional MAC protocols to improve the capacity of the wireless multi-hop infrastructures significantly by reducing wireless interference. All our techniques are shown to be highly efficient and practical. We demonstrated the performance potential of our techniques by implementing them in commodity off-the-self wireless hardware.

We investigated the use of directional communication to improve the duration and quality of connecitivity of vehicular clients to the wireless infrastructure and proposed practical beam steering and AP selection algorithms. We conducted extensive inter-vehicular communication experiments and studied the performance potential of using directional communication for V2V links. The techniques proposed provide practical solutions supported by thorough experimental study to improve connectivity in highly mobile scenarios.

We believe that the techniques proposed in this dissertation would be useful in providing high speed, reliable and low-cost ubiquitous broadband wireless access to users in the future.

Bibliography

[1] 802.11 Phocus Array Antenna System by Fidelity Comtech. http://www.fidelity-comtech.com/.

[2] Ad hoc On Demand Routing Protocol (Internet RFC 3561). http://www.ietf.org/rfc/rfc3561.txt.

[3] Atheros Communications. http://www.atheros.com.

[4] Chaska Wireless Solutions. http://www.chaska.net.

[5] Dedicated Short Range Communications. http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[6] FON. http://www.fon.com/.

[7] Garmin GPS System. http://www.garmin.com/.

[8] GLPK: GNU Linear Programming Kit. http://www.gnu.org/software/glpk/glpk.html.

[9] Google Street View. http://maps.google.com/help/maps/streetview/.

[10] Honeywell HMC6352 Digital Compass Solution. http://www.ssec.honeywell.com/magnetic/datasheets/HMC6352.pdf.

[11] Kismet Wireless. http://www.kismetwireless.net/.

[12] MADWIFI Project. http://sourceforge.net/projects/madwifi/.

[13] Meraki Networks. http://meraki.com/.

[14] Meru Networks. http://www.merunetworks.com/index.shtml.

[15] NYCwireless Pebble Linux. http://www.nycwireless.net/pebble.

[16] OpenHAL. http://madwifi.org/wiki/OpenHAL.

[17] Router Board - RB14. http://www.routerboard.com/rb11.html.

[18] Soekris Engineering. http://www.soekris.com/.

[19] Telecommunications and Information Exchange between Roadside and Vehicle Systems – 5.9 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical layer (PHY) Specifications. ASTM International, www.astm.org.

[20] The CoMo project. http://como.intel-research.net/.

[21] The Network Simulator ns-2. http://www.isi.edu/nsnam/ns/.

[22] Tropos Networks. http://www.tropos.com.

[23] VANET: Workshop on Vehicular Ad hoc Networks. http://www.sigmobile.org/events/workshops.html.

[24] War Driving. http://www.wardriving.com/.

[25] WiFi Maps. http://wifimaps.zhrodague.net/.

[26] Wireless Geographic Logging Engine. http://www.wigle.net/.

[27] A. Acharya, A. Misra, and S. Bansal. MACA-P: A MAC for Concurrent Transmissions in Multi-Hop Wireless Networks. In *First IEEE Percom 2003 Conference*, pages 505–508, March 2003.

[28] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *First Annual International Conference on Broadband Networks (Broadnets)*, San Jose, California, USA, October 2004.

[29] A. Akella, G. Judd, P. Steenkiste, and S. Seshan. Self Management in Chaotic Wireless Deployments. In *11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[30] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks*, 47(4):445–487, March 2005.

[31] D. Alderson, L. Li, W. Willinger, and J. Doyle. Understanding Internet topology: principles, models, and validation. *IEEE/ACM Transactions on Networking*, 13(6):1205–1218, 2005.

[32] M. Alichery, R. Bhatia, and L. Li. Joint Channel Assignment and Routing for throughput optimization in Multi-Radio wireless mesh networks. In *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[33] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal of Optimization*, 5:13–51, 1995.

[34] T. Anderson and D. Darling. Asymptotic theory of certain 'goodness-of-fit' criteria based on stochastic processes. *Annals of Mathematical Statistics*, 23:193–212, 1952.

[35] T. Aubrey and P. White. A Comparison of Switched Pattern Diversity Antennas. In *IEEE Vehicular Technology Conference (VTC)*, 1993.

[36] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2004.

[37] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM*, pages 775–784, 2000.

[38] C. Balanis. Smart antennas for future reconfigurable wireless communication networks. Tech report. Arizona State University, April 2000.

[39] L. Bao and L. Garcia-Luna-Aceves. Transmission scheduling in ad hoc networks with directional antennas. In *ACM Mobicom*, 2002.

[40] S. J. Benson and Y. Ye. DSDP5: Software for semidefinite programming. Technical Report ANL/MCS-P1289-0905, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, Sept. 2005. Submitted to ACM Transactions on Mathematical Software.

[41] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[42] J. Bicket. Bit-rate selection in wireless networks. *MIT Masters Thesis*, 2005.

[43] S. Biswas, R. Thatchikou, and F. Dion. Vehicle-to-vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety. In *Communications Magazine*, pages 74–82, 2006.

[44] J. Bruck, J. Gao, and A. Jiang. Localization and Routing in Sensor Networks by Local Angle Information. In *ACM MOBIHOC*, May 2005.

[45] M. Buettner, E. Anderson, G. Yee, D. Saha, A. Sheth, D. Sicker, and D. Grunwald. A Phased Array Antenna Testbed for Evaluating Directionality in Wireless Networks. In *MobiEval : System Evaluation for Mobile Platform*, San Juan, Puerto Rico, June 2007.

[46] J. Burgess, G. Bissias, M. D. Corner, and B. N. Levine. Surviving Attacks on Disruption-Tolerant Networks without Authentication. In *Proc. of The ACM International Symposium on Mobile Ad hoc Networking and Computing (Mobihoc)*, 2007.

[47] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *12th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2006.

[48] J. Camp, J. Robinson, C. Steger, and E. W. Knightly. Measurement Driven Deployment of a Two-Tier Urban Mesh Access Network. In *MOBISYS*, Uppasala, Sweden, June 2006.

[49] J. M. Carey and D. Grunwald. Enhancing WLAN Security with Smart Antennas: A Physical Layer Response for Information Assurance. In *IEEE Vehicular Technology Conference (VTC)*, 2004.

[50] R. Chandra, P. Bahl, and P. Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In *Proceedings of the IEEE INFOCOM*, 2004.

[51] Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi. Ieee 802.11 based vehicular communication simulation design for ns-2. In *VANET '06: Proceedings of the 3rd ACM international workshop on Vehicular ad hoc networks*, Los Angeles, Calfornia, 2006.

[52] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *MOBISYS*, pages 233–245, 2005.

[53] D. Choffnes and F. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, Cologne, Germany, 2005.

[54] R. R. Choudhury, T. Ueda, J. Bordim, and N. Vaidya. BeamNet: Ad Hoc Networking Testbed Using Beamforming Antennas. In *IEEE Vehicular Technology Conference (VTC)*, 2005.

[55] R. R. Choudhury and N. Vaidya. Deafness: A MAC Problem in Ad Hoc Networks when using Directional Antennas. In *10th IEEE International Conference on Network Protocols (ICNP)*, 2004.

[56] R. R. Choudhury and N. H. Vaidya. Impact of directional antennas on ad hoc routing. In *Personal and Wireless Communication (PWC)*, September 2003.

[57] R. R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc networks. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[58] Cirond Technologies Inc. Channel Overlap Calculations for 802.11b Networks, 2002. White Paper.

[59] A. Coja-Oghlan, C. Moore, and V. Sanwalani. MAX k-CUT and Approximating the Chromatic Number of Random Graphs. In *ICALP*, 2003.

[60] C. Cordeiro, H. Gossain, and D. Agarwal. A Directional Antenna Medium Access Control Protocol for Wireless Ad Hoc Networks. *Brazilian Telecommunications Society, Special Issue on Sensors and Ad Hoc Networks*, To Appear.

[61] P. Costa, D. Frey, M. Migliavacca, and L. Mottola. Towards Lightweight Information Dissemination in Inter-Vehicular Networks. In *VANET '06: Proceedings of the 3rd ACM international workshop on Vehicular ad hoc networks*, Los Angeles, Calfornia, 2006.

[62] A. Das, H. Alazemi, R. Vijayakumar, and S.Roy. Optimization Models for Fixed Channel Assignment in Wireless Mesh Networks with Multiple Radios. In *Proceedings of the International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005.

[63] S. M. Das, H. Pucha, K. Papagiannaki, and Y. C. Hu. Studying Wireless Routing Link Metric Dynamics. In *IMC*, 2007.

[64] D. Dhoutaut, A. Regis, and F. Spies. Impact of radio propagation models in vehicular ad hoc networks simulations. In *VANET '06: Proceedings of the 3rd ACM international workshop on Vehicular ad hoc networks*, Los Angeles, Calfornia, 2006.

[65] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode. Location-aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication. *IEEE Journal on Selected Areas in Communications (JSAC)*, 15(8), 2007.

[66] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode. VITP: An Information Transfer Protocol for Vehicular Computing. In *2nd ACM International Workshop on Vehicular Ad-Hoc Networks (VANET)*, 2005.

[67] Y. Ding, C. Wang, and L. Xiao. A Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks. In *4th ACM International Workshop on Vehicular Ad-Hoc Networks (VANET)*, 2007.

[68] F. Dobias and W. Grabow. Adaptive array antennas for 5.8 GHz vehicle to roadside communication. In *IEEE Vehicular Technology Conference (VTC)*, 1994.

[69] T. Elbatt, T. Anderson, and B. Ryu. Performance evaluation of multiple access protocols for ad hoc networks using directional antennas. In *WCNC*, 2003.

[70] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *ACM SIGCOMM*, 2003.

[71] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *Proc. Sigcomm*, pages 251–262, 1999.

[72] W. Franz, R. Eberhardt, and T. Luckenbach. FleetNet - Internet on the Road. In *8th World Congress on Intelligent Transportation Systems (ITS)*, 2001.

[73] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algoritmica*, 18, 1997.

[74] R. Gass, J. Scott, and C. Diot. Measurements of in-motion 802.11 networking. In *WMCSA '06: Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications*, Washington, DC, USA, 2006.

[75] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, and G. Marfia. Vehicular Grid Communications: The Role of the Internet Infrastructure. In *Wireless Internet Conference (WICON)*, 2006.

[76] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *International Conference on Intelligent Robots and Systems*, October 2001.

[77] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42(6), 1995.

[78] M. Gong, S. Midkiff, and S. Mao. A Combined Proactive Routing and Multi-Channel MAC Protocol for Wireless Ad Hoc Networks. In *Second Annual International Conference on Broadband Networks (Broadnets)*, 2005.

[79] H. Gossain, C. Cordeiro, D. Cavalcanti, and D. Agarwal. The Deafness Problems and Solutions in Wireless Ad Hoc Networks using Directional Antennas. In *IEEE Workshop on Wireless Ad Hoc and Sensor Networks, in conjunction with IEEE Globecom*, November 2004.

[80] GPS Errors and Estimating Receiver Accuracy. http://edu-observatory.org/gps/gpsaccuracy.html.

[81] M. Grotschel, L. Lovasz, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. *Springer-Verlag*, 1987.

[82] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[83] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *5th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2007)*, San Juan, Puerto Rico, June 2007.

[84] G. Hamerly and C. Elkan. Learning the K in K-mean. *Advances in Neural Information Processing Systems (MIT Press)*, 16, 2004.

[85] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *ACM MOBICOM*, Seattle, Washington, August 1999.

[86] A. Hertz and D. de Werra. Using Tabu Search Techniques for Graph Coloring. *Computing*, 39(4), 1987.

[87] Z. Huang, C. Shen, C. Srisathapornphat, and C. Jaikaeo. A busy-tone based directional MAC protocol for ad hoc networks. In *Milcom*, 2002.

[88] F. Hui and P. Mohapatra. Experimental characterization of multi-hop communications in vehicular ad hoc network. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, Cologne, Germany, 2005.

[89] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[90] IEEE Std. 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[91] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of Interference on Multi-hop Wireless Network Performance. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2003.

[92] A. Kashyap, S. Ganguly, and S. R. Das. A Measurement-Based Approach to Modeling Link Capacity in 802.11-based Wireless Networks. In *Proceedings of MobiCom*, 2007.

[93] A. Kaul, K. Ramachandran, P. Shankar, S. Oh, M. Gruteser, I. Seskar, and T. Nadeem. Effect of antenna placement and diversity on vehicular network communications. In *Proc. IEEE SECON Conference*, 2007.

[94] W. Kellerer. (Auto)Mobile Communication in a Heterogeneous and Converged World. *IEEE Personal Communications*, 8(6):41–47, 2001.

[95] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rossel, P. Vortisch, S. Assenmacher, and F. Busch. Enabling efficient and accurate large-scale simulations of VANETs for vehicular traffic management. In *4th ACM International Workshop on Vehicular Ad-Hoc Networks (VANET)*, 2007.

[96] B. Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[97] Y. Ko, V. Shankarkumar, and N. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *19th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2000.

[98] T. Korakis, G. Jakllari, and L. Tassiulas. A MAC protocol for full exploitation of directional antennas in ad hoc wireless networks. In *ACM Mobihoc*, June 2003.

[99] P. Krishnan, A. S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A system for LEASE: System for location estimation assisted by stationary emitters for indoor RF wireless networks. In *IEEE Infocom*, Hongkong, March 2004.

[100] V. Kumar and S. R. Das. Performance of Dead Reckoning-Based Location Service for Mobile Ad Hoc Networks. *Wireless Communications and Mobile Computing Journal*, 4(2), 2004.

[101] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. *SIGMETRICS Perform. Eval. Rev.*, 33(1), 2005.

[102] P. Kyasanur and N.H.Vaidya. Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces. In *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[103] P. Kyasanur and N. H. Vaidya. Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 10(1):31–43, 2006.

[104] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *ACM MOBICOM*, Atlanta, GA, September 2002.

[105] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. Efficient Data Harvesting in Mobile Sensor Platforms. In *Second IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS)*, 2006.

[106] X. Lin and S. Rasool. A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks. In *Proceedings of the IEEE INFOCOM*, 2007.

[107] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley*, University of California Press, 1967.

[108] R. Mahajan, J. Zahorjan, and B. Zill. Understanding wifi-based connectivity from moving vehicles. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, San Diego, California, USA, 2007.

[109] R. Maheshwari, H. Gupta, and S. R. Das. Mutichannel MAC Protocols for Wireless Networks. In *Proceedings of the International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2006.

[110] M. K. Marina and S. Das. A Topology Control Approach to Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Second Annual International Conference on Broadband Networks (Broadnets)*, 2005.

[111] A. Mishra, M. ho Shin, and W. A. Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *23rd*

Annual IEEE Conference on Computer Communications (INFOCOM), 2004.

[112] R. Montemanni, D. Smith, and S. Allen. Lower bounds for fixed spectrum frequency assignment. *Annals of Operations Research*, 107, October 2001.

[113] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto. CarNet: A Scalable Ad HocWireless Network System. In *9th ACM SIGOPS European Workshop*, 2000.

[114] M. Moske, H. Fubler, H. Hartenstein, and W. Franz. Performance Measurements of a Vehicular Ad Hoc Network. In *Vehicular Technology Conference (VTC)*, 2004.

[115] A. Nandan, S. Das, G. Pau, M. Sanadidi, and M. Gerla. Cooperative downloading in Vehicular Ad Hoc Networks. In *2nd Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2005.

[116] A. Nasipuri and R. E. Najjar. Experimental Evaluation of an Angle Based Indoor Localization System. In *WiNMee*, Boston, MA, April 2006.

[117] A. Nasipuri, S. Ye, J. You, and R. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. In *IEEE Wireless Communication and Networking Conference (WCNC)*, pages 1214–1219, September 2000.

[118] V. Navda, A. Kashyap, and S. R. Das. Design and evaluation of iMesh: An infrastructure-mode wireless mesh network. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2005.

[119] M. Neufeld and D. Grunwald. Using phase array antennas with the 802.11 MAC protocol. In *1st International Conference on Broadband Networks (BroadNets)*, 2004.

[120] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall. Improved Access Point Selection. In *4th International Conference on Mobile Systems, Applications, and Services (Mobisys)*, June 2006.

[121] D. Niculescu. Interference Map for 802.11 Networks. In *IMC*, 2007.

[122] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *IEEE Globecom*, November 2001.

[123] D. Niculescu and B. Nath. VOR Base Stations for Indoor 802.11 Positioning. In *ACM MOBICOM*, 2004.

[124] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for Automobile Users. In *23rd Annual IEEE Conference on Computer Communications (INFOCOM)*, 2004.

[125] J. Ott and D. Kutscher. A Disconnection-Tolerant Transport for Drive-thru Internet Environments. In *24th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2005.

[126] J. Padhye, S. Agarwal, V. Padmanaban, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *Internet Measurement Conference (IMC)*, 2005.

[127] Phocus Array Antenna System: Manual. Fidelity Comtech Inc. 2006.

[128] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *ACM MOBICOM*, Rome, Italy, July 2001.

[129] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A General Model of Wireless Interference. In *Proceedings of MobiCom*, 2007.

[130] A. H. M. Rad and V. Wong. Joint Channel Allocation, Interface Assignment and MAC Design for Multi-Channel Wireless Mesh Networks. In *Proceedings of the IEEE INFOCOM*, 2007.

[131] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Proceedings of the IEEE INFOCOM*, 2006.

[132] R. Ramanathan. On the performance of Ad hoc networks with beamforming antennas. In *ACM Mobihoc*, pages 95–105, October 2001.

[133] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad Hoc Networking with Directional Antennas: A Complete System Solution. *IEEE Journal on Selected Areas in Communication (JSAC)*, 23(3):496–506, 2005.

[134] A. Raniwala and T. Chiueh. Architechture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proceedings of the IEEE INFOCOM*, 2005.

[135] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 8(2), 2004.

[136] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-Based Models of Delivery and Interference in Static Wireless Networks. In *ACM SIGCOMM*, 2006.

[137] C. Robinson, L. Caminiti, D. Caveney, and K. Laberteaux. Efficient coordination and transmission of data for vehicular safety applications. In *VANET '06: Proceedings of the 3rd ACM international workshop on Vehicular ad hoc networks*, Los Angeles, Calfornia, 2006.

[138] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy. Experimenting with a Multi-Radio Mesh Networking Testbed. In *(WiNMee Workshop)*, 2005.

[139] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *ACM MOBICOM*, pages 166–179, 2001.

[140] R. Sengupta and Q. Xu. DSRC for Safety Systems. In *Intellimotion–Research Updates in Intelligent Transportation Systems*, 2004.

[141] J. Shi, T. Salonidis, and E. Knightly. Starvation Mitigation Through Multi-Channel Coordination in CSMA Multi-hop Wireless Networks. In *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2006.

[142] J. P. Singh, N. Bambos, B. Srinivasan, and D. Clawin. Wireless LAN Performance Under Varied Stress Conditions in Vehicular Traffic Scenarios. In *Vehicular Technology Conference (VTC)*, 2002.

[143] J. So and N. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. In *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.

[144] M. Stephens. EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association*, 69:730–737, 1974.

[145] K. Sundaresan and R. Sivakumar. On the medium access control problem in ad-hoc networks using directional antennas. In *ACM Mobihoc*, June 2003.

[146] M. Takai, J. Martin, R. Bagrodia, and A. Ren. Directional Virtual Carrier Sensing for directional antennas in mobile ad hoc networks. In *ACM Mobihoc*, pages 39–46, June 2002.

[147] J. Tang, G. Xue, and W. Zhang. Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks. In *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[148] S. N. Technologies. Qualnet simulator version 3.7. www.scalable-networks.com.

[149] P. M. Vaidya. A New Algorithm for Minimizing Convex Functions Over Convex Sets. *Mathematical Programming*, 73(3), 1996.

[150] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar. Component Based Channel Assignment in Single Radio, Multichannel Ad hoc Networks. In *Proceedings of the Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2006.

[151] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10:91–102, January 1992.

[152] J. H. Winters. Signal Acquisition and Tracking with Adaptive Arrays in the Digital Mobile Radio System IS-54 with Flat Fading. *IEEE Transactions on Vehicular Technology*, 42(4):377–384, 1993.

[153] J. H. Winters. Smart antennas for Wireless Systems. *IEEE Personal Communications*, 5(1):23–27, 1998.

[154] J. H. Winters and M. J. Gans. Versus Phased Arrays in Mobile Radio Systems. *IEEE Transactions on Vehicular Technology*, 48(2):353–362, 1999.

[155] H. Wu, M. Palekar, R. Fujimoto, R. Guensler, M. Hunter, J. Lee, and J. Ko. An empirical study of short range communications for vehicles. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, Cologne, Germany, 2005.

[156] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)*, 2000.

[157] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-vehicle safety messaging in DSRC. In *1st ACM international workshop on Vehicular ad hoc networks (VANET)*, 2004.

[158] Q. Xue and A. Ganz. Temporal Topology Control in Multi-channel Multihop Wireless Access Networks. In *Second Annual International Conference on Broadband Networks (Broadnets)*, 2005.

[159] R. M. Yadumurthy, A. C. H, M. Sadashivaiah, and R. Makanaboyina. Reliable MAC Broadcast Protocol in Directional and Omni-Directional Transmissions for Vehicular Ad Hoc Networks. In *2nd ACM International Workshop on Vehicular Ad-Hoc Networks (VANET)*, 2005.

[160] Y. Yamasaki, M. Yasunaga, Y. Murakami, and H. Moribe. A Beam Controlled Array Antenna for Road to Vehicle Communications. In *IEEE Intelligent Transportation Systems*, 2001.

[161] M. Youssef, A. Agrawala, and U. Shankar. WLAN location determination via clustering and probability distributions. Technical report, University of Maryland, College Park, MD, March 2003.

[162] T. Yum and K. Hung. Design Algorithms for multihop packet radio networks with multiple directional antennas. *IEEE Transactions on Communications*, 40(11), 1992.

[163] X. Zhang, J. K. Kurose, B. N. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, Montreal, Quebec, Canada, 2007.